

Continuous-Time Models of Arrival Times and Optimization Methods for Variable Selection

by

Michael Lindon

Department of Statistical Science
Duke University

Date: _____

Approved:

Surya T. Tokdar, Supervisor

Merlise A. Clyde

Mike West

Jennifer M. Groh

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2018

ABSTRACT

Continuous-Time Models of Arrival Times and Optimization
Methods for Variable Selection

by

Michael Lindon

Department of Statistical Science
Duke University

Date: _____

Approved:

Surya T. Tokdar, Supervisor

Merlise A. Clyde

Mike West

Jennifer M. Groh

An abstract of a dissertation submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2018

Copyright © 2018 by Michael Lindon
All rights reserved except the rights granted by the
Creative Commons Attribution-Noncommercial Licence

Abstract

This thesis naturally divides itself into two sections. The first two chapters concern the development of Bayesian semi-parametric models for arrival times. Chapter 2 considers Bayesian inference for a Gaussian process modulated temporal inhomogeneous Poisson point process, made challenging by an intractable likelihood. The intractable likelihood is circumvented by two novel data augmentation strategies which result in Gaussian measurements of the Gaussian process, connecting the model with a larger literature on modelling time-dependent functions from Bayesian non-parametric regression to time series. A scalable state-space representation of the Matern Gaussian process in 1 dimension is used to provide access to linear time filtering algorithms for performing inference. An MCMC algorithm based on Gibbs sampling with slice-sampling steps is provided and illustrated on simulated and real datasets. The MCMC algorithm exhibits excellent mixing and scalability.

Chapter 3 builds on the previous model to detect specific signals in temporal point patterns arising in neuroscience. The firing of a neuron over time in response to an external stimulus generates a temporal point pattern or “spike train”. Of special interest is how neurons encode information from dual simultaneous external stimuli. Among many hypotheses is the presence multiplexing - interleaving periods of firing as it would for each individual stimulus in isolation. Statistical models are developed to quantify evidence for a variety of experimental hypotheses. Each experimental hypothesis translates to a particular form of intensity function for the dual stimuli

trials. The dual stimuli intensity is modelled as a dynamic superposition of single stimulus intensities, defined by a time-dependent weight function that is modelled non-parametrically as a transformed Gaussian process. Experiments on simulated data demonstrate that the model is able to learn the weight function very well, but other model parameters which have meaningful physical interpretations less well.

Chapters 4 and 5 concern mathematical optimization and theoretical properties of Bayesian models for variable selection. Such optimizations are challenging due to non-convexity, non-smoothness and discontinuity of the objective. Chapter 4 presents advances in continuous optimization algorithms based on relating mathematical and statistical approaches defined in connection with several iterative algorithms for penalized linear regression. I demonstrate the equivalence of parameter mappings using EM under several data augmentation strategies - location-mixture representations, orthogonal data augmentation and LQ design matrix decompositions. I show that these model-based approaches are equivalent to algorithmic derivation via proximal gradient methods. This provides new perspectives on model-based and algorithmic approaches, connects across several research themes in optimization and statistics, and provides access, beyond EM, to relevant theory from the proximal gradient and convex analysis literatures.

Chapter 5 presents a modern and technologically up-to-date approach to discrete optimization for variable selection models through their formulation as mixed integer programming models. Mixed integer quadratic and quadratically constrained programs are developed for the point-mass-Laplace and g-prior. Combined with warm-starts and optimality-based bounds tightening procedures provided by the heuristics of the previous chapter, the MIQP model developed for the point-mass-Laplace prior converges to global optimality in a matter of seconds for moderately sized real datasets. The obtained estimator is demonstrated to possess superior predictive performance over that obtained by cross-validated lasso in a number of real datasets.

The MIQCP model for the g-prior struggles to match the performance of the former and highlights the fact that the performance of the mixed integer solver depends critically on the ability of the prior to rapidly concentrate posterior mass on good models.

To my family and friends

Contents

Abstract	iv
List of Figures	xii
List of Abbreviations and Symbols	xiv
Acknowledgements	xvi
1 Introduction	1
2 A Continuous-Time Model of Arrival Times	7
2.1 The Inhomogeneous Poisson Point Process	7
2.1.1 The Likelihood Function	7
2.1.2 Thinning and Superposition	8
2.2 The Gaussian Process Modulated Inhomogeneous Poisson Point Process	10
2.2.1 Gaussian Process Intensity Functions	10
2.2.2 Data Augmentation Strategies	11
2.2.3 State-Space Representation of Gaussian Processes	16
2.2.4 Forward Filtering Backward Sampling	19
2.3 Full Model Specification	23
2.4 Gibbs Sampling	24
2.5 Illustrations	27
2.5.1 Simulated Data	27
2.5.2 Coal Mining Disasters	27

2.6	Discussion	28
2.7	A Novel Implementation of Gaussian Processes	30
2.7.1	Motivation	30
2.7.2	Conditional Distributions Under State-Space Construction	32
2.7.3	The Persistent AVL-Tree	34
2.7.4	Example Implementation	35
3	Detecting Multiplexing in Neuronal Spike Trains	38
3.1	Statistical Model for Multiplexing	40
3.2	Data Augmentation	42
3.3	Complete Model Description	44
3.4	The Euler-Maruyama Approximation	46
3.4.1	Forward Filtering	47
3.4.2	Backwards Sampling	48
3.4.3	Stability	48
3.5	Gibbs Sampling	49
3.5.1	Steps for Single Trial Specific Parameters	49
3.5.2	Steps for Augmented Realizations	50
3.5.3	Steps for Dual Trial Specific Parameters	51
3.6	Model Performance on Simulated Data	53
3.7	A Simplified Model	56
3.8	Cell YKIC092311_1 609Hz 24° 742Hz -6°	58
4	Continuous Optimization for Variable Selection	61
4.1	Spike and Slab Priors	61
4.2	Theory Based on Compatibility Condition	62
4.3	Theory Based on η -Null Consistency Condition	70

4.4	Continuous Optimization Methods	73
4.5	Introduction	73
4.6	Preliminaries	76
4.6.1	MM algorithm	76
4.6.2	EM algorithm	77
4.6.3	Lipschitz-Gradient Functions	78
4.7	Algorithm Comparisons	79
4.7.1	Proximal Gradient Method	79
4.7.2	Orthogonal Data Augmentation	80
4.7.3	Location Mixtures of Spherical Normals	82
4.7.4	LQ-decompositions of the Design Matrix	84
4.8	Comparative Discussion	85
4.9	Experimentation for Non-Convex Problems	86
5	Discrete Optimization for Variable Selection	91
5.1	Discrete Optimization	91
5.2	Branch and Bound	94
5.3	Mixed Integer Programming	99
5.3.1	Introduction to Solving Mixed Integer Programs	100
5.4	A MIQP for Best Subset Selection	103
5.4.1	Big-M Transformation	104
5.4.2	Monotonicity vs Continuous Relaxation	105
5.5	Optimality Based Bounds Tightening	106
5.6	Mixed Integer Models for Point Mass-Laplace Mixture Priors	107
5.6.1	Example: Diabetes Dataset	109
5.6.2	Example: Ozone Dataset	114

5.7	Mixed Integer Models for Zellner’s G-Prior	115
5.7.1	Parameter Expansion	117
5.7.2	Piecewise Linear Approximations	121
5.7.3	Model Space Priors	123
5.7.4	Specially Ordered Sets of Type I/II	124
5.7.5	Heuristics	126
5.7.6	Bound Tightening	127
5.7.7	Example: Diabetes Dataset	128
6	Concluding Remarks	133
A	Figures	139
	Bibliography	144
	Biography	152

List of Figures

2.1	Thinning a homogeneous Poisson point process	9
2.2	Posterior summary of intensity function for simulated dataset I with true intensity $\lambda(t) = 400\sigma(-1 + 3 \sin(2\pi 3t))$	29
2.3	Posterior summary of intensity function for coal mining disasters dataset	29
3.1	Posterior summaries of hyperparameters for simulated dataset I . . .	55
3.2	Posterior summaries of dual trial specific parameters for simulated dataset I	56
3.3	Posterior summaries of dual trial specific parameters for simulated dataset II	57
3.4	Posterior summaries for cell YKIC092311_1 609hz 24° 742hz -6° . . .	59
3.5	Raw data for cell YKIC092311_1 609Hz 24° 742Hz -6°.	60
4.1	Proximal functions for various penalties	87
4.2	Performance of rcd, gcd and proximal gradient w.r.t. ccd on dataset 1	89
4.3	Performance of rcd, gcd and proximal gradient w.r.t. ccd on dataset 2	89
4.4	Performance of rcd, gcd and proximal gradient w.r.t. ccd on dataset 3	90
5.1	An illustration of monotonicity-based branch and bound	97
5.2	Effect of tighter coefficient bounds on progress of optimality gap . . .	108
5.3	Predictor correlations in diabetes dataset.	109
5.4	Coefficient bounds obtained through optimality-based bounds tightening of model (5.9) for diabetes dataset	112
5.5	Estimated coefficients resulting from point-mass-Laplace mixture prior and cross-validated lasso for diabetes dataset	112

5.6	Coefficient bounds obtained through optimality-based bounds tightening of model (5.6) for diabetes dataset	114
5.7	Coefficient bounds obtained through optimality-based bounds tightening of model (5.9) for ozone dataset	115
5.8	Predictor correlations in Ozone dataset	116
5.9	Estimated coefficients resulting from point-mass-Laplace mixture prior and cross-validated lasso for ozone dataset	117
5.10	Piecewise linear approximation and polyhedral relaxation of logarithmic term	120
5.11	Coefficient bounds obtained through optimality-based bounds tightening of model (5.32) for diabetes dataset	130
5.12	Progress of the optimality gap for model (5.32) on diabetes dataset	130
A.1	Additional posterior summaries of simulated dataset I	140
A.2	Additional posterior summaries of coal mining distasters dataset	141
A.3	Posterior summaries of A-trial intensity parameters for cell YKIC092311_1 609Hz 24° 742Hz -6°	142
A.4	Posterior summaries of B-trial intensity parameters for cell YKIC092311_1 609Hz 24° 742Hz -6°	143

List of Abbreviations and Symbols

Symbols

Put general notes about symbol usage in text here. Notice this text is double-spaced, as required.

$N(\mathcal{T})$	Number of time-points in interval \mathcal{T} .
$ S $	Cardinality of set S .
$\mathcal{D}_{[a,b]}$	A distribution \mathcal{D} truncated to the interval $[a, b]$.
$\mathcal{N}(\mu, \sigma^2)$	A Normal distribution with mean μ and variance σ^2 .
$\mathcal{PG}(b, c)$	A Polya-Gamma distribution with parameters $b > 0$ and $c \in \mathbb{R}$.
$\mathcal{PPP}(\lambda, \mathcal{D})$	Inhomogeneous Poisson point process with intensity λ and mark distribution \mathcal{D} (optional).
$Ga(a, b)$	A gamma distribution with shape a and rate b .
$\mathcal{DE}(\lambda)$	A double-exponential (Laplace) distribution with rate λ .
$\mathcal{B}(a, b)$	A Beta distribution with shapes a and b .
$Bern(\theta)$	A Bernoulli distribution with probability θ .
$IG(a, b)$	Inverse gamma distribution with shape a and scale b .
$Dir(\alpha_1, \dots, \alpha_k)$	A Dirichlet distribution with concentration parameters $\alpha_i > 0$.

Abbreviations

Long lines in the `sybollist` environment are single spaced, like in the other front matter tables.

MCMC	Markov chain Monte Carlo.
GP	Gaussian process.
PPP	Poisson point process.
ccd	Cyclic coordinate descent.
gcd	Greedy coordinate descent.
rcd	Randomized coordinate descent.
BnB	Branch and bound.
MIP	Mixed integer program/programming.
MIQP	Mixed integer quadratic program/programming.
MIQCP	Mixed integer quadratically constrained program/programming.
MINLP	Mixed integer non-linear program/programming.

Acknowledgements

I would like to thank each of my committee members for their friendly encouragement and patience while completing my PhD. I have had the pleasure of working with each of them and each have distinctly supported and shaped my progress, for which I am very grateful. I hope that they will enjoy reading this thesis. In addition to my committee members, I would like to acknowledge the support provided by other students in the PhD program. In particular my friend Kaoru Irie, with whom I have enjoyed many a helpful conversation about research.

I am thankful to Duke Graduate School for providing me with a James B. Duke fellowship, to the BEST foundation for the BEST awarded which provided me with funding over the summer of 2017 and to SAMSI for funding me as a year-long research assistant in their 2016-2017 program on optimization. Coming to Duke has opened up many doors to me and I am thankful to those that brought me here.

1

Introduction

Chapters 2 and 3, which construct continuous time Bayesian semi-parametric models for arrival times and temporal point patterns, can be read completely in isolation from chapters 4 and 5, which deal with mathematical optimization of problems arising in variable selection. The reader should feel free to start with the half that most interests them.

- Chapter 2: “*A Continuous-Time Model of Arrival Times*”.

This chapter develops a Bayesian semi-parametric model of arrival times or temporal point patterns. The observed times are modelled as a point process in 1 dimension - a realization from an inhomogeneous Poisson point process. Inference in these models is focussed on learning the *intensity function*, from which all properties of the Poisson point process are derived. The intensity function is modelled nonparametrically as a transformed Gaussian process which provides support over a rich family of possible intensity functions. The challenge with such “*doubly stochastic*” Poisson process models is that the resulting likelihood is intractable - it contains an integral over the function being modelled as a

Gaussian process. Typical approaches to this challenge are through *discretization*, using an approximate likelihood resulting from a discrete approximation to the integral. This requires an undesirable arbitrary choice of discretization, something which has the potential to introduce bias into the analysis. Similar approaches bin the data into a partition of the time over which the process is observed, proceeding to then model the data as counts. This also requires an arbitrary choice of partition, which may also introduce bias into the analysis. This is a larger issue in chapter 3, where choosing a partition of time may obscure temporal signals in the point pattern. A recent alternative to discretization is a data augmentation approach based on *thinning*, augmenting an unobserved realization from an additional inhomogeneous Poisson point process which can be regarded as a byproduct of the data generating process of the observed point pattern. The likelihood under the complete data contains only a tractable integral over a homogeneous Poisson point process. Previous works developing this approach were lacking attractive MCMC algorithms for performing inference, requiring many tuning parameters and still complicated proposals for the intensity function. The attractiveness of the model proposed in this chapter is perhaps its simplicity. In addition to augmenting the thinned realization, each realization is elevated to a *marked* inhomogeneous Poisson point process, where each point in addition to a time has an associated Gaussian random variable. The latter is related to *probit* and *Polya-gamma* data augmentation strategies, reducing the problem to essentially having Gaussian distributed measurements of the intensity function over time. This is attractive because it transforms an unfamiliar problem into a familiar problem, namely, a problem where one has Gaussian distributed measurements of a function over time which has received a lot of attention in the nonparametric regression and time series literature. This enables many possibilities in modelling the tempo-

ral dependence of the intensity function. The approach adopted in this chapter is to model the intensity function via a Gaussian process that admits an exact state-space representation in one dimension by additionally modelling first and second derivatives. This allows inference to be performed in linear time by using filtering algorithms that exploit the Markov conditional independence structure of the state-space representation.

- Chapter 3: “*Detecting Multiplexing in Neuronal Spike Trains*”.

This chapter leverages the scalability and flexibility of the previously developed model for detecting time-dependent signals in temporal point patterns arising in neuroscience. The experimental goal is to better understand how neurons encode information from multiple simultaneous external stimuli - behaviour that is currently poorly understood. By recording the firing times of a neuron presented with single stimulus and then comparing these with the firing times of the same neuron presented with multiple stimuli it is hoped that their juxtaposition will reveal the relationships between single and multiple stimuli firing rates. By modelling the temporal point patterns as realizations from an inhomogeneous Poisson point process, different experimental hypotheses about the firing rate can be translated to different forms for the intensity function. One such hypothesis amounts to the intensity function for the dual stimuli trials being a dynamic superposition of the intensity functions for the single stimulus trials. The most common analyses in neuroscience record multiple spike-trains and regard these as realizations from the same inhomogeneous Poisson point process - pooling them together to form a sufficient statistic. If the temporal dependence of the superposition is not the same across repeated measurements, then any temporal signal is lost when aggregating trials. In addition previous analyses also binned firing times into an arbitrary partition of the time interval

of observation. If the bin width is of the order of the temporal dependence, then the temporal signal can be lost. The model developed in this chapter addresses all of these issues. Experiments on simulated and real data reveal that the model is capable of learning the temporal dependence of the superposition very well, but other model parameters with meaningful physical interpretations are difficult to learn from such data.

- Chapter 4: “*Continuous Optimization for Variable Selection*”.

Continuous optimization refers to mathematical optimization problems in which the parameters are elements of a continuum such as \mathbb{R}^p . This does not mean that the objective function is a continuous function. In fact, the optimization problems which are motivated as finding MAP estimates in Bayesian variable selection models are typically non-convex, non-smooth and even discontinuous. These applications fall beyond classical optimization theory and require new tailored algorithms. This chapter initially began by exploring the utility of orthogonal data augmentation in developing an EM algorithm for finding good quality local solutions. After some literature review, it became clear that the same algorithm had been discovered and re-discovered in many disciplines from a variety of different motivations. This chapter begins by providing theoretical support for the MAP estimate under the point-mass-Laplace mixture prior as an estimator is only attractive if it is both computationally tractable and possesses desirable theoretical properties. The first approach adapts theory for Lasso by assuming a compatibility criterion on the design matrix, allowing bounds on sparsity, estimation error and predictive error that hold with high probability to be derived. The resulting bounds reduce to existing bounds for the lasso as special cases and also provide an upper bound on the sparsity of the estimated regression vector - a bound which is not available for the lasso.

The second approach reaches similar results through different assumptions, namely, the η -Null consistency criterion and the restricted invertibility factor of the design matrix. Following these results the performance of the proximal gradient and coordinate descent methods are evaluated on simulated datasets. Surprisingly the coordinate descent methods appear to find better quality local optima.

- Chapter 5: “*Discrete Optimization for Variable Selection*”.

The theoretical properties derived in the previous chapter are for the MAP estimate, in other words the global optimum of the optimization formulation. Heuristics such as the proximal gradient and coordinate descent methods provide no guarantees of finding the globally optimal solution and can frequently get stuck in very poor local optima. As such their practical use carries a high risk, as one has no idea of the quality of the solution returned. This chapter on global discrete optimization combines the heuristics of the previous chapter with algorithms that are guaranteed to find the globally optimal solution or, if stopped early, provide quantitative statements about how optimal is the best solution found so far. These problems are NP-hard and have exponential worst-case complexity but can actually work surprisingly well for certain problems. This chapter will illustrate when one can expect these algorithms to work well. In particular, the mixed integer quadratic program developed for obtaining the MAP estimate under point-mass-Laplace mixture priors converges rapidly in a matter of seconds for moderately sized real datasets, making it a realistic and practically viable approach. The mixed integer quadratically constrained program developed for the g-prior, however, fails to converge in a reasonable amount of time. This chapter begins by recalling the history of exact discrete optimization methods applied to variable selection and proceeds to present a

modern state-of-the-art approach based on mixed integer programming, generalizing earlier work.

A Continuous-Time Model of Arrival Times

2.1 The Inhomogeneous Poisson Point Process

2.1.1 The Likelihood Function

The first two chapters of this thesis concern the modelling of temporal point-patterns - sets of event times, denoted $\{t_i\}_{i=1}^n$, observed over an interval of time $\mathcal{T} \subset \mathbb{R}$. For point patterns there are in fact two objects to be modelled. The first is the number of points in the interval \mathcal{T} , denoted $N(\mathcal{T})$, and secondly the times themselves. For this reason a *realization* from a point process is commonly denoted as an ordered pair $\xi = (n, \{t_i\}_{i=1}^n)$ which makes the two quantities explicit. Sets are used for the times instead of an ordered tuples because the ordering of the times is irrelevant. A typical assumption is to model the point-pattern as a realization from an inhomogeneous (or non-homogeneous) Poisson point process, popular for its simplicity and to some extent its tractability. It is defined by an *intensity function* $\lambda : \mathcal{T} \rightarrow \mathbb{R}^+$ that is locally integrable, $\int_B \lambda(\xi) d\xi \leq \infty$ for all bounded $B \in \mathcal{T}$, defining an *intensity measure* $\Lambda(B) = \int_B \lambda(\xi) d\xi$. The number of points in the interval of observation $N(\mathcal{T})$ is under this assumption a Poisson distributed random variable with rate $\Lambda(\mathcal{T}) = \int_{\mathcal{T}} \lambda(s) ds$ and the times are assumed to be iid random variables from from a distribution with a

probability density $p(t) = \lambda(t)/\Lambda(\mathcal{T})$ for $t \in \mathcal{T}$. From exchangeability it follows that $p(\{t_i\}_{i=1}^n) = n! \prod_{i=1}^n \lambda(t_i)/\Lambda(\mathcal{T})$, as there are $n!$ different orderings of times which result in the set $\{t_i\}_{i=1}^n$. It then follows that the likelihood for λ is given by

$$\begin{aligned} p((n, \{t_i\}_{i=1}^n)|\lambda) &= p(\{t_i\}_{i=1}^n|n, \lambda)p(n|\lambda) \\ &= n! \prod_{i=1}^n \frac{\lambda(t_i)}{\Lambda(\mathcal{T})} \frac{\Lambda(\mathcal{T})^n}{n!} e^{-\Lambda(\mathcal{T})} \\ &= \prod_{i=1}^n \lambda(t_i) e^{-\int_{\mathcal{T}} \lambda(\xi) d\xi}. \end{aligned} \tag{2.1}$$

In the data augmentation strategies to follow it is useful to introduce the idea of a *marked* point process. Under an enumeration of the points each point i has an associated time t_i and mark m_i , and so a realization from a marked point process is denoted $\xi = (n, \{(t_i, m_i)\}_{i=1}^n)$. If the mark distribution for m_i is conditionally independent of $\{t_j\}_{j \neq i}$ given t_i with density $p(m_i|t_i)$ then the likelihood is simply

$$p((n, \{(t_i, m_i)\}_{i=1}^n)|\lambda) = \prod_{i=1}^n p(m_i|t_i) \lambda(t_i) e^{-\int_{\mathcal{T}} \lambda(\xi) d\xi}. \tag{2.2}$$

There are three important operations for manipulating point processes, namely, *mapping*, *thinning* and *superposition*. The former is of no importance in the applications modelled in this thesis but the latter two warrant some discussion.

2.1.2 Thinning and Superposition

Thinning provides a way to sculpt new from existing point processes. In particular suppose $\xi = (n, \{(t_i, m_i)\}_{i=1}^n)$, is a realization from an inhomogeneous marked Poisson point process with intensity $\lambda(\cdot)$ and mark distribution $m_i =$ “accept” with probability $\alpha(t_i)$ or $m_i =$ “reject” with probability $1 - \alpha(t_i)$, where $\alpha : \mathbb{R} \rightarrow [0, 1]$. Using the notation $|S|$ to denote the cardinality of the set S and a subscript i to denote the i 'th element of a tuple, let $\xi_A = (|A|, A)$ where $A = \{(t, m)_1 : (t, m) \in \{(t_i, m_i)\}_{i=1}^n, m_i =$

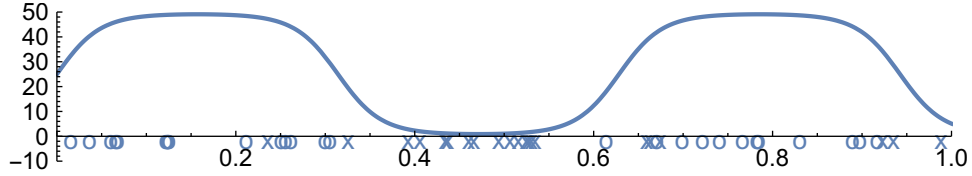


FIGURE 2.1: Thinning a homogeneous Poisson point process with intensity $\Lambda = 50$ to obtain an inhomogeneous Poisson point process with intensity $\lambda(t) = \Lambda\sigma(4\sin(10t))$, where σ is the logistic function. Points marked with “o” are accepted, whereas those marked with “x” are rejected.

“accept”} and let $\xi_R = (|R|, R)$ where $R = \{(t, m)_1 : (t, m) \in \{(t_i, m_i)\}_{i=1}^n, m_i =$ “reject”}. Then ξ_A and ξ_R are realizations from inhomogeneous Poisson point process with intensity functions $\alpha(\cdot)\lambda(\cdot)$ and $(1 - \alpha(\cdot))\lambda(\cdot)$ respectively. The earliest proof of this result is due to Lewis and Shedler (1979), but an alternative proof based on densities can be found in Rao (2012). This provides a very convenient way to generate realizations over an interval \mathcal{T} from an inhomogeneous Poisson point process with intensity satisfying $\sup\{\lambda(t), t \in \mathcal{T}\} < \infty$. Let $\Lambda = \sup\{\lambda(t), t \in \mathcal{T}\}$ and generate a realization from a homogeneous Poisson point process with constant intensity function Λ . This is achieved by drawing a Poisson random variable n with rate $\Lambda\mathcal{T}$ and then simulating n uniform random variables over \mathcal{T} . Let the latter be denoted $\{u_i\}_{i=1}^n$. Retain each $u \in \{u_i\}_{i=1}^n$ with probability $\lambda(u)/\Lambda$. The set of retained times then forms a realization from an inhomogeneous Poisson point process with rate $\lambda(\cdot)$ by thinning. This process is illustrated in figure 2.1.

Superposition on the other hand is in some sense the reverse of thinning. Given two realizations $\xi_A = (n_a, A)$ and $\xi_B = (n_b, B)$ from inhomogeneous Poisson point processes with intensity functions $\lambda_A(\cdot)$ and $\lambda_B(\cdot)$ respectively, then $\xi_{A \cup B} := (n_a + n_b, A \cup B)$ is a realization from an inhomogeneous Poisson point process with intensity function $\lambda_A(\cdot) + \lambda_B(\cdot)$. A formal proof of the superposition theorem can be found in the classic text of Kingman (1993).

2.2 The Gaussian Process Modulated Inhomogeneous Poisson Point Process

2.2.1 Gaussian Process Intensity Functions

In order to avoid making particular strong assumptions about the intensity function, such as restricting it to belong to a small parametric family, it is often preferable to model it nonparametrically as further realization from another stochastic process, one supporting a rich class of functions. Such a model is called a *doubly stochastic* Poisson process or Cox process (see Cox (1955)). If a Bayesian nonparametric approach is adopted, modelling $\lambda(t) = \exp(f(t))$ with a Gaussian process prior on f , then the resulting model is called a log-Gaussian cox process. The log-Gaussian cox process has proven to be a very popular model, most certainly due to its flexibility as the Gaussian process prior has support over a rich class of functions (see Tokdar and Ghosh (2007), Choi and Schervish (2007) and van der Vaart and van Zanten (2008)). A Gaussian process is defined in terms of its mean function $\mu(\cdot)$ and covariance kernel $K(\cdot, \cdot)$, possessing the characteristic property that if $f \sim \mathcal{GP}(\mu(\cdot), K(\cdot, \cdot))$, then for any finite number of times $t_{1:p}$ the function values follow a finite-dimensional multivariate Gaussian distribution $f_{1:p} \sim \mathcal{N}_p(\mu, K)$ where $\mu_i = \mu(t_i)$ and $K_{ij} = K(t_i, t_j)$. For further reading on Gaussian processes the reader is referred to Rasmussen and Williams (2005).

A major challenge with the doubly stochastic Poisson process is the presence of an intractable integral over an infinite dimensional random function in the likelihood. Adams et al. (2009b) overcome this integral by reintroducing the thinned realization in a data augmented model. Conditional on the observed and thinned realizations the likelihood is that of a homogeneous Poisson point process and the integral is over a constant intensity function. In order to generate the thinned realization, however, an upper bound on the intensity function is required. This is not available in the

log-Gaussian cox process where $\lambda(t) = \exp(f(t))$ and so the authors instead choose to model the intensity function as $\lambda(t) = \Lambda\sigma(f(t))$, where σ is the sigmoid or logistic function and Λ is a non-negative scaling constant. The authors name this parameterization as the sigmoidal Gaussian cox process and propose a Metropolis-Hastings based MCMC algorithm to marginalize over the number and location of the thinned times. A more efficient MCMC algorithm was later provided by Rao (2012), who demonstrated that it is possible to sample the entire thinned realization in one step within a Gibbs sampling algorithm, resulting in faster convergence due to the global nature of the updates. When sampling the function f conditional on thinned and observed realizations, Adams et al. (2009b) use Hamiltonian Monte-Carlo (Duane et al. (1987)), whereas Rao (2012) uses elliptical slice sampling (Murray et al. (2010)). The model introduced in this chapter provides two contributions. The first contribution concerns two data augmentation strategies that result in multivariate Gaussian full conditional distribution for the function f . The second uses the exact state-space representation of the Matern covariance Gaussian process in one dimension due to Hartikainen and Särkkä (2010), which enables linear-time filtering algorithms to be used for likelihood evaluations and sampling of f from its full conditional distribution.

2.2.2 Data Augmentation Strategies

The motivation of the data augmentation strategies proposed in this section is to take an unfamiliar problem and transform it into a familiar one. The current problem essentially concerns the learning of a function where the likelihood for this function is atypical. It is more common to see inference on functions being performed in applications such as nonparametric regression, where one possesses noisy observations of the function over a set of inputs, for which there already exists a large base of literature. In the inhomogeneous Poisson point process, however, information about

the intensity function is not provided by noisy observations but by the presence *or absence* of time-points. The proposed data augmentation strategies connect with existing literature on nonparametric regression and time-series by imputing noisy observations of the intensity function through the information provided by the presence or absence of time-points.

Let the intensity function be parameterized as $\lambda(t) = \Lambda\sigma(\mu + f(t))$, where Λ is a non-negative scaling constant and $\sigma : \mathbb{R} \rightarrow [0, 1]$. Let $\xi_a = (n_a, A)$, where $A = \{t_i^a\}_{i=1}^{n_a}$, denote the observed data which will be modelled as a realization from an inhomogeneous Poisson point process with rate $\lambda(t)$ and let $\xi_r = (n_r, R)$, where $R = \{t_i^r\}_{i=1}^{n_r}$, denote an unobserved realization from an inhomogeneous Poisson point process with rate $\Lambda - \lambda(t)$. The likelihood then reduces to that of a homogeneous Poisson point process

$$p(\xi_a, \xi_r | f, \Lambda, \mu) = \prod_{t \in A} \sigma(\mu + f(t)) \prod_{t \in R} (1 - \sigma(\mu + f(t))) \Lambda^{n_a + n_r} e^{-\Lambda T}. \quad (2.3)$$

With a Gaussian process prior on f , it would indeed be possible to sample the function values over $A \cup R$ via Hamiltonian Monte-Carlo or elliptical slice sampling but it is possible to get a convenient conjugate update for f via data augmentation strategies. Recall that augmenting a model of data Y_{obs} and parameters θ with missing data Y_{mis} is a valid data augmentation when $p(Y_{obs} | \theta) = \int p(Y_{obs}, Y_{mis} | \theta) dY_{mis}$. The two data augmentation approaches that are considered are the probit data augmentation of Albert and Chib (1993) and the Polya-Gamma data augmentation of Polson et al. (2013). These approaches target different link functions but both result in tractable full-conditionals for f . It is my experience that the Polya-Gamma approach results in visibly better mixing, perhaps by nature of it being a scale-mixture as opposed to a location-mixture, but at an increased computational cost of generating the Polya-Gamma random variables. It is not clear whether this pays off overall.

Probit Data Augmentation

Suppose $\sigma(\cdot) = \Phi(\cdot)$ where the latter is the standard normal cumulative distribution function. I shall denote this model the probit Gaussian Cox process. The data generating process for ξ_a and ξ_r can be viewed as thinning a homogeneous Poisson point process with rate Λ by assigning a time-point t to ξ_a with probability $\lambda(t)/\Lambda = \Phi(\mu + f(t), 1)$, otherwise assigning it to ξ_r . As noted by Albert and Chib (1993) this is probabilistically equivalent to drawing a random variable $m \sim \mathcal{N}(\mu + f(t), 1)$ and assigning t to ξ_a when $m > 0$, otherwise assigning it to ξ_r . To see this observe that $\Phi(\mu + f(t), 1) = \int 1_{[0, \infty]}(m) \mathcal{N}(m | \mu + f(t), 1) dm$. By Bayes theorem it follows that the distribution of m conditional on t is $\mathcal{N}_{[0, \infty]}(\mu + f(t), 1)$ if t were assigned to ξ_a and $\mathcal{N}_{(-\infty, 0]}(\mu + f(t), 1)$ otherwise, where the subscripted interval denotes an interval of truncation. It is useful to reintroduce these m 's back into the model in the form of *unobserved* marks.

Suppose that ξ_a and ξ_r are modelled as *marked* inhomogeneous Poisson point processes with mark distributions $\mathcal{N}_{[0, \infty]}(\mu + f(t), 1)$ and $\mathcal{N}_{(-\infty, 0]}(\mu + f(t), 1)$ respectively. The likelihood is then

$$\begin{aligned}
 p(\xi_a, \xi_r | f, \Lambda, \mu) &= \prod_{t \in \{t_i^a\}_{i=1}^{n_a}} \Phi(\mu + f(t)) \prod_{t \in \{t_i^r\}_{i=1}^{n_r}} (1 - \Phi(\mu + f(t))) \Lambda^{n_a + n_r} e^{-\Lambda T} \\
 &\quad \prod_{(t, m) \in A} \frac{\mathcal{N}(m | \mu + f(t), 1)}{\Phi(\mu + f(t))} 1_{[0, \infty]}(m) \\
 &\quad \prod_{(t, m) \in B} \frac{\mathcal{N}(m | \mu + f(t), 1)}{1 - \Phi(\mu + f(t))} 1_{(-\infty, 0]}(m).
 \end{aligned} \tag{2.4}$$

The convenient property of this data augmented model is that $\Phi(\mu + f(t))$ and $1 - \Phi(\mu + f(t))$ terms cancel with the normalizing constants of the mark distributions, resulting in a full conditional for f given by

$$p(f | -) \propto \prod_{(t, m) \in A} \mathcal{N}(m | \mu + f(t), 1) \prod_{(t, m) \in B} \mathcal{N}(m | \mu + f(t), 1) p(f). \tag{2.5}$$

Under a Gaussian process prior for f , the full conditional distribution over values $\{t_i^a\}_{i=1}^{n_a} \cup \{t_i^r\}_{i=1}^{n_r}$ is a tractable multivariate normal. It is intuitive to regard each mark m_i as providing a noisy measurement or observation of $\mu + f$ at the time t_i with unit variance. The full conditionals for the unobserved marks are simply their mark distributions. The tractability of all full conditionals presents the opportunity of developing a Gibbs sampling algorithm to generate draws of f , μ and Λ from the posterior distribution. An alternative data augmentation strategy is also available when σ is the logistic function.

Polya-Gamma Data Augmentation

Suppose $\sigma(\cdot)$ is the logistic function as in the sigmoidal Gaussian cox process. Theorem 1 of Polson et al. (2013) states that

$$\sigma(x) = \frac{e^x}{1 + e^x} = \frac{1}{2} e^{-x/2} \int_0^\infty e^{-\omega x^2/2} p(\omega) d\omega, \quad (2.6)$$

where $p(\omega)$ is the probability density function of a $\mathcal{PG}(1, 0)$ (Polya-Gamma) random variable. Moreover the conditional distribution

$$p(\omega|x) = \frac{e^{-\omega x^2/2} p(\omega)}{\int_0^\infty e^{-\omega x^2/2} p(\omega) d\omega}, \quad (2.7)$$

obtained by treating the integrand in as an unnormalized joint density in (x, ω) , is a $\mathcal{PG}(1, x)$ random variable. Similarly

$$1 - \sigma(x) = \frac{1}{1 + e^x} = \frac{1}{2} e^{-x/2} \int_0^\infty e^{-\omega x^2/2} p(\omega) d\omega. \quad (2.8)$$

As with the probit data augmentation approach, consider modelling ξ_a and ξ_r as marked inhomogeneous Poisson point processes where the mark distributions are

$\mathcal{PG}(1, \mu + f(t))$. The likelihood then becomes

$$\begin{aligned}
p(\xi_a, \xi_r | f, \Lambda, \mu) &= \prod_{t \in \{t_i^a\}_{i=1}^{n_a}} \frac{1}{2} e^{(\mu+f(t))/2} \int_0^\infty e^{-\omega(\mu+f(t))^2/2} p(\omega) d\omega, \\
&\prod_{t \in \{t_i^r\}_{i=1}^{n_r}} \frac{1}{2} e^{-(\mu+f(t))/2} \int_0^\infty e^{-\omega(\mu+f(t))^2/2} p(\omega) d\omega \\
&\prod_{(t,m) \in A} \frac{e^{-m(\mu+f(t))^2/2} p(m)}{\int_0^\infty e^{-m(\mu+f(t))^2/2} p(m) dm}, \\
&\prod_{(t,m) \in R} \frac{e^{-m(\mu+f(t))^2/2} p(m)}{\int_0^\infty e^{-m(\mu+f(t))^2/2} p(m) dm}, \\
&\Lambda^{n_a+n_r} e^{-\Lambda T}
\end{aligned} \tag{2.9}$$

where $p(m)$ is the density of a $\mathcal{PG}(1, 0)$ in m . As before with the probit data augmentation, a nice property about the Polya-Gamma augmented model is that the integrals in the logistic link function cancel with the normalizing constants of the mark distributions, resulting in the full conditional for f given by

$$p(f | -) \propto \prod_{(t,m) \in A} e^{-\frac{m}{2} (\frac{1}{2m} - \mu - f(t))^2} \prod_{(t,m) \in R} e^{-\frac{m}{2} ((-\frac{1}{2m}) - \mu - f(t))^2} p(f), \tag{2.10}$$

resulting once again a multivariate normal full conditional distribution for f values over $\{t_i^a\}_{i=1}^{n_a} \cup \{t_i^r\}_{i=1}^{n_r}$ under a Gaussian process prior assumption. For each pair $(t, m) \in A$ it is intuitive to regard $1/(2m)$ as providing a noisy measurement or observation of $\mu + f$ at time t with observation variance $1/m$. Similarly for each pair $(t, m) \in R$ it is intuitive to regard $-1/(2m)$ as providing a noisy measurement or observation of $\mu + f$ at time t with measurement variance $1/m$. It will later provide simplification and aid intuition to define new quantities $\{y_i\}_{i=1}^{n_a+n_r}$ and $\{\sigma_i^2\}_{i=1}^{n_a+n_r}$ in terms of the marks $\{m_i^a\}_{i=1}^{n_a} \cup \{m_i^r\}_{i=1}^{n_r}$ and define full conditionals with respect to these newly defined quantities instead of the original marks. The full conditional distributions for the marks are simply $\mathcal{PG}(1, \mu + f(t))$ random variables.

2.2.3 State-Space Representation of Gaussian Processes

The data augmentation strategies outlined in the previous section result in a tractable multivariate Gaussian full-conditional distribution for f evaluated at times in $\{t_i^a\}_{i=1}^{n_a} \cup \{t_i^r\}_{i=1}^{n_r}$. As a result, the computational complexity of sampling f from its full-conditional distribution is $\mathcal{O}((n_a + n_r)^3)$. The computational burden is the price one commonly pays for working with Gaussian processes. To address this issue a number of approximations have been introduced in the literature, some are numerical, some are mathematical. The former usually deal with numerical approximations to the linear algebra computations required in the multivariate normal. Popular approaches are the low-rank Cholesky decomposition, the randomized singular value decomposition of Halko et al. (2011) or approximate factorizations within a Krylov subspace Chow and Saad (2014).

The latter concerns developing mathematical models that approximate the original. Approximations of this ilk include the predictive process of Banerjee et al. (2008) and nearest neighbour approaches of Datta et al. (2016) to name a few. All of these methods are general purpose in the sense that covariance kernel used does not really matter. There are, however, particular kinds of Gaussian process that possess specific favourable properties. Take the exponential covariance kernel for instance $K(x, y) = \rho^2 \exp(-|x - y|/\tau)$. Sample paths of this GP are nowhere differentiable but the precision matrix is sparse, betraying conditional independence properties of the function values. Indeed this GP can be constructed from a continuous-time AR(1) model, possessing the Markov conditional independence property. To see this let $f_i = f(t_i)$ and consider the state equation

$$f_i = \phi_i f_{i-1} + \epsilon_i, \tag{2.11}$$

where $\phi_i = \exp(-|t_i - t_{i-1}|/\tau)$ and $\epsilon_i \sim \mathcal{N}(0, \rho^2(1 - \exp(-|t_i - t_{i-1}|/\tau)^2))$. A little algebra together with $f_1 \sim \mathcal{N}(0, \rho^2)$ shows that $f_{1:n} \sim \mathcal{N}(0, K)$ with $K_{ij} =$

$\rho^2 \exp(-|t_i - t_j|/\tau)$). In this special case, therefore, a state-space model gives rise to a Gaussian process. This means that instead of using the general $\mathcal{O}(n^3)$ machinery for GPs, inference can be performed in $\mathcal{O}(n)$ complexity using the Kalman filter, a specialized algorithm that exploits the Markov property.

Depending on the application, one might desire to have smooth sample paths. A common choice is with the squared exponential covariance kernel $K(x, y) = \rho^2 \exp(-(x - y)^2/\tau^2)$, which is a limiting case of the Matern covariance kernel

$$K_\nu(x, y) = \rho^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{|x - y|}{\tau} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{|x - y|}{\tau} \right), \quad (2.12)$$

in the limit $\nu \rightarrow \infty$, where Γ is the gamma function and K_ν is the modified Bessel function of the second kind. The exponential covariance kernel is also a special case of the Matern with $\nu = 1/2$. A natural question to ask is whether there exists a happy medium somewhere between $\nu = 1/2$ and $\nu = \infty$. Sample paths of 1-dimensional GPs with the Matern- ν covariance kernel are $\lceil \nu \rceil - 1$ times differentiable and, using the results of Hartikainen and Särkkä (2010), have exact $(\nu + 1/2)$ -dimensional state-space representations. I shall use a $\nu = 5/2$ Matern covariance kernel, providing sample paths that are 2-times differentiable, but also because the state-space has dimension 3 as this allows highly optimized linear algebra routines to be used. It is the case that for small fixed size matrices there exist highly optimized linear algebra routines compared to those that handle matrices of arbitrary size. This is because many operations on 3 dimensional matrices admit closed form expressions, and also because 3 dimensional matrices have received a lot of attention computationally due to the large number of physical systems that are simulated in 3-dimensional Euclidean space.

Let $F_i = (f(t_i), f'(t_i), f''(t_i))^T$ and consider following stochastic differential equa-

tion

$$\frac{dF(t)}{dt} = AF(t) + Lw(t),$$

$$\text{where } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3\lambda^3 & -3\lambda^2 & -3\lambda \end{bmatrix}, \quad L = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (2.13)$$

$\lambda = \sqrt{5}/\tau$ and $w(t)$ is a white noise process with spectral density $q = (16/3)\rho^2\lambda^5$. The observation of Hartikainen and Särkkä (2010) is that the first element of the solution to the equation (2.13) is, probabilistically, a zero-mean Gaussian process with Matern $\nu = 5/2$ covariance kernel. The continuous time model in equation (2.13) can be converted to a discrete time model on a finite number of times $t_{1:n}$ via

$$F_i = \Phi_i F_{i-1} + \epsilon_i, \quad (2.14a)$$

$$F_1 \sim \mathcal{N}(0, Q_\infty), \quad (2.14b)$$

$$\text{where } \Phi_i = \exp(A\Delta_i), \quad (2.14c)$$

$$\Delta_i = t_i - t_{i-1}, \quad (2.14d)$$

$$\epsilon_i \sim \mathcal{N}(0, Q_i), \quad (2.14e)$$

$$Q_i = \int_0^{\Delta_i} \exp(A\tau) L q L^T \exp(A\tau)^T d\tau, \quad (2.14f)$$

$$Q_\infty = \int_0^\infty \exp(A\tau) L q L^T \exp(A\tau)^T d\tau. \quad (2.14g)$$

The reader can find more details on these steps in (Bar-Shalom et al., 2002, Chapter 4). The matrices F_i , Q_i and Q_∞ admit closed form mathematical expressions as

$$\Phi_i = e^{-\Delta_i \lambda} \begin{pmatrix} \left(\frac{\Delta_i^2 \lambda^2}{2} + \Delta_i \lambda + 1 \right) & (\lambda \Delta_i^2 + \Delta_i) & \frac{1}{2} \Delta_i^2 \\ -\frac{1}{2} \Delta_i^2 \lambda^3 & (-\Delta_i^2 \lambda^2 + \Delta_i \lambda + 1) & \left(\Delta_i - \frac{\Delta_i^2 \lambda}{2} \right) \\ \left(\frac{\Delta_i^2 \lambda^4}{2} - \Delta_i \lambda^3 \right) & (\Delta_i^2 \lambda^3 - 3\Delta_i \lambda^2) & \left(\frac{\Delta_i^2 \lambda^2}{2} - 2\Delta_i \lambda + 1 \right) \end{pmatrix}, \quad (2.15)$$

$$Q_\infty = \begin{pmatrix} \rho^2 & 0 & -\frac{1}{3}\lambda^2\rho^2 \\ 0 & \frac{\lambda^2\rho^2}{3} & 0 \\ -\frac{1}{3}\lambda^2\rho^2 & 0 & \lambda^4\rho^2 \end{pmatrix}, \quad (2.16)$$

although the expression for Q_i is a bit too cumbersome to state here.

With the complicated details now stated, it is in fact very simple to validate this construction by showing that the correct marginal distribution for $f_{1:n}$ is recovered, considering $f'_{1:n}$ and $f''_{1:n}$ to be augmented data. From $F_1 \sim \mathcal{N}(0, Q_\infty)$ and looking at $Q_{\infty,11} = \rho^2$ it is clear that the marginal distribution for f_1 is $\mathcal{N}(0, \rho^2)$. From properties of the stationary covariance matrix it follows that $\Phi_i Q_\infty \Phi_i^T + Q_i = Q_\infty$ and so by a similar argument the marginal distribution for f_i is also $\mathcal{N}(0, \rho^2)$ for $i \in \{1, \dots, n\}$. The covariance between F_i and F_{i-j} is given by $\Phi_i \Phi_{i-1} \dots \Phi_{i-j+1} Q_\infty = \exp(A(t_i - t_{i-j})) Q_\infty$. The element in row 1 column 1 of $\exp(A(t_i - t_{i-j})) Q_\infty$ is identically

$$\rho^2 \left(1 + \lambda(t_i - t_{i-j}) + \frac{\lambda}{3}(t_i - t_{i-j})^2 \right) e^{-\lambda(t_i - t_{i-j})}, \quad (2.17)$$

which is exactly the covariance defined by the Matern $\nu = 5/2$ covariance kernel. This shows that the state-space construction gives rise to the correct marginal distribution for $f_{1:n}$ and that adding $f'_{1:n}$ and $f''_{1:n}$ constitutes a valid data augmentation.

2.2.4 Forward Filtering Backward Sampling

In general sampling from the posterior resulting from a Gaussian process prior, or evaluating the marginal likelihood of the data are operations with cubic complexity. The state-space construction makes possible the use of the *forward filtering backwards sampling* algorithm, for which the complexity is only linear. The Polya-Gamma and Probit data augmentations can be regarded as imputing noisy observations of the function F over time. Consider the following state-space model

$$y_i = \mu + HF_i + \nu_i \quad (\text{Observation Equation}), \quad (2.18)$$

$$F_i = \Phi_i F_{i-1} + \epsilon_i \quad (\text{State Equation}), \quad (2.19)$$

where $H = (1, 0, 0)$, $\nu \sim \mathcal{N}(0, \sigma_i^2)$, Φ_i and ϵ_i as defined before.

Ignoring all hyperparameters for simplicity, the posterior distribution for $F_{1:n}$ can be factored in compositional form as

$$p(F_{1:n}|y_{1:n}) = p(F_n|y_{1:n}) \prod_{i=1}^{n-1} p(F_i|F_{i+1:n}, y_{1:n}). \quad (2.20)$$

Sampling $F_{1:n}$ could then be sampled compositionally from the factored distributions in (2.21). Looking at these factors more closely reveals through the observation

$$p(F_i|F_{i+1:n}, y_{1:n}) \propto p(F_{i+1}|F_i)p(y_i|F_i)p(F_i|y_{1:i}), \quad (2.21)$$

that $p(F_i|F_{i+1:n}, y_{1:n}) = p(F_i|F_{i+1}, y_{1:i})$ i.e. F_i is conditionally independent of $F_{i+2:n}$ given F_{i+1} and $y_{1:i}$. Forward filtering constructs in one pass through the data the conditional distributions $p(F_i|F_{i+1}, y_{1:i})$, the product of which define the joint posterior. The backward sampling step, initialized by sampling $F_n \sim p(F_n|y_{1:n})$, samples in one pass the F_i 's from their conditional distributions $p(F_i|F_{i+1}, y_{1:i})$.

Forward Filtering

Suppose it is known that $F_i|y_{1:i} \sim \mathcal{N}(m_i, M_i)$, then

$$\begin{pmatrix} y_{i+1} \\ F_{i+1} \\ F_i \end{pmatrix} | y_{1:i} \sim \mathcal{N} \left(\begin{pmatrix} \mu + H\Phi_i m_i \\ \Phi_{i+1} m_i \\ m_i \end{pmatrix}, O_i \right), \quad (2.22)$$

where

$$O_i = \begin{pmatrix} \sigma_{i+1}^2 + H(\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1})H^T & H(\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1}) & H\Phi_{i+1}M_i \\ \Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1})H^T & \Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1} & \Phi_{i+1}M_i \\ M_i\Phi_{i+1}H^T & M_i\Phi_{i+1}^T & M_i \end{pmatrix}. \quad (2.23)$$

The new m_{i+1} and M_{i+1} , in $F_{i+1}|y_{1:i+1} \sim \mathcal{N}(m_{i+1}, M_{i+1})$, are then derived from multivariate normal theory as

$$\begin{aligned} m_{i+1} &= \Phi_{i+1}m_i + \\ & (\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1})H^T [\sigma_{i+1}^2 + H(\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1})H^T]^{-1} (y_{i+1} - (\mu + H\Phi_{i+1}m_i)) \end{aligned} \quad (2.24)$$

and

$$\begin{aligned}
M_{i+1} &= \Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1} - \\
&(\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1})H^T[\sigma_{i+1}^2 + H(\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1})H^T]^{-1}H(\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1}).
\end{aligned} \tag{2.25}$$

Some computation can be saved by recognizing that multiplying by $H = (1, 0, 0)$ merely corresponding to taking a slice of a matrix. This recursion definition is initialized with the base case

$$m_1 = Q_\infty H^T [\sigma_1^2 + H Q_\infty H^T]^{-1} (y_1 - \mu), \tag{2.26}$$

$$M_1 = Q_\infty - Q_\infty H^T [\sigma^2 + H Q_\infty H^T]^{-1} H Q_\infty. \tag{2.27}$$

Backwards Sampling

The conditional distributions can then be obtained from (2.22) using multivariate normal theory, giving

$$\begin{aligned}
F_i | F_{i+1}, y_{1:i} &\sim \mathcal{N}(m_i + M_i \Phi_{i+1}^T [\Phi_{i+1} M_i \Phi_{i+1}^T + Q_{i+1}]^{-1} (F_{i+1} - \Phi_{i+1} m_i) \\
M_i - M_i \Phi_{i+1}^T &[\Phi_{i+1} M_i \Phi_{i+1}^T + Q_{i+1}]^{-1} \Phi_{i+1} M_i).
\end{aligned} \tag{2.28}$$

Starting with a draw of F_n from $F_n | y_{1:n} \sim \mathcal{N}(m_n, M_n)$, function values F_i can be sampled given F_{i+1} in linear time via (2.28).

Marginal Likelihood Evaluation

The Gaussian process prior has associated hyperparameters of characteristic time-scale τ and variance ρ^2 , upon which one would like to place priors. At times it may be useful to evaluate the marginal likelihood, $p(y_{1:n} | \mu, \rho^2, \tau)$, having integrated out $f_{1:n}$. This too can be written in compositional form as $p(y_1 | \mu, \rho^2, \tau) \prod_{i=1}^{n-1} p(y_{i+1} | y_{1:i}, \mu, \rho^2, \tau)$ and conveniently these factors are produced as a byproduct of the forward-filtering stage as

$$\begin{aligned}
y_{i+1} | y_{1:i}, \mu, \rho^2, \tau &\sim \mathcal{N}(\mu + H \Phi_i m_i, \sigma_{i+1}^2 + H(\Phi_{i+1} M_i \Phi_{i+1}^T + Q_{i+1}) H^T), \\
y_1 | \mu, \rho^2, \tau &\sim \mathcal{N}(\mu, \sigma_1^2 + H Q_\infty H^T).
\end{aligned} \tag{2.29}$$

This means that evaluating the marginal likelihood has complexity $\mathcal{O}(n)$ instead of $\mathcal{O}(n^3)$. In addition a nice unit test to check for the validity of the mathematics and code is to compare the numerical value of the marginal likelihood computed in the manner described previously with the value computed by evaluating $\mathcal{N}_n(Y|\mu, \Sigma + K)$, where $\Sigma = \text{Diag}(\sigma_1^2, \dots, \sigma_n^2)$.

Marginal Posterior for μ

First, a note on model parameterization. Note that the mean parameter μ which occurs in the observation equation could have equally been placed in the state equation. Although the posterior is unaffected, different model parameterizations lead to Markov chains with different mixing properties. A discussion of this can be found in (Prado and West, 2010, Chapter 7). Combining the two different parameterizations would be a neat application of the ancillary-sufficient interweaving strategy Yu and Meng (2011). Instead, the approach adopted here is to sample (F, μ) as a block by first sampling $\mu|y_{1:n}, \rho^2, \tau$ and then $F_{1:n}|\mu, y_{1:n}, \rho^2, \tau$. The conditional distribution for μ is available in closed form. To derive this it is necessary to reformulate the forward filtering steps a little because currently each m_i is an expression that depends on μ . Suppose m_i is of the form $m_i = C_i + D_i\mu$, then

$$m_{i+1} = C_{i+1} + D_{i+1}\mu_i$$

$$\text{where } C_{i+1} = (I - F_{i+1}H)\Phi_{i+1}C_i + F_{i+1}y_{i+1}, \tag{2.30}$$

$$D_{i+1} = (I - F_{i+1}H)\Phi_{i+1}D_i - F_{i+1},$$

$$F_{i+1} = (\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1})H^T[\sigma_{i+1}^2 + H(\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1})H^T]^{-1}.$$

This is clearly the case for m_1 with $C_1 = Q_\infty H^T[\sigma_1^2 + HQ_\infty H^T]^{-1}y_1$ and $D_1 = -Q_\infty H^T[\sigma_1^2 + HQ_\infty H^T]^{-1}$ and so it is true for all $i \in \{1, \dots, n\}$ by induction. Suppose the prior distribution on μ is $\mathcal{N}(\tilde{\mu}, \sigma_\mu^2)$, then the conditional distribution

$\mu|\xi_a, \xi_r, \rho^2, \tau \sim \mathcal{N}(c, v^2)$ where

$$v^2 = \left(\sum_{i=1}^{n-1} \frac{H\Phi_{i+1}D_i}{\sigma_{i+1}^2 + H(\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1})H^T} + \frac{1}{\sigma_1^2 + HQ_\infty H^T} + \frac{1}{\sigma_\mu^2} \right)^{-1} \quad (2.31)$$

$$c = v^2 \left(\sum_{i=1}^{n-1} \frac{H\Phi_{i+1}D_i}{\sigma_{i+1}^2 + H(\Phi_{i+1}M_i\Phi_{i+1}^T + Q_{i+1})H^T} \left(\frac{y_{i+1} - H\Phi_{i+1}C_i}{H\Phi_{i+1}D_i} \right) + \frac{y_1}{\sigma_1^2 + HQ_\infty H^T} + \frac{\tilde{\mu}}{\sigma_\mu^2} \right).$$

The mathematics and code can be easily unit tested by comparing numerical values of c and v^2 with their $\mathcal{O}(n^3)$ counterparts of $v^2 = (1/\sigma_\mu^2 + 1^T(\Sigma + K)^{-1}1)^{-1}$ and $c = v^2(1^T(\Sigma + K)^{-1}Y + \tilde{\mu}/\sigma_\mu^2)$.

2.3 Full Model Specification

It is time to give a complete description to the proposed model for modelling arrival times. Suppressing the augmentation of f' and f'' , the model reads

$$\xi_a|\Lambda, f, \mu \sim \mathcal{PPP}(\Lambda\sigma(\mu + f(\cdot)), \mathcal{D}_a),$$

$$\xi_r|\Lambda, f, \mu \sim \mathcal{PPP}(\Lambda - \Lambda\sigma(\mu + f(\cdot)), \mathcal{D}_r),$$

$$f|\rho^2, \tau \sim \mathcal{GP}(0, K_{\rho^2, \tau}^{\nu=5/2}(\cdot, \cdot)),$$

$$\mu \sim \mathcal{N}(0, \sigma_\mu^2), \quad (2.32)$$

$$\Lambda \sim Ga(l_1, l_2),$$

$$\rho^2 \sim IG\left(\frac{r_1}{2}, \frac{r_2}{2}\right),$$

$$\tau \sim Ga(t_1, t_2),$$

where \mathcal{D}_a and \mathcal{D}_r are the appropriate mark distributions depending on which link function is used. $\xi_a = (n_a, \{t_i^a, m_i^a\}_{i=1}^{n_a})$ where t_i^a and m_i^a are the time and mark for time-point i in the point pattern ξ_a . Similarly $\xi_r = (n_r, \{t_i^r, m_i^r\}_{i=1}^{n_r})$ where t_i^r and m_i^r are the time and mark for point i in the point pattern ξ_r . When σ is the probit function, \mathcal{D}_a is $\mathcal{N}_{[0, \infty)}(\mu + f(\cdot), 1)$ and \mathcal{D}_r is $\mathcal{N}_{(-\infty, 0]}(\mu + f(\cdot), 1)$. When σ is the logistic function, both \mathcal{D}_a and \mathcal{D}_r are $\mathcal{PG}(1, \mu + f(\cdot))$ distributions.

As alluded to earlier it is intuitive to regard the pairs of times and marks contained in ξ_a and ξ_r as providing noisy observations of $\mu + f$ over time. At present one has an enumeration of the points in ξ_a and an enumeration of the points in ξ_r . It is more convenient when stating the full conditional distributions to have an enumeration of the union of these points and, for the purposes of the state-space representation, it is helpful to choose an enumeration such that the time points are ordered. For this reason it is helpful to define three new quantities $t_{1:(n_a+n_r)}$, $y_{1:(n_a+n_r)}$ and $\sigma_{1:(n_a+n_r)}^2$, and specify full conditional distributions in terms of these instead of the original marks. It is also sometimes helpful to use these quantities to define $\Sigma = \text{Diag}(\sigma_1^2, \dots, \sigma_{n_a+n_r}^2)$ and K - the covariance matrix resulting from the Matern covariance kernel.

In the case when σ is the logistic function, define two new sets $\mathcal{Y}^a = \{(t_i^a, 1/(2m_i^a), 1/m_i^a)\}_{i=1}^{n_a}$ and $\mathcal{Y}^r = \{t_i^r, -1/(2m_i^r), 1/m_i^r\}$. The first element of the 3-tuple is once again a time, the second element corresponds to a noisy observation and the third element corresponds to an observation variance. Let $\mathcal{Y} = \mathcal{Y}^a \cup \mathcal{Y}^r$ and let $\mathcal{Y}_{1:n_a+n_b}$ denote an enumeration of the elements of this set such that the times are ordered i.e. $\mathcal{Y}_{1,1} \leq \mathcal{Y}_{2,1} \leq \dots \leq \mathcal{Y}_{(n_a+n_b),1}$. The noisy observations $y_{1:(n_a+n_b)}$ can then be defined as $\mathcal{Y}_{1:(n_a+n_b),2}$, with observation variances $\sigma_{1:(n_a+n_b)}^2$ defined as $\mathcal{Y}_{1:(n_a+n_b),3}$, occurring at times $t_{1:n_a+n_r}$ defined as $\mathcal{Y}_{1:(n_a+n_b),1}$.

The case when σ is the probit function is considerably easier. It is the same as the previous definition except $\mathcal{Y}^a = \{(t_i^a, m_i^a, 1)\}_{i=1}^{n_a}$ and $\mathcal{Y}^r = \{t_i^r, m_i^r, 1\}$.

2.4 Gibbs Sampling

A cyclical Gibbs sampler can be constructed by iteratively sampling the following conditional distributions.

$\xi_r | \Lambda, f, \mu$

ξ_r is a new independent realization of a marked inhomogeneous Poisson point process with rate $\Lambda - \Lambda\sigma(\mu + f(\cdot))$ and mark distribution \mathcal{D}_r . When σ is the probit function, $m_i^a | t_i^a \sim \mathcal{N}_{(-\infty, 0]}(\mu + f(t_i^a), 1)$ and when σ is the logistic function $m_i^a | t_i^a \sim \mathcal{PG}(\mu + f(t_i^a))$

$\{m_i^a\}_{i=1}^{n_a} | \{t_i^a\}_{i=1}^{n_a}, f, \mu$

The unobserved marks of ξ_a are simulated from the mark distribution D_a . When σ is the probit function, $m_i^a | t_i^a \sim \mathcal{N}_{[0, \infty)}(\mu + f(t_i^a), 1, 0)$ and when σ is the logistic function $m_i^a | t_i^a \sim \mathcal{PG}(\mu + f(t_i^a))$.

$\tau | \xi_a, \xi_r, \mu, \rho^2$

$$p(\tau | \xi_a, \xi_r, \mu, \rho^2) \propto p(\xi_a, \xi_r | \mu, \rho^2, \tau) p(\tau) \quad (2.33)$$

$$\propto \mathcal{N}(y_{1:n_a+n_r} | 1\mu, \Sigma + K) p(\tau) \quad (2.34)$$

There is no closed form mathematical express for the conditional distribution for τ . As this is 1-dimensional, however, it is easy to complete via slice-sampling (see Neal (2003)) which only requires the ability to evaluate the distribution up to a constant of proportionality. The marginal likelihood $p(\xi_a, \xi_r | \mu, \tau, \rho^2)$ is easily evaluated in linear time by running the forward filtering steps described in section 2.2.4 on \mathcal{Y} .

$\rho^2 | \xi_a, \xi_r, \mu, \tau$

This step, just as for τ , is completed via slice-sampling.

$(F, \mu) | \xi_a, \xi_r, \rho^2, \tau$

First μ is drawn from $\mu | \xi_a, \xi_r, \rho^2, \tau$ via the method described in section 2.2.4 on the \mathcal{Y} . Conditional on the new value of μ , F is sampled via forwards filtering backwards sampling on \mathcal{Y} .

$\Lambda|\xi_a, \xi_r$

The $Ga(l, l\Lambda_0)$ prior is conjugate for the homogeneous Poisson point process with rate Λ . It follows that

$$\Lambda|-\sim Ga(n_a + n_r + l_1, \mathcal{T} + l_2) \quad (2.35)$$

Comments and Alternatives

The reader may wonder how it is justified to sample from $\rho^2|\xi_a, \xi_r, \mu, \tau$ instead of $\rho^2|\xi_a, \xi_r, F, \mu, \tau$ i.e. without conditioning on F . Similarly for the τ step. The above steps are best viewed as sampling blocks $(F, \tau^2)|-$, $(F, \rho^2)|-$ and $(F, \mu)|-$. It is simply the case that drawing from $F|\tau, -$ and $F|\rho^2, -$ is unnecessary because the next step draws from a distribution that does not condition on F .

There is only one tuning parameter in the proposed MCMC algorithm, namely, the length of the window in the “stepping-out” procedure of slice sampling. A poorly tuned length may result in many likelihood evaluations for sampling the full conditionals of ρ^2 and τ . Although likelihood evaluations under this model are inexpensive, there are alternative MCMC approaches for which this issue does not arise. This can be resolved by moving to a discrete prior on τ and sampling ρ^2 from $\rho^2|F, \tau$ instead of $\rho^2|\tau, \xi_a, \xi_r, \mu$. The full conditional distribution $\rho^2|F, \tau$ is a tractable inverse-gamma distribution. It is an open question whether the MCMC exhibits better mixing when sampling ρ^2 from $\rho^2|F, \tau$ or from $\rho^2|\tau, \mu, \xi_a, \xi_r$ via slice-sampling. In addition, moving to a discrete prior on τ loses very little in terms of model flexibility. What is gained is that the sampling step for τ becomes easy as $\tau|\rho^2, \mu, \xi_a, \xi_r$ is also a discrete distribution, requiring a fixed number of likelihood evaluations which can be computed if desired in parallel.

2.5 Illustrations

2.5.1 Simulated Data

A single realization from an inhomogeneous Poisson point process with rate $\lambda(t) = 400\sigma(-1 + 3\sin(2\pi 3t))$ is observed. The prior on μ is $\mathcal{N}(0, 1/4)$ so that with approximately 0.95 prior probability μ is in the interval $[-1, 1]$. The prior on ρ^2 is $Ga_{[0,4]}(4/2, 1/2)$ which gives approximately 0.9 prior probability that $\rho^2 < 1$. In general it is a good idea to constrain μ and the GP variance ρ^2 because μ and f are mapped through the link function, so there is no reason for these terms to be excessively large. Without informative priors for these parameters, the posterior could potentially support very large values. The prior on Λ non-informative is $Ga(2, l_2)$ with l_2 chosen such that with 0.95 prior probability $\Lambda < 400$. The prior on τ is $Ga(2, t_2)$ such that with 0.95 prior probability $\tau < 1$. A shape parameter of 2 is used instead of 1 because concentrating prior mass at zero for the time-scale can lead to over-fitting. The posterior summaries of all parameters are shown in figure A.1. The posterior summaries show very good mixing properties of the Markov chain and good learning of the parameters μ and Λ . Posterior credible intervals of $\lambda(t)$ are shown in figure 2.2, showing that the intensity function is also learned very well.

2.5.2 Coal Mining Disasters

This dataset records in continuous time a sequence of 191 coal mining explosions resulting in 10 or more fatalities over a period starting from March 15 1851 until March 22 1962. Since its introduction in Maguire et al. (1952) and subsequent correction by Jarrett (1979), it has become a popular test dataset in the point process literature and has been studied in the works of Adams et al. (2009b) and Rao (2012). The data is translated and scaled so that times occur on the unit interval. The priors used in the analysis are the same as in the previous section. Trace-plots displaying

excellent mixing and posterior summaries are shown in figure A.2. The estimated intensity function is shown in figure 2.3, which corroborates the analysis in previous works.

2.6 Discussion

The previous sections described a continuous-time Bayesian model for arrival times. Based on the generative idea of thinning a homogeneous Poisson point process, the intractable likelihood of the inhomogeneous Poisson point process is circumvented. Parameterizing the intensity function with the logistic or probit link function and augmenting the observed and thinned points to have Polya-Gamma or truncated Gaussian marks connects this model with the larger literature on nonparametric regression and time-series, allowing great flexibility in how the intensity function is modelled over time. The approach proposed in this chapter has been to model the linear term of the intensity function as a particular Gaussian process which admits an exact state-space representation in one dimension. This circumvents the cubic complexity usually associated with Gaussian processes and allows draws from the posterior and evaluations of the marginal likelihood to be computed using filtering algorithms with computationally complexity that is linear in the number of time-points. The blocked Gibbs sampler exhibits excellent mixing and scales linearly in the problem size.

Before this chapter closes it is necessary to discuss some further details that have so far been postponed. Most importantly, the conditional distributions under the state-space representation required for predicting the intensity function at arbitrary new time points, as is required during the thinning process. Necessarily, the appropriate datastructure for storing realized function values that makes prediction at new time points performant by exploiting the temporal nature of the data. Most interestingly, from a computer science perspective, a novel implementation of Gaussian

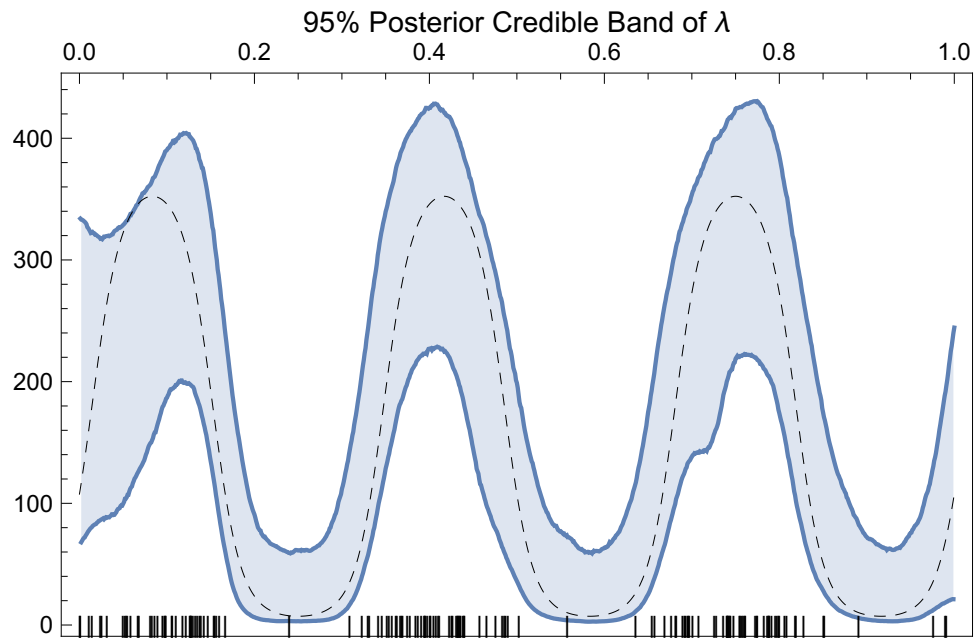


FIGURE 2.2: 95% Posterior credible band for the intensity function. The true intensity function $\lambda(t) = 400\sigma(-1 + 3 \sin(2\pi 3t))$ is shown by the dashed black line. The observed data is shown by the black ticks on the x-axis.

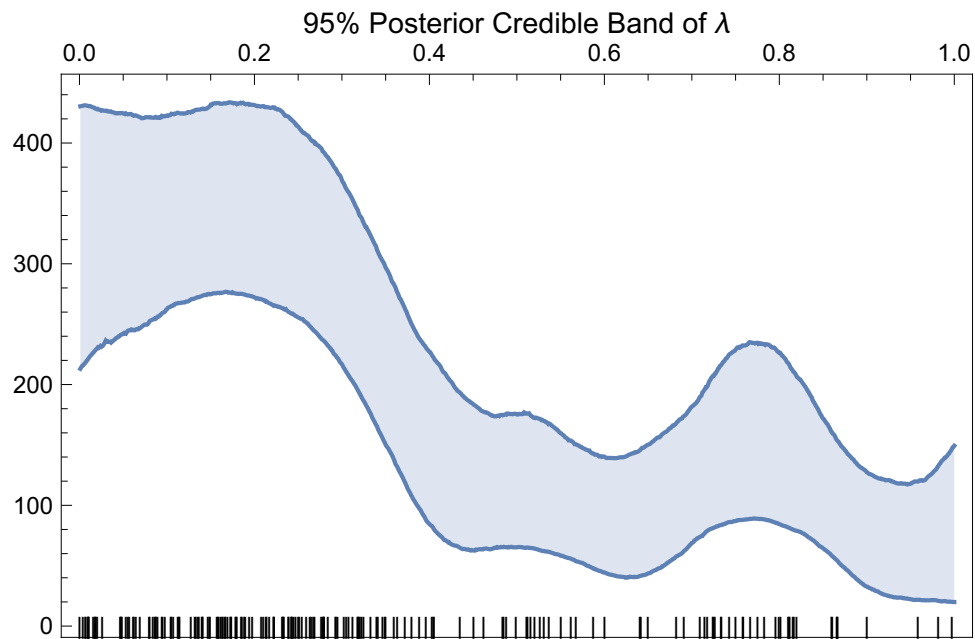


FIGURE 2.3: 95% Posterior credible band for the intensity function on the coal mine dataset. The observed data is shown by the black ticks on the x-axis.

processes that uses *lazy evaluation*, narrowing the disconnect between working with Gaussian processes mathematically and computationally.

2.7 A Novel Implementation of Gaussian Processes

2.7.1 Motivation

The term *expressiveness* in the context of software engineering is the ability of a language to reduce the number of transformations that a problem description must undergo before it can be solved computationally. Usually there is some disconnect between the mathematical description of the problem and how it is coded in a program. Ideally one would like the code to be as close to the mathematics as possible. This disconnect is inconveniently large when working with Gaussian processes. They are commonly used to model functions, yet are never implemented as such computationally. The usual argument for this is that Gaussian processes are inherently infinite dimensional objects, as for every $t \in \mathbb{R}$ one must associate $f(t)$, and so it is not possible to work with them directly. The usual problem transformation is to specify ahead of time a finite discretization on which to learn and predict the function and work with the discretized form of f computationally. This amounts to storing function values in an array. This suffices in the simple applications, such as a two-part Gaussian process regression where one has a set of observations on a set of inputs and wishes to predict the function on a new set of inputs. In this case the required discretization of f is known ahead of time, but matters become complicated and messy in applications where it is not. For more complicated operations with Gaussian processes the approach based on discretization soon becomes inelegant.

Consider the following motivation. It is very clear from the mathematics how to generate a realization via thinning. One might write a function accepting as input an intensity function, an upper bound thereupon, and an interval over which to simulate. When f is not a function but a discretized version the thinning procedure

must undergo a transformation to deal with the discretized version, diverging once again away from the mathematics. In addition, the thinning procedure requires knowing the function value on randomly selected number of inputs that is not known in advance. There is a further problem. Once the value of $f(t)$ has been drawn or *realized* at a time t , this function value must be conditioned upon when realizing further f values at subsequent times. Informally once the value of f has been seen at an input, then it is not possible to unsee it. The implication of this is that one must maintain a record of all values for times upon which f has been realized. In the following application this leads to an enormous amount of book-keeping. During the thinning operation values of f need to be realized on uniformly drawn times to compute the acceptance probability. Not all of these time are accepted and so the times upon which f has been realized is not equal to the collection of times contained in all observed and augmented realizations. This problem becomes exacerbated in the following application where the model contains up to 60 different realizations from different inhomogeneous Poisson point processes. This soon forces one to search for a more elegant implementation of Gaussian processes than the common discretized implementation.

Clearly the best implementation would allow f to remain a function as there is then no divergence of the code from the mathematics. After all, much effort has gone into modelling the arrivals in continuous-time without discretization, it would be nice to avoid discretization in the computation also. The usual objection in imperative languages is that it is not possible work with the infinite dimensional object f , but in functional languages working with infinite dimensional objects is not just common, it is the norm. The difference is in the method of evaluation. In imperative languages, assignments are evaluated *eagerly* whereas in functional languages assignments are evaluated *lazily*. Lazy evaluation means that the computation is delayed until is it required. This allows one to work with the entire natural numbers an infinite lists of

MCMC draws for example. It is perfect for implementing Gaussian processes, as f can be defined for all t , but computation is only performed when evaluating $f(t)$ for a time $t \in \mathbb{R}$. This section provides an implementation that allows random functions f to be sampled from $p(f)$ or $p(f|Y)$ in a manner that is no different from generating a random variate from a uniform distribution. The random function is a function just like any other, it can be evaluated at *any* input t or passed around to other parts of code. The output of the MCMC provides a linked list of *functions*, not a linked list of arrays corresponding to f values on a random discretization. This implementation is much closer to the mathematics requiring no problem transformation at all, resulting in very clean code.

2.7.2 Conditional Distributions Under State-Space Construction

The first consideration is a datastructure in which to store key-value pairs $(t_i, F(t_i))$, where $F(t) = (f(t), f'(t), f''(t))^T$, corresponding to previously realized function values. The desiderata of the datastructure will soon become clear after studying the conditional distributions under the state-space representation of the Gaussian process. Suppose the function F has been learned on a set of times T providing a set of function values $\mathcal{F} = \{F(t) : t \in T\}$. Given a new time t_n one would like to draw $F_n = F(t_n)$ from $p(F_n|\mathcal{F})$ i.e. conditioning on all previously learned values. From the Markov property of the continuous-time state-space representation, the conditional distribution for F_n is determined by its nearest neighbours in time only. Let $t_a = \min\{t \in T : t > t_n\}$ and $t_b = \max\{t \in T : t < t_n\}$ (a and b for after and before), with associated $F_a = F(t_a)$ and $F_b = F(t_b)$. It follows from the Markov property

that F_n is conditionally independent of $\mathcal{F} \setminus \{F_a, F_b\}$ given F_a and F_b . Let

$$\Phi_a = \exp(A(t_a - t_n)) \quad (2.36)$$

$$\Phi_n = \exp(A(t_n - t_b)) \quad (2.37)$$

$$\Phi_{ab} = \exp(A(t_a - t_b)) \quad (2.38)$$

$$Q_a = \int_0^{t_a - t_n} \exp(A\tau) LqL^T \exp(A\tau)^T d\tau, \quad (2.39)$$

$$Q_n = \int_0^{t_n - t_b} \exp(A\tau) LqL^T \exp(A\tau)^T d\tau, \quad (2.40)$$

$$Q_{ab} = \int_0^{t_a - t_b} \exp(A\tau) LqL^T \exp(A\tau)^T d\tau, \quad (2.41)$$

denote the required auto-regression and innovation matrices as in equations 2.14, then the joint distribution of F_a and F_n conditional on the previous value F_b is given by

$$\begin{pmatrix} F_a \\ F_n \end{pmatrix} | F_b \sim \mathcal{N} \left(\begin{pmatrix} \Phi_a \Phi_n F_b \\ \Phi_n F_b \end{pmatrix}, \begin{pmatrix} \Phi_a Q_n \Phi_a^T + Q_a & \Phi_a Q_n \\ Q_n \Phi_a^T & Q_n \end{pmatrix} \right). \quad (2.42)$$

This distribution can undergo some simplification. It is also true by direct forward simulation that $F_a | F_b \sim \mathcal{N}(\Phi_{ab} F_b, Q_{ab})$ which can be compared with the marginal in (2.42). It follows that terms can be simplified as $\Phi_a \Phi_n = \exp(A(t_a - t_n)) \exp(A(t_n - t_b)) = \exp(A(t_a - t_b)) = \Phi_{ab}$ and $\Phi_n Q_n \Phi_n^T + Q_a = Q_{ab}$. The conditional distribution is then available through multivariate normal theory as

$$F_n | F_b, F_a \sim \mathcal{N}(\Phi_n F_b + Q_n \Phi_a^T Q_{ab}^{-1} (F_a - \Phi_{ab} F_b), Q_n - Q_n \Phi_a^T Q_{ab}^{-1} \Phi_a Q_n). \quad (2.43)$$

If the set $\{t \in T : t > t_n\}$ is empty but $\{t \in T : t < t_n\}$ is not then the function has been learned already at a time before but not after. In this case the conditional distribution is that of forward simulation

$$F_n | F_b \sim \mathcal{N}(\Phi_n F_b, Q_n). \quad (2.44)$$

If the set $\{t \in T : t < t_n\}$ is empty but $\{t \in T : t > t_n\}$ is not then the function has been already learned at a time point after but not before. In this case the conditional distribution is that of reverse simulation

$$\begin{aligned} F_n | F_a &\sim \mathcal{N}(Q_\infty \Phi_a^T Q_\infty^{-1} F_a, Q_\infty - Q_\infty \Phi_n^T Q_\infty^{-1} \Phi_n Q_\infty) \\ &= \mathcal{N}(\Phi_a^R F_a, Q_a^R), \end{aligned} \tag{2.45}$$

where $\Phi_{a,ij}^R = \Phi_{a,ij}$ for (i, j) in $\{(1, 1), (1, 3), (2, 2), (2, 1), (2, 3)\}$ and $\Phi_{a,ij}^R = -\Phi_{a,ij}$ elsewhere. Informally this is because the sign of the first derivative changes when time is reversed. Q_a^R is defined similarly. If both sets are empty, then the function has not been learned anywhere yet. In this case, the distribution of F_n is the stationary distribution.

$$F_n \sim \mathcal{N}(0, Q_\infty). \tag{2.46}$$

Clearly in order to efficiently realize F at a new time point t , it is algorithmically necessary to efficiently retrieve its nearest neighbours in time from the data structure. In addition, as the F is progressively learned through its repeated use in different MCMC steps, it is necessary to be able to efficiently insert new key-value pairs into the datastructure while preserving the ability to quickly retrieve nearest neighbours of future time points.

2.7.3 The Persistent AVL-Tree

When talking about key-value pairs in the current context the keys are times on the real line, and the values are the corresponding function values. As they are real valued, the keys possess a natural ordering. This is very important to quickly retrieve nearest neighbours. The appropriate datastructure is any implementation of a “sorted-map”. In C++ this is implemented using red-black tree, whereas in Clojure it is implemented as a persistent (immutable) AVL-tree.

The tree structure of the AVL-tree, based on the ordering of its keys, allows nearest neighbours to be looked up with $\mathcal{O}(\log N)$ complexity, where N is the total

number of key-value pairs stored. Once the function value has been realized at this new time, the key-value pair $(t_n, F(t_n))$ must be added to the datastructure, as it is necessary to condition upon it when realizing F at subsequent new times. Insertion into the AVL-tree is an operation that also enjoys logarithmic complexity. Moreover, in contrast to other types of binary tree, the AVL-tree is *self-balancing* meaning that the maximum depth is kept as small as possible. This ensures logarithmic complexity of retrieval of nearest neighbours, or indeed the function values if one asks for a time that has already been realized, as F is progressively realized through multiple usages in the code.

2.7.4 Example Implementation

Listing 2.1 shows the definition of a function which produces lazily evaluated realizations of F , written in Clojure.

```

1 (defn sample-gp [known-values gp-var gp-time-scale]
2   (let [cond-set (atom (into (avl/sorted-map) known-values))]
3     (with-meta
4       (fn [t]
5         (if (contains? @cond-set t) (@cond-set t)
6           (let [bef (avl/nearest @cond-set < t)
7                 aft (avl/nearest @cond-set > t)
8                 has-bef (not (nil? bef))
9                 has-aft (not (nil? aft))
10                has-both (and has-bef has-aft)]
11             (cond
12               has-both (let [[tb Fb] bef
13                             [ta Fa] aft
14                             delay-v (- t tb)
15                             delay-n (- ta t)
16                             Ft (sample-sandwich Fb delay-v Fa delay-n gp-var gp-time-scale )]
17                           (do (swap! cond-set assoc t Ft) Ft))
18               has-bef (let [[tb Fb] bef
19                             delay (- t tb)
20                             Ft (sample-neighbour Fb delay gp-var gp-time-scale)]
21                         (do (swap! cond-set assoc t Ft) Ft))
22               has-aft (let [[ta Fa] aft
23                             delay (- ta t)
24                             Ft (sample-neighbour Fa delay gp-var gp-time-scale)]
25                         (do (swap! cond-set assoc t Ft) Ft))
26               :else (let [Ft (sample-mvn (statcov gp-var gp-time-scale))]
27                       (do (swap! cond-set assoc t Ft) Ft))))))
28     {:var gp-var :time-scale gp-time-scale :data cond-set}))

```

Listing 2.1: A function for sampling a lazily evaluated realization of a Gaussian process

The `sample-gp` function is an example of a *closure*, a concept introduced by Peter Landin in 1964 and popularized by the scheme programming language. It is a function that returns a new function with its own environment, in which an AVL-tree

is bound to the name `cond-set` (short for conditioning set - it will store the realized times and function values). The function that it returns is defined between lines 4 and 27. Whenever this new function is evaluated at a new time t , it first checks to see if this key is present in the AVL-tree i.e. if the function has already been previously evaluated at this input. If it is present, the function returns $F(t)$, if not it attempts to find the keys nearest to t , in logarithmic time, and binds these to `bef` and `aft` (previously “b” and “a”). Conditions that can arise are that t has a nearest neighbour both before and after, only before or after t , or there are currently no keys in the AVL-tree. In the case when t is sandwiched between to other times already present in the AVL-tree, then a value for $F(t)$ is drawn from equation (2.43). The other cases are handled similarly. This new key-value pair $(t, F(t))$ is inserted into the AVL-tree (in logarithmic time) before returning the result to the user.

The behaviour to the user is as follows

```
core> (def mygp (sample-gp {} 3 1))
#core/mygp
core> (mygp 0.5)
#vector [2.5855374199430243,0.741972115586161,-10.952079103500354]
core> (mygp 10.5)
#vector [-1.1879519809380346,3.6235009428526337,-13.156076870588237]
core> (mygp 10.51)
#vector [-1.1523807108872246,3.4913880171082368,-14.086361936311798]
```

i.e. `mygp` is a lazily evaluated realization of a Gaussian process with variance 3 and time-scale 1. It behaves like any other function in clojure. By inspecting the meta-data it is possible to see the internal AVL-tree become updated with new key-value pairs as the function is subsequently evaluated.

```
core> (def mygp (sample-gp 3 1))
#core/mygp
core> (meta mygp)
{:var 3, :time-scale 1, :data #atom[{} 0x5164dd9a]}
core> @(:data (meta mygp))
{}
core> (mygp 1)
#vector [0.3017980988600516,2.101152522085235,2.327574588156968]
core> @(:data (meta mygp))
{1 #vector [0.3017980988600516,2.101152522085235,2.327574588156968]}
core> (mygp 1.5)
#vector [1.40519429958541,2.1969074173882324,-1.312884773740397]
core> @(:data (meta mygp))
{1 #vector [0.3017980988600516,2.101152522085235,2.327574588156968],
 1.5 #vector [1.40519429958541,2.1969074173882324,-1.312884773740397]}
```

This corresponds to a realization of F from its prior distribution. The same code can be used to generate lazily evaluated realizations of F from the posterior distribution

also. The procedure is in two parts. Suppose one has a set of observations $\{(t_i, y_i)\}_{i=1}^n$. The FFBS algorithm is first used to realize F at times $\{t_i\}_{i=1}^n$. These times and function values are used to initialize the AVL-tree, passed to the `sample-gp` function in the first `known-values` argument. For a new time $t \notin \{t_i\}_{i=1}^n$ the posterior distribution is conditionally independent of $\{y_i\}_{i=1}^n$ given $\{F_i\}_{i=1}^n$ and so the conditional distribution is $p(F(t)|F_a, F_b)$ as before.

The main points are that the self-balancing AVL-tree maintains logarithmic complexity of insertion and lookup operations as more and more function values are realized. This means that looking up $F(t)$ for a previously evaluated t in addition to sampling a new value $F(t)|F_a, F_b$ for new a t is fast. All the book-keeping, associated with keeping track of times upon which F has been realized, is handled internally and is hidden from the user. The function F behaves like any other function in the language and can be passed to other functions such as the thinning function used to generate realizations. Moreover, the reference to the AVL-tree is *atomic*, meaning that this function is thread safe should one wish to use it in a parallel application. In the following application sampling the large number of augmented independent realizations as well as their hyperparameters can be performed in parallel. Thread safety in this context means that there is no race condition on the AVL tree.

Detecting Multiplexing in Neuronal Spike Trains

This chapter is a continuation of the former by using the previous model in an application within neuroscience. In this application the arrival times are the times at which a neuron (cell) discharges (fires) in response to an external stimulus. The recorded point-pattern is referred to in the literature as a *spike train*. Understanding temporal patterns in spike trains is the focus of *neural encoding* - how the brain is able to encode information from the external environment. Encoding of information can be at the individual cell level, through patterns in the spike train, or at a population level with multiple cells encoding the information as an ensemble. How the brain encodes information from an external single stimulus is considered to be well understood, yet this is not the case when presented multiple external stimuli. The following application is motivated by the desire to understand this better.

The experimental setup, with some simplification of the details, is as follows. A subject is played a sound at a frequency of A Herz and multiple spike-trains are recorded from a single cell. The subject is subsequently played a sound at a frequency of B Herz and multiple spike-trains are recorded from the same cell. The subject is then played both sounds simultaneously and multiple spike trains are recorded

from the same cell. Each spike train measurement will be referred to as a *trial*. The single sound trials will be referred to as A and B, whereas the dual sound are simply referred to as dual sound trials. The collection of A_n trials, B_n B trials and n_d dual trials will be referred to as a *triplet*, which corresponds to a single cell. Following data collection, the analysis concerns discovering relationships between the dual and single sound trials.

There are multiple hypotheses for how the brain encodes multiple information, based on empirical observations. A simple hypothesis is that within a population some cells fire like A and some like B. The respective proportions is something that needs to be learned, and this proportion may depend on the frequencies of each sound. An alternative is that each cell fires at an intermediate rate, somewhere between the firing rate of each single sound trial, yet this hypothesis is less supported empirically because behaviour is different compared to when the subject is played a single sound at a frequency somewhere between sounds A and B. A more novel and exciting hypothesis is the possibility of temporal *multiplexing* - a technique used in telecommunications to transmit multiple signals across a single channel. This behaviour is categorized by interleaving periods of firing like A and firing like B. If such behaviour is present then it is desired to learn the period of multiplexing, how quickly the cell switches between firing like A and then firing like B. This behaviour will also be referred to as *switching* or *dynamic*.

The most common analyses in neuroscience are not able to detect within trial dynamic behaviour. It is most common to regard repeated recorded spike trains as repeated measurements of the same object, aggregating them together to form a sufficient statistic. When repeated trials are pooled together, this smooths out any trial specific temporal dependence. In addition, the data is not modelled in continuous-time. Instead, an arbitrary choice of bin-width is made and the time-points are binned to form count data - the number of time-points in each bin. Avoiding the

arbitrary choice of bin-width is already desirable, but if the period of multiplexing is shorter than the bin-width then this transformation of the data will obscure the temporal behaviour. For these reasons new statistical tools are required.

There are two sides of the analysis. One side is to build a classifier that is able to detect the presence of dynamic behaviour. The other side concerns learning properties of the cell. For instance, suppose the cell encodes information by sometimes firing like A and sometimes like B. If some trials are like A and other are like B, then it is desired to understand with what probability a new trial will be like A or like B. Similarly dynamic behaviour might only be present in a subset of the trials, then it is desired to understand with what probability a new trial will exhibit dynamic behaviour. The period of multiplexing might also be different between trials, some trials may multiplex at different rates, and so the cell distribution of multiplexing periods is also desirable to learn.

The statistical challenge is to develop a model that is flexible enough to quantify evidence for each of these different hypotheses, while at the same time avoiding the dangers of making a model that is too flexible. The latter concerns over-fitting the data and having too many parameters to be meaningfully learned. Indeed the analysis is ambitious, attempting to learn many quantities of interest from data with only moderate signal to noise ratio. The model presented in this chapter learns on simulated datasets some quantities of interest well, but other quantities not so well.

3.1 Statistical Model for Multiplexing

The variety of different types of data necessitates a specific notation. There are three kinds of trial A, B, D , and for each trial kind there A_n, B_n and D_n spike trains recorded respectively. Each spike train is modelled as a realization from an inhomogeneous Poisson point process, for which additional augmented realizations are necessary. Instead of having four subscripts separated by commas the following

notational rules are adopted. The left superscript denotes the type of trial, the right superscript denotes the trial number, the left subscript denotes the realization type and the right subscript finally denotes the point in an enumeration of the point pattern. If no left superscript is present, it corresponds to a dual trial. The reason for doing this is that some derived quantities will not be indexed by all four parameters and so if one were using subscripts one would end up with quantities possessing variable numbers of subscripts, of which the meaning is difficult to keep track. Although having left and right sub and superscripts is unsightly, it is at least consistent.

To begin, each A and B single sound trial is modelled as a realization from an inhomogeneous Poisson point process with rate ${}^A\lambda(t) = {}^A\Lambda({}^A\mu + {}^A f(t))$ and ${}^B\lambda(t) = {}^B\Lambda\sigma({}^B\mu + {}^B f(t))$ respectively. Each dual trial is modelled as a realization from an inhomogeneous Poisson point process with rate $\lambda^i(t) = \alpha^i(t){}^A\lambda(t) + (1 - \alpha^i(t)){}^B\lambda(t)$ where each realization has its own *weight function* $\alpha^i : \mathcal{T} \rightarrow [0, 1]$ structured as $\alpha^i(t) = \sigma(\mu^i + g^i(t))$. The link function σ once again can be the probit or logistic function. The weight function describes mathematically the extent to which the dual sound trial behaviour is weighted toward single sound trials A or B. The different experimental hypotheses translate to different forms of $\alpha^i(t)$. There is a time-averaged or static component μ^i and a dynamic component $g^i(t)$. To accommodate purely static behaviour $g^i(t)$ is modelled as a mixture of the zero function ($\delta(t) = 0 \quad \forall t \in \mathcal{T}$) and a *zero-mean* Matern Gaussian process with weights $(1 - \theta)$ and θ respectively. Dual sound trials for which $g^i(t)$ is non-zero will be referred to as *dynamic*, otherwise *static*. The parameter θ is a probability of generating dynamic trials that is intrinsic to each cell and is a quantity of interest. This motivates a non-informative $\mathcal{B}(1, 1)$ prior over θ .

Statements about the firing rate translate to statements about the intensity function. To illustrate the behaviour that this model can capture consider the following

examples. When $g^i(t) = 0$ for all t then the dual sound intensity is a static superposition of the single sound intensities with weights $\sigma(\mu^i)$ and $(1 - \sigma(\mu^i))$. When μ^i is large in magnitude and positive the trial is firing like A, whereas when it is large in magnitude and negative the trial is firing like B. When μ^i is moderate in magnitude the firing rate is somewhere between that of A and B. μ^i is of course allowed to vary between trials, hence the superscript i . There is a hypothesis that the cell sometimes fires like A and sometimes like B. In order to learn how the cell decides new μ^i values the distribution of μ^i 's is modelled as a Dirichlet process mixture of normals. When $g^i(t)$ is non-zero, the firing rate is between the firing rate of A and B with dynamic dependence. The multiplexing hypothesis corresponds to functions $g^i(t)$ that exhibits high and low swings with period behaviour. The period may vary between trials and so in order to learn the distribution of periods, the time-scale of the Gaussian process is modelled with a discrete prior combined with a Dirichlet prior on the probability vector. Learning the probability vector corresponds to the time-scales favoured by the cell.

3.2 Data Augmentation

Each trial, modelled as a realization from an inhomogeneous Poisson point process, requires additional unobserved realizations to be augmented to the model in order to make the likelihood tractable. Let ${}^A_a\xi^i = ({}^A_n^i, \{({}^A_t_j^i, {}^A_m_j^i)\}_{j=1}^{{}^A_n^i})$ denote the realization from an inhomogeneous Poisson point process with intensity ${}^A\lambda$ and appropriate mark distribution, corresponding to the i 'th A trial with unobserved marks. As before, an additional realization ${}^A_r\xi^i$ from an inhomogeneous Poisson point process with intensity ${}^A\Lambda - {}^A\lambda$ and appropriate mark distribution is required to make the likelihood tractable. The realizations for the B trials ${}^B_a\xi^i$ and ${}^B_r\xi^i$ are defined similarly for $i \in \{1, \dots, {}^Bn\}$.

The realization corresponding to the i 'th observed dual sound trial is denoted ${}^D_o \xi^i$ and is modelled as inhomogeneous Poisson point processes with rate $\lambda^i(t) = \alpha^i(t)^A \lambda(t) + (1 - \alpha^i(t))^B \lambda(t)$, namely, a superposition of two intensities. By the superposition theorem for the Poisson point process, see Kingman (1993), this can be modelled in a generative way by assuming that $\{{}^D_o t_j^i\}_{j=1}^{Dn^i} = \{{}^D_1 t_j^i\}_{j=1}^{Dn^i} \cup \{{}^D_4 t_j^i\}_{j=1}^{Dn^i}$ where $\{{}^D_1 t_j^i\}_{j=1}^{n_1^i}$ and $\{{}^D_4 t_j^i\}_{j=1}^{n_4^i}$ are realizations of inhomogeneous Poisson point processes with rates $\alpha^i(t)^A \lambda(t)$ and $(1 - \alpha^i(t))^B \lambda(t)$ respectively. The knowledge of whether a time ${}^D t_j^i$ came from the former or latter realization is assumed lost in the observed data and must be imputed. These realizations contain information about ${}^A \lambda$, ${}^B \lambda$ and α^i and so to get a tractable full conditional for g^i , $({}^A f, {}^A \mu)$ and $({}^B f, {}^B \mu)$ the same strategy of adding latent truncated-normal or Polya-gamma marks must be used. Moreover, an intractable integral remains in the likelihood. Additional augmented realizations are required in order to remove it, which motivates the following set of augmented realizations

$${}^D_1 \xi^i | {}^A \Lambda, {}^A f, {}^A \mu, \mu^i, g^i \sim PPP(\alpha^i(t)^A \lambda(t), {}^A \mathcal{D}_+, \mathcal{D}_+, \mathcal{D}_1), \quad (3.1)$$

$${}^D_2 \xi^i | {}^A \Lambda, {}^A f, {}^A \mu, \mu^i, g^i \sim PPP((1 - \alpha^i(t))^A \lambda(t), {}^A \mathcal{D}_+, \mathcal{D}_-), \quad (3.2)$$

$${}^D_3 \xi^i | {}^A \Lambda, {}^A f, {}^A \mu \sim PPP({}^A \Lambda - {}^A \lambda(t), {}^A \mathcal{D}_-), \quad (3.3)$$

$${}^D_4 \xi^i | {}^B \Lambda, {}^B f, {}^B \mu, \mu^i, g^i \sim PPP((1 - \alpha^i(t))^B \lambda(t), {}^B \mathcal{D}_+, \mathcal{D}_-, \mathcal{D}_4), \quad (3.4)$$

$${}^D_5 \xi^i | {}^B \Lambda, {}^B f, {}^B \mu, \mu^i, g^i \sim PPP(\alpha^i(t)^B \lambda(t), {}^B \mathcal{D}_+, \mathcal{D}_+), \quad (3.5)$$

$${}^D_6 \xi^i | {}^B \Lambda, {}^B f, {}^B \mu \sim PPP({}^B \Lambda - {}^B \lambda(t), {}^B \mathcal{D}_-), \quad (3.6)$$

for all $i \in \{1, \dots, Dn\}$. The intensities sum to ${}^A \Lambda + {}^B \Lambda$ and so the likelihood reduces to that of two homogeneous Poisson point processes. and for which

When the link function is the probit function the mark distributions ${}^{\mathcal{X}} \mathcal{D}_+$ is $m|t \sim \mathcal{N}_{[0,\infty)}({}^{\mathcal{X}} \mu + {}^{\mathcal{X}} f(t), 1)$, ${}^{\mathcal{X}} \mathcal{D}_-$ is $m|t \sim \mathcal{N}_{(-\infty,0]}({}^{\mathcal{X}} \mu + {}^{\mathcal{X}} f(t), 1)$ for $\mathcal{X} \in \{A, B\}$. \mathcal{D}_+^i is $w|t \sim \mathcal{N}_{[0,\infty)}(\mu^i + g^i(t), 1)$ and \mathcal{D}_-^i is $w|t \sim \mathcal{N}_{(-\infty,0]}(\mu^i + g^i(t), 1)$. In the

logistic case ${}^{\mathcal{X}}\mathcal{D}_+ = {}^{\mathcal{X}}\mathcal{D}_-$ is $m|t \sim \mathcal{PG}(1, {}^{\mathcal{X}}\mu + {}^{\mathcal{X}}f(t))$ for $\mathcal{X} \in \{A, B\}$. Similarly $\mathcal{D}_+^i = \mathcal{D}_-^i$ is $w|t \sim \mathcal{PG}(1, \mu^i + g^i(t))$. The mark distributions \mathcal{D}_1 and \mathcal{D}_4 are trivial discrete distributions $\mathbb{P}[l = 1|t] = 1$ and $\mathbb{P}[l = 0] = 1$ respectively. Augmenting points in these realizations with a binary label to record from which realization a time-point arises is useful for the purposes of imputation as it is assumed that ${}^D_o\xi^i = ({}^D_1n^i, \{{}^D_1t_j^i, {}^D_1m_j^i, {}^D_1w_j^i, {}^D_1l_j^i\}_{j=1}^{D_1n^i} \cup \{{}^D_4t_j^i, {}^D_4m_j^i, {}^D_4w_j^i, {}^D_4l_j^i\}_{j=1}^{D_4n^i})$. The labels are unobserved but when imputed allow the realizations ${}^D_1\xi^i$ and ${}^D_4\xi^i$ to be reconstructed.

3.3 Complete Model Description

The full model is quite large and so it is best to state it in parts. The realizations and parameters associated with the single sound trials are modelled as

$${}^{\mathcal{X}}\xi^i | {}^{\mathcal{X}}\Lambda, {}^{\mathcal{X}}f, {}^{\mathcal{X}}\mu \sim PPP({}^{\mathcal{X}}\lambda(t), {}^{\mathcal{X}}\mathcal{D}_+), \quad (3.7)$$

$${}^{\mathcal{X}}\xi^i | {}^{\mathcal{X}}\Lambda, {}^{\mathcal{X}}f, {}^{\mathcal{X}}\mu \sim PPP({}^{\mathcal{X}}\Lambda - {}^{\mathcal{X}}\lambda(t), {}^{\mathcal{X}}\mathcal{D}_-), \quad (3.8)$$

$${}^{\mathcal{X}}\lambda(t) = {}^{\mathcal{X}}\Lambda \sigma({}^{\mathcal{X}}\mu + {}^{\mathcal{X}}f(t)), \quad (3.9)$$

$${}^{\mathcal{X}}\Lambda \sim Ga(l_1, l_2), \quad (3.10)$$

$${}^{\mathcal{X}}\mu \sim \mathcal{N}(0, \sigma_\mu^2), \quad (3.11)$$

$${}^{\mathcal{X}}f | {}^{\mathcal{X}}\rho^2, {}^{\mathcal{X}}\tau \sim \mathcal{GP}(0, K_{{}^{\mathcal{X}}\rho^2, {}^{\mathcal{X}}\tau}^{\nu=5/2}(\cdot, \cdot)), \quad (3.12)$$

$${}^{\mathcal{X}}\tau \sim Ga(t_1, t_2), \quad (3.13)$$

$${}^{\mathcal{X}}\rho^2 \sim IG(r_1/2, r_2/2), \quad (3.14)$$

for all $\mathcal{X} \in \{A, B\}$ and $i \in \{1, \dots, {}^{\mathcal{X}}n\}$.

The observed and augmented realizations associated with the dual sound trials are modelled as

$${}^D_1\xi^i|{}^A\Lambda, {}^Af, {}^A\mu, \mu^i, g^i \sim PPP(\alpha^i(t) {}^A\lambda(t), {}^A\mathcal{D}_+, \mathcal{D}_+^i, \mathcal{D}_1), \quad (3.15)$$

$${}^D_2\xi^i|{}^A\Lambda, {}^Af, {}^A\mu, \mu^i, g^i \sim PPP((1 - \alpha^i(t)) {}^A\lambda(t), {}^A\mathcal{D}_+, \mathcal{D}_-^i), \quad (3.16)$$

$${}^D_3\xi^i|{}^A\Lambda, {}^Af, {}^A\mu \sim PPP({}^A\Lambda - {}^A\lambda(t), {}^A\mathcal{D}_-), \quad (3.17)$$

$${}^D_4\xi^i|{}^B\Lambda, {}^Bf, {}^B\mu, \mu^i, g^i \sim PPP((1 - \alpha^i(t)) {}^B\lambda(t), {}^B\mathcal{D}_+, \mathcal{D}_-^i, \mathcal{D}_4), \quad (3.18)$$

$${}^D_5\xi^i|{}^B\Lambda, {}^Bf, {}^B\mu, \mu^i, g^i \sim PPP(\alpha^i(t) {}^B\lambda(t), {}^B\mathcal{D}_+, \mathcal{D}_+^i), \quad (3.19)$$

$${}^D_6\xi^i|{}^B\Lambda, {}^Bf, {}^B\mu \sim PPP({}^B\Lambda - {}^B\lambda(t), {}^B\mathcal{D}_-), \quad (3.20)$$

for all $i \in \{1, \dots, {}^Dn\}$ with

${}^D_o\xi^i = ({}^D_1n^i + {}^D_4n^i, \{({}^D_1t_j^i, {}^D_1m_j^i, {}^D_1w_j^i, {}^D_1l_j^i \}_{j=1}^{{}^Dn^i} \cup \{ ({}^D_4t_j^i, {}^D_4m_j^i, {}^D_4w_j^i, {}^D_4l_j^i) \}_{j=1}^{{}^Dn^i})$. The priors on parameters associated with the dual sound trials are given by

$$\alpha^i(t) = \sigma(\mu^i + g^i(t)), \quad (3.21)$$

$$g^i|\gamma^i, {}^D\rho^2, {}^D\tau^i \sim \gamma^i \mathcal{GP}(0, K_{{}^D\rho^2, {}^D\tau^i}^{\nu=5/2}(\cdot, \cdot)) + (1 - \gamma^i)\delta, \quad (3.22)$$

$$\gamma^i|\theta \sim \text{Bern}(\theta), \quad (3.23)$$

$$\theta \sim \mathcal{B}(b_1, b_2), \quad (3.24)$$

$$p({}^D\tau^i|p) \propto \sum_{i=1}^k p_k \delta({}^D\tau^i - \tau_k), \quad (3.25)$$

$$p \sim \text{Dir}(a_1, \dots, a_k), \quad (3.26)$$

$${}^D\rho^2 \sim \text{IG}(r/2, r\rho_0^2/2), \quad (3.27)$$

$$\mu^i|\tilde{\mu}^i \sim \mathcal{N}(\tilde{\mu}^i, c\sigma_\mu^2), \quad (3.28)$$

$$\tilde{\mu}^1, \dots, \tilde{\mu}^{{}^Dn} \sim \mathcal{G} \quad (3.29)$$

$$\mathcal{G} \sim \mathcal{DP}(a, \pi_0), \quad (3.30)$$

where the base measure π_0 is a $\mathcal{N}(0, \sigma_\mu^2)$ distribution and c a positive constant less than 1.

3.4 The Euler-Maruyama Approximation

Before discussing MCMC for this model it is necessary to talk about the computational burden and approximations that can be used to improved performance. A back-of-the-envelope calculation shows for each D trial contributes an expected number ${}^A\Lambda$ of noisy measurements of ${}^A f$ and ${}^B\Lambda$ of ${}^B f$, given all augmented realizations. Similarly each A trial contributes an expected number of ${}^A\Lambda$ noisy measurements of ${}^A f$, and each B trial contributes an expected number of ${}^B\Lambda$ noisy measurements of ${}^B f$. With 15 trials of each type with typical values of ${}^A\Lambda \approx {}^B\Lambda \approx 400$, this amounts to approximately 12000 data points each. These noisy measurements are uniformly distributed in time over \mathcal{T} , forming a very dense random discretization in time. Given the proximity of subsequent time point, the Euler Maruyama approximation, see (Kloeden and Platen, 2011, Chapter 9), to equation (2.13) becomes almost exact numerically. The Euler-Maruyama approximation is essentially a first order approximation to the solution, resulting in the approximate state equation

$$\begin{aligned}
 F_i &= \tilde{\Phi}_i F_{i-1} + \epsilon_i \\
 \epsilon_i &\sim \mathcal{N}(0, \tilde{Q}_i)
 \end{aligned}$$

where

$$\tilde{\Phi}_i = \begin{pmatrix} 1 & \Delta_i & 0 \\ 0 & 1 & \Delta_i \\ -3\lambda^3\Delta_i & -3\lambda^2\Delta_i & 1 - 3\lambda\Delta_i \end{pmatrix}$$

$$\tilde{Q}_i = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & (16/3)\rho^2\lambda^5\Delta_i \end{pmatrix}$$

$$\Delta_i = t_i - t_{i-1}$$

$$\lambda = \sqrt{5}/\tau.$$
(3.31)

These approximate matrices $\tilde{\Phi}_i$ and \tilde{Q}_i can also be motivated as performing series expansions of Φ_i and Q_i , defined in equation (2.14), and retaining terms up to leading

order in Δ_i/τ . With such a dense random grid, Δ_i/τ is typically very tiny and this approximation performs very well. Φ_i and Q_i can then be replaced everywhere by the approximations. The computational advantage comes from the zeros present in these matrices. It allows some linear algebra operations to be avoided, replaced by expressions derived by hand. This mainly avoids the computation of inverses as it turns out these matrices are diagonal.

3.4.1 Forward Filtering

To illustrate this consider the forward filtering stage. Let M_i be of the form

$$M_i = \begin{pmatrix} M_{i,11} & 0 & M_{i,13} \\ 0 & M_{i,22} & 0 \\ M_{i,31} & 0 & M_{i,33} \end{pmatrix}. \quad (3.32)$$

It follows from the form of $\tilde{\Phi}_{i+1}$ that $\tilde{\Phi}_{i+1}M_i\tilde{\Phi}_{i+1} + \tilde{Q}_{i+1}$ is a diagonal matrix give by

$$\tilde{\Phi}_{i+1}M_i\tilde{\Phi}_{i+1} + \tilde{Q}_{i+1} = \begin{pmatrix} M_{i,11} & 0 & 0 \\ 0 & M_{i,22} & 0 \\ 0 & 0 & \frac{16}{3}\Delta_{i+1}\rho^2\lambda^5 + M_{i,33}(1 - 3\lambda\Delta_{i+1})^2 \end{pmatrix}, \quad (3.33)$$

which leads to a complete forward filtering update of

$$M_{i+1} = \begin{pmatrix} M_{i,11} - \frac{M_{i,11}^2}{\sigma_{i+1}^2 + M_{i,11}} & 0 & 0 \\ 0 & M_{i,22} & 0 \\ 0 & 0 & \frac{16}{3}\Delta_{i+1}\rho^2\lambda^5 + M_{i,33}(1 - 3\Delta_{i+1}\lambda)^2 \end{pmatrix}. \quad (3.34)$$

M_1 is clearly of this form,

$$M_1 = \begin{pmatrix} \rho^2 - \frac{\rho^4}{\rho^2 + \sigma^2} & 0 & -\frac{1}{3}\lambda^2\rho^2 \\ 0 & \frac{\lambda^2\rho^2}{3} & 0 \\ -\frac{1}{3}\lambda^2\rho^2 & 0 & \lambda^4\rho^2 \end{pmatrix}, \quad (3.35)$$

and so it follows that deriving M_{i+1} from M_i corresponds to merely updating the diagonals of a diagonal matrix. Similar arguments lead to the expression for m_{i+1} as

$$m_{i+1} = \begin{pmatrix} \frac{M_{i,11}(y_{i+1}-\mu-m_{i,1}-m_{i,2}\Delta_{i+1})}{M_{i,11}+\sigma_{i+1}^2} + m_{i,1} + m_{i,2}\Delta_{i+1} \\ m_{i,2} + m_{i,3}\Delta_{i+1} \\ -\lambda^3 m_{i,1}\Delta_{i+1} - 3\lambda^2 m_{i,2}\Delta_{i+1} + m_{i,3}(1-3\Delta_{i+1}\lambda) \end{pmatrix} \quad (3.36)$$

3.4.2 Backwards Sampling

The backwards sampling steps also enjoy simplified expression without the need for linear algebra. In particular $F_i|F_{i+1}, y_{1:i+1} \sim (c_i, V_i)$ where

$$c_i = \begin{pmatrix} F_{i+1,1} - m_{i,2}\Delta_{i+1} \\ F_{i+1,2} - m_{i,3}\Delta_{i+1} \\ \frac{3M_{i,33}(1-3\Delta_{i+1}\lambda)(F_{i+1,3} + \lambda^3 m_{i,1}\Delta_{i+1} + 3\lambda^2 m_{i,2}\Delta_{i+1} - m_{i,3}(1-3\Delta_{i+1}\lambda))}{3M_{i,33}(1-3\Delta_{i+1}\lambda)^2 + 16\lambda^5 \rho^2 \Delta_{i+1}} + m_{i,3} \end{pmatrix} \quad (3.37)$$

and

$$V_i = \begin{pmatrix} 0 & 0 & M_{i,13} \\ 0 & 0 & 0 \\ M_{i,31} & 0 & M_{i,33} - \frac{3M_{i,33}^2(1-3\Delta_{i+1}\lambda)^2}{16\Delta_{i+1}\rho^2\lambda^5 + 3M_{i,33}(1-3\Delta_{i+1}\lambda)^2} \end{pmatrix}, \quad (3.38)$$

where of course $M_{i,31} = M_{i,13} = 0$ for all i except $i = 1$. There is also an argument for using the Euler Maruyama approximation for numerical stability.

3.4.3 Stability

Equation (2.43) requires taking the inverse of the matrix Q_{ab} (in the sense of solving a linear system). When the times t_a and t_b are close, this matrix becomes very poorly conditioned and loses numerical accuracy. This is also the case when evaluating $p(F_{1:n}|\rho^2, \tau)$ as this involves terms like $\|F_{i+1} - A_{i+1}F_i\|_{Q_{i+1}^{-1}}^2$. These become unstable when times are very close to each other as the conditional distribution essentially degenerates down onto the third dimension and so it is better to use the Euler Maruyama approximation to avoid numerical instabilities. The conditional density

for $F_{1:n}$ can then be approximated as

$$\begin{aligned}
p(F_{1:n}|\rho^2\tau) &\approx \left(\frac{1}{\rho^2\lambda^5}\right)^{1/2} \exp\left(-\frac{1}{2}\frac{3}{16\rho^2\lambda^5}(f_i'' - (-\lambda^3\Delta f_{i-1} - 3\lambda^2\Delta_i f'_{i-1} + (1 - 3\lambda\Delta_i)f''_{i-1}))^2\right) \\
&+ \frac{1}{|V_\infty|^{1/2}} \left(\frac{1}{\rho^2}\right)^{3/2} \exp\left(-\frac{1}{2\rho^2}F_1^T V_\infty^{-1} F_1\right),
\end{aligned} \tag{3.39}$$

where $V_\infty = Q_\infty/\rho^2$. This can be used with conjugate inverse gamma priors on ρ^2 to get an inverse-gamma full conditional distribution. The form of $F(t)|F_a, F_b$ under the Euler Maruyama approximation is simple, but a little cumbersome and is omitted for the sake of brevity.

3.5 Gibbs Sampling

When discussing the MCMC implementation it helps to break it down into parts. To begin, consider the steps associated with the parameters of the single sound trials.

3.5.1 Steps for Single Trial Specific Parameters

The Gibbs steps for sampling conditional distributions of ${}^{\mathcal{X}}F, {}^{\mathcal{X}}\mu, {}^{\mathcal{X}}\rho^2$ and ${}^{\mathcal{X}}\tau$ for $\mathcal{X} \in \{A, B\}$ are much the same as in section 2.4, with the modification that dual trials now also contribute noisy measurements of ${}^{\mathcal{X}}\mu + {}^{\mathcal{X}}f$. All of these measurements must be collected and enumerated, preferably in a sorted order in order to use the state-space representation. When the logistic link function is used, let

$$\begin{aligned}
{}^A S &= \bigcup_{i=1}^{A_n} \{(A t_j^i, 1/(2_a^A m_j^i), 1/a^A m_j^i)\}_{j=1}^{A_n^i} \cup \{(A t_j^i, 1/(2_r^A m_j^i), 1/r^A m_j^i)\}_{i=1}^{A_n^i}, \\
{}^A D &= \bigcup_{i=1}^{D_n} \bigcup_{k=1}^3 \{(D t_j^i, 1/(2_k^D m_j^i), 1/k^D m_j^i)\}_{j=1}^{D_n^i}, \\
{}^A \mathcal{Y} &= {}^A S \cup {}^A D,
\end{aligned} \tag{3.40}$$

then ${}^A \mathcal{Y}$ corresponds to a set of noisy measurements of ${}^A\mu + {}^A f$. The first element of each 3-tuple is a time, the second is a measurement the third is the measurement vari-

ance. These need to be sorted before using the state-space representation, so for this reason let ${}^A N = |{}^A \mathcal{Y}|$ and ${}^A \mathcal{Y}_{1:AN}$ denote an enumeration of the elements of ${}^A \mathcal{Y}$ such that the times are ordered i.e. ${}^A \mathcal{Y}_{1,1} \leqslant {}^A \mathcal{Y}_{2,1}, \dots, \leqslant {}^A \mathcal{Y}_{AN,1}$. An ordered sequence of times can then be defined as ${}^A t_{1:AN} = {}^A \mathcal{Y}_{1:AN,1}$ with corresponding measurements ${}^A y_{1:AN} = {}^A \mathcal{Y}_{1:AN,2}$ and measurement variances ${}^A \sigma_{1:AN}^2 = {}^A \mathcal{Y}_{1:AN,3}$. The newly defined ${}^A t_{1:AN}$, ${}^A y_{1:AN}$ and ${}^A \sigma_{1:AN}^2$ can then be used to sample the conditional distributions $({}^A F, {}^A \tau)|-$, $({}^A F, {}^A \rho^2)|-$ and $({}^A F, {}^A \mu)|-$ as in section 2.4. For the parameters associated with ${}^B \lambda$ the definition of the previously defined quantities changes only slightly. The only modification is

$$\begin{aligned}
{}^B S &= \bigcup_{i=1}^{B_n} \{({}_a^B t_j^i, 1/(2_a^B m_j^i), 1/_a^B m_j^i)\}_{j=1}^{B_n^i} \cup \{({}_r^B t_j^i, 1/(2_r^B m_j^i), 1/_r^B m_j^i)\}_{j=1}^{B_n^i}, \\
{}^B D &= \bigcup_{i=1}^{D_n} \bigcup_{k=4}^6 \{({}_k^D t_j^i, 1/(2_k^D m_j^i), 1/_k^D m_j^i)\}_{j=1}^{D_n^i}, \\
{}^B \mathcal{Y} &= {}^B S \cup {}^B D,
\end{aligned} \tag{3.41}$$

from which the process is the same as before. The only difference when the probit link function is used is that $\{({}_x^{\mathcal{X}} t_j^i, 1/(2_x^{\mathcal{X}} m_j^i), 1/_x^{\mathcal{X}} m_j^i)\}_{j=1}^{\mathcal{X} n^i}$ is replaced by $\{({}_x^{\mathcal{X}} t_j^i, {}_x^{\mathcal{X}} m_j^i, 1)\}_{j=1}^{\mathcal{X} n^i}$ for each realization. The conditional distribution for ${}^{\mathcal{X}} \Lambda$ is

$${}^{\mathcal{X}} \Lambda| - \sim Ga(l_1 + {}^{\mathcal{X}} N, l_2 + {}^{\mathcal{X}} n \mathcal{T} + {}^D n \mathcal{T}) \tag{3.42}$$

3.5.2 Steps for Augmented Realizations

The realizations ${}^A \xi_r^i$ for $i \in \{1, \dots, {}^A n\}$, ${}^B \xi_r^i$ for $i \in \{1, \dots, {}^B n\}$ and ${}^D \xi_k^i$ for $i \in \{1, \dots, {}^D n\}$ and $k \in \{2, 3, 5, 6\}$ are independent realizations of marked inhomogeneous Poisson point processes defined in 3.7 and 3.15. The steps for sampling marks for the remaining dual sounds trials are as follows

$${}^D l_j^i | {}^D t_j^i, {}^A \lambda(\cdot), {}^B \lambda(\cdot), \alpha^i(\cdot) \sim \text{Bern} \left(\frac{\alpha^i({}_o^D t_j^i) {}^A \lambda({}_o^D t_j^i)}{(\alpha^i({}_o^D t_j^i) {}^A \lambda({}_o^D t_j^i) + (1 - \alpha^i({}_o^D t_j^i)) {}^B \lambda({}_o^D t_j^i)} \right), \tag{3.43}$$

and

$$\begin{aligned}
& \mathcal{D}_+^A: \quad {}_o^D m_j^i | {}_o^D t_j^i, {}^A \lambda(\cdot), {}^B \lambda(\cdot), \alpha^i(\cdot), {}_o^D l_j^i = 1 \sim \mathcal{D}_+^A \\
& \mathcal{D}_+^B: \quad {}_o^D m_j^i | {}_o^D t_j^i, {}^A \lambda(\cdot), {}^B \lambda(\cdot), \alpha^i(\cdot), {}_o^D l_j^i = 0 \sim {}^B \mathcal{D}_+ \\
& \mathcal{D}_+^i: \quad {}_o^D w_j^i | {}_o^D t_j^i, {}^A \lambda(\cdot), {}^B \lambda(\cdot), \alpha^i(\cdot), {}_o^D l_j^i = 1 \sim \mathcal{D}_+^i \\
& \mathcal{D}_-^i: \quad {}_o^D w_j^i | {}_o^D t_j^i, {}^A \lambda(\cdot), {}^B \lambda(\cdot), \alpha^i(\cdot), {}_o^D l_j^i = 0 \sim \mathcal{D}_-^i.
\end{aligned} \tag{3.44}$$

The newly sampled labels determine ${}_1^D \xi^i$ and ${}_4^D \xi^i$. Explicitly let $O^i = \{({}_o^D t_j^i, {}_o^D m_j^i, {}_o^D w_j^i)\}_{j=1}^{D n^i}$ and define subsets $L^i = \{(t, m, w, l) \in O^i : l = 1\}$ and $R^i = \{(t, m, w, l) \in O^i : l = 0\}$, then ${}_1^D \xi^i$ and ${}_4^D \xi^i$ are given by

$$\begin{aligned}
{}_1^D \xi^i &= (|L^i|, L^i), \\
{}_4^D \xi^i &= (|R^i|, R^i).
\end{aligned} \tag{3.45}$$

3.5.3 Steps for Dual Trial Specific Parameters

For each dual trial i , define ${}^D \mathcal{Y}^i = \bigcup_{k \in \{1,2,4,5\}} \{({}_k^D t_j^i, {}_k^D w_j^i, 1)\}_{j=1}^{D n^i}$ and $N^i = |{}^D \mathcal{Y}^i|$. These correspond to all noisy measurements of the function g^i across all realizations. Once again let ${}^D \mathcal{Y}_{1:N^i}^i$ denote an enumeration such that the times are ordered, the times $t_{1:N^i}^i$, measurements $y_{1:N^i}^i$ and variances $(\sigma^2)_{1:N^i}^i$ can then be defined as the first, second and third elements of these ordered 3-tuples. It is also helpful to let these quantities define $\Sigma^i = \text{Diag}((\sigma^2)_1^i, \dots, (\sigma^2)_{N^i}^i)$ and K^i - the covariance matrix resulting from the Matern covariance kernel on times $t_{1:N^i}^i$. As the state-space construction requires augmenting the first and second derivatives of the function g^i , $G^i(t) = (g^i(t), (g^i)'(t), (g^i)''(t))$.

Conditional $(G_{1:D_n}, {}^D\rho^2)|-$

The variance term ${}^D\rho^2$ enters into the likelihood through all dual trials for which $\gamma^i = 1$, therefore the full conditional is

$$p({}^D\rho^2|-) \propto \prod_{i:\gamma^i=1} \prod_{k \in \{1,2,4,5\}} p({}^D_k\xi^i | {}^D\rho^2, {}^D\tau^i, \gamma^i, \mu^i, {}^A\lambda, {}^B\lambda) p({}^D\rho^2), \quad (3.46)$$

$$\propto \prod_{i:\gamma^i=1} \mathcal{N}(y_{1:N^i}^i | 1\mu^i, \Sigma^i + K^i) p({}^D\rho^2), \quad (3.47)$$

which can be evaluated in linear time by running forward filtering on each required ${}^D\mathcal{Y}^i$. A new value of ${}^D\rho^2$ is sampled via slice sampling. Sampling from $G_{1:D_n} | {}^D\rho^2, -$ needn't be performed in the cyclical Gibbs sampler.

Conditional $(G^i, {}^D\tau^i)|-$

The conditional distribution when $\gamma_i = 1$

$$p({}^D\tau^i|-) \propto \prod_{k \in \{1,2,4,5\}} p({}^D_k\xi^i | {}^D\rho^2, {}^D\tau^i, \mu^i, \gamma^i, {}^A\lambda, {}^B\lambda) p(\tau_i^2), \quad (3.48)$$

$$\propto \mathcal{N}(y_{1:N^i}^i | 1\mu^i, \Sigma^i + K^i) p({}^D\tau^i), \quad (3.49)$$

which can be computed in linear time by forward filtering on ${}^D\mathcal{Y}^i$. When $\gamma^i = 0$ it is merely the prior distribution This is also sampled via slice sampling and in the cyclical Gibbs sampler the draw of G^i conditional on ${}^D\tau^i$ needn't be performed.

Conditional $(G^i, \mu^i)|-$

Starting with $\mu^i|-$, when $\gamma^i = 0$ the full conditional is $\mathcal{N}(c^i, (v^2)^i)$ where

$$(v^2)^i = \sum_{j=1}^{N^i} \frac{1}{(\sigma^2)_j^i} + \frac{\tilde{1}}{\sigma_\mu^2} \quad (3.50)$$

$$c^i = (v^2)^i \left(\sum_{j=1}^{N^i} \frac{y_j^i}{(\sigma^2)_j^i} + \frac{\tilde{\mu}}{\sigma_\mu^2} \right).$$

When $\gamma^i = 1$ the full conditional is normal with mean and variance computed as in section 2.2.4. In the cyclical Gibbs sampler G^i conditional on μ^i is not actually drawn.

Conditional $G^i, \gamma^i | -$

$\gamma^i | -$ is a Bernoulli distribution with probability p^i given by

$$p^i = O^i / (1 + O^i),$$

$$O^i = \frac{\mathcal{N}(y_{1:N^i}^i | \mu^i, \Sigma^i + K^i)}{\mathcal{N}(y_{1:N^i}^i | \mu^i, \Sigma^i)} \frac{\theta}{1 - \theta}, \quad (3.51)$$

where the numerator is computed by forward filtering on ${}^D\mathcal{Y}^i$. G^i is then sampled from the conditional distribution $G^i | \gamma^i, -$ by FFBS on ${}^D\mathcal{Y}^i$.

Conditional $\tilde{\mu}^1, \dots, \tilde{\mu}^{D_n} | \mu^1, \dots, \mu^{D_n}, -$

The priors used are conjugate and so in this case the $\tilde{\mu}_1, \dots, \tilde{\mu}_p$ can be sampled via algorithm 3 of Neal (2000).

Conditional $\theta | \gamma^{1:D_n}, -$

The beta prior is conjugate resulting in the beta full conditional

$$\theta | \gamma^{1:D_n}, - \sim \mathcal{B}e(b_1 + \sum_{i=1}^{D_n} \gamma^i, D_n - \sum_{i=1}^{D_n} \gamma^i + b_2). \quad (3.52)$$

Conditional $p | {}^D\tau^{1:D_n}, -$

The Dirichlet prior is conjugate to the discrete distribution on ${}^D\tau^i$, so the full conditional for p is $Dir(a_1 + c_1, \dots, a_k + c_k)$ where $c_i = \sum_{j=1}^{D_n} 1\{{}^D\tau^j = \tau_i\}$.

3.6 Model Performance on Simulated Data

The behaviour of the model will now be demonstrated on two simulated datasets. In addition to being able to correctly classify dynamic vs static α^i 's for each trial, of

particular interest is the ability to learn properties of the cell, namely, the probability that a new trial will be dynamic and the cell's distribution of μ^i 's and $D\tau^i$'s. With this information one would be able to make quantitative statements about $\alpha^i(\cdot)$ for a new trial. It seems that the distribution of the μ^i 's is learned well, but θ and the distribution over $D\tau^i$ not so well. There is a tendency of the model to over-fit by making trials dynamic when they should be static, and concentrating posterior mass on smaller time-scales. In order to isolate and study this behaviour, the intensities $A\lambda$ and $B\lambda$ are considered known and fixed for the purposes of experimentation.

Consider the following simulated data. 15, 20 and 20 trials are generated for types A, B and D respectively. The A and B single sound trials are realizations from homogeneous Poisson point processes with rates 400 and 100 respectively, observed over the unit interval. Each dual sound trial i is generated from an inhomogeneous Poisson point process where α^i is dynamic with probability 0.5. When dynamic, $\alpha^i(t) = 0.5 + 0.4 \sin(2\pi(\phi_i + t)/T_i)$ where T_i is a period sampled uniformly on the interval $[0.4, 1]$ and ϕ_i is a phase shift sampled uniformly from $[0, T_i]$. When static, $\alpha^i(t) = c_i$ where c_i is drawn randomly from a mixture of two uniform distributions with support $[0.2, 0.3]$ and $[0.7, 0.8]$ with equal weight. In total there are 14 dynamic trials out of 20. The MCMC is run for 8000 iterations following 2000 burnin iterations and every 10th iterate is retained. b_1 and b_2 are taken to be unity so that the prior on θ is uniform. The first comment is that the posterior density for θ , shown in figure 3.1, is more skewed to 1 than expected. Figure 3.2 displays typical posterior summaries for a truthfully dynamic and static trial. In both cases the 95% credible interval correctly covers the true weight function, but the static trials are classified as dynamic. Indeed most posterior draws of weight functions for static trials are dynamic. Moreover, the time-scales for the static trials are also estimated to be small, which is not expected.

Consider the probability vector in the discrete prior on the time-scale, upon

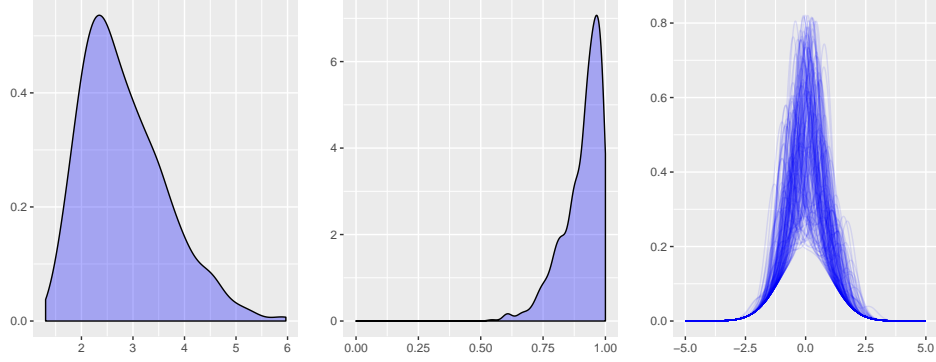


FIGURE 3.1: (Left) Posterior density for ${}^D\rho^2$. (Middle) Posterior density for θ . (Right) Density functions $p(\mu_{D_{n+1}}|\tilde{\mu}_{1:D_n})$ using posterior draws of $\tilde{\mu}_{1:D_n}$

which a Dirichlet prior was placed. Averaging each element across the MCMC draws results in $[0.18, 0.53, 0.19, 0.04, 0.05]$. Now consider the posterior draws of the time-scale for the static trial shown in figure 3.2. The relative proportions of draws is almost identical to the probability vector p in the prior, revealing the the likelihood for the static trials is rather indifferent to the time-scale. When dynamic trials strongly favour short time-scales, and static trials are indifferent toward time-scales, the probability vector p is pulled toward loading on smaller time-scales. Generating new α^i 's from the posterior predictive would result more dynamic weight functions, but would spike-trains generated from the posterior predictive look any different from those upon which the model was trained?

Before assigning too much weight to this analysis, however, there is clearly an issue regarding the mixing of γ^i . This is not unexpected for a data augmented model. This cannot reliably be used to classify trials or for learning a cell's propensity to create dynamic trials through learning the parameter θ . In order to resolve this issue the model was simplified.

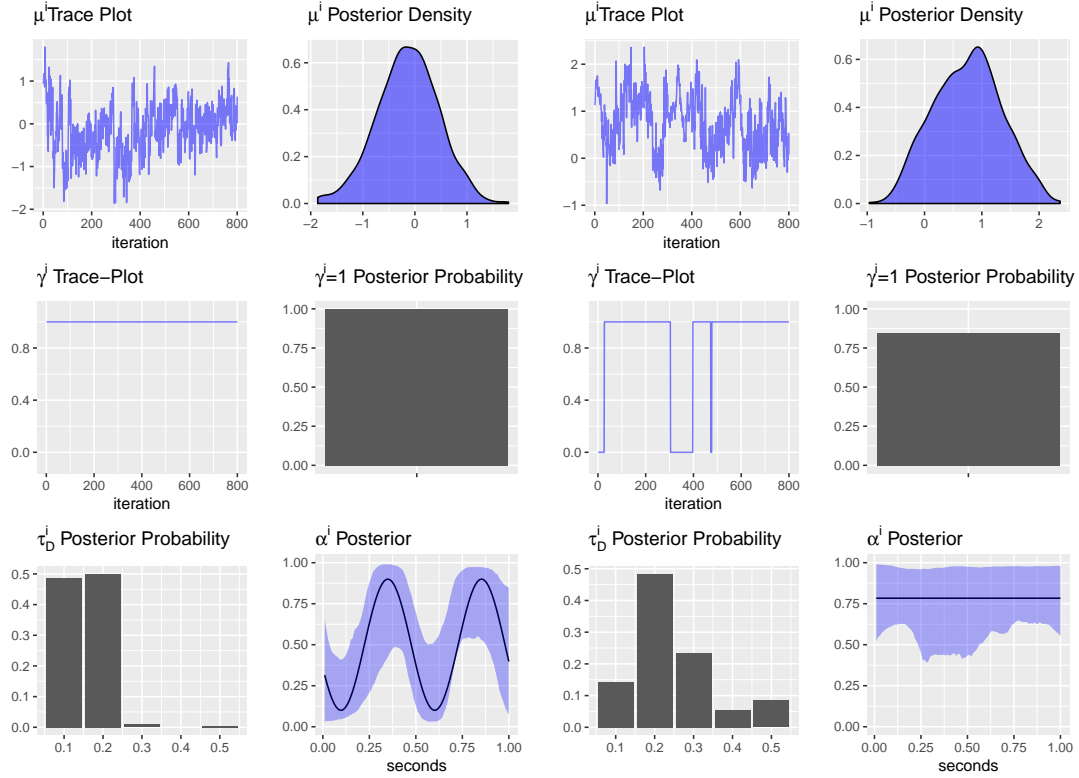


FIGURE 3.2: Posterior summaries of parameters associated with dual sound trial (left two columns) $i = 9$ (right two columns) $i = 14$. The true $\alpha^i(\cdot)$'s are represented with black lines.

3.7 A Simplified Model

The mixture prior on the g^i 's is actually a remnant from an earlier model where the time-scales for each dual trial were a common parameter given a continuous prior. Having moved to a discrete prior on the time-scale, it is considerably simpler to specify a discrete prior on the inverse of the time-scale with a point mass at zero. This would also result in static α^i 's. The Dirichlet prior could then be used to give aprior probability $1/2$ to the point mass at zero, and distribute the remaining mass across the remaining point masses. Learning a cell's propensity to generate flat α^i 's is then possible by considering the posterior over p . Working with a prior on the inverse time-scale with a point mass at zero is a little awkward numerically, and so instead

consider a model which places a discrete prior on the time-scale with a point mass at a very large value. The prior considered is taken, somewhat arbitrarily, to have point masses at $[0.1, 0.2, 0.5, 1, 100]$. It is expected that the posterior probability that $D\tau^i = 100$ is high for the static trials. In addition a new dataset was generated. The conditions are exactly the same as the previous dataset, but is now more balanced with 10 static and 10 dynamic trials.

Figure 3.3 shows typical posterior summaries for a dynamic and static trial. As before, the posterior for the weight functions concentrates well around the true functions but the time-scale is poorly learned. Averaging each element of the probability vector p across MCMC draws results in $[0.03, 0.71, 0.16, 0.07, 0.04]$. The posterior probability of $D\tau^i$ values is also very similar to p for the static trials, as shown in figure 3.2, adding support to the idea that the likelihood function for static trials varies very slowly as a function of the time-scale. At this point, attempting to classify trials as dynamic or static based on the posterior of their time-scale parameter is abandoned.

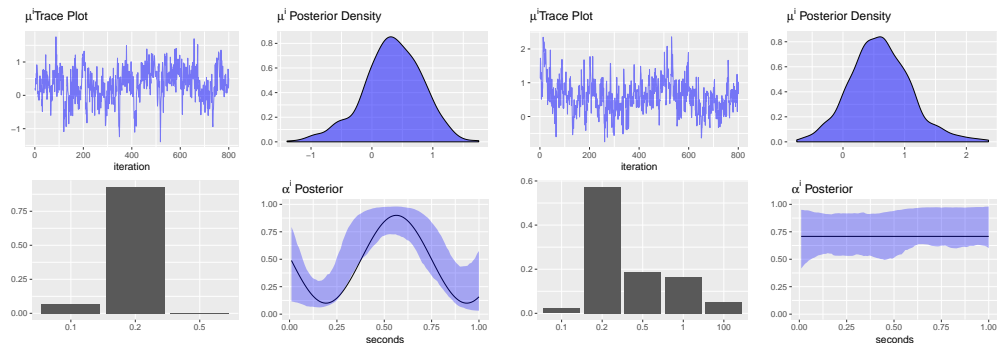


FIGURE 3.3: Posterior summaries of parameters associated with dual sound trial (left) $i = 17$ (right) $i = 2$. The true $\alpha^i(\cdot)$ is shown in black in the bottom right figure.

3.8 Cell YKIC092311_1 609Hz 24° 742Hz -6°

Previous analyses identified this cell under these conditions as exhibiting the most dynamic behaviour among all triplets. The goal is merely to learn the weight functions and not to attempt to classify them as switching or dynamic. In addition, learning the cell's time-scale distribution is abandoned and the probability vector p is fixed with no Dirichlet prior. The discrete prior consists of locations $[0.05, 0.1, 0.2, 0.3, 100]$ with equal probabilities. The two frequencies are 609 and 742 Herz, at locations 24 and -6 degrees respectively. The measured spike trains are shown in figure 3.5. An informative prior on ${}^A\Lambda$ and ${}^B\Lambda$ of $Ga(15, 1/20)$ was specified using domain specific knowledge, putting approximately 0.95 prior probability between 170 and 470. The variance, σ_μ^2 , of the prior for ${}^A\mu$ and ${}^B\mu$ and the base measure in the Dirichlet process was taken to be 0.25. The prior on the GP variance was an inverse Gamma with shape 2 and scale 0.5 truncated to the interval $[0, 4]$. The purpose of constraining the means and variances of the GP are that the function is composed with the link function, and so beyond a point it is unnecessary for the function values to be any larger. The prior on the time-scale of Af and Bf was taken to be a $Ga(5, 15)$ distribution truncated to the interval $[0.05, \infty]$ in order to eliminate very small time-scales which resulting in over-fitting. This puts approximately 0.95 prior probability between 0.1 and 0.7. The MCMC was run for 10000 iterations of which every 10'th iterate was retained. Posterior summaries for ${}^x\mu, {}^x\rho^2, {}^x\tau, {}^xf, {}^x\Lambda$ and ${}^x\lambda$ for $\mathcal{X} \in \{A, B\}$ are shown in figures A.3 and A.4. As is clear from figure A.4, the posterior favours smaller time-scales even for the single sound intensity, hence why the truncation was necessary in the prior to avoid over-fitting. The estimated single sound intensities and dual sound weight functions are shown in figure 3.5. Under this analysis, the flattish nature of the weight functions is most consistent with the intermediary static superposition hypothesis.

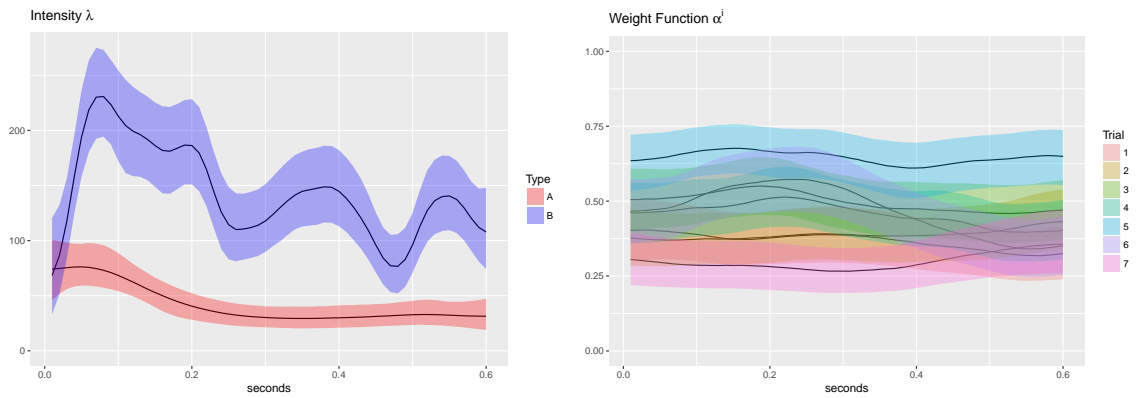


FIGURE 3.4: Posterior summaries for cell YKIC092311.1 609hz 24° 742hz -6° . (Left) Estimated intensity functions $^A\lambda$ and $^B\lambda$. Black lines represent posterior median values and colour bands represent 0.95 posterior probability credible intervals. (Right) Estimated weight functions α^i for each dual trial 1 through 7. Black lines represent posterior median values and colour bands represent 0.5 posterior probability credible intervals.

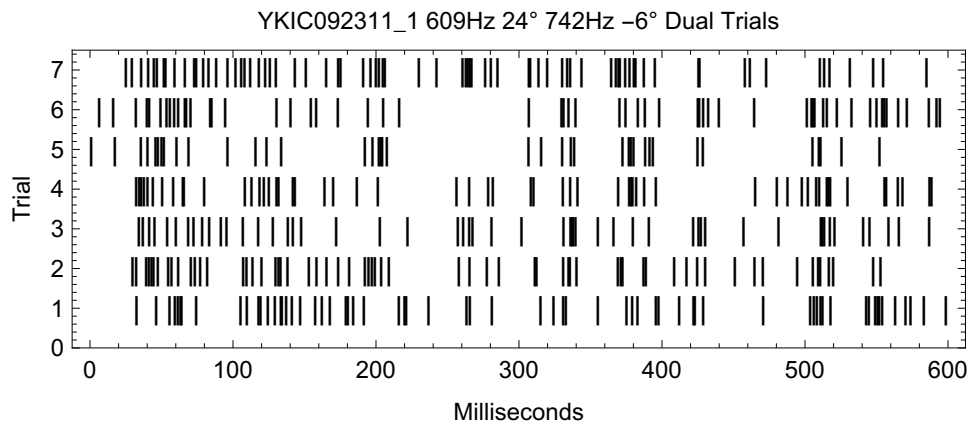
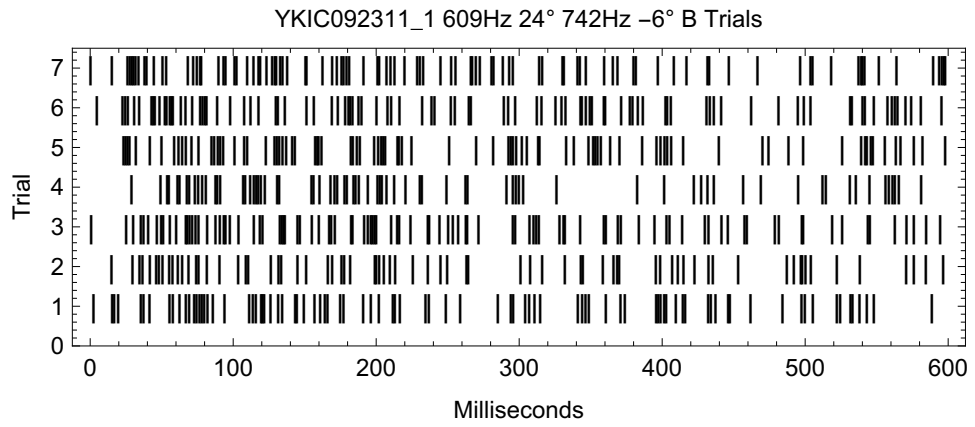
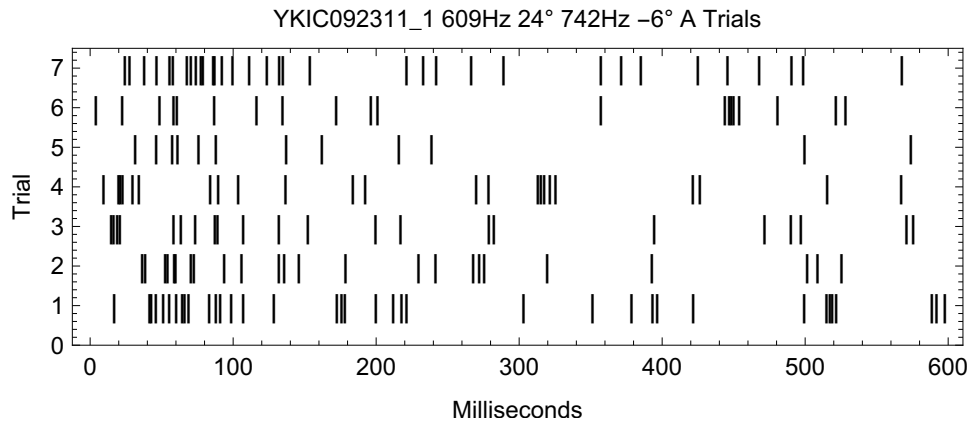


FIGURE 3.5: Raw data for cell YKIC092311_1 609Hz 24° 742Hz -6°

Continuous Optimization for Variable Selection

4.1 Spike and Slab Priors

This chapter is primarily devoted to spike and slab models. Initially introduced by Mitchell and Beauchamp (1988), the spike and slab prior constituted a prior on the regression coefficients which was a mixture of a point mass at zero and a diffuse uniform prior. These two-component mixture priors have received a lot of attention in the variable selection literature (see Ishwaran and Rao (2005), George and McCulloch (1993), Clyde et al. (1996)) and remain popular to date. The specific model that is studied in this chapter is the following spike and slab model

$$\begin{aligned}
 Y|\beta &\sim \mathcal{N}(X\beta, \sigma^2) \\
 \beta_i|\gamma_i &\sim \gamma_i \mathcal{DE}(\lambda_1/\sigma) + (1 - \gamma_i)\delta_0 \\
 \gamma_i &\sim \text{Bern}(\theta),
 \end{aligned}
 \tag{4.1}$$

although many of the following algorithms are also valid for other choices of slab density. Initially θ is considered a fixed hyperparameter, but this assumption is relaxed later. The properties of the posterior resulting from this prior have been studied

in Castillo et al. (2015). This chapter considers the MAP (maximum a posteriori) estimator of (4.1). It is organized as follows. The chapter begins with a study of the theoretical properties of this estimator, generalizing existing results for lasso based on ℓ_2 -regularity conditions of the design matrix quantified by the *compatibility number* (see Buehlmann and van de Geer (2011)). Thereafter, an alternative approach deriving similar results based on the general theory of Zhang and Zhang (2012) is provided. After the initial theory the remaining focus is on optimization methods for computing this estimator. This chapter focusses only on continuous optimization methods, which provide only a locally optimal solution. This leads to an interesting survey of algorithms that have arisen in different disciplines. In particular four algorithms are studied; the proximal gradient method, orthogonal data augmentation, LQ decompositions of the design matrix and location mixtures of spherical normals which exploit the separability of the penalty. All of these algorithms shall prove to result in identical parameter mappings. The next chapter considers discrete optimization, algorithms which use the local solutions found from algorithms in this chapter as starting points to find the globally optimal solution.

4.2 Theory Based on Compatibility Condition

Optimization Formulation

The problem of finding the global posterior mode of equation (4.1) can be written as the following mathematical program

$$\begin{aligned}
 & \underset{\beta, \gamma}{\text{Minimize}} && F(\beta, \gamma) := \frac{1}{2\sigma^2} \|Y - X\beta\|_2^2 - \left(\log \left(\frac{\theta}{1-\theta} \right) + \log \left(\frac{\lambda_1}{\sigma} \right) \right) \sum_i \gamma_i + \frac{\lambda_1}{\sigma} \|\beta\|_1 \\
 & \text{subject to} && \gamma \in \{0, 1\}^p, \\
 & && (\beta_i \neq 0 \wedge \gamma_i = 1) \vee (\beta_i = 0 \wedge \gamma_i = 0),
 \end{aligned}
 \tag{4.2}$$

which corresponds to minimizing the negative of the log-posterior. This in turn can be rewritten in the form of a penalized regression

$$L(\beta) := \frac{1}{2\sigma^2} \|Y - X\beta\|_2^2 + \lambda_0 \|\beta\|_0 + \frac{\lambda_1}{\sigma} \|\beta\|_1, \quad (4.3)$$

where $\lambda_0 = -(\log(\theta/(1-\theta)) + \log(\lambda_1/\sigma))$. This clearly contains both ℓ_0 and ℓ_1 (lasso) regularized least squares as special cases. This model is of interest to Bayesian statisticians who wish to model sparsity through the prior $p(\gamma)$, and apply modest shrinkage to the non-zero coefficients through the Laplace slab distribution. As the lasso is obtained through the special case of $\lambda_0 = 0$, it is interesting to see how available results for lasso can be generalized for (4.3).

Generalizing Results for Lasso to Spike and Slab

In the literature concerning theoretical results of the lasso it is actually more common to study the following formulation

$$\hat{\beta} := \arg \min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|_2^2/2n + \lambda'_0 \|\beta\|_0 + \lambda'_1 \|\beta\|_1, \quad (4.4)$$

where $\lambda'_0 = \sigma^2 \lambda_0/n$ and $\lambda'_1 = \sigma \lambda_1/n$. This is simply rescaling equation (4.3) by (σ^2/n) . I follow this approach only for the sake of continuity in some of the constants that have been derived in other works and I shall drop the primes on λ'_0 and λ'_1 here-out. The following results assume that the truth is linear, i.e. that $Y = X\beta^0 + \varepsilon$ for some true but unknown β^0 , with noise terms $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ for a known variance σ^2 . Moreover it is assumed that the columns of the design matrix have been scaled as $\|x_i\|_2 = \sqrt{n}$ for each column $i \in \{1, \dots, p\}$. It follows immediately from the definition of $\hat{\beta}$ in (4.4) that

$$\|Y - X\hat{\beta}\|_2^2/2n + \lambda_0 \|\hat{\beta}\|_0 + \lambda_1 \|\hat{\beta}\|_1 \leq \|Y - X\beta^0\|_2^2/2n + \lambda_0 \|\beta^0\|_0 + \lambda_1 \|\beta^0\|_1. \quad (4.5)$$

Equation (4.5) can be rearranged to yield the commonly named *basic inequality*

$$\|X(\hat{\beta} - \beta^0)\|_2^2/2n + \lambda_0 \|\hat{\beta}\|_0 + \lambda_1 \|\hat{\beta}\|_1 \leq \varepsilon^T X(\hat{\beta} - \beta^0)/n + \lambda_0 \|\beta^0\|_0 + \lambda_1 \|\beta^0\|_1, \quad (4.6)$$

which isolates the only random term on the right hand side. Buehlmann and van de Geer (2011) call this the *empirical process* term. One approach to deal with this is to bound the empirical process term from above with high probability. Indeed, the extended Hölders inequality can be used to provide an upper bound as $|\epsilon^T X(\hat{\beta} - \beta^0)| \leq \|X^T \epsilon\|_\infty \|\hat{\beta} - \beta^0\|_1$. The next step is to make suitable assumptions on the noise term to bound $\|X^T \epsilon\|_\infty$ with high probability. This can be achieved by assuming the noise variables are *sub-Gaussian* and using the tail probabilities to get the bound.

Definition 1. A random variable $X \in \mathbb{R}$ is said to be *sub-Gaussian with variance proxy* σ^2 , denoted $X \sim \text{subG}(\sigma^2)$, if $\mathbb{E}[X] = 0$ and

$$\mathbb{E}[\exp(sX)] \leq \exp\left(\frac{\sigma^2 s^2}{2}\right) \quad (4.7)$$

Corollary 2. Let $X \sim \text{subG}(\sigma^2)$, then for any $t > 0$

$$\mathbb{P}[|X| > t] \leq 2 \exp(-t^2/(2\sigma^2)) \quad (4.8)$$

Proof. Choose any $s > 0$. The result can be obtained using the Chernoff bound as follows

$$\begin{aligned} \mathbb{P}[X > t] &= \mathbb{P}[\exp(Xs) > \exp(ts)] \\ &\leq \frac{\mathbb{E}[\exp(Xs)]}{\exp(ts)} \quad (\text{Markov's Inequality}) \\ &\leq \exp(s^2\sigma^2/2 - ts) \quad (\text{Sub-Gaussian}). \end{aligned} \quad (4.9)$$

The last line holds for any $s > 0$ and so the bound can be tightened by using the value of s for which the right hand side achieves a minimum. This occurs at $\hat{s} = -t/2\sigma^2$, which provides an upper bound on the right hand side of $\exp(-t^2/(2\sigma^2))$. The result then follows from $\mathbb{P}[|X| > t] = \mathbb{P}[X > t] + \mathbb{P}[X < -t]$ \square

Corollary 3. Let X_1, \dots, X_p denote independent $\text{subG}(\sigma^2)$ random variables and let $a \in \mathbb{R}^p$ be an arbitrary vector, then $\sum_{i=1}^p a_i x_i \sim \text{subG}(\sigma^2 \|a\|_2^2)$.

Proof.

$$\begin{aligned}
\mathbb{E}[\exp(sa^T X)] &= \prod_{i=1}^p \mathbb{E}[\exp(sa_i x_i)] \quad (\text{Independence}) \\
&\leq \prod_{i=1}^p \exp(s^2 a_i^2 \sigma^2 / 2) \quad (\text{Sub-Gaussian}) \\
&= \exp(s^2 \|a\|_2^2 / 2)
\end{aligned} \tag{4.10}$$

□

With the definition of sub-Gaussian random variables in mind attention can be restricted to a set upon which the empirical process term is bounded with high probability.

Lemma 4. *Let $\mathcal{F} = \{\|X^T \epsilon\|_\infty / n \leq \mathcal{T}\}$ where $\epsilon_1, \dots, \epsilon_p$ are independent $\text{subG}(\sigma^2)$ random variables and $\mathcal{T} = \sigma \sqrt{\frac{t^2 + 2 \log p}{n}}$, then*

$$\mathbb{P}[\mathcal{F}] \geq 1 - 2 \exp(-t^2/2) \tag{4.11}$$

Proof.

$$\begin{aligned}
1 - \mathbb{P}[\mathcal{F}] &= \mathbb{P}[\max |X_i^T \epsilon| \geq n\mathcal{T}] \\
&\leq \sum_{i=1}^p \mathbb{P}[|X_i^T \epsilon| \geq n\mathcal{T}] \quad (\text{Union Bound}) \\
&\leq 2p \exp(-n^2 \mathcal{T}^2 / (2n\sigma^2)) \quad (\text{Sub-Gaussian Tail Probability}) \\
&= 2 \exp(-t^2/2)
\end{aligned} \tag{4.12}$$

□

It is now possible to state the main result.

Theorem 5. *Let $Y = X\beta^0 + \epsilon$ where $\epsilon_1, \dots, \epsilon_n$ are independent $\text{subG}(\sigma^2)$ random variables and X is such that the columns have been scaled to have ℓ_2 -norm of \sqrt{n} .*

Let $\hat{\beta}$ defined as in equation (4.4) and let $S_0 = \{i : \beta_i^0 \neq 0\}$. With $\lambda_1 \geq 2\sigma\sqrt{\frac{t^2+2\log p}{n}}$ the following bounds

$$\|\hat{\beta}\|_0 \leq \left(1 + \frac{\lambda_1^2}{2\lambda_0\phi^2(S_0; \lambda_0, \lambda_1)}\right) |S_0|, \quad (4.13)$$

$$\|\hat{\beta} - \beta^0\|_1 \leq \left(\frac{2\lambda_0}{\lambda_1} + \frac{\lambda_1}{\phi^2(S_0; \lambda_0, \lambda_1)}\right) |S_0|, \quad (4.14)$$

$$\|X(\hat{\beta} - \beta^0)\|_2^2/n \leq \frac{D}{D-1} \left(\frac{\lambda_1^2 D}{\phi^2(S_0; \lambda_0, \lambda_1)} + 2\lambda_0\right) |S_0| \quad \forall D > 1, \quad (4.15)$$

hold on a set \mathcal{F} with $\mathbb{P}[\mathcal{F}] \geq 1 - 2\exp(-t^2/2)$, where $\phi^2(S_0; \lambda_0, \lambda_1)$ is a modified compatibility number defined as

$$\begin{aligned} \phi(S; \lambda_0, \lambda_1) = \\ \inf \left\{ \frac{|S|^{1/2}\|Xu\|_2}{n^{1/2}\|u_S\|_1} : 2\lambda_0\|u_{S^c}\|_0 + \lambda_1\|u_{S^c}\|_1 \leq 3\lambda_1\|u_S\|_1 + 2\lambda_0\|u_S\|_0 \right\}, \end{aligned} \quad (4.16)$$

Proof. The set $\mathcal{F} = \{\|X^T\epsilon\|_\infty/n \leq \sigma\sqrt{\frac{t^2+2\log p}{n}}\}$ has a probability of at least $1 - 2\exp(-t^2/2)$ by lemma 4. On \mathcal{F} the empirical process term can be bounded in the basic inequality given by (4.6), yielding

$$\|X(\hat{\beta} - \beta^0)\|_2^2/n + 2\lambda_0\|\hat{\beta}\|_0 + 2\lambda_1\|\hat{\beta}\|_1 \leq \lambda_1\|\hat{\beta} - \beta^0\|_1 + 2\lambda_0\|\beta^0\|_0 + 2\lambda_1\|\beta^0\|_1. \quad (4.17)$$

The $l1$ norms in equation (4.17) can be manipulated through the observation that

$$\|\hat{\beta}\|_1 = \|\hat{\beta}_{S_0}\|_1 + \|\hat{\beta}_{S_0^c}\|_1 \geq \|\beta_{S_0}^0\|_1 - \|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1 + \|\hat{\beta}_{S_0^c}\|_1, \quad (4.18)$$

$$\|\hat{\beta} - \beta^0\|_1 = \|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1 + \|\hat{\beta}_{S_0^c}\|_1, \quad (4.19)$$

to give

$$\begin{aligned} & \|X(\hat{\beta} - \beta^0)\|_2^2/n + 2\lambda_0\|\hat{\beta}\|_0 + 2\lambda_1\|\beta_{S_0}^0\|_1 - 2\lambda_1\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1 + 2\lambda_1\|\hat{\beta}_{S_0^c}\|_1 \\ & \leq \lambda_1\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1 + \lambda_1\|\hat{\beta}_{S_0^c}\|_1 + 2\lambda_0\|\beta^0\|_0 + 2\lambda_1\|\beta^0\|_1. \end{aligned} \quad (4.20)$$

This simplifies to

$$\begin{aligned} & \|X(\hat{\beta} - \beta^0)\|_2^2/n + 2\lambda_0\|\hat{\beta}\|_0 + \lambda_1\|\hat{\beta}_{S_0^c}\|_1 \\ & \leq 3\lambda_1\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1 + 2\lambda_0\|\beta^0\|_0, \end{aligned} \quad (4.21)$$

providing the inequality

$$2\lambda_0\|\hat{\beta}\|_0 + \lambda_1\|\hat{\beta}_{S_0^c}\|_1 \leq 3\lambda_1\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1 + 2\lambda_0\|\beta^0\|_0. \quad (4.22)$$

Because $\beta_{S_0^c}^0 = 0$ equation (4.22) can be written as

$$2\lambda_0\|\hat{\beta}_{S_0^c} - \beta_{S_0^c}^0\|_0 + \lambda_1\|\hat{\beta}_{S_0^c} - \beta_{S_0^c}^0\|_1 \leq 3\lambda_1\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1 + 2\lambda_0\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_0. \quad (4.23)$$

This shows that the estimation error $\Delta := \hat{\beta} - \beta^0$ satisfies $2\lambda_0\|\Delta_{S_0^c}\|_0 + \lambda_1\|\Delta_{S_0^c}\|_1 \leq 3\lambda_1\|\Delta_{S_0}\|_1 + 2\lambda_0\|\Delta_{S_0}\|_0$. Returning to equation (4.21) and substituting equation (4.19) gives

$$\begin{aligned} & \|X(\hat{\beta} - \beta^0)\|_2^2/n + 2\lambda_0\|\hat{\beta}\|_0 + \lambda_1\|\hat{\beta} - \beta^0\|_1 \\ & = \|X(\hat{\beta} - \beta^0)\|_2^2 + 2\lambda_0\|\hat{\beta}\|_0 + \lambda_1\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1 + \lambda_1\|\hat{\beta}_{S_0^c}\|_1 \\ & \leq 4\lambda_1\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1 + 2\lambda_0\|\beta^0\|_0 \end{aligned} \quad (4.24)$$

At this point, one might try to rid of the $\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1$ term on the RHS by incorporating it somehow into the $\|X(\hat{\beta} - \beta^0)\|_2^2/n$ term on the LHS. The Cauchy-Schwarz inequality could be used to bound $\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1$ from above by $\sqrt{S_0}\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_2$, but this still needs to be related to $\|X(\hat{\beta} - \beta^0)\|_2$. It is for this reason that the modified compatibility criterion is required. As seen from equation (4.23), the estimation error $\Delta \in \{u \in \mathbb{R}^p : 2\lambda_0\|u_{S_0^c}\|_0 + \lambda_1\|u_{S_0^c}\|_1 \leq 3\lambda_1\|u_{S_0}\|_1 + 2\lambda_0\|u_{S_0}\|_0\}$, and so it follows that $\|\hat{\beta}_{S_0} - \beta_{S_0}^0\|_1 \leq |S_0|^{1/2}\|X\hat{\beta} - X\beta^0\|_2/(n^{1/2}\phi(S_0; \lambda_0; \lambda_1))$. Equation 4.26 then becomes

$$\begin{aligned} & \|X(\hat{\beta} - \beta^0)\|_2^2/n + 2\lambda_0\|\hat{\beta}\|_0 + \lambda_1\|\hat{\beta} - \beta^0\|_1 \\ & \leq 4\lambda_1 \frac{|S_0|^{1/2}\|X(\hat{\beta} - \beta^0)\|_2}{n^{1/2}\phi(S; \lambda_0, \lambda_1)^{1/2}} + 2\lambda_0\|\beta^0\|_0 \end{aligned} \quad (4.25)$$

From the identity $4uv \leq u^2 + v^2$ it follows that

$$\left(1 - \frac{1}{D}\right) \|X(\hat{\beta} - \beta^0)\|_2^2/n + 2\lambda_0 \|\hat{\beta}\|_0 + \lambda_1 \|\hat{\beta} - \beta^0\|_1 \leq \left(\frac{\lambda_1^2 D}{\phi^2(S; \lambda_0, \lambda_1)} + 2\lambda_0\right) |S_0|, \quad (4.26)$$

for any $D > 0$. D can be chosen to tighten the bounds. In particular for $D = 1$

$$\|\hat{\beta}\|_0 \leq \left(1 + \frac{\lambda_1^2}{2\lambda_0 \phi^2(S; \lambda_0, \lambda_1)}\right) |S_0| \quad (4.27)$$

and

$$\|\hat{\beta} - \beta^0\|_1 \leq \left(\frac{2\lambda_0}{\lambda_1} + \frac{\lambda_1}{\phi^2(S; \lambda_0, \lambda_1)}\right) |S_0|. \quad (4.28)$$

For $D \geq 1$

$$\|X(\hat{\beta} - \beta^0)\|_2^2/n \leq \frac{D}{D-1} \left(\frac{\lambda_1^2 D}{\phi^2(S; \lambda_0, \lambda_1)} + 2\lambda_0\right) |S_0|. \quad (4.29)$$

□

Equations (4.14) and (4.15) are direct generalizations of results for lasso under the same set of assumptions for the spike and slab prior. In particular with $\lambda_0 = 0$, $D = 1$ and $D = 4$ they reduce to inequalities found in theorem 6.1 of Buehlmann and van de Geer (2011). Theorem 5 provides an additional bound, however, on the sparsity of the estimated regression vector, which is not present in the analysis for the lasso. It is known that lasso typically over-estimates the number of active predictors, and so having an upper bound on the sparsity of the estimator is useful. The new modified compatibility number also warrants some discussion. When $\lambda_0 = 0$, the modified compatibility number reduces to the original compatibility number present in other works such as in Buehlmann and van de Geer (2011) and Castillo et al. (2015). The original compatibility number $\phi(S; 0, \lambda_1)$ is actually independent of λ_1 as it cancels out, and so for the sake of simplicity the original compatibility number can be denoted $\phi(S; 0, 1)$. There are two relevant questions. The first is how does $\phi(S; \lambda_0, \lambda_1)$ relate

to $\phi(S; 0, 1)$. The second question is how does $\phi(S; \lambda_0, \lambda_1)$ behave as λ_0 and λ_1 are varied. In particular, one would like to choose a specific form of λ_0 and λ_1 in terms of n and p to try and recover an oracle inequality, but as $\phi(S_0; \lambda_0, \lambda_1)$ appears in the bounds in theorem 5 it is not clear how to do this.

Recall that the compatibility number was motivated by a desire to bound the ℓ_1 norm of the estimation error $\hat{\beta}_{S_0} - \beta_{S_0}^0$ in terms of $\|X\hat{\beta} - X\beta^0\|_2^2$. In this sense the compatibility number corresponds to an ℓ_2 -regularity condition on the design matrix. For a given set S it might be possible to find a constant c such that $\|u_S\|_1 \leq |S|^{1/2}\|Xu\|_2^2/(n^{1/2}c)$ for all $u \in \mathbb{R}^p$, but for this to hold for all $u \in \mathbb{R}^p$ the constant c would likely need to be very small. The analysis for Lasso shows, however, that $\hat{\beta}_{S_0} - \beta_{S_0}^0 \in O(S_0) \subset \mathbb{R}^p$ where $O(S) = \{u : \|u_{S^c}\|_1 \leq 3\|u_S\|_1\}$, and so one only needs a constant c such that $\|u_{S_0}\|_1 \leq |S_0|^{1/2}\|Xu\|_2^2/(n^{1/2}c)$ holds for all vectors in $O(S_0)$. As $O(S_0)$ is a subset of \mathbb{R}^p , then the constant does not to be as small. This is the original compatibility number. The analysis for the point-mass-Laplace mixture prior, however, showed that $\hat{\beta}_{S_0} - \beta_{S_0}^0 \in M(S_0; \lambda_0, \lambda_1) \subset \mathbb{R}^p$, where $M(S; \lambda_0, \lambda_1) = \{u : 2(\lambda_0/\lambda_1)(|S^c| - |S|) + \|u_{S^c}\|_1 \leq 3\|u_S\|_1\}$. Under the assumption that the true vector β^0 is more sparse than it is dense i.e. $|S_0| < |S_0^c|$, then $M(S_0; \lambda_0, \lambda_1) \subseteq O(S_0)$. To see this let $x \in M(S_0; \lambda_0, \lambda_1)$. Because the term $2(\lambda_0/\lambda_1)(|S^c| - |S|)$ is positive by assumption, it follows that $\|x_{S^c}^c\|_1 \leq 3\|x_S\|_1$ and so $x \in O(S_0)$. The modified compatibility number results in a tighter bound therefore, because $\|u_{S_0}\|_1 \leq |S_0|^{1/2}\|Xu\|_2^2/(n^{1/2}c)$ need only hold for u in the smaller set $M(S_0; \lambda_0, \lambda_1)$. Alternatively it is clear from the definition that $\phi(S_0; \lambda_0, \lambda_1) \geq \phi(S_0; 0, 1)$ as the infimum is over a smaller set. Increasing λ_0 decreases the size of $M(S_0; \lambda_0, \lambda_1)$ and the value of $1/\phi(S_0; \lambda_0, \lambda_1)$. The implications for the bounds in theorem 5 then are that there is a trade-off between increasing λ_0 and decreasing $1/\phi(S_0; \lambda_0, \lambda_1)$. In addition, $1/\phi(S_0; \lambda_0, \lambda_1) \leq 1/\phi(S_0; 0, 1)$ when $|S_0| < |S_0^c|$ allows the following result to be stated.

Theorem 6. *Under the assumptions of theorem 5 with $\lambda_1 = 2\sigma\sqrt{2\tau\log(p)/n}$, $\lambda_0 = \sigma^2(\log p)/n$ and the additional assumption that $|S_0| < |S_0^c|$, then $\hat{\beta}$ achieves the oracle inequality*

$$\|X(\hat{\beta} - \beta^0)\|_2^2/n \leq \text{const.} \frac{\log p}{n} \sigma^2 |S_0|, \quad (4.30)$$

on a set \mathcal{F} with $\mathbb{P}[\mathcal{F}] \geq 1 - 2\exp(-(\tau - 1)\log p)$.

Proof. From theorem 5 with $t^2 = (2\tau - 2)\log p$

$$\begin{aligned} \|X(\hat{\beta} - \beta^0)\|_2^2/n &\leq \frac{D}{D-1} \left(\frac{\lambda_1^2 D}{\phi^2(S; \lambda_0, \lambda_1)} + 2\lambda_0 \right) |S_0| \\ &\leq \frac{D}{D-1} \left(\frac{4(2\tau \log p)D}{n\phi^2(S; 0, 1)} + 2\frac{\log p}{n} \right) \sigma^2 |S_0| \\ &\leq \text{const.} \frac{\log p}{n} \sigma^2 |S_0|. \end{aligned}$$

□

4.3 Theory Based on η -Null Consistency Condition

An alternative approach to understanding the theoretical properties of the point estimator obtained from (4.4) is in the context of the general theory for concave regularizers by Zhang and Zhang (2012). This was the approach used by Rockova and George (2016) to establish theoretical properties for their spike and slab lasso prior, which differs from the prior studied here as another Laplace distribution is used instead of a point mass at zero. As there are many popular regularizers in the modern literature, Zhang and Zhang (2012) derive general results that hold under mild assumptions of the regularizer used. If the regularizer used is denoted $g(x)$, the authors assume that

1. $g(0) = 0$ (zero)
2. $g(-x) = g(x)$ (symmetric)
3. $g(x) \leq g(x + y) \forall x, y > 0$ (nondecreasing)
4. $g(x + y) \leq g(x) + g(y)$ (subadditive).

These assumptions hold true for ℓ_0 and ℓ_1 penalties and for combinations of them. In addition, it is assumed that the regularizer satisfies the η -null consistency (η -NC) condition

Definition 7. Let $\eta \in (0, 1]$. A regularizer g satisfies the η -NC condition if

$$\min_{\beta \in \mathbb{R}^p} \|\epsilon/\eta - X\beta\|_2^2/(2n) + g(\beta) = \|\epsilon/\eta\|_2^2/(2n). \quad (4.31)$$

The 1 - NC condition means that if the true regression vector $\beta^0 = 0$, then the global optimizer $\hat{\beta} = 0$. I am using a shorthand where if the argument of g is a vector, like $g(\beta)$, this is understood to mean $\sum_{i=1}^p g(\beta_i)$. It should be clear from the context depending on whether the argument is a vector or scalar. Given a regularizer g , the threshold level is defined as

$$\lambda^* = \inf_{t>0} \left\{ \frac{t}{2} + \frac{g(t)}{t} \right\}, \quad (4.32)$$

named so because the solution to $\arg \min_t \{(z - t)^2/2 + g(t)\} = 0$ iff $|z| \leq \lambda^*$. For the point-mass-Laplace mixture prior, the threshold level $\lambda^* = \sqrt{2\lambda_0} + \lambda_1$. The assumptions on the design matrix are that the *restricted invertibility factor* is non-zero, defined as

Definition 8. For $q \geq 1$, $S \subset \{1, \dots, p\}$, and $\xi > 0$, the *restricted invertibility factor* is defined as

$$RIF_q(\xi, S) = \inf \left\{ \frac{|S|^{1/q} \|X^T X u\|_\infty}{n \|u\|_q} : g(u_{S^c}) < \xi g(u_S) \right\}. \quad (4.33)$$

In order to import the results of Zhang and Zhang (2012), all that must be shown is that the η -NC condition holds with high probability for $g(\beta) = \lambda_0 \|\beta\|_0 + \lambda_1 \|\beta\|_1$.

Theorem 9. *Let the penalty $g(\beta) = \lambda_0 \|\beta\|_0 + \lambda_1 \|\beta\|_1$ with $\lambda_1 \geq \frac{\sigma}{\eta} \sqrt{\frac{t^2 + 2 \log p}{n}}$. Assume $\epsilon_1, \dots, \epsilon_n$ are independent subG(σ^2) random variables. Then the η -NC condition holds with probability at least $1 - 2 \exp(-t^2/2)$.*

Proof. Let $\hat{\beta} = \arg \min\{(1/2n)\|\epsilon/\eta - X\beta\|_2^2 + g(\beta)\}$. Then by definition

$$(1/2n)\|\epsilon/\eta - X\hat{\beta}\|_2^2 + g(\hat{\beta}) \leq (1/2n)\|\epsilon/\eta - X\beta\|_2^2 + g(\beta) \quad \forall \beta \in \mathbb{R}^p \quad (4.34)$$

As the RHS holds for all $\beta \in \mathbb{R}^p$, it also holds for $\beta = 0_p$, the vector of zeros. It follows that

$$(1/2n)\|\epsilon/\eta - X\hat{\beta}\|_2^2 + g(\hat{\beta}) \leq (1/2n)\|\epsilon/\eta\|_2^2. \quad (4.35)$$

It remains to show that LHS \geq RHS.

$$\begin{aligned} (1/2n)\|\epsilon/\eta - X\hat{\beta}\|_2^2 + g(\hat{\beta}) &= (1/2n)\|\epsilon/\eta\|_2^2 + (1/2n)\hat{\beta}^T X^T X \hat{\beta} - \epsilon^T X \hat{\beta}/(n\eta) + g(\hat{\beta}) \\ &\geq (1/2n)\|\epsilon/\eta\|_2^2 - |\epsilon^T X \hat{\beta}/(n\eta)| + g(\hat{\beta}) \\ &\geq (1/2n)\|\epsilon/\eta\|_2^2 - \|\epsilon^T X/(n\eta)\|_\infty \|\hat{\beta}\|_1 + g(\hat{\beta}). \end{aligned}$$

It follows from the assumption of sub-Gaussian errors and the form of λ_1 that $\mathbb{P}[\|\epsilon^T X/(n\eta)\|_\infty \leq \lambda_1] \geq 1 - 2 \exp(-t^2/2)$. Therefore

$$\begin{aligned} (1/2n)\|\epsilon/\eta - X\hat{\beta}\|_2^2 + g(\hat{\beta}) &\geq (1/2n)\|\epsilon/\eta\|_2^2 - \lambda_1 \|\hat{\beta}\|_1 + \lambda_1 \|\hat{\beta}\|_1 + \lambda_0 \|\hat{\beta}\|_0 \\ &\geq (1/2n)\|\epsilon/\eta\|_2^2, \end{aligned}$$

with probability at least $1 - 2 \exp(-t^2/2)$. □

With the definitions of the threshold λ^* , the restricted invertibility factor $RIF_q(\xi, S)$ and conditions under when the η -NC condition is expected to hold with high probability it is possible to state the main result

Theorem 10. Let $\hat{\beta}$ be defined as in equation (4.4), β^0 the true regression vector, $S_0 = \{i : \beta_i^0 \neq 0\}$, $Y = X\beta + \epsilon$ where $\epsilon_1, \dots, \epsilon_n$ are independent subG(σ^2) random variables, $\eta \in (0, 1]$, $\xi = (1 + \eta)/(1 - \eta)$ and $\lambda_1 \geq \frac{\sigma}{\eta} \sqrt{\frac{t^2 + 2 \log p}{n}}$. Then for all $q \geq 1$ the following bounds

$$\|\hat{\beta} - \beta^0\|_q \leq \frac{(1 + \eta)}{RIF_q(\xi, S_0)} (\sqrt{2\lambda_0} + \lambda_1) |S_0|^{1/q}, \quad (4.36)$$

$$\|X\hat{\beta} - X\beta^0\|_2^2/n \leq 2\xi \left(2 \vee \frac{(1 + \eta)}{RIF_1(\xi, S_0)} \right) (\sqrt{2\lambda_0} + \lambda_1)^2 |S_0|, \quad (4.37)$$

hold with probability at least $1 - 2 \exp(-t^2/2)$.

The proof uses theorem 9 to assume that the η -NC condition holds with high probability, the rest is then a direct consequence of Theorem 1 of Zhang and Zhang (2012). Obtaining a bound on the sparsity of $\hat{\beta}$ requires more work, but is possible through theorem 2 of Zhang and Zhang (2012). Setting $\lambda_0 = (\sigma^2 \log p)/(\eta^2 n)$ attains the oracle inequality with high probability as before. The next section switches attention to optimization algorithms for obtaining locally optimal solutions to problem (4.3).

4.4 Continuous Optimization Methods

4.5 Introduction

The optimization problem (4.3) is a member of the following class of problems

$$\text{minimize } F(\beta) := f(\beta) + g(\beta), \quad (4.38)$$

where $f(\beta)$ is a differentiable term and $g(\beta)$ is separable penalty, meaning that $g(\beta) = \sum_{i=1}^p h_i(\beta_i)$ for functions h_1, \dots, h_p , possibly non-differentiable, possibly discontinuous and possibly non-convex. For the point-mass-Laplace prior $g(\beta) = \lambda_0 \|\beta\|_0 + \lambda_1 \|\beta\|_1 = \sum_{i=1}^p \lambda_0 1\{\beta_i \neq 0\} + \lambda_1 |\beta_i|$. I will use the same function name g for

the univariate case as it should be clear from the context of whether the argument is a vector or scalar. The MAP estimate of logistic regression models under generalized double Pareto priors of Armagan et al. (2013) is another example of an optimization problem that falls within this class. The possibly non-smooth, possibly discontinuous term presents significant challenges and in general an analytic solution to equation (4.38) will not be available. The primary structural property that is exploited by most algorithms is the separability of the penalty. This is useful when the solution to the univariate or orthogonal case is analytically tractable, provided by the proximal operator of the penalty defined as

$$\text{prox}_{tg}(x) = \arg \min_y \left\{ \frac{1}{2t}(y - x)^2 + g(y) \right\}. \quad (4.39)$$

A catalogue of penalties and their proximal operators can be found in the review paper by Polson et al. (2015). When $f(\beta)$ corresponds to a univariate regression, equation (4.38) can be re-expressed in the form required by equation (4.39), and so the minimizer can be computed through use of the proximal operator. When $f(\beta)$ corresponds to a p – *dimensional* orthogonal regression, (4.38) reduces to p *independent* univariate regressions. Therefore, assuming that g has a tractable proximal operator, there are at least two immediate ways of tackling the minimization in equation (4.38). The first approach is to condition on all but one variable and solve the resulting univariate conditional minimization problem. This is then repeated for other variables in a cyclic, greedy or random way. The second approach is to approximate the function $f(\beta)$ by a separable function $\tilde{f}(\beta)$ and solve the p resulting univariate minimization problems simultaneously.

Coordinate descent is an algorithm in the former category. It was largely ignored until Friedman et al. (2007) found it to be effective in ℓ_1 –regularized problems. It remains competitive and is implemented in the popular R package glmnet for fitting lasso constrained linear models described in Friedman et al. (2010). Non-asymptotic

rates of convergence for this method, when applied to convex problems, are given in Saha and Tewari (2013). A closely related algorithm is greedy coordinate descent, where the variable to be updated is chosen in a greedy manner, also known as Gauss-Southwell selection. The variable to update is selected as the one out of p candidates for which solving the univariate minimization problem provides the greatest reduction in the objective. The greedy coordinate descent is known to converge faster than random coordinate descent for convex problems, see Nutini et al. (2015), but requires p more computation per iteration.

For non-convex problems one cares arguably less about the rate of convergence to a local optimum and more about the ability of the algorithm to find the global optimum. There are few theoretical results about the optimality of the previous algorithms but common belief suggests that updating variables one at a time is very susceptible to entrapment in local modes. For this reason, this chapter focusses on algorithms of the second category that update all variables simultaneously. It began initially with the development of two new algorithms motivated by the expectation-minimization principle of Dempster et al. (1977). The first algorithm was based on the orthogonal data augmentation idea of Ghosh and Clyde (2011), while the second exploited a location mixture representation of the multivariate Gaussian distribution. Comparing these algorithms with each other, the proximal gradient method (see Rockafellar (1970)) and the iterative algorithm of Figueiredo and Nowak (2003) led to a surprising observation that all four algorithms are, in fact, identical.

In order to compare the four algorithms considered in this chapter, a common perspective with which to view them is needed. For the purposes of comparison it is most convenient to consider each algorithm as a *majorization-minimization* (MM) algorithm, reviewed in section 4.6.1. Although the proximal gradient method is more commonly motivated by showing that the fixed-point of the proximal gradient map satisfies the desired optimality condition for the minimum of a convex composite non-

smooth objective function, this motivation does not hold for non-convex problems. Instead it can be motivated as an MM algorithm by considering functions with Lipschitz continuous derivative, which is outlined in section 4.6.3. This allows it to be compared most readily with the EM algorithms arising from data augmentation techniques, which are themselves special cases of MM algorithms operating within a probabilistic context, as demonstrated in section 4.6.2. The following subsections review the relevant theory for MM/EM algorithms and Lipschitz gradient functions which is required in section 4.7 to demonstrate that all four algorithms result in identical parameter mappings.

4.6 Preliminaries

4.6.1 MM algorithm

The majorization-minimization (MM) algorithm is a general method for generating a *descent* algorithm. A function $Q(\cdot|\beta^{(n)}) : \mathcal{Q} \rightarrow \mathbb{R}$ is said to *majorize* a function $f : \mathcal{Q} \rightarrow \mathbb{R}$ at $\beta^{(n)} \in \mathcal{Q}$ if

1. $f(\beta^{(n)}) = Q(\beta^{(n)}|\beta^{(n)})$ (tangency)
2. $f(\beta) \leq Q(\beta|\beta^{(n)}) \forall \beta \in \mathcal{Q}$ (domination)

The MM algorithm constitutes two steps. First by majorizing the objective about the current iterate $\beta^{(n)}$ with a function $Q(\cdot|\beta^{(n)})$ (surrogate), and then minimizing this surrogate to obtain the next iterate $\beta^{(n+1)}$. The tangency and domination properties alone are sufficient to guarantee decrease in the objective.

$$f(\beta^{(n+1)}) \leq Q(\beta^{(n+1)}|\beta^{(n)}) \leq Q(\beta^{(n)}|\beta^{(n)}) = f(\beta^{(n)}). \quad (4.40)$$

How well suited the MM principle is to any given problem depends upon how easy it is to find a readily minimizable surrogate, thus converting a hard optimization

problem into a sequence of simpler ones. For the class of \mathcal{C}^1 functions with Lipschitz continuous derivative there is a surrogate which has strong connections with first-order gradient descent methods. Numerous examples in statistics and machine learning can be found in Lange (2016).

4.6.2 EM algorithm

The EM algorithm of Dempster et al. (1977) is an example of an MM algorithm where the objective function to be minimized is defined within the context of a statistical model. The surrogate Q is achieved by taking expectations and applying Jensen's inequality. Formally let $f(\beta)$ correspond to the negative log density $-\log p(\beta)$, where $p(\beta)$ is the marginal distribution obtained from some joint distribution $p(\beta, \eta)$. Then

$$\begin{aligned} f(\beta) &= -\log \int \frac{p(\beta, \eta)}{p(\eta|\beta^{(n)})} p(\eta|\beta^{(n)}) d\eta \\ &\leq -\mathbb{E}[\log p(\beta, \eta)|\beta^{(n)}] + \mathbb{E}[\log p(\eta|\beta^{(n)})|\beta^{(n)}], \end{aligned} \tag{4.41}$$

where I have used concavity of the log function and Jensen's inequality with the expectation taken with respect to the conditional distribution for η given $\beta^{(n)}$. The right hand side of equation (4.41) possesses the desired tangency and domination properties required for the MM algorithm, and so minimizing these terms with respect to β to obtain the next iterate results in an algorithm possessing the descent property. Technically both terms constitute the Q function in the MM algorithm but as the second term is only an additive constant it does not affect the minimization step and is typically dropped. There is a freedom as to how one chooses this joint distribution, which usually takes the form of data augmentation or parameter expansion, and is most useful when the joint distribution is much more amenable to minimization than the resulting marginal. This chapter will visit three distinct data augmentation ideas which are useful for penalized linear least squares problems.

4.6.3 Lipschitz-Gradient Functions

In order to motivate the proximal gradient method as an MM algorithm the following results for Lipschitz gradient functions are required. A differentiable function $f : \mathbb{R}^p \rightarrow \mathbb{R}$ is said to be ρ -smooth if it satisfies

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\rho}{2} \|y - x\|_2^2 \quad \forall y, x \in \mathbb{R}^p. \quad (4.42)$$

A differentiable function $f : \mathbb{R}^p \rightarrow \mathbb{R}$ is said to be *Lipschitz gradient* if ∇f is Lipschitz continuous. We will use the result that a Lipschitz gradient function with parameter L is L -smooth. This follows from application of Taylor's theorem and the Cauchy-Schwarz inequality

$$\begin{aligned} f(y) &= f(x) + \langle \nabla f(x), y - x \rangle + \int_0^1 \langle \nabla f(x + \alpha(y - x)) - \nabla f(x), y - x \rangle d\alpha \\ &\leq f(x) + \langle \nabla f(x), y - x \rangle + \int_0^1 \|\nabla f(x + \alpha(y - x)) - \nabla f(x)\|_2 \|y - x\|_2 d\alpha \\ &\leq f(x) + \langle \nabla f(x), y - x \rangle + L \|y - x\|_2^2 \int_0^1 \alpha d\alpha \\ &= f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|_2^2 \end{aligned} \quad (4.43)$$

The right hand side of inequality (4.43) clearly satisfies the desired tangency and domination properties of a majorizing function. It's connection with gradient descent methods can be readily seen by rewriting it in the following form with $x = \beta^{(n)}$

$$f(\beta) \leq f(\beta^{(n)}) - \frac{1}{2L} \|\nabla f(\beta^{(n)})\|_2^2 + \frac{L}{2} \left\| \beta - \left(\beta^{(n)} - \frac{1}{L} \nabla f(\beta^{(n)}) \right) \right\|_2^2. \quad (4.44)$$

The minimization of the surrogate in equation (4.44) yields the next iterate as $\beta^{(n+1)} = \beta^{(n)} - \frac{1}{L} \nabla f(\beta^{(n)})$, which is recognizable as a gradient step with step size $1/L$. The proximal gradient method can be motivated as an MM algorithm similarly.

4.7 Algorithm Comparisons

In the next section I will demonstrate the main result that all four iterative algorithms result in identical updating rules. For algorithms that operate within a probabilistic framework I assume that the prior has the form $p(\beta) \propto \exp(-g(\beta))$ where g is the penalty in the optimization formulation. Furthermore I assume for simplicity and without loss of generality that the observation variance $\sigma^2 = 1$. Notationally I use $\lambda_i(A)$ to denote the i 'th largest eigenvalue of the matrix A .

4.7.1 Proximal Gradient Method

The proximal gradient method concerns convex minimization of functions that can be decomposed into a smooth and non-smooth part i.e.

$$\min_{\beta \in \mathbb{R}^p} F(\beta) := f(\beta) + g(\beta), \quad (4.45)$$

where f is, as before, assumed to be a continuous function with L-Lipschitz continuous derivative and g a function, possibly non-smooth, with a tractable proximal operator. Given a current parameter value $\beta^{(n)}$ one can apply the MM principle using the surrogate derived from the Lipschitz continuity of the derivative to obtain the proximal gradient method

$$\begin{aligned} \beta^{(n+1)} &= \arg \min_{\beta \in \mathbb{R}^p} \left(f(\beta^{(n)}) + \langle \nabla f(\beta^{(n)}), \beta - \beta^{(n)} \rangle + \frac{1}{2t} \|\beta - \beta^{(n)}\|_2^2 + g(\beta) \right) \\ &= \arg \min_{\beta \in \mathbb{R}^p} \left(g(\beta) + \frac{1}{2t} \|\beta - (\beta^{(n)} - t \nabla f(\beta^{(n)}))\|_2^2 \right) \\ &= \text{prox}_{tg}(\beta^{(n)} - t \nabla f(\beta^{(n)})) \end{aligned} \quad (4.46)$$

with a step size $t \leq \frac{1}{L}$. The following rate of convergence is due to Beck and Teboulle (2009). For convex problems, the step-size $t = 1/L$ guarantees the following rate of

convergence in the objective

$$F(\beta^{(n)}) - F^* \leq \frac{1}{2nt} \|\beta^{(0)} - \beta^*\|_2^2, \quad (4.47)$$

where F^* is the solution to equation (4.45) and β^* the argument that achieves this minimum. This implies that the number of iterations required to obtain an ϵ -optimal solution is at most $\lceil \frac{1}{2\epsilon t} \|\beta^{(0)} - \beta^*\|_2^2 \rceil$. Note that the proximal gradient method applied to the least squares problem considered here reduces to

$$\beta^{(n+1)} = \text{prox}_{tg}(\beta^{(n)} - tX^T(X\beta^{(n)} - Y)). \quad (4.48)$$

4.7.2 Orthogonal Data Augmentation

Ghosh and Clyde (2011) introduced a novel data augmentation technique known as orthogonal data augmentation, whereby added to the original design matrix X_o are additional rows X_a such that $X_o^T X_o + X_a^T X_a = LI_p$ for $L \geq \lambda_1(X_o^T X_o)$. The data augmented model is then

$$\begin{aligned} Y_o | \beta, \sigma^2 &\sim \mathcal{N}(X_o \beta, \sigma^2) \\ Y_a | \beta, \sigma^2 &\sim \mathcal{N}(X_a \beta, \sigma^2) \end{aligned} \quad (4.49)$$

The “complete” design matrix $X_c^T = [X_o^T, X_a^T]$ together with the observed and augmented data Y_o and Y_a then forms an orthogonal regression problem which permits closed form expressions for posterior model and variable inclusion probabilities (conditional on the observation variance σ^2). This was applied successfully in conjunction with Markov chain Monte Carlo (MCMC) to marginalize over the augmented data and obtain Rao-Blackwellized estimates of posterior model and inclusion probabilities and is implemented in the R package BoomSpikeSlab by Scott (2017).

The same data augmentation technique can be applied in the context of optimization to generate an iterative algorithm for solving (penalized) least squares problems. By observing that an orthogonal penalized least squares problem reduces

to p univariate optimization problems, the algorithm iterates between updating the augmented data Y_a and the regression coefficients β . The surrogate is obtained by taking the expectation of the negative log-posterior under the complete data with respect to the augmented data Y_a conditional on the current value of the parameters $\beta^{(n)}$

$$Q(\beta|\beta^{(n)}) = \mathbb{E}[-\log p(\beta|Y_o, Y_a)|\beta^{(n)}] \propto \frac{L}{2}\|\beta - \frac{1}{L}(X_o^T Y_o + X_a^T \mathbb{E}[Y_a|\beta^{(n)}])\|_2^2 + g(\beta). \quad (4.50)$$

The expected value of Y_a given $\beta^{(n)}$ is simply $X_a\beta^{(n)}$ and so the surrogate provided by the missing data is

$$Q(\beta|\beta^{(n)}) = \frac{L}{2}\|\beta - \frac{1}{L}(X_o^T Y_o + X_a^T X_a\beta^{(n)})\|_2^2 + g(\beta). \quad (4.51)$$

Although motivated very differently, a little algebra shows that the surrogate provided by the orthogonal data augmentation idea is identical to the surrogate provided by Lipschitz continuity of the derivative as in gradient descent

$$\begin{aligned} \frac{1}{L}(X_o^T Y_o + X_a^T X_a\beta^{(n)}) &= \frac{1}{L}(X_o^T Y_o + (LI - X_o^T X_o)\beta^{(n)}) \\ &= \beta^{(n)} - \frac{1}{L}X_o^T(X_o\beta^{(n)} - Y_o). \end{aligned} \quad (4.52)$$

The surrogate is then minimized to obtain the new iterate as

$$\beta^{(n+1)} = \text{prox}_{\frac{1}{L}g} \left(\beta^{(n)} - \frac{1}{L}X_o^T(X_o\beta^{(n)} - Y_o) \right). \quad (4.53)$$

The condition for $L \geq \lambda_1(X_o^T X_o)$ from the orthogonal data augmentation perspective to ensure positive definiteness of $LI_p - X_o^T X_o$ and consequently a real valued X_a is equivalent to the condition that L be greater than or equal to the Lipschitz constant of the derivative of $1/2\|Y_o - X_o\beta\|_2^2$, namely, $\lambda_1(X_o^T X_o)$. This shows that for linear least squares problems with arbitrary penalties that the orthogonal data augmentation

and proximal gradient algorithms are essentially equivalent. This algorithm was also subsequently published by Xiong et al. (2016) but the authors fail to identify the connection with other methods.

4.7.3 Location Mixtures of Spherical Normals

Note that the Lipschitz surrogate in (4.44) is, up to negative logarithm, proportional to the kernel of a Normal with a mean $\beta^{(n)} - (1/L)\nabla f(\beta^{(n)})$ and a covariance matrix $(1/L)I_p$. One might ask if the original density might be expressed as a mixture of such Normals, and that forming the aforementioned surrogate be viewed as conditioning on the mixing parameter. For multivariate likelihoods this is indeed the case. Assume that the likelihood $L(\beta) = \mathcal{N}(\beta|\mu, \Sigma)$, where the latter represents the density function of a multivariate normal with mean μ and covariance matrix Σ . Then, for any $t \leq \lambda_p(\Sigma)$, the posterior density can be regarded as the marginal arising from the joint

$$p(\beta|Y) \propto \int \mathcal{N}(\beta|\mu + z, tI_p)\mathcal{N}(z|0, \Sigma - tI_p)p(\beta)dz, \quad (4.54)$$

that is the multivariate likelihood can be expressed as mixing a spherical normal over a latent location parameter. The parameter expansion is nice because now the parameters of interest appear in the likelihood of a spherical normal distribution. This means, assuming independent priors specified on the individual components of β , that the individual β_i s are conditionally independent of each other given z . Conditioning on z then reduces once again a multivariate optimization problem to p univariate optimization problems. The only restriction is that $t \leq \lambda_p(\Sigma)$ to ensure positive semi-definiteness of the covariance matrix of z . Let us consider an EM algorithm, treating z as the missing “data”, and show how this too is just another disguised version of the proximal gradient method.

Let us start by forming the surrogate for the EM algorithm

$$Q(\beta|\beta^{(n)}) = \mathbb{E}[-\log p(\beta, z|Y)|\beta^{(n)}] \propto \frac{1}{2t} \|\beta - (\mu + \mathbb{E}[z|\beta^{(n)}])\|_2^2 + g(\beta), \quad (4.55)$$

where

$$z|\beta \sim N((t^{-1}I + (\Sigma - tI)^{-1})^{-1}t^{-1}(\beta - \mu), (t^{-1}I + (\Sigma - tI)^{-1})^{-1}). \quad (4.56)$$

It is useful at this stage to use the Sherman-Morrison-Woodbury formula to simplify the above as

$$z|\beta \sim N((I - t\Sigma^{-1})(\beta - \mu), t - t\Sigma^{-1}t). \quad (4.57)$$

It follows that

$$Q(\beta|\beta^{(n)}) \propto \frac{1}{2t} \|\beta - (\beta^{(n)} - t\Sigma^{-1}(\beta^{(n)} - \mu))\|_2^2 + g(\beta), \quad (4.58)$$

for which minimization provides the next iterate as

$$\beta^{(n+1)} = \text{prox}_{tg}(\beta^{(n)} - t\Sigma^{-1}(\beta^{(n)} - \mu)). \quad (4.59)$$

This is identical to the proximal gradient method by observing that $\nabla f(\beta^{(n)}) = \Sigma^{-1}(\beta^{(n)} - \mu)$ for $f(\beta) = (\beta - \mu)^T \Sigma^{-1}(\beta - \mu)$. For linear least squares problems, for which the corresponding likelihood function is $L(\beta) = \mathcal{N}(\beta|(X^T X)^{-1}X^T Y, (X^T X)^{-1})$, equation (4.59) reduces to

$$\beta^{(n+1)} = \text{prox}_{tg}(\beta^{(n)} - tX^T(X\beta^{(n)} - Y)), \quad (4.60)$$

as expected. Note that the condition arising from positive semi-definiteness of the covariance matrix for z , namely, $t \leq \lambda_p(\Sigma)$ is the same condition arising from the Lipschitz inequality, namely, that the inverse step-size $1/t$ be greater than or equal to the Lipschitz constant, which in this case is $\lambda_1(\Sigma^{-1})$ and specifically $\lambda_1(X^T X)$ in the linear least squares context.

4.7.4 LQ-decompositions of the Design Matrix

This data augmentation strategy due to Figueiredo and Nowak (2003) appeared in the context of wavelet based image processing where the design matrix is a product of a matrix H with an orthogonal matrix W corresponding to the discrete wavelet transform. The authors introduced missing data Z such that the augmented model is

$$\begin{aligned} Y|Z &\sim \mathcal{N}(HZ, I_p - tHH^T) \\ Z|\beta &\sim \mathcal{N}(W\beta, tI_p) \end{aligned} \tag{4.61}$$

where the covariance matrices are carefully chosen such that marginalization over Z recovers the original distribution for $Y|\beta$. Once again this places the parameters of interest in the context of orthogonal regression, resulting in p tractable univariate minimizations for penalties possessing tractable proximal operators. Although the form of X as $X = HW$ was determined by the application, this data augmentation strategy can be applied quite generally as one can express an arbitrary design matrix X as a product of a lower triangular matrix H and an orthogonal matrix W via the LQ-decomposition. Establishing the surrogate for the EM algorithm is achieved by computing the expectation over Z conditional on Y and the current iterate $\beta^{(n)}$

$$Q(\beta|\beta^{(n)}) = \mathbb{E}[-\log p(\beta|Y, Z)|Y, \beta^{(n)}] \propto \frac{1}{2t} \|\beta - W^T \mathbb{E}[Z|Y, \beta^{(n)}]\|_2^2 + g(\beta), \tag{4.62}$$

where the conditional expectation is given by

$$Z|Y, \beta \sim N(\Omega(t^{-1}W\beta + H^T(I_p - tHH^T)^{-1}Y), \Omega), \tag{4.63}$$

with

$$\Omega = (t^{-1} + H^T(I_p - tHH^T)^{-1}H)^{-1}. \tag{4.64}$$

The expressions in equations (4.63) and (4.64) simplify considerably through the repeated application of the Sherman-Morrison-Woodbury identity and are key to

revealing the surrogate as the same as that developed in section. In particular, repeated application shows that the covariance matrix can be expressed as

$$\Omega = (t^{-1} + H^T(I_p - tHH^T)^{-1}H)^{-1} = t - tH^T Ht, \quad (4.65)$$

and the mean as

$$\begin{aligned} \Omega(t^{-1}W\beta + H^T(I_p - tHH^T)^{-1}Y) &= (t - tH^T Ht)(t^{-1}W\beta + H^T(I_p - tHH^T)^{-1}Y) \\ &= W\beta + tH^T(I_p - tHH^T)^{-1}Y \\ &\quad - tH^T HW\beta - t^2H^T HH^T(I_p - tHH^T)^{-1}Y \\ &= (W - tH^T HW)\beta + tH^T(I_p - tHH^T)(I_p - tHH^T)^{-1}Y \\ &= (W - tH^T HW)\beta + tH^T Y. \end{aligned} \quad (4.66)$$

It follows that the term involving the expectation in equation (4.62) can be rewritten as

$$\begin{aligned} W^T \mathbb{E}[Z|Y, \beta^{(n)}] &= (W^T W - tW^T H^T HW)\beta^{(n)} + tW^T H^T Y \\ &= \beta^{(n)} - tX^T(X\beta^{(n)} - Y), \end{aligned} \quad (4.67)$$

and so the final minimization operation of the EM algorithm can be seen as a proximal gradient step

$$\beta^{(n+1)} = \text{prox}_{t\mathcal{g}}(\beta^{(n)} - tX^T(X\beta^{(n)} - Y)), \quad (4.68)$$

as expected. Once again, the condition that $I_p - tHH^T \geq 0$ requires that $t \leq 1/\lambda_1(HH^T) = 1/\lambda_1(X^T X)$.

4.8 Comparative Discussion

The last section considered three competing data augmentation strategies from statistics and signal processing used to generate iterative algorithms for solving penalized least squares problems and compared them to the proximal gradient method for composite non-smooth optimization. The previous section demonstrated that all three

algorithms, though motivated by three distinct data augmentation ideas, result in parameter mappings that are identical to each other and to those resulting from the proximal gradient method. For convex problems, it follows that they all exhibit the same rate of convergence. None of these algorithms as presented so far are, however, optimal. In particular a lower bound on the complexity for any first order optimization algorithm is given by

$$F(\beta^{(n)}) - F^* \geq \frac{3L}{32(n+2)^2} \|\beta^{(0)} - \beta^*\|_2^2, \quad (4.69)$$

see for example Nesterov (2014, chap. 2). An optimal first order algorithm would attain the lower bound on the convergence rate up to a constant factor, whereas the algorithms presented so far have a suboptimal convergence rate of $\mathcal{O}(1/n)$. Beck and Teboulle (2009) and Nesterov (2013) give acceleration strategies for the proximal gradient method which achieve the optimal rate of convergence in (4.69) up to a constant factor. The established connection between the three data augmentation algorithms and the proximal gradient method means that they can share in this acceleration also. The data augmentation strategies are restricted to Gaussian likelihoods, whereas the proximal gradient method is more general, requiring only that the function $f(\beta)$ is Lipschitz gradient, and can therefore be applied to other problems such as penalized logistic regression.

When the problem is non-convex, one cares more about the quality of the local solution obtained, not how fast it gets there. There is little literature establishing properties of these algorithms for non-convex problems and so this chapter concludes with some experimentation.

4.9 Experimentation for Non-Convex Problems

The proximal gradient method, randomized coordinate descent (CD), cyclic greedy descent (CDC) and greedy coordinate descent (GCD) will be required in the next

section to get good quality initial solutions. The proximal operator of $g(\beta) = \lambda_0\|\beta\|_0 + \lambda_1\|\beta\|_1$ is

$$\text{prox}_{t_g}(z) = \text{sign}(z)(|z| - t\lambda_1)1\{|z| > \sqrt{2t\lambda_0} + \lambda_1\}, \quad (4.70)$$

and is shown in figure 4.1. It is similar to lasso but adds an additional region of

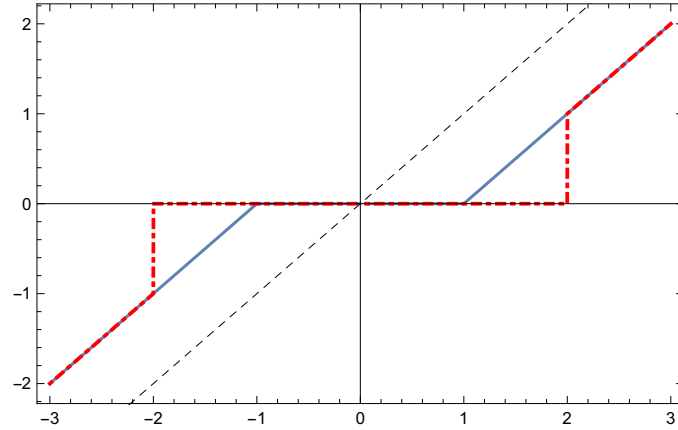


FIGURE 4.1: Proximal functions for $t = 1$ and (black,dashed) $g(x) = 0$, (blue solid) $g(x) = |x|$ and (red, dot-dashed) $g(x) = |x| + 0.51\{x \neq 0\}$

hard-thresholding, controlled by λ_0 . This helps to understand the properties of the penalty. It is known that lasso typically includes too many predictors. In addition choosing the tuning parameter λ_1 in lasso involves a trade-off between selection and shrinkage. Making λ_1 too large gets good selection, but shrinks coefficients heavily. Making λ_1 too small, however, does not shrink coefficients too heavily but too many predictors are included. The point-mass-Laplace mixture prior has an additional degree of freedom in that λ_0 can be used to control selection, and λ_1 can be used to control shrinkage. It is a generalization of both ℓ_1 and ℓ_0 penalized regression. For $f(\beta) = \|Y - X\beta\|_2^2/(2n)$, the gradient vector $\nabla f(\beta) = (1/n)X^T(X\beta - Y)$ and the Lipschitz constant is $L_f = \lambda_{\max}(X^T X)/n$. The proximal gradient update is then

$$\beta \leftarrow \text{prox}_{t_g}(\beta - (t/n)X^T(X\beta - Y)), \quad (4.71)$$

where the step size $t < 1/L_f$. For coordinate descent methods, the update of variable β_i holding all $\beta_{\setminus i}$ is

$$\beta_i \leftarrow \text{prox}_{1g}(x_i^T(Y - \sum_{j \neq i} x_j \beta_j)/n). \quad (4.72)$$

To assess how susceptible the four algorithms (ccd, rcd, gcd and proximal gradient) are to entrapment by local modes, 100 datasets are replicated under the same conditions. Let ccd be the reference algorithm. For each replica, the difference in the objective between the value returned by each algorithm and the reference is recorded and the 100 instances are used to form box-plots. An algorithm exhibiting better performance than cyclic gradient descent should result in more negative values. All algorithms were seeded with an initial $\beta = 0_p$ and run until the absolute relative change in the objective fell below a tolerance of 0.0001. The penalties used are $\lambda_1 = \sigma\sqrt{(2\log p)/n}$ and $\lambda_0 = (\sigma^2/n)\log p$.

Dataset 1

$n = 300$, $p = 80$, $X_i \sim \mathcal{N}(0, \Sigma)$ for $i \in \{1, \dots, p\}$ with $\Sigma_{ij} = 0.95^{i-j}$, $Y = X\beta^0 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, I_n)$ and $\beta_i^0 = 1$ with probability $1/8$ and zero otherwise. Figure 4.2 shows that the coordinate descent algorithms all perform similarly, whereas the proximal gradient method performs much worse.

Dataset 2

This dataset uses a real design matrix X from the diabetes dataset of Efron et al. (2004), available in the R package “lars”. 350 rows are randomly selected and the columns are standardized to have zero mean and \sqrt{n} euclidean norm. There are $p = 64$ predictors and $\beta_i^0 = 1$ with probability $10/p$ and zero otherwise. The data is simulated as $Y = X\beta^0 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 3I_n)$. Figure

4.3 indicates that the proximal gradient method exhibits poorer performance than

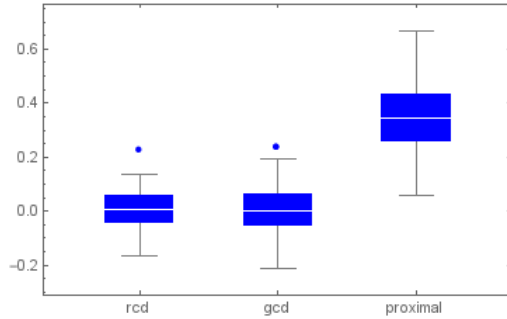


FIGURE 4.2: Difference in objective value between solution from cyclic coordinate descent (ccd) and solution provided by randomized coordinate descent (rcd), greedy coordinate descent (gcd) and proximal gradient method (proximal). 0.025, 0.25, 0.5, 0.75, 0.975 quantiles are -0.15, -0.04, 0.01, 0.06, 0.13 for rcd, -0.18, -0.06, 0.00, 0.06, 0.19 for gcd and 0.15, 0.26, 0.34, 0.43, 0.53 proximal.

cyclic gradient descent on this dataset, with greedy coordinate descent performing marginally better.

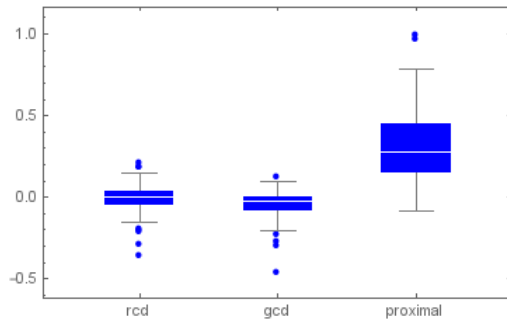


FIGURE 4.3: Difference in objective value between solution from cyclic coordinate descent (ccd) and solution provided by randomized coordinate descent (rcd), greedy coordinate descent (gcd) and proximal gradient method (proximal). 0.025, 0.25, 0.5, 0.75, 0.975 quantiles are -0.23, -0.05, 0.00, 0.03, 0.18 for rcd, -0.28, -0.08, -0.02, 0.00, 0.09 for gcd and -0.07, 0.15, 0.28, 0.44, 0.87 for proximal.

Dataset 3

This design matrix for this dataset has been studied in Yang et al. (2016). $n = 300$, $p = 80$ and the rows are simulated iid from a zero-mean multivariate normal with

covariance matrix

$$\Sigma = \begin{bmatrix} 1 & \mu & \mu & \dots & \mu \\ \mu & 1 & 0 & \dots & 0 \\ \mu & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu & 0 & 0 & \dots & 1, \end{bmatrix}$$

where $\mu = 10/(20\sqrt{80})$. $Y|\beta \sim \mathcal{N}(X\beta, 3I_n)$ where β_i is simulated from a standard normal with probability $10/p$ and zero otherwise. This design matrix was initially studied in Wainwright (2009) and was demonstrated to be particularly challenging for lasso. As shown in figure 4.4, the coordinate descent methods almost always return the same solution, whereas the proximal gradient method almost always returns something worse than cyclic gradient descent.

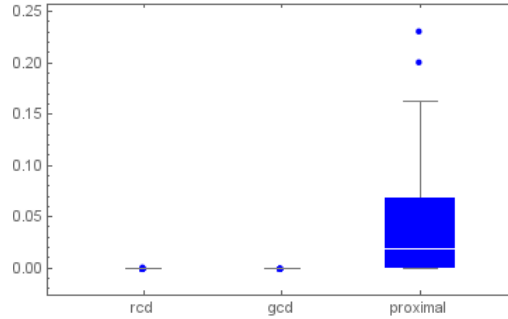


FIGURE 4.4: Difference in objective value between solution from cyclic coordinate descent (ccd) and solution provided by randomized coordinate descent (rcd), greedy coordinate descent (gcd) and proximal gradient method (proximal). 0.025, 0.25, 0.5, 0.75, 0.975 quantiles are 0.00, 0.00, 0.00, 0.00, 0.00 for rcd, 0.00, 0.00, 0.00, 0.00, 0.00 for gcd and 0.00, 0.00, 0.02, 0.07, 0.16 for proximal.

Discrete Optimization for Variable Selection

5.1 Discrete Optimization

Selecting a single model with respect to some criterion, $p(\gamma|Y)$ for instance, is a *discrete* optimization problem. Discrete optimization refers to a class of problems in which the domain \mathcal{D} of the objective function is a finite or countably infinite set. In addition, there may be certain *constraints* required by the problem that partition the domain into a *feasible set* \mathcal{S} , the set of elements in the domain satisfying all constraints, and its complement. A discrete optimization problem can then be represented abstractly as follows

$$\begin{aligned} & \underset{x}{\text{Minimize}} && F(x) \\ & \text{subject to} && x \in \mathcal{S}. \end{aligned}$$

In classical non-discrete optimization, \mathcal{D} is typically a continuous metric space. The distance metric is important as it defines the notion of a local neighbourhood. In addition, continuity and differentiability of $F(x)$ provide information about what the objective may look like over these neighbourhoods. The basic intuition behind

iterative gradient methods is to consider directions which decrease the objective most rapidly, based on derivative information contained within a local neighbourhood of the current point.

The discrete optimization case is considerably harder. When \mathcal{S} is discrete, it may be harder to define a reasonable notion of distance and hence the meaning of a local neighbourhood. If a neighbourhood is well defined, with no discrete counterpart of continuity or differentiability it is not possible to infer how the function may change over these neighbourhoods. This lack of information makes discrete optimization problems more challenging and limits the number of methods with which one can find a solution. In principle it is possible to evaluate the objective for all values of x in S and choose the one achieving $\min\{F(x) : x \in \mathcal{S}\}$. This *brute force* approach soon becomes computationally prohibitive as the problem size grows and one must use a method that exploits more problem structure. In general these methods can be classified into two categories. *Exact* methods which compute the globally optimal solution and *heuristics* which find a feasible solution with no guarantees of optimality. Both types have their merits and can be argued to be complementary to one another. Heuristics are typically very fast algorithms which provide a good quality sub-optimal solution. Yet, without any knowledge of the solution that the heuristic will provide, it is difficult to supply any properties that the solution may possess, statistical or otherwise. Exact methods on the other hand are guaranteed to find the globally optimal solution, for which the properties may be easier to ascertain, yet usually require considerably more computational effort.

Among some of the most popular general-purpose heuristics are genetic (Holland (1992)), tabu-search (Glover (1989), Glover (1990)), variable neighbourhood search (Mladenović and Hansen (1997)), multi-start greedy (Laguna and Marti (1999)), ant colony (Dorigo and Di Caro (1999)) and simulated annealing (Kirkpatrick et al. (1983)) algorithms. Overviews of these popular methods and many more can be

found in the excellent texts of Aarts and Lenstra (1997) and Gendreau and Potvin (2010). Early heuristics in the statistics community for variable selection problems focused on *stepwise* methods, dating back at least as early as Efroymson (1960). This algorithm interleaved forward and backward selection steps, using p -values under F -tests between models to select the predictor to be added or removed. This can also be viewed as a greedy neighbourhood search algorithm on the residual sum of squares (see Christensen (1987)). Although it is easy to create examples for which stepwise methods do not select the optimal model (see Hocking and Hocking (1976)), it is sometimes possible to quantify how bad the model selected by these methods can be. *Submodularity* of the objective function is a useful property for understanding the theoretical properties of greedy methods, and is used by Das and Kempe (2008) and Das and Kempe (2011) to derive a constant-factor approximation in the best subset selection problem. Stepwise and greedy procedures can be generalized by regarding them as neighbourhood search algorithms on $\{0, 1\}^p$, where the Hadamard distance is the appropriate metric. A recognized failure of these deterministic algorithm is their inability to recover from earlier mistakes and escape local optima. Such observations motivated the development of stochastic neighbourhood search algorithms, such as Shotgun stochastic search of Hans et al. (2007), which makes local stochastic moves in neighbourhoods of Hadamard distance 2.

Exact methods on the other hand can usually be categorized as either *dynamic programming* (Bellman (1954), Bellman (1957)) or *branch and bound* algorithms. Both are divide-and-conquer approaches, differing in how subproblems are related to each other. In branch and bound algorithms, the domain is recursively partitioned into smaller independent subproblems. In dynamic programming algorithms, however, the problem admits a special substructure that allows its solution to be computed via solving smaller *overlapping* subproblems. In this manner, the solution is “built-up” where solutions to subproblems are computed with aid of previously

found solutions to the subsubproblems. Branch and bound will be a core concept in the following chapters and therefore requires elaboration.

5.2 Branch and Bound

The earliest usages of branch and bound algorithms are believed to be Land and Doig (1960), who were working on discrete programming problems, and Little et al. (1963), who were working on the travelling salesman problem. Examples of varied statistical applications solved by branch and bound can be found in Brusco and Stahl (2005). Otherwise known as *partial enumeration*, branch and bound is able to find the global optimum without evaluating each point in the feasible set. It adaptively partitions the domain through a branching process, in which it eliminates the possibility of finding the global optimum through bounding. The branching process can be viewed as creating subproblems in which additional constraints are added, requiring the solution to be in a restricted part of the domain. Solving the original problem is then equivalent to solving these subproblems, but some of these subproblems may be skipped because they have no feasible solutions or via bounding. An upper bound on the globally optimal value is usually achieved by evaluating the objective at a feasible point, chosen through a heuristic in order to get a good upper bound. Computation of a lower bound on the optimal value for a subproblem is achieved through solving a *relaxation*.

Definition 11 (Relaxation). *An optimization problem $\min\{F_R(x) : x \in \mathcal{D}_R\}$ is a relaxation of $\min\{F(x) : x \in \mathcal{D}\}$ if $F_R(x) \leq F(x)$ for all $x \in \mathcal{D}$ and/or $\mathcal{D} \subseteq \mathcal{D}_R$. A feasible set \mathcal{D}_R is relaxation of a feasible set \mathcal{D} if $\mathcal{D} \subseteq \mathcal{D}_R$.*

A good relaxation is one that provides a tight lower bound and is easy to compute. If a lower bound L on the subproblem corresponding to $\min\{F(x) : x \in A \subset \mathcal{D}\}$ is found to be greater than the current upper bound on the global solution, then the

global optimum does not occur in A and the subproblem can be skipped. Using the notation $P_F(\mathcal{S})$ to denote the problem (or “program”) of finding $\min\{F(x) : x \in \mathcal{S}\}$, a general branch and bound algorithm can be described in pseudocode as follows

```

Data: An initial problem  $P_F(\mathcal{S} \cap \mathcal{D})$ 
Result:  $\hat{x}$  such that  $F(\hat{x}) \leq F(x)$  for all  $x \in \mathcal{S}$ 
1 Initialize:
2  $M \leftarrow P_F(\mathcal{S} \cap \mathcal{D})$  ;                               /* datastructure to store problems */
3  $U \leftarrow \infty$  ;                                       /* set upper bound to infinity */
4  $\tilde{x} \leftarrow Nil$  ;                                     /* best solution found so far */
5 while  $M$  is not empty do
6   Take a problem  $P_F(\mathcal{S} \cap \mathcal{A})$  from  $M$ ;
7   if  $\mathcal{A} \cap \mathcal{S} \neq \emptyset$  then
8      $y \leftarrow$  solution to relaxation of  $P_F(\mathcal{S} \cap \mathcal{A})$ ;
9     if  $F(y) < U$  then
10      if  $y \in \mathcal{S}$  then
11         $U \leftarrow F_R(y)$ ;                               /* update upper bound */
12         $\tilde{x} \leftarrow y$  ;                               /* update best feasible solution */
13      else
14        Partition  $\mathcal{A} = \bigcup_{i=1}^n \mathcal{A}_i$  ;                 /* branch */
15         $M \leftarrow M \cup (\bigcup P_F(\mathcal{S} \cap \mathcal{A}_i))$  ;   /* Add subproblems */
16      end
17    end
18  end
19 end
20 return  $\tilde{x}$ 

```

Algorithm 1: A generic branch and bound algorithm

The behaviour of the algorithm depends critically on the branching/partitioning rule used to create subproblems (line 14) and how the next subproblem to be solved is selected from the collection stored in M (line 6). If M is a stack with the LIFO (last in, first out) property, then the order of subproblems visited follows a *depth-first* traversal of the tree created by the branching process. The advantage of this strategy is that it yields feasible solutions very quickly, providing an upper bound U early on. Alternatively, if there is some domain specific information that allows the probability of finding the globally optimal solution in a certain partition to be

estimated, then the next subproblem can be selected as the one that appears most fruitful. In this case M is a priority-queue and results in a *best-first* search strategy. There are many more search strategies and a theoretical comparison of some of the most popular can be found Ibaraki (1976) and Linderoth and Savelsbergh (1999).

Subproblems can be *pruned* or *fathomed* (skipped) if they are *infeasible*, meaning there are no feasible solutions in this part of the domain (line 7), or are *dominated* by an upper bound on the optimal value of the initial problem, meaning it is not possible to find the optimal solution to the initial problem in this part of the domain (line 9).

The earliest exact methods for variable selection were branch and bound algorithms of Beale et al. (1967), Hocking and Leslie (1967), LaMotte and Hocking (1970) and Furnival and Wilson (1974), the legacy of which can be found in the R package “leaps” by Lumley (2017). Each algorithm considered the residual sum of squares as the objective function and obtained bounds through its *monotonicity* property $RSS(A) \leq RSS(B)$, where A is any set of predictors and $B \subset A$. This earlier work serves as a nice example with which to illustrate the concepts of branch and bound.

Branch and Bound for Residual Sum of Squares

As an example, suppose one wishes to find the model that minimizes the residual sum of squares with size 2 out of a possible $p = 5$ predictors. The problem may be formulated as follows

$$\begin{aligned}
 & \underset{\gamma}{\text{Minimize}} && F(\gamma) := \|Y - X_{\gamma}(X_{\gamma}^T X_{\gamma})^{-1} X_{\gamma}^T Y\|_2^2 \\
 & \text{subject to} && \gamma \in \{0, 1\}^p \\
 & && \sum_{i=1}^p \gamma_i = 2.
 \end{aligned} \tag{5.1}$$

The domain can be partitioned by branching on a predictor i , creating two subproblems in which constraints $\gamma_i = 1$ and $\gamma_i = 0$ are added. In order to quickly compute a lower bound on the solution of these subproblems, a relaxation needs to be defined. Recall to obtain a relaxation, one can either provide a function that lower bounds the objective function or enlarge the feasible set. In this case, a relaxation can be obtained by retaining the same objective function but enlarging the feasible set by removing the $\sum_{i=1}^p \gamma_i = 2$ constraint. Without this constraint, the subproblem has no combinatorial complexity and is trivial to minimize, namely, by the RSS under the largest model satisfying the $\gamma_i = 1$ or $\gamma_i = 0$ constraints. It is illustrative to step through algorithm 1 generating the BnB tree shown in figure 20.

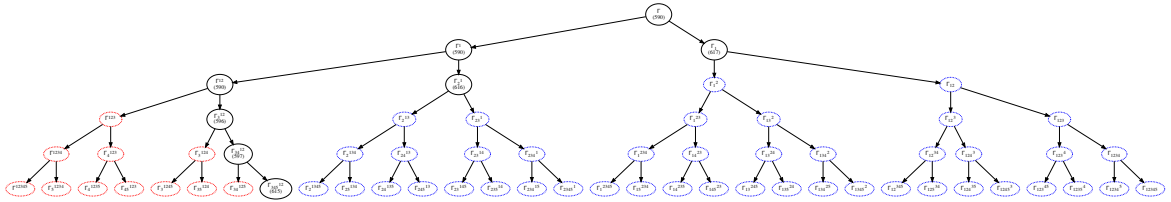


FIGURE 5.1: A Branch and bound tree generated from algorithm 1. Red nodes are pruned through infeasibility, blue nodes are pruned through bounding

It is useful to introduce a shorthand notation $\Gamma_{i\dots k}^{l\dots n} = \{\gamma \in \{0, 1\}^p : \gamma_a = 0 \forall a \in \{i, \dots, k\}, \gamma_b = 1 \forall b \in \{l \dots n\}\}$, and each node labelled $\Gamma_{i\dots k}^{l\dots n}$ corresponds to the subproblem $P_F(\mathcal{S} \cap \Gamma_{i\dots k}^{l\dots n})$ in addition to the value of the relaxation if computed. The root node Γ is relaxed to obtain a value of the RSS under the full model at 590. Branching is performed on predictor 1 and two subproblems with additional constraints $\gamma_1 = 1$ and $\gamma_1 = 0$ are added to the stack M . ‘Popping the top problem off of the stack corresponds to the subproblem with constraint $\gamma_1 = 1$, Γ^1 , for which the relaxation is the same as the previous node. Branching is performed on predictor 2 and two subproblems are added to the stack. Problem Γ^{12} is popped from the top of the stack, for which the relaxation is the same as the previous nodes, and

branching occurs on predictor 3, creating two subproblems added to the stack. The next problem is now Γ^{123} . Models in this part of the domain include three predictors, which are infeasible. Nothing is done with this node and it, along with the subtree, is passively pruned. The next problem corresponds to Γ_3^{12} , for which the relaxation is 596. Branching occurs on predictor 4 and two subproblems are added to the stack. The next problem is infeasible as it contains three predictors, and so this node along with its subtree is pruned. The next problem is Γ_{34}^{12} , for which the relaxation is 597. Branching occurs on predictor 5 and two subproblems are added to the stack. The next problem is infeasible and is pruned. The relaxation of the next problem is 615 and provides a feasible solution to the original problem, namely $(1, 1, 0, 0, 0)$. The upper bound is updated to 615 and the best feasible solution acquired so far is recorded. The next problem is Γ_2^1 for which the relaxation is 616. This is greater than the current upper bound $U = 615$ and so the optimal solution cannot be found in this subtree. This node is pruned by dominance. The next problem is Γ_1 for which the relaxation is 617. As before, this is pruned by dominance. The stack is now empty and the result that $(1, 1, 0, 0, 0)$ provides the optimal objective value of 615 is returned.

This basic illustration was able to guarantee the optimal model of size 2 in 6 function evaluations instead of 10. It masks, however, an important point regarding the decision on which predictor to branch. This branching strategy followed an arbitrary order, whereas typically information from the solution to the relaxation is used to decide on what predictor to branch. The performance of this BnB algorithm is limited by the tightness of the relaxation. The solution to the relaxation provides a lower bound on a subproblem, and subproblems are skipped when this lower bound is greater than current best upper bound. If it were possible to obtain a greater (tighter) lower bound on a subproblem, then more of them could be skipped. It is possible to get a much tighter lower bound by modelling 5.1 as a *mixed integer*

quadratic program.

5.3 Mixed Integer Programming

Recent work by Bertsimas et al. (2016) has demonstrated that mixed integer quadratic programming models are very effective in solving problem 5.1. A mixed integer quadratic program describes problems of the form

$$\begin{aligned} \underset{x}{\text{Minimize}} \quad & x^T H x + c^T x, \\ \text{subject to} \quad & Ax \leq b, \\ & x_i \in \mathbb{Z}, \forall i \in \mathcal{I}, \end{aligned}$$

where $I \subset \{1, \dots, p\}$ is an index set, $H \in \mathbb{R}^p$ a symmetric positive-semidefinite matrix, $c \in \mathbb{R}^p$, $A \in \mathbb{R}^{k \times p}$ and $b \in \mathbb{R}^k$. It is *mixed* because there are both real-valued as well as integer variables. Special cases include quadratic programs ($\mathcal{I} = \emptyset$), linear programs ($H = 0$, $\mathcal{I} = \emptyset$) and mixed integer linear programs ($H = 0$), from which it follows that MIQPs are NP-hard due to Kannan and Monma (1978). The performance of the MIQP formulation is attributed to a combination of the theoretical advancements of MIO solvers with enhanced hardware. Bixby (2012) compared the performance of twelve consecutive versions of CPLEX, a mixed integer solver from IBM (2011), dating from 1991 to 2007, on a set of mixed integer problems on the same computer and reported a speed up of 29000. This impressive speedup is due to theoretical and practical developments in the area of mixed integer programming such as cutting plane methods (Marchand et al. (2002)), column generation (Barnhart et al. (1996)), integration with constraint programming to handle disjunctive constraints, preprocessing and automatic model reformulation. In combination with general speedups in computer hardware, solving mixed integer optimization problems is dramatically faster than it was 30 years ago. It is easy to verify, however,

that the relaxation provided by the MIQP is much tighter than that provided by monotonicity in the last section.

5.3.1 Introduction to Solving Mixed Integer Programs

An excellent introduction to solving mixed integer nonlinear convex/nonconvex programs, of which MIQP is a special case, is given in Belotti et al. (2012). The most basic algorithm for solving MIQP problems is a branch and bound algorithm, which dates back to the work of Dakin (1965). A relaxation of a MIQP problem is obtained by relaxing the integrality constraints on the integer variables, allowing them to be real-valued, satisfying any additional bounds present in the problem. This is sometimes referred to as a *continuous-relaxation* or a *QP-relaxation* as the relaxation is a quadratic programming problem. A lower bound on a subproblem is then obtained by solving this quadratic program via any appropriate algorithm, such as the QP-simplex, interior point, or barrier algorithm to name a few. Branching is then performed on one of the integer variables, creating two subproblems into which the set of feasible integers for that variable is partitioned. Selecting an integer variable to branch on has received much attention as it is critical to the performance of the solver. A nice review of the most commonly used strategies can be found in Achterberg et al. (2005). The idea of branching on the most fractional variable under the solution to the relaxation, the variable furthest from integrality, known as *maximum fractional branching* or *most infeasible branching* is apparently no better than randomly selecting a variable. Instead, the two most commonly used strategies are *strong branching* and *pseudocost branching*. Both of these approaches try to estimate the increase in the lower bound for branching on a fractional variable, and choose the one that provides the greatest increase. For any subtree starting at a parent node, corresponding to a particular subset of the domain $A \subset \mathcal{D}$, one has a lower bound on all the feasible solutions in A provided by the relaxation. When

A is partitioned into two new subsets A_+ and A_- via branching, corresponding to two child nodes, one obtains an increased lower bound on all the feasible solutions in A provided by the minimum of the two relaxation solutions provided by each child node. The idea is to make this lower bound increase as fast as possible so that as many nodes can be pruned via comparison with the upper bound as possible. In the notation of Belotti et al. (2012), introduce degradation estimates D_i^+ and D_i^- for the increase in lower bound for branching left and right on variable i . A good variable to branch on would be one for which both degradation estimates are large. A commonly used function to score variables is

$$s_i = \mu \min(D_i^+, D_i^-) + (1 - \mu) \max(D_i^+, D_i^-), \quad (5.2)$$

where μ is close to 1. Branching is then performed on the variable with the highest score.

If the current lower bound for the current node is L , strong branching computes the lower bounds L_i^+ and L_i^- by solving the relaxation of each subproblem for all choices of possible branching variable. Degradation estimates are then no longer estimates but exactly $D_i^+ = L_i^+ - L$ and $D_i^- = L_i^- - L$. This results in a branch and bound tree with very few nodes but is heavy in computation, as each branching decision requires solving a number of quadratic programs. In short, it takes a peak at all possible branching options and chooses the one which helps the lower bound increase the fastest.

Pseudocost branching on the other hand is an alternative to strong branching which aims to avoid the heavy computation. It attempts to estimate the increase in lower bound by looking at historical increases whenever a variable has previously been branched upon. In particular, if variable i has been branched upon n_i times in the history of the algorithm, one can estimate the increase in the lower bound per

unit change in x_i by averaging across previous instances:

$$\Delta_L^+ = \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{L_{ji}^+ - L_j}{\lceil y_{ji} \rceil - y_{ji}}, \quad \Delta_L^- = \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{L_{ji}^- - L_j}{\lfloor y_{ji} \rfloor - y_{ji}}, \quad (5.3)$$

where L_j is the lower bound of parent node j , y_j is argument for which the relaxation of node j attains a minimum, L_{ji}^+ is the lower bound on the child node attained from branching left on variable i and L_{ji}^- is similar. If the relaxation to the current problem attains at minimum at y , then degradation estimates

$$D_i^+ = \Delta_L^+(\lceil y_i \rceil - y_i) \quad D_i^- = \Delta_L^-(\lfloor y_i \rfloor - y_i) \quad (5.4)$$

can be used. These ideas can be combined to form hybrid approaches, or generalized to an approach called *reliability branching* (Achterberg et al. (2005)).

Common node selection strategies are *depth-first*, *best-first* and *least-discrepancy* search, with no single strategy being uniformly better than the others. Depth first search has two favourable points. The first is that it dives down the tree very quickly to obtain a feasible solution and hence an upper bound early on in the beginning of the algorithm. This helps subsequent nodes to be pruned. In addition, sometimes the BnB algorithm is not allowed to run until the optimality gap (upper bound minus lower bound) becomes zero, but instead until it falls within some tolerance or a time limit is exceeded. In this case it is useful to be able to return the best feasible solution found so far. With respect to memory, depth first search is also efficient. When there are only p binary variables to branch on, then there are at most p subproblems stored in M as the algorithm attempts to branch left until a node is pruned or infeasible and then backtracks.

Best first search on the other hand has exponential worst case complexity. It selects the next problem to solve which has the smallest lower bound among all subproblems stored in M . This has the potential to store all subproblems in M . Suppose best first search decides to branch left on a node, but the children have a

lower bound that is greater than all the other nodes in M , so it branches on one of the other nodes and the same thing happens. In the worst case this branches on all nodes and adds them to M . If, however, our relaxation is perfect and the solution to the relaxation is no longer a lower bound but actually the solution to the subproblem, then best first search branches optimally and creates the tree with the smallest number of nodes. Best first search therefore works well when the relaxation is tight.

Least- or limited-discrepancy search is a very interesting idea which uses a heuristic to drive the search strategy. It is based on the intuition that a heuristic might have succeeded in locating the global optimum if it were not for a small number of mistakes along the way. Consider a greedy forward selection procedure that selects predictors 3, 1 and 4 out of a possible 5. This can be regarded as branching left on 3, then on 1, then on 4 and then branching right on 2 and then 5. A path of discrepancy n is a set of branching decisions which differ from those taken by the heuristic n times. For example left on 3, right on 1, left on 4, right on 2 and then right on 5 is a path of discrepancy one. Whereas left on 3, right on 1, left on 4, left on 2 and right on 5 is a path of discrepancy 2. Limited discrepancy search traverses the tree in order of increasing discrepancies. As the discrepancy number increases, the heuristic is trusted less and less. It embodies the idea that the global optimum may be reached in a sequence of steps that is similar to those taken by the heuristic. Depending on its implementation, its memory efficiency falls between that of depth first and best first search.

5.4 A MIQP for Best Subset Selection

Bertsimas et al. (2016) introduced a mixed integer quadratic program equivalent of 5.1.

$$\begin{aligned}
& \underset{\beta, \gamma}{\text{Minimize}} && F(\beta) := \|Y - X\beta\|_2^2, \\
& \text{subject to} && \gamma \in \{0, 1\}^p, \\
& && \sum_{i=1}^p \gamma_i = k, \\
& && (\beta_i \neq 0 \wedge (1 - \gamma_i) = 0) \vee (\beta_i = 0 \wedge (1 - \gamma_i) \neq 0).
\end{aligned} \tag{5.5}$$

Problem (5.5) isn't technically a MIQP, as the last constraint is not linear. In fact, the feasible set defined by this disjunction is not even convex. There are special ways to incorporate logical constraints into problem by performing special branching procedures but discussion of these is deferred to a later chapter. Instead it is useful to relax this constraint to a linear constraint via a *big-M* transformation.

5.4.1 Big-M Transformation

The big-M transformation is a neat trick to relax disjunctions to linear inequalities, allowing the problem to be solved as a MIQP. It requires the knowledge of upper and lower bounds on the parameters, the regression coefficients in this case. Using these upper and lower bounds, equation (5.5) can be reformulated as

$$\begin{aligned}
& \underset{\beta, \gamma}{\text{Minimize}} && F(\beta) := \|Y - X\beta\|_2^2, \\
& \text{subject to} && \gamma \in \{0, 1\}^p, \\
& && \sum_{i=1}^p \gamma_i = k, \\
& && \underline{\beta}_i \gamma_i \leq \beta_i \leq \bar{\beta}_i \gamma_i.
\end{aligned} \tag{5.6}$$

When $\gamma_i = 1$, β_i is free to be within the interval $[\underline{\beta}_i, \bar{\beta}_i]$, otherwise $\beta_i = 0$. Naturally, the lower and upper bounds must not render the globally optimal solution for β infeasible, yet choosing tighter bounds results in better performance. The reason

for this is that choosing larger bounds increases the size of the feasible set, which degrades the quality of the relaxation, as minimizing a function over a larger set is guaranteed to achieve a smaller value. These upper and lower bounds can be estimated from the data using optimization, a technique known as *optimality based bounds tightening*. A discussion of this is deferred to a later section in order to illustrate an important point.

5.4.2 Monotonicity vs Continuous Relaxation

The reason why the MIQP based branch and bound outperforms the monotonicity based branch and bound is that the relaxation is *always* tighter. To illustrate this, suppose an example dataset has been generated where $n = 100$, $p = 60$, the rows of the design matrix are simulated from a zero-mean multivariate normal with covariance matrix $\Sigma_{ij} = 0.7^{|i-j|}$ and observations $y_i|\beta \sim N(x_i^T\beta, 1)$, where the true regression vector has elements 1 for $i \in \{10, 20, 30, 40, 50, 60\}$ and zero otherwise. Suppose the cardinality constraint, k , is 6 and a node has been reached through branching that corresponds to a subproblem of (5.6) with constraints $\gamma_i = 1$ for $i \in \{10, 30\}$ and $\gamma_i = 0$ for $i \in \{20, 40\}$. The choice of node here is arbitrary, it is just for the purposes of illustration. The optimal model in $\Gamma_{20,40}^{10,30}$ subject to the cardinality constraint contains predictors 7, 10, 30, 50, 57 and 60, for which the regression coefficients are 0.48, 1.05, 0.76, 0.95, -0.40 and 1.03, with an RSS value of 251.53. A lower bound on the RSS for the optimal model in $\Gamma_{20,40}^{10,30}$ can be obtained via monotonicity, i.e. solving the relaxation obtained by removing the cardinality constraint, by evaluating the RSS at γ^a where $\gamma_i^a = 1$ for all $i \notin \{20, 40\}$ and zero otherwise. This gives a lower bound of 28.26. In contrast suppose one places lower and upper bounds on each regression coefficient of -1.5 and 1.5 respectively. Adding these constraints does not change the optimal solution of the subproblem, as the optimal solution remains feasible. The lower bound obtained from solving the continuous relaxation of (5.6)

is 150.43. With moderate bounds on the regression coefficients therefore, the lower bound provided by the continuous relaxation is much tighter than the relaxation obtained by removing the cardinality constraint.

As the lower and upper bounds for β tighten around their values under the optimal model, the continuous relaxation becomes perfect, and the lower bound becomes equal to the solution of the subproblem. As they become larger, however, the lower bound provided by the continuous relaxation becomes identical to the lower bound provided by monotonicity. Recall that the relaxation when wanting to use monotonicity was obtained by simply removing the cardinality constraint. Having large bounds essentially does the same thing. A very large lower or upper bound on β_i multiplied by a small fractional value of γ_i is still large. The feasible set in \mathbb{R}^p for β defined by the constraints is then so large that the constraints are practically irrelevant. It is clear, therefore, that in order to get better performance than the branch and bound based on monotonicity one only need specify finite bounds on the regression coefficients, so long as they do not render the globally optimal solution for β infeasible. These bounds can be obtained through optimality based bounds tightening.

5.5 Optimality Based Bounds Tightening

Generally speaking, if one is trying to maximize an objective function $F(x)$ over a feasible set \mathcal{S} and one already has a feasible solution \tilde{x} , obtained perhaps through a heuristic, providing an upper bound $\tilde{F} := F(\tilde{x})$, then one can obtain upper and lower bounds on x_i by solving

$$\overline{x}_i = \max\{x_i : x \in \mathcal{S}, F(x) \leq \tilde{F}\}, \quad (5.7a)$$

$$\underline{x}_i = \min\{x_i : x \in \mathcal{S}, F(x) \leq \tilde{F}\}. \quad (5.7b)$$

In practise the feasible set \mathcal{S} may be non-convex and so the bounds are typically optimized over a convex relaxation of \mathcal{S} . With respect to the best subset selection problem, the feasible set is defined by constraints on the model size. Maximizing/minimizing β_i over this set would lead to another discrete optimization problem, so instead one can solve a relaxation obtained by removing this constraint.

$$\bar{\beta}_i = \max\{\beta_i : F(\beta) \leq \tilde{F}\}, \quad (5.8a)$$

$$\underline{\beta}_i = \min\{\beta_i : F(\beta) \leq \tilde{F}\}, \quad (5.8b)$$

as it advocated in Bertsimas et al. (2016). How to obtain these bounds is described later through optimality based bounds tightening. To conclude the example of best subset selection, two full runs of the algorithm are demonstrated with $\bar{\beta}_i = -\underline{\beta}_i = 1.5$ and $\bar{\beta}_i = -\underline{\beta}_i = 300$, the latter of which emulates the behaviour of the branch and bound based on monotonicity. The same dataset as described earlier is used, with the exception of reducing the number of observations to $n = 70$ in order to increase the runtime, otherwise the BnB with regression coefficient bounds at 1.5 converges too fast. In addition the model size was increased to $k = 10$. Figure 5.5 shows the optimality gap of both runs over time. When $\bar{\beta}_i = -\underline{\beta}_i = 1.5$, the BnB algorithm converges to global optimality in under **6 seconds**, whereas when $\bar{\beta}_i = -\underline{\beta}_i = 300$ to emulate lower bounds equivalent to those obtained by monotonicity, the BnB algorithm takes 524 seconds.

5.6 Mixed Integer Models for Point Mass-Laplace Mixture Priors

In order to include the ℓ_1 norm in the objective into a mixed integer quadratic program, one must write $s_i = |\beta_i|$ and add the constraints $-s_i \leq \beta_i \leq s_i$. The MIQP

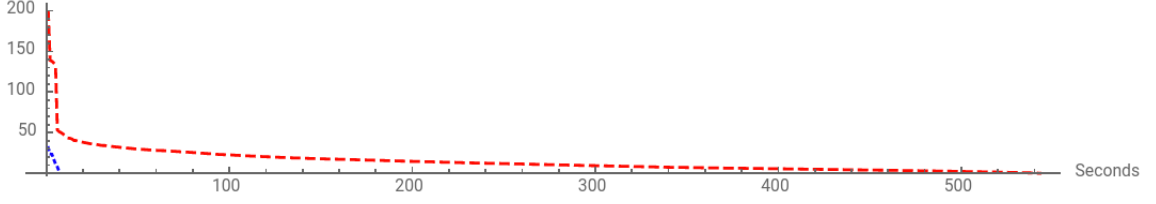


FIGURE 5.2: Optimality Gap (upper minus lower bound) when the lower and upper bounds on the regression coefficients have absolute value (blue, short-dashed) 1.5 and (red, long-dashed) 300.

can then be formulated as

$$\begin{aligned}
 & \underset{s, \beta, \gamma}{\text{Minimize}} && F(s, \beta, \gamma) := (1/2n) \|Y - X\beta\|_2^2 + \lambda_1 \sum_{i=1}^p s_i + \lambda_0 \sum_{i=1}^p \gamma_i, \\
 & \text{subject to} && -\underline{\beta}_i \gamma_i \leq \beta_i \leq \bar{\beta}_i \gamma_i \quad \forall i \in \{1, \dots, p\}, \\
 & && -s_i \leq \beta_i \leq s_i \quad \forall i \in \{1, \dots, p\}, \\
 & && s_i \geq 0 \quad \forall i \in \{1, \dots, p\}, \\
 & && \underline{k} \leq \sum_{i=1}^p \gamma_i \leq \bar{k}, \\
 & && \gamma_i \in \{0, 1\} \quad \forall i \in \{1, \dots, p\},
 \end{aligned} \tag{5.9}$$

where \underline{k} and \bar{k} are lower and upper bounds on the model size. The can initially be taken to be 0 and p respectively, but can be tightened using bound tightening procedures. It may be tempting to think that the Big-M constraint could use s_i instead, that is, $-s_i \gamma_i \leq \beta_i \leq \gamma_i s_i$. This would be very convenient, as it would avoid estimation of lower and upper bounds $\underline{\beta}_i$ and $\bar{\beta}_i$, but it is neither a linear nor a quadratic constraint as the second order matrix is not positive semi-definite. These bounds must be estimated through optimality based bounds tightening.

5.6.1 Example: Diabetes Dataset

Once again consider the diabetes dataset of Efron et al. (2004). The dataset consists of $n = 442$ observations and $p = 64$ predictors. A training/test partition of the data was made by randomly selecting 350 rows to train on. The correlation among predictors is shown in figure 5.3. The response and design matrix columns are stan-

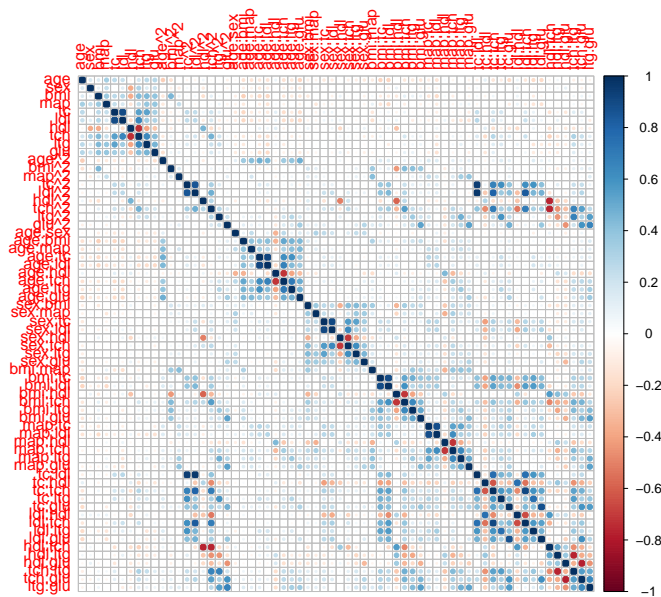


FIGURE 5.3: Predictor correlations in diabetes dataset.

standardized to have zero mean and ℓ_2 -norm of \sqrt{n} . To get some perspective 10-fold cross validated lasso was performed with R package `glmnet`. The optimal value of λ_1 obtained by cross validation is 0.017, resulting in a regression vector with 31 non-zero elements for which the largest element is 0.35 in absolute value and achieving an out of sample squared predictive error of 51.87. For the point-mass-Laplace model, I take in this case a subjective approach to selecting λ_0 and λ_1 . An estimate $\hat{\sigma}^2$ is obtained from fitting the full model, providing a value of 0.44. This determines λ_0 as $-(\hat{\sigma}^2/n) \log(p)$. Given, that λ_1 in lasso is struggling to achieve a balance between selection and shrinkage, I expect that λ_1 need not be so large for the point-mass-

Laplace prior as λ_0 will take care of the selection, so I make the subjective choice that $\lambda_1 = 0.01$ i.e. smaller than the value obtained by cross validation previously. Given that the largest element selected by lasso is 0.35 in absolute value, I am also willing to make the assumption that all the coefficients should lie in the interval $[-4, 4]$ (for the Big-M transformation). With these values, it takes no longer than **11 seconds** to solve the MIQP, resulting in a regression vector with 7 non-zero elements, achieving a squared predictive error of 46.59. This is usually a first pass at solving the MIQP, as typically one may wish to avoid specifying a hard constraint such as $\beta_i \in [-4, 4]$. A more thorough approach follows.

Optimality Based Bound Tightening

The greedy coordinate descent heuristic produces a regression vector with 7 non-zero elements obtaining an objective value of 0.26. Let $\tilde{F} = 0.26$ and initially let $\bar{\beta}_i = \infty$, $\underline{\beta}_i = -\infty$ for all $i = 1, \dots, p$ and $\underline{z} = 0$ and $\bar{z} = p$. Then an upper bound on β_i can be obtained from solving

$$\begin{aligned}
& \underset{\beta_i}{\text{Maximize}} && \beta_i, \\
& \text{subject to} && F(s, \beta, \gamma) \leq \tilde{F} \\
& && -\underline{\beta}_i \gamma_i \leq \beta_i \leq \bar{\beta}_i \gamma_i \quad \forall i \in \{1, \dots, p\}, \\
& && -s_i \leq \beta_i \leq s_i \quad \forall i \in \{1, \dots, p\}, \\
& && s_i \geq 0 \quad \forall i \in \{1, \dots, p\}, \\
& && \underline{k} \leq \sum_{i=1}^p \gamma_i \leq \bar{k},
\end{aligned} \tag{5.10}$$

and similarly minimizing to get a lower bound on β_i . Clearly the globally optimal solution must be a member of the set $\{(s, \beta, \gamma) \in \mathbb{R}^+ \times \mathbb{R}^p \times \{0, 1\}^p : F(s, \beta, \gamma) \leq \tilde{F}\}$. It must also be in the original feasible set. Equation (5.10) is maximizing β_i over

a relaxation of the intersection between these two sets. The relaxation provided by relaxing integrality constraints is used so that an upper bound is obtained by solving a QCP and not a MIQCP. One also have two choices. Solving this problems serially, means that the bounds get updated for sequential indices i , meaning that the upper and lower bounds obtained are tighter. Alternatively, one can solve these p quadratic programs in parallel keeping the upper and lower bounds at plus and minus infinity. Solving each of these quadratically constrained programs takes approximately 0.01 to 0.02 seconds for the diabetes dataset. After looping serially through the coefficients, the upper and lower bounds are displayed in figure 5.4 in blue.

Given tightened bounds on the β_i 's it is now possible to get an upper bound on the maximum model size by maximizing

$$\begin{aligned}
& \underset{\gamma \in \{0,1\}^p}{\text{Maximize}} && \sum_{i=1}^p \gamma_i, \\
& \text{subject to} && F(s, \beta, \gamma) \leq \tilde{F} \\
& && -\underline{\beta}_i \gamma_i \leq \beta_i \leq \bar{\beta}_i \gamma_i \quad \forall i \in \{1, \dots, p\}, \\
& && -s_i \leq \beta_i \leq s_i \quad \forall i \in \{1, \dots, p\}, \\
& && s_i \geq 0 \quad \forall i \in \{1, \dots, p\}, \\
& && \underline{k} \leq \sum_{i=1}^p \gamma_i \leq \bar{k},
\end{aligned} \tag{5.11}$$

and similarly minimizing to get a lower bound. This gives an upper bound of 12 and a lower bound of 1. One can keep iterating between tightening bounds on β_i 's and then on model sizes but subsequent improvements are modest. Fig. 5.4 shows the bounds tightened on the regression coefficients having been tightened a second time after tightening \bar{k} and \underline{k} .

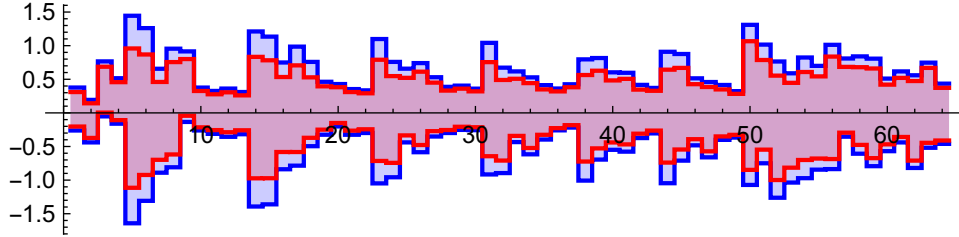


FIGURE 5.4: Intervals of permissible coefficient values defined after (Blue) first bound tightening (Red) second bound tightening of model (5.9) on diabetes dataset.

Results and Discussion

Using the bounds achieved by optimality based bounds tightening and warm-starting the solver with the solution found by the heuristic, it takes a mere **4 seconds** to solve the MIQP, exploring a total of 10562 nodes. In addition, the bounds obtained this way are guaranteed not to render the global solution infeasible. As noted before the estimator obtained from solving (5.9) obtains a squared predictive error of 46.59, whereas the lasso estimator obtains 51.87. The estimated regression vector is shown in figure 5.5. It illustrates a common phenomenon that lasso tends to include too many predictors.

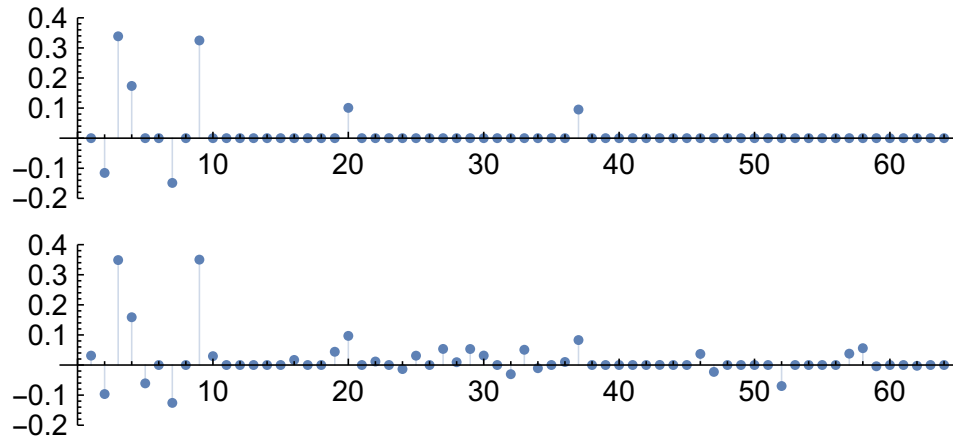


FIGURE 5.5: Coefficients estimated from (bottom) 10-fold cross validated lasso (top) solving (5.9)

It is an interesting question to ask how this method relates to the MIQP formu-

lation of best subset selection by Bertsimas et al. (2016). Firstly, the heuristic used by the authors, which they call a “discrete first order algorithm”, simply exploits the Lipschitz gradient property of the objective, and so is yet again a discovery of the same algorithm discussed in the previous chapter. Consider solving (5.6) with $k = 7$. To be generous, the globally optimal solution is computed first and now assumed to be found by the heuristic. Performing bound tightening on (5.6) provides the bounds shown in figure 5.6, which are much larger than the bounds computed for (5.9) as (5.6) contains no regularization term. Using the bounds found by bound tightening and warm-starting the solver with the globally optimal solution, it takes the solver 30 seconds to certify optimality, exploring a total of 55929 nodes. In short, it takes much longer to solve (5.6) than (5.9).

Recall that branch and bound performs well when the globally optimal solution has an objective value that is much less than the rest. In this case pruning is very effective. If there are many local solutions with objective values very close to that attained by the global solution, then the branch and bound algorithm is inefficient at pruning. In the variable selection context, branch and bound performs well when there is a dominant model. Adding the ℓ_1 and ℓ_0 penalties helps to differentiate the good models from the bad models, and hence increases the difference in their objective values. It was shown in Castillo et al. (2015) that point-mass-Laplace mixture priors enjoy optimal rates of posterior contraction, that is, they concentrate posterior mass on good models very quickly. In contrast, the best subset selection problem contains no prior or penalty, resulting in feasible models having similar objective values. This is evident from looking at the output of the solver. Solving (5.6) explores 55929 nodes in 30 seconds, whereas solving (5.9) explores only 10562 nodes in 4 seconds. The solver was able to prune the tree much quicker in problem (5.9) than in (5.6). In addition, (5.9) does not require an a priori constraint on the model size.

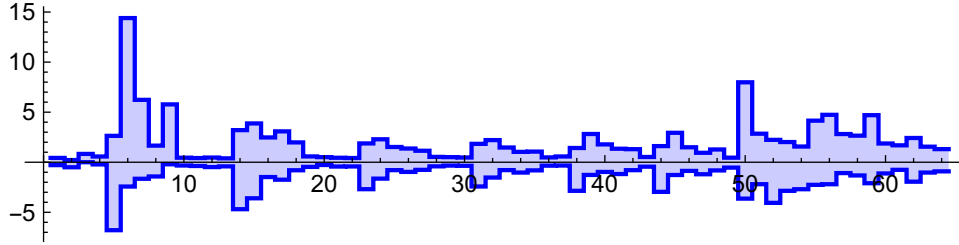


FIGURE 5.6: Intervals of permissible coefficient values defined after bound tightening of (5.6)

5.6.2 Example: Ozone Dataset

This example considers the Ozone dataset of Breiman and Friedman (1985). It consists of 366 daily maximum one-hour-averaged ozone measurements with twelve predictors. Of the twelve predictors three are factors, concerning time and date for book-keeping, and eight are meteorological covariates. These are pressure (Pr), wind speed (WS), humidity (Hu), temperature at Sandburg California (TS), temperature at El Monte California (TE), inversion base height (IH), pressure gradient (PG), inversion base temperature (IT) and visibility (Vi). After removing missing data there are only 203 observations. The data was randomly partitioned into training and test data by selecting 150 measurements to train on. It is common in previous analyses to drop the second temperature measurement. Models considered contain the remaining eight meteorological covariates, all two way interactions between them and their quadratic terms resulting in $p = 44$ predictors. The correlations between predictor is shown in figure 5.8. The predictors and the response were centered and scaled to have ℓ_2 -norm of \sqrt{n} .

It is my practical experience that choosing the *universal threshold* of $\lambda_1 = \sigma\sqrt{2\log(p)/n}$ appears to be too strong. Performing cross validated lasso results in a tuning parameter of $\lambda_1 = 0.002$. As in the analysis of the diabetes dataset, λ_1 need not be as large in the point-mass Laplace mixture prior and so this parameter was taken as 0.001. An estimate of σ^2 was obtained from fitting a linear model on

the full model, defining λ_0 as $-(\hat{\sigma}^2/n) \log(p)$.

The greedy gradient descent heuristic identified a model with 5 predictors, of which only 2 are present in the optimal model - containing predictors Hu^2 , PG^2 , IT , Hu:PG , Hu:IT , TS:PG , TS:IT with the colon denoting interaction. Optimality based bound tightening procedures resulted in the bounds displayed in figure in blue, where solving 5.10 for each index took less than 0.01 seconds. The bounds on the model size were then tightened to 1 and 17 by solving 5.11. A further round of bound tightening on the covariates resulted in the bounds shown in figure in red. The MIQP solver was seeded with the solution obtained from the heuristic and took **11 seconds** to solve to global optimality. The squared out of sample predictive error of the estimator obtained was 11.37, compared with 11.55 of cross validated lasso.

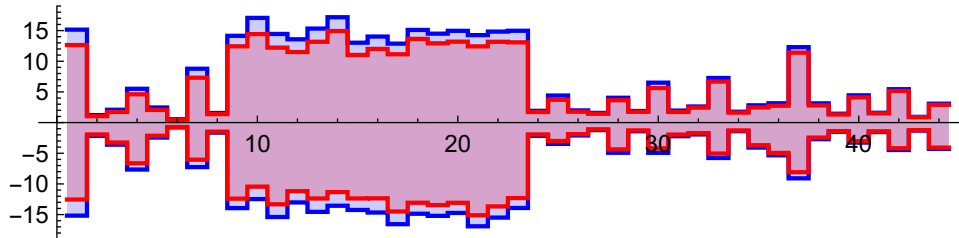


FIGURE 5.7: Intervals of permissible coefficient values in Ozone dataset defined after (Blue) first bound tightening (Red) second bound tightening of model (5.9).

5.7 Mixed Integer Models for Zellner's G-Prior

In addition to spike-and-slab priors Zellner's g-prior remains popular for its computational tractability and desirable properties. Consider a canonical linear model $Y|\gamma, \alpha, \beta, \sigma^2 \sim N(1\alpha + X_\gamma\beta_\gamma, \sigma^2)$, where γ is a model index, α is an intercept term, X_γ and β_γ are the design matrix and regression vector under model γ . A typical set of priors might be the g-prior for the regression coefficients and reference priors for

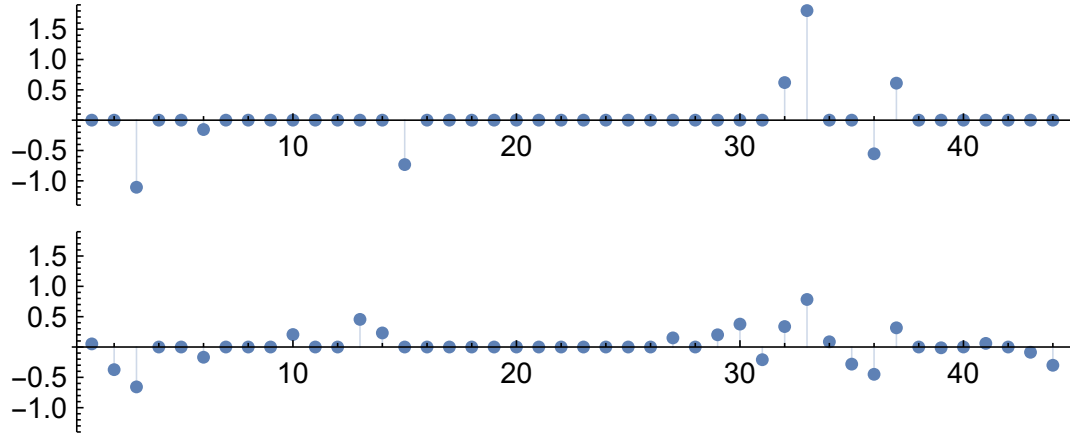


FIGURE 5.9: Coefficients estimated from (bottom) 10-fold cross validated lasso (top) solving (5.9) for Ozone dataset.

in the risk inflation criterion of Foster and George (1994) as well as using mixtures of g-priors in Liang et al. (2008). Despite the attractive properties of the resulting posterior, extracting information therefrom remains a challenging task. Enumeration of the model space soon becomes computationally prohibitive with p and $MCMC$ approaches, the computational complexity of which is studied in Yang et al. (2016), can be slow to converge and exhibit poor mixing. This section develops a mixed integer quadratically constrained programming model for selecting the optimal model under the g-prior.

5.7.1 Parameter Expansion

The posterior probability of a model γ resulting from priors (5.12) is given by

$$\begin{aligned}
 \log p(\gamma|Y) &= k - \frac{1}{2} \log(1 + g) \sum_{i=1}^p \gamma_i \\
 &\quad - \frac{n-1}{2} \log \left(1 + g \frac{\|Y_c - X_\gamma \hat{\beta}_\gamma\|_2^2}{\|Y_c\|_2^2} \right) \\
 &\quad + \log p(\gamma),
 \end{aligned} \tag{5.14}$$

where k is a constant of proportionality and $\hat{\beta}_\gamma = (X_\gamma^T X_\gamma)^{-1} X_\gamma^T Y_c$. It is assumed that the data has been centered $Y_c = Y - \bar{Y}$ in addition to the columns of design matrix so that $1^T X = 0_p$. In what follows the subscript c will be dropped from Y_c and Y shall refer to the centered data unless stated otherwise. In order to find the modal model $\hat{\gamma} := \arg \max_{\gamma \in \{0,1\}^p} \log p(\gamma|Y)$, it is in fact useful to re-introduce the regression vector β through the following observation.

Lemma 12.

$$\max_{\gamma \in \{0,1\}^p} \log p(\gamma|Y) = \max_{(\gamma, \beta) \in \mathcal{P}} \log L(\gamma, \beta), \quad (5.15a)$$

$$\text{where } \mathcal{P} = \{(g, b) \in \{0, 1\}^p \times \mathbb{R}^p : b_i = 0 \text{ if } g_i = 0 \quad (5.15b)$$

$$\forall i \in \{1, \dots, p\}\},$$

$$\begin{aligned} L(\gamma, \beta) = & -\frac{1}{2} \log(1 + g) \sum_{i=1}^p \gamma_i + \log p(\gamma) \quad (5.15c) \\ & - \frac{n-1}{2} \log \left(1 + g \frac{\|Y - X\beta\|_2^2}{\|Y\|_2^2} \right) \end{aligned}$$

Proposition 12 states that maximizing the posterior over models is equivalent to maximizing a new function $L(\gamma, \beta)$ over a feasible set \mathcal{P} which encodes the constraints that β_i cannot be nonzero unless $\gamma_i = 1$.

Proof. Let $\Phi(\beta) = \|Y - X\beta\|_2^2$ and consider a specific γ . Minimizing $\Phi(\beta)$ over feasible set $\mathcal{P}_\gamma = \{\beta \in \mathbb{R}^p : \beta_i = 0 \text{ if } \gamma_i = 0\}$ results in a minimal value of $\|Y - X\hat{\beta}_\gamma\|_2^2$ where $\hat{\beta}_\gamma$ is the OLS estimator of β under model γ , padded with zeros to make a p -dimensional vector. By monotonicity of the logarithm and the observation that $\Phi(\hat{\beta}_\gamma) \leq \Phi(\beta) \forall \beta \in \mathcal{P}_\gamma$, it follows that

$$-\log \left(1 + g\Phi(\hat{\beta}_\gamma)/\|Y\|_2^2 \right) \geq -\log \left(1 + g\Phi(\beta)/\|Y\|_2^2 \right) \forall \beta \in \mathcal{P}_\gamma,$$

and consequently

$$\log p(\gamma|Y) = \max_{\beta \in \mathcal{P}_\gamma} L(\beta, \gamma).$$

The result then follows from maximizing the lhs over $\gamma \in \{0, 1\}^p$. □

There are two main challenges with the parameter expanded formulation. Neither the (negative) objective function, nor the feasible set is convex. The disjunctive constraints can once again be overcome with a Big-M transformation, but the logarithmic term of the objective requires special attention.

Approximating the Logarithmic Term

The main challenge is the non-concave term resulting from the composition of the negative logarithm with a quadratic. One approach is to approximate this term using piecewise linear functions. It is easier to build piecewise linear approximations to univariate rather than multivariate functions over a bounded domain, and so it is useful to further reformulate the problem with the introduction of an auxiliary variable z as follows

$$\begin{aligned}
 \text{Maximize} \quad & -\frac{\log(1+g)}{2} \sum_{i=1}^p \gamma_i - \frac{n-1}{2} \log(z) + \log p(\gamma) \\
 \text{s.t.} \quad & -\gamma_i \underline{\beta}_i \leq \beta_i \leq \bar{\beta}_i \gamma_i \\
 & 1 + g \frac{\|Y - X\beta\|_2^2}{\|Y\|_2^2} \leq z \\
 & \underline{z} \leq z \leq \bar{z},
 \end{aligned} \tag{5.16}$$

The objective is maximized by making z small, which in turn drives the residual sum of squares toward smaller values through the second constraint in (5.32). The initial lower and upper bounds on z are provided by the observation that the residual sum of squares is minimized under the full model and maximized under the null model and can be specified as $1 + g\|Y - X(X^T X)^{-1} X^T Y\|_2^2 / \|Y\|_2^2$ and $1 + g$ respectively. This conveniently defines an initial interval over which the logarithm must be approximated by a piecewise linear function, although this interval can be tightened.

Once the logarithmic term is replaced by a piecewise linear function, the program falls into the category of a mixed integer quadratically constrained program.

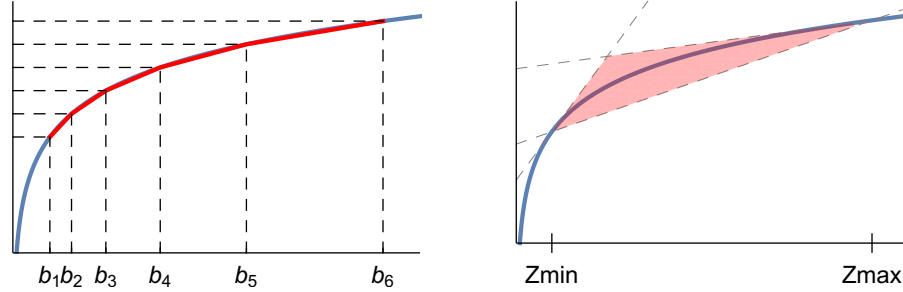


FIGURE 5.10: (left) A piecewise linear approximation to the logarithm defined on breakpoints $\{b_i\}_{i=1}^6$ (right) a polyhedral relation of the constraint $z_1 = \log z_2$ for $z_2 \in [z_{\min}, z_{\max}]$.

An alternative to approximations through piecewise linear functions is using spatial branch and bound methods. The introduction of the auxiliary variable z also helps in applying these techniques, although we shall now call this variable z_2 and introduce an additional variable z_1

$$\begin{aligned}
 \text{Maximize} \quad & -\frac{\log(1+g)}{2} \sum_{i=1}^p \gamma_i - \frac{n-1}{2} z_1 + \log p(\gamma) \\
 \text{s.t.} \quad & -\gamma_i \underline{\beta}_i \leq \beta_i \leq \bar{\beta}_i \gamma_i \\
 & z_1 = \log z_2 \\
 & 1 + g \frac{\|Y - X\beta\|_2^2}{\|Y\|_2^2} \leq z_2 \\
 & \underline{z} \leq z_2 \leq \bar{z},
 \end{aligned} \tag{5.17}$$

So far this has done nothing but replace a non-concave term in the objective with a non-convex constraint. This constraint can, however, be relaxed to a set of linear constraints through linearization, increasing the feasible set as illustrated in figure 5.10. Relaxing integrality constraints on the binary decision variables γ_i and solving the continuous relaxation over this enlarged convex feasible set provides a valid upper

bound on the original maximization problem. Spatial branch and bound methods can now branch on z_2 in addition to the binary γ_i 's, creating subproblems into which the interval of feasible values for z_2 is partitioned. In each subproblem, the interval of feasible values for z_2 is smaller than the parent problem, and so polyhedral relaxation better approximates the true non-convex feasible set. Computing the upper bound for each subproblem requires solving a quadratically constrained program. While this can be implemented with minimal effort, some additional care is required about when to branch on the continuous z_2 or the binary γ_i 's, for which a number of branching strategies are discussed in Belotti et al. (2012).

The approach of solving a QCP at each subproblem is valid but is not pursued here. Instead, one can go one step further and linearize all non-linear constraints, including the quadratic constraint, solving an LP at each subproblem. As linearizing the quadratic constraint increases the size of the feasible set, the upper bound provided by the LP-relaxation is not as tight as that provided by the QCP-relaxation. It is easier, however, to solve an LP than a QCP, which may result in less computation time overall. This is the approach used by the open-source solver Couenne of Belotti et al. (2009), a general purpose mixed integer nonlinear program solver against which the formulation based on piecewise linear approximations will be compared. For the latter the Gurobi Optimization (2016) solver is used.

5.7.2 Piecewise Linear Approximations

Approximating non-linear and/or non-concave terms via piecewise linear functions allows the model to stay within the definition of a MIQCP, and use optimized solvers, without the need to reach for more complicated MINLP techniques. To construct a piecewise linear approximation $\hat{f}(z)$ to $f(z)$ over $[z_{\min}, z_{\max}]$ one may start by defining d segments through a collection of breakpoints $z_{\min} = b^0 < b^1 < \dots < b^d = z_{\max}$.

The approximating function \hat{f} is then given by

$$\hat{f}(z) = f(b^{k-1}) + \left(\frac{f(b^k) - f(b^{k-1})}{b^k - b^{k-1}} \right) (z - b^{k-1}), \quad z \in [b^{k-1}, b^k], \quad (5.18)$$

or equivalently

$$\hat{f}(z) = m_k z + a_k, \quad z \in [b^{k-1}, b^k], \quad (5.19)$$

where $m_k = (f(b^k) - f(b^{k-1})) / (b^k - b^{k-1})$ and $a_k = f(b^{k-1}) - m_k b^{k-1}$. The piecewise linear behaviour can then be encoded in at least two different ways, either with the introduction of new binary variables or using specially ordered sets of type II Beale and Tomlin (1970), discussed in section 5.7.4. There are multiple ways to model piecewise linear functions using binary variables including, but not limited to, the *multiple choice*, *disaggregated convex combination* and models *incremental models*, a review of which can be found in Belotti et al. (2012). According to the computational experiments conducted by Vielma et al. (2010) the *multiple choice model* exhibits the best performance for small to moderate numbers of break points (≈ 16). This approach introduces d binary variables z_k , where $z_k = 1$ indicates that z is in the interval $[b^{k-1}, b^k]$, and d continuous variables w_k , where $w_k = z$ if z is in the interval $[b^{k-1}, b^k]$ and zero otherwise. These requirements can be encoded through the following constraints

$$\begin{aligned} \sum_{k=1}^d z_k &= 1, \quad z_k \in \{0, 1\} \\ \sum_{k=1}^d w_k &= z, \\ b^{k-1} z_k &\leq w_k \leq b^k z_k. \end{aligned} \quad (5.20)$$

How to choose the breakpoints and number of segments is also a consideration as there is a trade-off between acquiring a good approximation and increasing the problem size through additional binary variables. Some approaches try to minimize the

number of segments used whilst others try to minimize the approximation error for a given number of segments. The approach used in this paper is simple; for a given number of segments d , partition the interval $[\log(z_{\min}), \log(z_{\max})]$ into d intervals of equal length, defining $d + 1$ equally spaced grid-points. The breakpoints are then defined as the image of the grid-points under the exponential function, as illustrated in figure 5.10. An alternative approach using specially ordered sets of type II instead of binary decision variables is discussed in section 5.7.4.

5.7.3 Model Space Priors

So far there has been no reference to the kind of priors over the model space. The easiest priors to incorporate into a mixed integer formulation are those of the form

$$p(\gamma) \propto \left(\frac{\theta}{1 - \theta} \right)^{|\gamma|} \mathbf{1}_{\{\underline{k} \leq |\gamma| \leq \bar{k}\}}(\gamma), \quad (5.21)$$

where θ is a prior inclusion probability and \underline{k} and \bar{k} are minimum and maximum model sizes respectively as this translates to a *linear* penalty on the model size on the log-scale and a *linear* constraints. Such sparsity priors have been studied in combination with Zellner's g-prior in Yang et al. (2016). Naturally it is possible to set $\bar{k} = p$, but choosing a smaller value if possible has computational advantages. Priors of the form $p(\gamma) = h(\sum_{i=1}^p \gamma_i)$ such as beta-binomial and Poisson priors can also be formulated as part of a mixed integer program. In order to achieve true upper bounds it is necessary that the continuous relaxation is concave. Unfortunately $h(\sum_{i=1}^p \gamma_i)$ is not concave when integrality constraints on the γ_i 's are relaxed. As done previously for the logarithm function, h could be approximated using piecewise linear functions. Alternatively one can exploit the property that there are only $p + 1$ model sizes, and so there are only $p + 1$ function values of h to model. This is a natural candidate for SOS constraints of type I, for which the construction is as follows

5.7.4 Specially Ordered Sets of Type I/II

Generally speaking a problem formulation may demand non-linear and even non-convex functions to be realistic. Fortunately, there are some tricks which, under certain circumstances, allow one to remain within the definition of a mixed integer (quadratically constrained) program, without reaching for my complicated MINLP methods. In the current context the prior on the model space may result in a non-linear penalty on the model size. Moreover, the penalty may be non-concave, ruling out the possibility of formulating the problem as a mixed integer non-linear concave program. In addition, the (negative) logarithm in equation (5.14) is a non-linear non-concave function. In this section we discuss the application of specially ordered sets (SOS) of type I and II to address these challenges.

Type I

When the non-linear function is a function of an integer decision variable assuming a finite number of possible values, then the function can be modelled using SOS constraints of type I. Priors that result in a penalty on the size of the model contribute a term $h(\sum_{i=1}^p \gamma_i)$ in the objective, where h is a possibly non-linear and non-convex function. Despite these difficulties it is possible to exploit the fact that this term can only assume at most $p + 1$ values, which can be modelled linearly in the following manner

$$h_0x_0 + h_1x_1 + \cdots + h_px_p = h\left(\sum_{i=1}^p \gamma_i\right) \quad \text{Function Row} \quad (5.22a)$$

$$0x_0 + 1x_1 + \cdots + px_p = \sum_{i=1}^p \gamma_i \quad \text{Reference Row} \quad (5.22b)$$

$$x_0 + x_1 + \cdots + x_p = 1 \quad \text{Convexity Row} \quad (5.22c)$$

$$x_k \geq 0$$

where $h_i = h(i)$. The idea is to replace all occurrences of $h(\sum_{i=1}^p \gamma_i)$ with $\sum_{i=0}^p h_i x_i$, while adding the reference and convexity row as constraints. Although the convexity row constraint could be achieved by requiring the x_k to be binary decision variables, modelling them as an SOS constraint of type I results in a more efficient branching progress, enforcing a single x_k be unity in the final solution with all others zero. Instead of branching on an individual binary x_k in which two subproblems are created in which $x_k = 0$ and $x_k = 1$, an index $k \in \{0, \dots, p\}$ is chosen and two subproblems are created in which $x_i = 0$ for all $i < k$ and $x_i = 0$ for all $i > k$. For more details, such as how to choose the aforementioned index the reader is directed toward Beale and Tomlin (1970).

Type II

The second type of SOS constraint differs from the first in that two neighbouring x_k are allowed to be non-zero in the final solution instead of just one. This is useful for modelling piecewise linear functions, as the sum-to-one constraint coded in the convexity row allows function values to be linearly interpolated between two break points. In contrast to the method presented in section 5.7.2, however, SOS II constraints do not use any binary decision variables. This constraint finds application in the current context by allowing a piecewise linear approximation to the logarithm to be built. Let $f(\cdot)$ denote a piecewise linear approximation to the logarithm function over a collection of breakpoints $\{b_k\}_{k=0}^d$ as described in equation (5.19). This can be encoded linearly using SOS type II constraints in the following way

$$f_0 z_0 + f_1 z_1 + \cdots + f_d z_d = f(z) \quad \text{Function Row} \quad (5.23a)$$

$$b_0 z_0 + b_1 z_1 + \cdots + b_d z_d = z \quad \text{Reference Row} \quad (5.23b)$$

$$z_0 + z_1 + \cdots + z_p = 1 \quad \text{Convexity Row} \quad (5.23c)$$

$$z_k \geq 0$$

where $f_i = f(b_i)$. The idea is to replace every occurrence of $f(z)$ with $\sum_{k=0}^d f_k z_k$, while adding the reference and convexity rows as constraints. Specifying (z_0, \dots, z_d) as SOS-II enforces the constraint that only two neighbouring (z_i, z_{i+1}) can be non-zero in the final solution. This satisfies $z = b_i z_i + b_{i+1}(1 - z_i)$ and $f(z) = f(b_i)z_i + f(b_{i+1})(1 - z_i)$, which linearly interpolates the function between the two breakpoints.

5.7.5 Heuristics

Once again it is possible to develop heuristics for the g-prior similar to those in the last chapter. Under independent Bernoulli priors with inclusion probability θ , maximizing $L(\beta, \gamma)$ in (5.15c) is equivalent to solving

$$\text{Minimize} \quad F(\beta) := \log \left(1 + g \frac{\|Y - X\beta\|_2^2}{\|Y\|_2^2} \right) + \Omega(\beta), \quad (5.24)$$

$$\text{where} \quad \Omega(\beta) = \lambda_0 \|\beta\|_0 \quad (5.25)$$

$$\lambda_0 = \frac{2}{n-1} \left(-\log \left(\frac{\theta}{1-\theta} \right) + \frac{1}{2} \log(1+g) \right), \quad (5.26)$$

for β as $\|\beta\|_0 = \sum_{i=1}^p \gamma_i$. Indeed if $\hat{\beta}$ is optimal under (5.24), then $(\hat{\beta}, \hat{\gamma} := I\{\beta_i \neq 0\})$ is optimal under (5.15c) and vice versa. This connects the problem with literature on penalized regression. Marjanovic et al. (2015) and Blumensath and Davies (2008) present iterative algorithms for finding local solutions in ℓ_0 -regularized least squares problems. The only complication is that the least squares term presents itself as an argument to a logarithmic expression. This can simply be overcome with with a

linearization step and motivated as an MM (Majorization-Minimization) algorithm as described in section . Observe that the function

$$\begin{aligned}
Q(\beta|\beta^{(t)}) &:= \log(z^{(t)}) + \lambda_0\|\beta\|_0 \\
&\quad + \frac{1}{z^{(t)}} \left(1 + g \frac{\|Y - X\beta\|_2^2}{\|Y\|_2^2} - z^{(t)} \right) \\
\text{where } z^{(t)} &= 1 + g \frac{\|Y - X\beta^{(t)}\|_2^2}{\|Y\|_2^2},
\end{aligned}$$

possesses the desired tangency and domination properties necessary for an MM algorithm. All that is left is to exploit the separable property of the ℓ_0 penalty, which admits a tractable minimization solution in the univariate or orthogonal case:

$$\begin{aligned}
\text{prox}_{t\Omega}(z) &= \arg \min_{x \in \mathbb{R}} \left\{ \frac{1}{2t}(z - x)^2 + \lambda_0 1\{x \neq 0\} \right\} \\
&= z 1\{|z| > \sqrt{2t\lambda_0}\}.
\end{aligned} \tag{5.27}$$

With this in mind one can define coordinate descent updates,

$$\begin{aligned}
\beta_i &\leftarrow \text{prox}_{\phi^{(t)}\Omega}(x_i^T(Y - \sum_{j \neq i} x_j \beta_j)/n), \\
\text{where } \phi^{(t)} &= \frac{z^{(t)}\|Y\|_2^2}{gn},
\end{aligned} \tag{5.28}$$

or proximal gradient updates

$$\begin{aligned}
\beta &\leftarrow \text{prox}_{\phi^{(t)}\Omega}(\beta - (1/L)X^T(X\beta - Y)), \\
\text{where } \phi^{(t)} &= \frac{z^{(t)}\|Y\|_2^2}{gL}, \\
L &= \lambda_{\max}(X^T X).
\end{aligned} \tag{5.29}$$

5.7.6 Bound Tightening

Let the objective in (5.32) be denoted $L(\gamma, \beta, z)$. Once one has a feasible solution $(\tilde{\gamma}, \tilde{\beta}, \tilde{z})$ has been found through a heuristic one can obtain a lower bound on (5.32)

by evaluating the objective at this feasible point $\tilde{L} = L(\tilde{\gamma}, \tilde{\beta}, \tilde{z})$. It follows that the globally optimal value for z , denoted \hat{z} , must satisfy

$$\hat{z} \leq \exp\left(\frac{-2\tilde{L}}{n-1}\right), \quad (5.30)$$

meaning that the upper bound z can be reduced to $\bar{z} = \min\{1 + g, \exp(-n\tilde{L}/(n-1))\}$.

When using independent Bernoulli priors over the model space, it follows from the same reasoning that

$$\bar{k} \leq \lceil \left(\frac{\tilde{L} + ((n-1)/2) \log \bar{z}}{\log(\theta/(1-\theta)) - \log(1+g)/2} \right) \rceil. \quad (5.31)$$

Upper and lower bounds on the regression coefficients can be computed once again by optimality based bounds tightening, similar to what was done in section 5.6.1

5.7.7 Example: Diabetes Dataset

Using independent Bernoulli priors with inclusion probability θ and SOSII constraints to model the piecewise linear function defined by a set of d breakpoints $\{b_i\}_{i=0}^d$ with corresponding function values $f_i = \log(b_i)$, the overall MIQCP model is

$$\begin{aligned}
& \text{Maximize} && \left(\log \frac{\theta}{1-\theta} - \frac{\log(1+g)}{2} \right) \sum_{i=1}^p \gamma_i - \frac{n-1}{2} \sum_{i=0}^d f_i z_i \\
& \text{s.t.} && -\gamma_i \underline{\beta}_i \leq \beta_i \leq \bar{\beta}_i \gamma_i \quad \forall i \in \{1, \dots, p\} \\
& && \underline{k} \leq \sum_{i=1}^p \gamma_i \leq \bar{k} \\
& && \sum_{i=0}^d b_i z_i = z \\
& && \sum_{i=0}^d z_i = 1 \\
& && z_i \geq 0 \quad \forall i \in \{0, \dots, d\} \\
& && \{z_0, \dots, z_d\} \text{ SOSII} \\
& && 1 + g \frac{\|Y - X\beta\|_2^2}{\|Y\|_2^2} \leq z \\
& && \underline{z} \leq z \leq \bar{z},
\end{aligned} \tag{5.32}$$

Using the same dataset as in section 5.6.1, the MM-cyclic coordinate descent algorithm identified a model with objective value -919.21 containing predictors 3 and 9. The MM-proximal gradient descent algorithm failed to escape from its initial value of the zero vector. In contrast shotgun stochastic search was run for 30 iterations identifying a model with objective value -917.04 containing predictors 3,4,5 and 9. This allows \bar{z} and \bar{k} to be tightened to 191.57 and 13 respectively. The lower bound \underline{z} computed using the RSS of the full model is 127.32. $d = 4$ linear segments are specified through a set of breakpoints [127.32, 141.01, 156.17, 172.97, 191.57] upon which the f_i values are [4.85, 4.95, 5.05, 5.15, 5.26]. Bound tightening on the regression coefficients results in the bounds shown in figure 5.11. The solver failed to converge to global optimality by 1390 seconds as shown in 5.12.

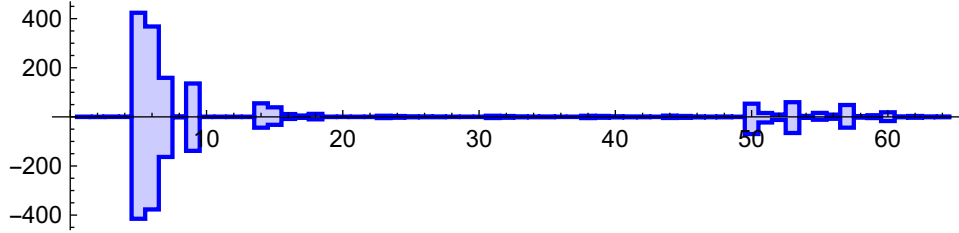


FIGURE 5.11: Intervals of permissible coefficient values defined after bound tightening for model (5.32)

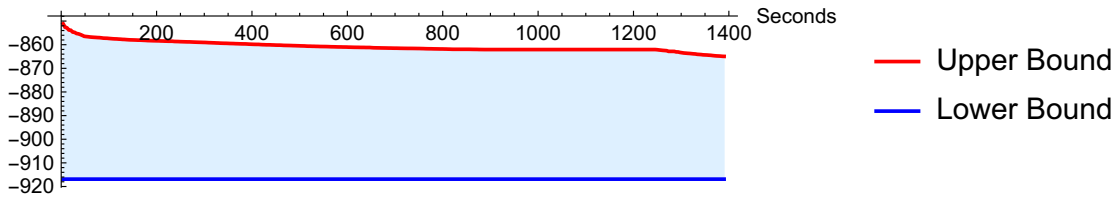


FIGURE 5.12: Progress of optimality gap over time.

The results are disappointing. The lower bound on the maximization problem is obtained through shotgun stochastic search already at -917.04 . The upper bound is provided by the minimum over all open subproblem relaxations. One possible explanation for the slow progress of the solver is that the poor bounds on the regression coefficients result in a very weak relaxation. The bounds provided by optimality based bound tightening are very large because the logarithm decreases very slowly as its argument increases. This means that when maximizing β_i , for example, it takes a long time due to the $-\log(\cdot)$ term before the objective falls below that obtained by the heuristic. Suppose one is somehow able to provide much tighter bounds on the regression coefficients then. Perhaps this would help. The top models identified by SSS have coefficient values that are all less than 1 in absolute value. It is interesting to see if reducing the feasible set by adding these constraints accelerates the progress of the solver. When bounding the coefficients to be inside $[-1, 1]$, the optimality gap is still only 5.15% at 1390 seconds (c.f. 5.66%), and so it appears that adding tighter bounds does not appear to help dramatically. For comparison, the Couenne solver

of Belotti et al. (2009), a general purpose MINLP solver, applied to the original problem only reaches an optimality gap of 11% after 1390 seconds.

Another possibility is that the way the piecewise linear approximation is handled through SOSII constraints is the source of slow performance. To investigate this one can replace logarithmic expression $\log(1 + g\|Y - X\beta\|_2^2/\|Y\|_2^2)$ with $1 + g\|Y - X\beta\|_2^2/\|Y\|_2^2$, which requires no piecewise linear approximation and hence no SOSII variables, and see how long the solver takes. This is now similar to the best subset selection problem except that the cardinality constraint of 13 is much larger than before and there is now a very modest penalty on the model size. Keeping the constraints on the coefficients at $[-1, 1]$ the solver converges in **1334 seconds**. This is an improvement, but it still takes a very long time. Another possibility is that there is a problem with this formulation, but running Couenne, a black-box MINLP solver, only reaches an optimality gap 11% in 1390 seconds.

A more likely explanation is that the posterior under g-prior concentrates mass on a single model at a much slower rate than the posterior resulting from a point-mass-Laplace mixture prior. Recall a branch and bound algorithm performs best when there is a dominant model, or more generally when the globally optimal solution of a maximization problem has an objective value that is much larger than all other locally optimal solutions. When this is the case, it is much easier to prune/fathom subproblems via bounding. This behaviour was already observed when comparing the performance of the mixed integer solver on the best subset selection problem against the posterior resulting from the point-mass-Laplace mixture prior. The former required the exploration of many more nodes in order to certify optimality. The posterior under the point-mass-Laplace mixture prior is known to contract at an optimal rate (see Castillo et al. (2015)) and it therefore makes sense that the mixed integer solver is able to certify optimality quickly. The problem is much worse for the g-prior for the following reason. In the best subset selection problem the objective

is the residual sum of squares. Consider two models of the same size and consider the difference in their residual sum of squares. This is the difference in objective for the best subset selection problem. The difference in their objective values under the MIQCP formulation for the g-prior is reduced due to the subadditivity of the logarithm. If the difference between their objective values was $A - B$ in the best subset selection problem, it is reduced to $\log(1 + A) - \log(1 + B)$ in the MIQCP formulation for the g-prior (assuming $g = n$ and $\|Y\|_2^2 = n$). The objective value across models becomes less separated. This can be argued to be a good thing from the perspective of handling model uncertainty, but it makes it difficult to select the modal model via discrete optimization.

6

Concluding Remarks

This thesis has presented research advances in Bayesian modelling of temporal point processes and mathematical optimization in variable selection.

- Chapter 2: “*A Continuous-Time Model of Arrival Times*”.

Chapter 2 presents an elegant model for modelling arrival times without the need for any binning discretization. It’s elegance lies in its simplicity, using data augmentation strategies that are familiar to Bayesian statisticians. The state-space representation of the Matern covariance Gaussian process in one dimension leads to a highly scalable model with a very practical implementation based on AVL-trees. The proposed Gibbs sampler with slice sampling steps exhibits excellent mixing as shown on a number of simulated and real datasets. The developed model could also readily find application in density estimation, a problem very similar to learning the intensity function. This connection has previously been exploited by Taddy and Kottas (2012) who model the normalized intensity as a Dirichlet process mixture of Betas. The idea is to model a set of iid observations from an unknown density as those that are accepted

from a rejection sampling procedure. Augmenting the rejected proposals and modelling the density in a similar fashion as $\Lambda\sigma(f)$ leads a model for density estimation similar to the *Gaussian Process Density Sampler* of Adams et al. (2009a), with obvious advantages. Naturally the scope is limited to densities with bounded support.

- Chapter 3: “*Detecting Multiplexing in Neuronal Spike Trains*”.

Chapter 3 attempted to explore how neurons encode information from dual stimuli by discovering relationships between the intensity of the dual stimuli trials to the intensities of the single sound trials. Early attempts to model the dual stimuli trials were through a continuous-time Markov chain, switching between states of firing like A and firing like B, using the uniformization based MCMC approach of Rao (2012). While this model is closest to the multiplexing hypothesis, it was later decided to relax the assumption of firing purely like A or purely like B. The dual intensity was subsequently modelled as a dynamic superposition of the intensity functions for the single stimulus trials. The weight function was modelled nonparametrically as a transformed mixture of Gaussian process and the zero function. It was attempted to learn a cell specific parameter corresponding to the probability of the cell to create dynamic dual stimuli trials. The MCMC algorithm for this model experienced difficulties with mixing, most surely due to the amount of missing data which is augmented on each iteration. It was decided to move from the probit link function to the logistic link function and use the Polya-Gamma data augmentation technique in the hope of better mixing. This did not result in a sufficient improvement in mixing. Overall I am not convinced that the Polya-Gamma data augmentation trumps the probit augmentation, as the time required to generate the Polya-Gamma random variables made a non-negligible contribu-

tion to the overall runtime of the MCMC. In addition to the issues surrounding mixing, there is a tendency to over-fit the data by favouring trials to be dynamic. The poor mixing prevented meaningful interpretation of the MCMC output and so the model was simplified by considering trials to be static if the posterior probability of their time-scale loaded heavily on long time-scales. In simulation studies this did not provide a good means of classification, as once again there is a tendency to over-fit the data by favouring small time-scales. The eventual model assumed all trials were dynamic and was able to provide good posterior credible intervals for the function values but neither the time-scale of the GP for each trial nor the cell's distribution of time-scales was learned well. For the triplet analysed, the weight functions did not exhibit much temporal dependence.

The stationary assumption of the Gaussian process could also be challenged. In the data for the single stimulus trials it is commonly observed that there is an initial rapid increase in firing rate, followed by a smaller less time dependent firing rate. This favours a smaller time-scale to begin with, followed by a longer time-scale. Modelling this with a stationary Gaussian process produces curves that are either fluctuating too much in the latter part or failing to adapt to the rapid increase in the former part. To overcome this I explored modelling the intensities using the nested GP model of Zhu and Dunson (2013), which also admits a state-space representation. In experimentation this failed to provide noticeable improvement over the stationary Matern GP.

- Chapter 4: *“Continuous Optimization for Variable Selection”*. Chapter 4 generalized existing results for Lasso, based on the compatibility criterion and sub-Gaussian error assumptions, for the point-mass-Laplace mixture prior which amounts to a penalty containing both ℓ_0 and ℓ_1 norms. The previous results for

Lasso are recovered as a special case when $\ell_0 = 0$. Additional theoretical results were provided using the results of Zhang and Zhang (2012), which assume the η -Null consistency criterion and nonzero restricted invertibility factor. While the former is a direct generalization of the “classical” proofs for lasso due to Buehlmann and van de Geer (2011), future work must consider which of these results is stronger and how these assumptions relate to many of the other assumptions used in developing theory for sparse estimation procedures, of which there are many.

Two new heuristic algorithms were presented for finding good quality local optima in nonconvex problems resulting from separable penalties, namely, through EM algorithms based on orthogonal data augmentation and location-mixtures representations. The resulting parameter mappings were found to be identical to a number of other algorithms proposed in the literature, namely, the classical proximal gradient method from convex analysis, LQ decompositions of the design matrix by Figueiredo and Nowak (2003) and the discrete-first order method proposed in Bertsimas et al. (2016). All of these algorithms attempt to exploit the separability of the prior and from different starting points end up at the same parameter mappings. The connection that I observe helps to provide new perspectives on each algorithm and provides access to theory beyond EM from convex analysis. While there is a lot of theory for the proximal gradient and coordinate descent methods applied to convex problems, there is little known about their performance when applied to nonconvex problems. Simulation studies suggested that coordinate descent methods are less susceptible to entrapment from suboptimal local modes in nonconvex problems.

- Chapter 5: “*Discrete Optimization for Variable Selection*”.

Chapter 5 explored the utility of some exact discrete optimization methods for

solving challenging mathematical optimization problems in variable selection. Since the early monotonicity-based branch and bound algorithms of the 1970s, exact methods have received little attention within the statistics community, while they have been actively developed outside. Mixed integer programming has established itself as a very powerful and flexible approach for solving a wide variety of discrete optimization problems. The performance of mixed integer solvers has increased dramatically over the past decades with remarkable speedups being provided by the development of new theory and methodologies but also through technological progress. The wide variety of optimization problems that can be solved through mixed integer programming means that the theory and methods continue to be actively developed, with solver performance ever improving. The MIQP model developed for obtaining the MAP estimate resulting from the point-mass-Laplace mixture prior converges in a matter of seconds for real moderately-sized problems. This chapter has demonstrated that solving some discrete optimization problems in statistics to global optimality is, in fact, practically viable. It encourages a renewed interest in these methods, with the possibility of them finding utility in many other areas of statistics. There are obvious extensions to group variable selection and there is active research on the development of mixed integer programming models for sparse classification, see Bertsimas et al. (2017). Beyond mixed integer methods, there is active research on solving high dimensional sparse regression through binary convex reformulations, see Bertsimas and Parys (2017). Through use of a novel cutting plane algorithm, the authors claim to be able to solve to global optimality sparse regression problems with a number of observations and predictors in the hundreds of thousands in a matter of seconds. Just as one should not expect all discrete optimizations to be impossible, one

should not expect all to be easy either. Indeed it was observed that the MIQCP model developed for the g-prior did not converge to global optimality in a practically reasonable amount of time. This is because the g-prior is rather objective and does not concentrate posterior mass on a single model at the same rate as the point-mass-Laplace mixture prior, making it difficult to solve through branch and bound. There are, however, some other strategies that could be tried to improve performance. Among models of the same size, the model with largest posterior probability is the model that minimizes the residual sum of squares. If one were search for the optimal model of a certain size k , one could solve the best subset selection problem with the constraint $\sum \gamma_i = k$. This would totally avoid the need to do a piecewise linear approximation to the logarithm term. Finding the globally optimal model would then correspond to finding the optimal model within each model size. If the number of possible model sizes could be restricted through optimality-based bounds tightening, and this number were small, then it could be implemented in the following way. Starting by finding the optimal model of the largest size, the optimal model of the second largest size could be found at a fraction of the total cost by implementing a *callback* to the solver. Implementing a callback allows one to add additional constraints without restarting the solver from the very beginning. Reducing the model size would render some previously feasible nodes infeasible. The solver is then able to solve the new problem in much less time using information from the previous problem. How well this works in practice is yet to be explored. In addition, the focus in this chapter has been on finding the mode of $p(\gamma|Y)$ for the g-prior. Had one sought the joint mode $p(\beta, \gamma|Y)$, then it would have been considerably easier.

Appendix A

Figures

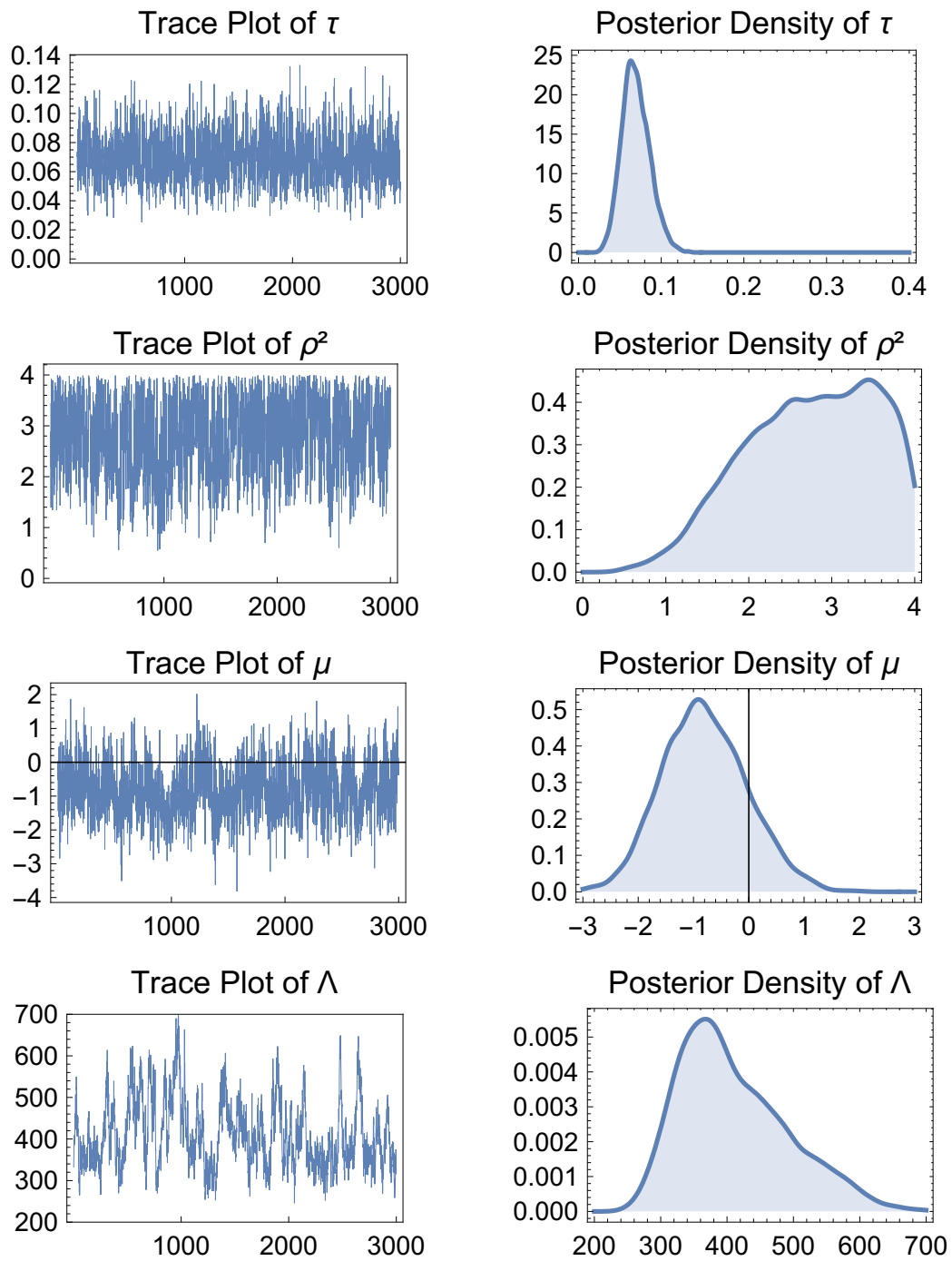


FIGURE A.1: Posterior summaries based on 3000 MCMC post burn-in draws on the simulated dataset I.

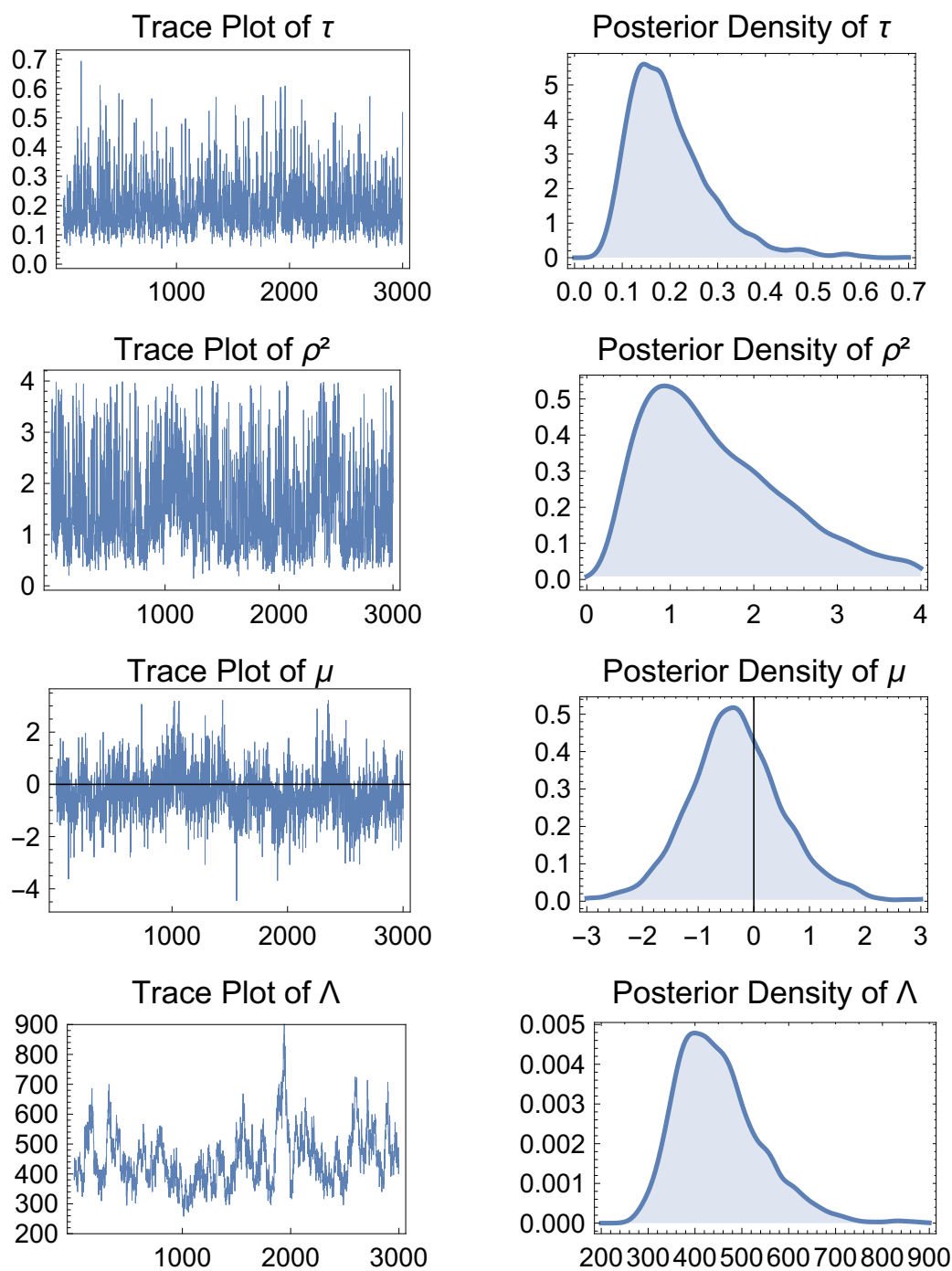


FIGURE A.2: Posterior summaries based on 3000 MCMC post burn-in draws on the coal mining dataset.

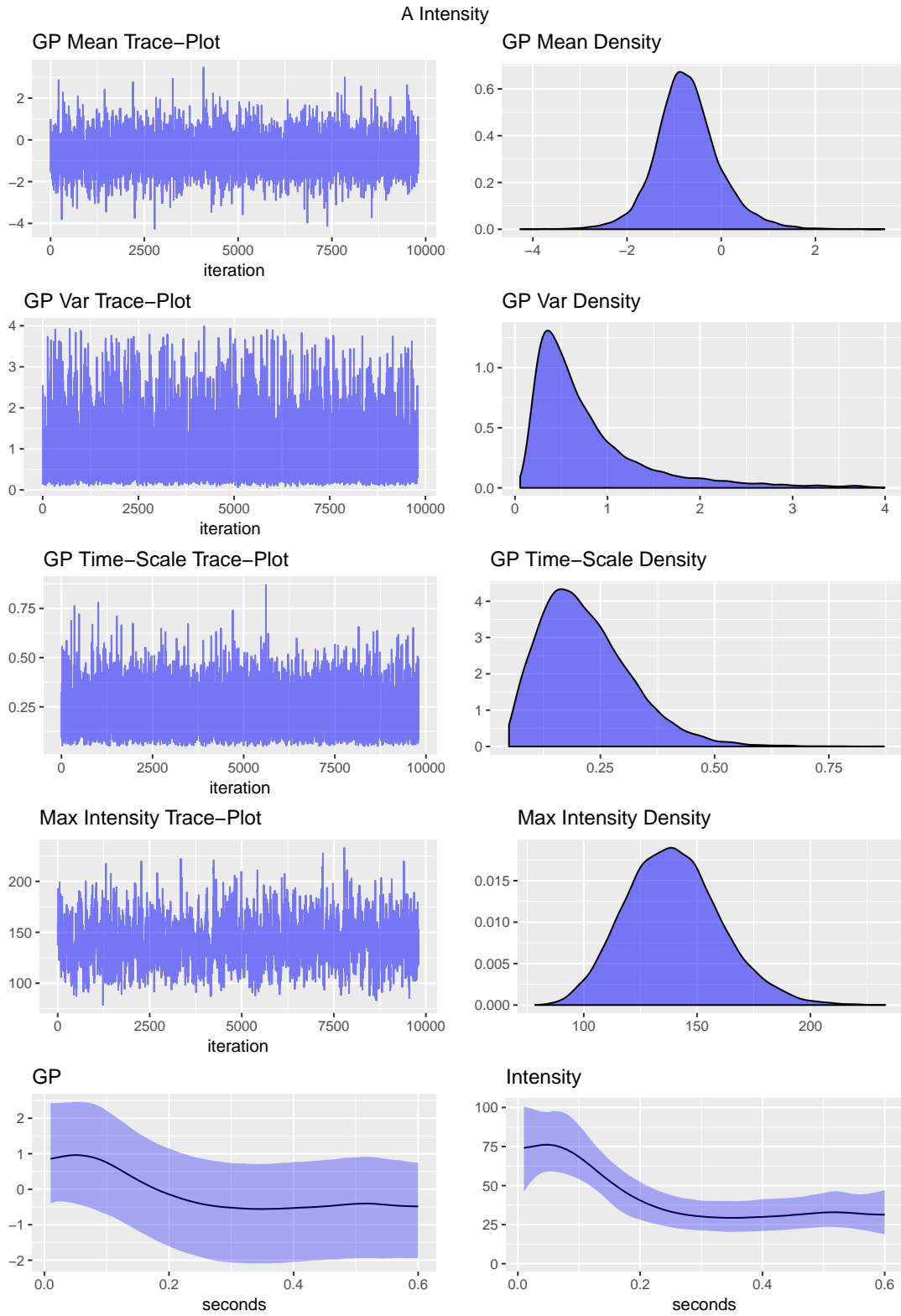


FIGURE A.3: Posterior summaries for the A trial intensity of cell YKIC092311.1 609Hz 24° 742Hz -6° based on retaining every 10th iterate from 10000 MCMC iterates

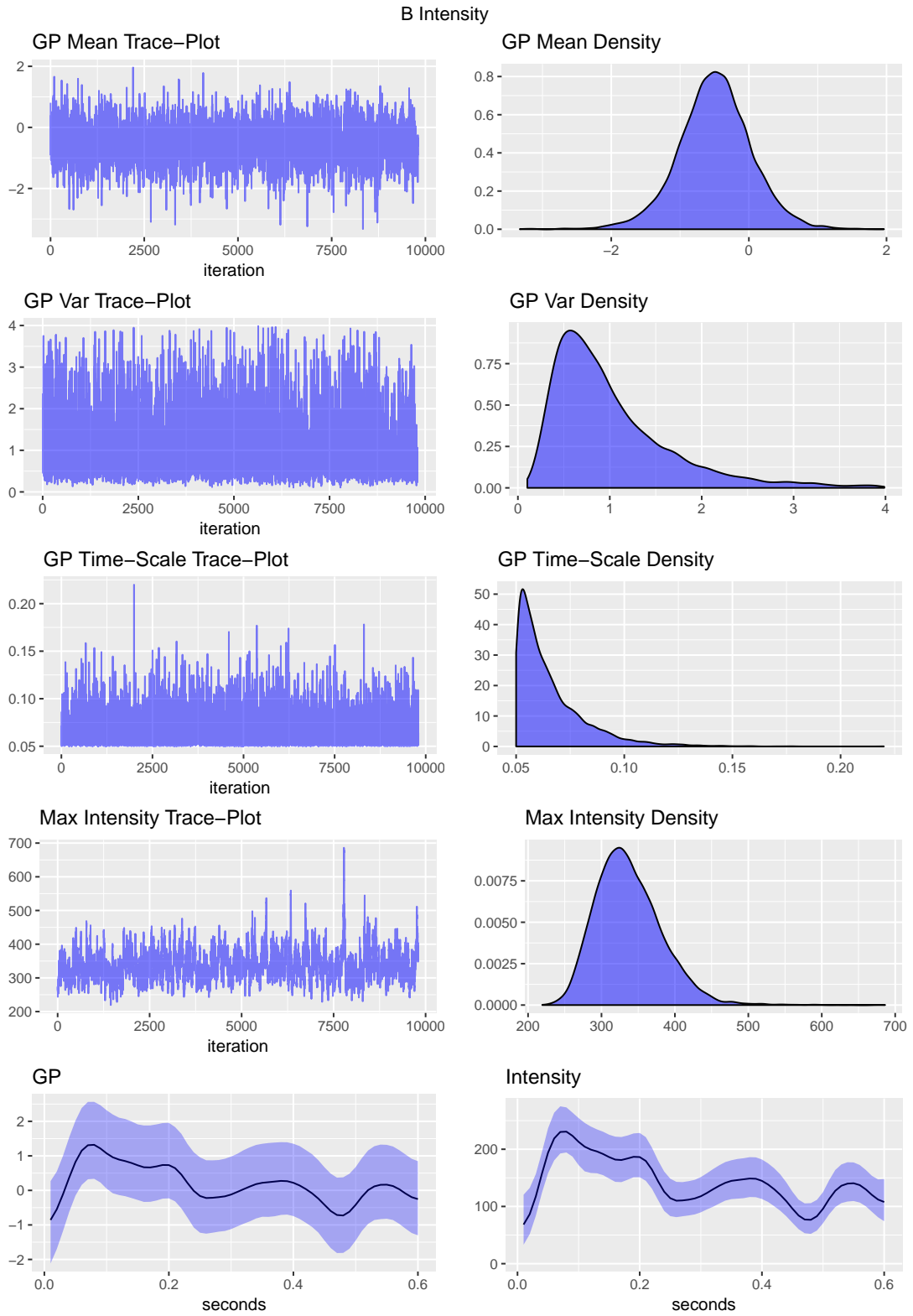


FIGURE A.4: Posterior summaries for the B trial intensity of cell YKIC092311.1 609Hz 24° 742Hz -6° based on retaining every 10th iterate from 10000 MCMC iterates

Bibliography

- Aarts, E. and Lenstra, J. K. (eds.) (1997), *Local Search in Combinatorial Optimization*, John Wiley & Sons, Inc., New York, NY, USA, 1st edn.
- Achterberg, T., Koch, T., and Martin, A. (2005), “Branching Rules Revisited,” *Oper. Res. Lett.*, 33, 42–54.
- Adams, R. P., Murray, I., and MacKay, D. J. C. (2009a), “Nonparametric Bayesian Density Modeling with Gaussian Processes,” .
- Adams, R. P., Murray, I., and MacKay, D. J. C. (2009b), “Tractable Nonparametric Bayesian Inference in Poisson Processes with Gaussian Process Intensities,” in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pp. 9–16, New York, NY, USA, ACM.
- Albert, J. H. and Chib, S. (1993), “Bayesian Analysis of Binary and Polychotomous Response Data,” *Journal of the American Statistical Association*, 88, 669–679.
- Armagan, A., Dunson, D. B., and Lee, J. (2013), “Generalized double pareto shrinkage,” *Statistica Sinica*, 23, 119–143.
- Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008), “Gaussian predictive process models for large spatial data sets,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70, 825–848.
- Bar-Shalom, Y., Kirubarajan, T., and Li, X.-R. (2002), *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, Inc., New York, NY, USA.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. (1996), “Branch-and-Price: Column Generation for Solving Huge Integer Programs,” *Operations Research*, 46, 316–329.
- Beale, E. and Tomlin, J. (1970), “Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables,” in *Proceedings of the 5th International Conference on Operations Research*, ed. J. In Lawrence, p. 447, Venice, Italy.

- Beale, E. M. L., Kendall, M. G., and Mann, D. W. (1967), “The Discarding of Variables in Multivariate Analysis,” *Biometrika*, 54, 357–366.
- Beck, A. and Teboulle, M. (2009), “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems,” *SIAM Journal on Imaging Sciences*, 2, 183–202.
- Bellman, R. (1954), “The theory of dynamic programming,” *Bull. Amer. Math. Soc.*, 60, 503–515.
- Bellman, R. (1957), *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1 edn.
- Belotti, P., Lee, J., Liberti, L., Margot, F., and Wächter, A. (2009), “Branching and bounds tightening techniques for non-convex MINLP,” *Optimization Methods and Software*, 24, 597–634.
- Belotti, P., Kirches, C., Leyffer, S., Linderoth, J. T., Luedtke, J., and Mahajan, A. (2012), “Mixed-Integer Nonlinear Optimization,” .
- Bertsimas, D. and Parys, B. V. (2017), “Sparse High-Dimensional Regression: Exact Scalable Algorithms and Phase Transitions,” .
- Bertsimas, D., King, A., and Mazumder, R. (2016), “Best subset selection via a modern optimization lens,” *Ann. Statist.*, 44, 813–852.
- Bertsimas, D., Pauphilet, J., and Parys, B. V. (2017), “Sparse Classification and Phase Transitions: A Discrete Optimization Perspective,” .
- Bixby, R. E. (2012), “A Brief History of Linear and Mixed-Integer Programming Computation,” *DOCUMENTA MATHEMATICA*, pp. 107–121.
- Blumensath, T. and Davies, M. E. (2008), “Iterative Thresholding for Sparse Approximations,” *Journal of Fourier Analysis and Applications*, 14, 629–654.
- Breiman, L. and Friedman, J. H. (1985), “Estimating Optimal Transformations for Multiple Regression and Correlation: Rejoinder,” *Journal of the American Statistical Association*, 80, 614–619.
- Brusco, M. and Stahl, S. (2005), *Branch-and-Bound Applications in Combinatorial Data Analysis (Statistics and Computing)*, Springer, 1 edn.
- Buehlmann, P. and van de Geer, S. (2011), *Statistics for High-Dimensional Data: Methods, Theory and Applications*, Springer Publishing Company, Incorporated, 1st edn.
- Castillo, I., Schmidt-Hieber, J., and van der Vaart, A. (2015), “Bayesian linear regression with sparse priors,” *Ann. Statist.*, 43, 1986–2018.

- Choi, T. and Schervish, M. J. (2007), “On posterior consistency in nonparametric regression problems,” *Journal of Multivariate Analysis*, 98, 1969 – 1987.
- Chow, E. and Saad, Y. (2014), “Preconditioned Krylov Subspace Methods for Sampling Multivariate Gaussian Distributions,” *SIAM Journal on Scientific Computing*, 36, A588–A608.
- Christensen, R. A. (1987), *Plane Answers to Complex Questions: The Theory of Linear Models*, Springer-Verlag New York, Inc., New York, NY, USA.
- Clyde, M., DeSimone, H., and Parmigiani, G. (1996), “Prediction via orthogonalized model mixing,” *Journal of the American Statistical Association*, 91, 1197–1208.
- Cox, D. R. (1955), “Some Statistical Methods Connected with Series of Events,” *Journal of the Royal Statistical Society. Series B (Methodological)*, 17, 129–164.
- Dakin, R. J. (1965), “A tree-search algorithm for mixed integer programming problems,” *The Computer Journal*, 8, 250–255.
- Das, A. and Kempe, D. (2008), “Algorithms for Subset Selection in Linear Regression,” in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC ’08, pp. 45–54, New York, NY, USA, ACM.
- Das, A. and Kempe, D. (2011), “Submodular meets Spectral: Greedy Algorithms for Subset Selection, Sparse Approximation and Dictionary Selection.” in *ICML*, eds. L. Getoor and T. Scheffer, pp. 1057–1064, Omnipress.
- Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016), “Hierarchical Nearest-Neighbor Gaussian Process Models for Large Geostatistical Datasets,” *Journal of the American Statistical Association*, 111, 800–812.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), “Maximum likelihood from incomplete data via the EM algorithm,” *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39, 1–38.
- Dorigo, M. and Di Caro, G. (1999), “New Ideas in Optimization,” chap. The Ant Colony Optimization Meta-heuristic, pp. 11–32, McGraw-Hill Ltd., UK, Maidenhead, UK, England.
- Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. (1987), “Hybrid Monte Carlo,” *Physics Letters B*, 195, 216 – 222.
- Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), “Least angle regression,” *Annals of Statistics*, 32, 407–499.
- Efroymson, M. A. (1960), “Multiple Regression Analysis,” in *Mathematical Methods for Digital Computers*, eds. A. Ralston and H. S. Wilf, John Wiley, New York.

- Figueiredo, M. A. T. and Nowak, R. D. (2003), “An EM algorithm for wavelet-based image restoration,” *IEEE Transactions on Image Processing*, 12, 906–916.
- Foster, D. P. and George, E. I. (1994), “The Risk Inflation Criterion for Multiple Regression,” *Ann. Statist.*, 22, 1947–1975.
- Friedman, J., Hastie, T., Hfling, H., and Tibshirani, R. (2007), “Pathwise coordinate optimization,” *Ann. Appl. Stat.*, 1, 302–332.
- Friedman, J., Hastie, T., and Tibshirani, R. (2010), “Regularization Paths for Generalized Linear Models via Coordinate Descent,” *Journal of Statistical Software, Articles*, 33, 1–22.
- Furnival, G. M. and Wilson, R. W. (1974), “Regressions by Leaps and Bounds,” *Technometrics*, 16, 499–511.
- Gendreau, M. and Potvin, J.-Y. (2010), *Handbook of Metaheuristics*, Springer Publishing Company, Incorporated, 2nd edn.
- George, E. I. and McCulloch, R. E. (1993), “Variable Selection via Gibbs Sampling,” *Journal of the American Statistical Association*, 88, 881–889.
- Ghosh, J. and Clyde, M. A. (2011), “RaoBlackwellization for Bayesian Variable Selection and Model Averaging in Linear and Binary Regression: A Novel Data Augmentation Approach,” *Journal of the American Statistical Association*, 106, 1041–1052.
- Glover, F. (1989), “Tabu SearchPart I,” *ORSA Journal on Computing*, 1, 190–206.
- Glover, F. (1990), “Tabu SearchPart II,” *ORSA Journal on Computing*, 2, 4–32.
- Gurobi Optimization, I. (2016), “Gurobi Optimizer Reference Manual,” .
- Halko, N., Martinsson, P. G., and Tropp, J. A. (2011), “Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions,” *SIAM Review*, 53, 217–288.
- Hans, C., Dobra, A., and West, M. (2007), “Shotgun Stochastic Search for Large p Regression,” *Journal of the American Statistical Association*, 102, 507–516.
- Hartikainen, J. and Särkkä, S. (2010), “Kalman filtering and smoothing solutions to temporal Gaussian process regression models,” in *2010 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 379–384.
- Hocking, R. R. and Hocking, R. R. (1976), “A Biometrics Invited Paper. The analysis and selection of variables in linear regression,” *Biometrics*, pp. 1–49.

- Hocking, R. R. and Leslie, R. N. (1967), “Selection of the Best Subset in Regression Analysis,” *Technometrics*, 9, 531–540.
- Holland, J. H. (1992), *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, MA, USA.
- Ibaraki, T. (1976), “Theoretical comparisons of search strategies in branch-and-bound algorithms,” *International Journal of Computer & Information Sciences*, 5, 315–344.
- IBM (2011), *IBM ILOG CPLEX Optimization Studio CPLEX User’s Manual*.
- Ishwaran, H. and Rao, J. S. (2005), “Spike and slab variable selection: Frequentist and Bayesian strategies,” *Ann. Statist.*, 33, 730–773.
- Jarrett, R. G. (1979), “A Note on the Intervals Between Coal-Mining Disasters,” *Biometrika*, 66, 191–193.
- Kannan, R. and Monma, C. L. (1978), “On the Computational Complexity of Integer Programming Problems,” in *Optimization and Operations Research*, eds. R. Henn, B. Korte, and W. Oettli, pp. 161–172, Berlin, Heidelberg, Springer Berlin Heidelberg.
- Kass, R. E. and Wasserman, L. (1995), “A Reference Bayesian Test for Nested Hypotheses and its Relationship to the Schwarz Criterion,” *Journal of the American Statistical Association*, 90, 928–934.
- Kingman, J. F. C. (1993), *Poisson processes*, vol. 3 of *Oxford Studies in Probability*, The Clarendon Press Oxford University Press, New York, Oxford Science Publications.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983), “Optimization by simulated annealing,” *SCIENCE*, 220, 671–680.
- Kloeden, P. E. and Platen, E. (2011), *Numerical Solution of Stochastic Differential Equations*, Springer, New York, corrected edn.
- Laguna, M. and Marti, R. (1999), “GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization,” *INFORMS Journal on Computing*, 11, 44–52.
- LaMotte, L. R. and Hocking, R. R. (1970), “Computational Efficiency in the Selection of Regression Variables,” *Technometrics*, 12, 83–93.
- Land, A. H. and Doig, A. G. (1960), “An Automatic Method of Solving Discrete Programming Problems,” *Econometrica*, 28, pp. 497–520.

- Lange, K. (2016), *MM Optimization Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA.
- Lewis, P. A. W. and Shedler, G. S. (1979), “Simulation of Nonhomogeneous Poisson Processes with Degree-Two Exponential Polynomial Rate Function,” *Operations Research*, 27, 1026–1040.
- Liang, F., Paulo, R., Molina, G., Clyde, M. A., and Berger, J. O. (2008), “Mixtures of g Priors for Bayesian Variable Selection,” *Journal of the American Statistical Association*, 103, 410–423.
- Linderoth, J. T. and Savelsbergh, M. W. P. (1999), “A Computational Study of Search Strategies for Mixed Integer Programming,” *INFORMS Journal on Computing*, 11, 173–187.
- Little, J. D. C., Murty, K. G., Sweeney, D. W., and Karel, C. (1963), “An Algorithm for the Traveling Salesman Problem,” *Oper. Res.*, 11, 972–989.
- Lumley, T. (2017), *leaps: Regression Subset Selection*, R package version 3.0.
- Maguire, B. A., Pearson, E. S., and Wynn, A. H. A. (1952), “The Time Intervals Between Industrial Accidents,” *Biometrika*, 39, 168–180.
- Marchand, H., Martin, A., Weismantel, R., and Wolsey, L. (2002), “Cutting planes in integer and mixed integer programming,” *Discrete Applied Mathematics*, 123, 397 – 446.
- Marjanovic, G., Ulfarsson, M. O., and Hero, A. O. (2015), “MIST: L0 sparse linear regression with momentum,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3551–3555.
- Mitchell, T. J. and Beauchamp, J. J. (1988), “Bayesian Variable Selection in Linear Regression,” *Journal of the American Statistical Association*, 83, 1023–1032.
- Mladenović, N. and Hansen, P. (1997), “Variable Neighborhood Search,” *Comput. Oper. Res.*, 24, 1097–1100.
- Murray, I., Adams, R. P., and MacKay, D. J. C. (2010), “Elliptical slice sampling,” 9, 541–548.
- Neal, R. M. (2000), “Markov Chain Sampling Methods for Dirichlet Process Mixture Models,” *Journal of Computational and Graphical Statistics*, 9, 249–265.
- Neal, R. M. (2003), “Slice sampling,” *Ann. Statist.*, 31, 705–767.
- Nesterov, Y. (2013), “Gradient methods for minimizing composite functions,” *Mathematical Programming*, 140, 125–161.

- Nesterov, Y. (2014), *Introductory Lectures on Convex Optimization: A Basic Course*, Springer Publishing Company, Incorporated, 1 edn.
- Nutini, J., Schmidt, M., Laradji, I. H., Friedlander, M., and Koepke, H. (2015), “Coordinate Descent Converges Faster with the Gauss-southwell Rule Than Random Selection,” in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, pp. 1632–1641, JMLR.org.
- Polson, N. G., Scott, J. G., and Windle, J. (2013), “Bayesian Inference for Logistic Models Using PlyaGamma Latent Variables,” *Journal of the American Statistical Association*, 108, 1339–1349.
- Polson, N. G., Scott, J. G., and Willard, B. T. (2015), “Proximal Algorithms in Statistics and Machine Learning,” *Statist. Sci.*, 30, 559–581.
- Prado, R. and West, M. (2010), *Time Series: Modeling, Computation, and Inference*, Chapman & Hall/CRC, 1st edn.
- Rao, V. A. (2012), “Markov chain Monte Carlo for continuous-time discrete-state systems,” Ph.D. thesis, University College London.
- Rasmussen, C. E. and Williams, C. K. I. (2005), *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press.
- Rockafellar, R. T. (1970), *Convex analysis*, Princeton Mathematical Series, Princeton University Press, Princeton, N. J.
- Rockova, V. and George, E. I. (2016), “The Spike-and-Slab LASSO,” *Journal of the American Statistical Association*, 0, 0–0.
- Saha, A. and Tewari, A. (2013), “On the Nonasymptotic Convergence of Cyclic Coordinate Descent Methods,” *SIAM Journal on Optimization*, 23, 576–601.
- Scott, S. L. (2017), *BoomSpikeSlab: MCMC for Spike and Slab Regression*, R package version 0.8.0.
- Taddy, M. A. and Kottas, A. (2012), “Mixture Modeling for Marked Poisson Processes,” *Bayesian Anal.*, 7, 335–362.
- Tokdar, S. T. and Ghosh, J. K. (2007), “Posterior consistency of logistic Gaussian process priors in density estimation,” *Journal of Statistical Planning and Inference*, 137, 34 – 42.
- van der Vaart, A. W. and van Zanten, J. H. (2008), “Rates of contraction of posterior distributions based on Gaussian process priors,” *Ann. Statist.*, 36, 1435–1463.

- Vielma, J. P., Ahmed, S., and Nemhauser, G. (2010), “Mixed-Integer Models for Non-separable Piecewise-Linear Optimization: Unifying Framework and Extensions,” *Operations Research*, 58, 303–315.
- Wainwright, M. J. (2009), “Information-Theoretic Limits on Sparsity Recovery in the High-Dimensional and Noisy Setting,” *IEEE Transactions on Information Theory*, 55, 5728–5741.
- Xiong, S., Dai, B., Huling, J., and Qian, P. Z. G. (2016), “Orthogonalizing EM: A Design-Based Least Squares Algorithm,” *Technometrics*, 58, 285–293.
- Yang, Y., Wainwright, M. J., and Jordan, M. I. (2016), “On the computational complexity of high-dimensional Bayesian variable selection,” *Ann. Statist.*, 44, 2497–2532.
- Yu, Y. and Meng, X.-L. (2011), “To Center or Not to Center: That Is Not the Question An Ancillarity Sufficiency Interweaving Strategy (ASIS) for Boosting MCMC Efficiency,” *Journal of Computational and Graphical Statistics*, 20, 531–570.
- Zhang, C.-H. and Zhang, T. (2012), “A General Theory of Concave Regularization for High-Dimensional Sparse Estimation Problems,” *Statist. Sci.*, 27, 576–593.
- Zhu, B. and Dunson, D. B. (2013), “Locally Adaptive Bayes Nonparametric Regression via Nested Gaussian Processes,” *J Am Stat Assoc*, 108, 10.1080/01621459.2013.838568, 25328260[pmid].

Biography

Michael Lindon was born on 4th April 1990 in Bournemouth, England. He received his BSc MPhys degree in Physics from the University of Warwick in 2012 and his MSc in Statistical Science from Duke University in 2014. He is expected to earn his PhD in Statistical Science from Duke University in 2018, after which he will join Tesla as a senior data scientist.