Bayesian Calibration of Flood Inundation Simulators Using an Observation of Flood Extent

By

Simon Peter Barratt Woodhead

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Science

March 2007

Department of Mathematics

Abstract

We develop a Bayesian framework for calibrating flood inundation simulators on an observation of flood extent, and making calibrated predictions of a future event. We illustrate the framework using the binary channel (BC) model for the likelihood of the observed flood extent given a simulation of flood extent. The BC model leads to poor results, and this motivates the search for a more appropriate likelihood model, which forms the basis for the rest of the thesis.

We extend the Ising model to regression on a binary image and review methods for dealing with the intractable normalising constant. We propose novel applications of path sampling, extend path sampling to sampling over areas, and develop approximations to path sampling. We also develop the heterogeneous binary channel (HBC) model to test the effect of heterogeneity and spatial dependence. We extend the hidden conditional autoregressive (HCAR) model to regression on a binary image. We show that the limit of the HCAR model as the parameters approach the boundary is the (improper) hidden intrinsic autoregressive (HIAR) model. We prove that the HIAR model can be used for calibration but not calibrated prediction. We develop a number of methods for improving mixing of the MCMC algorithm. We explore two extensions of the HCAR model. First the heterogeneous HCAR (HHCAR) model, which represents heterogeneity, and second the continuous HCAR (CHCAR) model, which uses continuous simulation values.

In conclusion, using our Bayesian framework we can replicate the results of less rigorous approaches, for example generalised likelihood uncertainty estimation (GLUE), and make probabilistic predictions which are not possible in these less rigorous approaches. Future work would further develop the likelihood models.

Acknowledgements

I would like to thank my two supervisors, Paul Bates and Peter Green, from whom I have been fortunate enough to receive expert advice in both geography and statistics. Both have been patient and supportive during the course of my PhD. I am also grateful to Stuart Coles who supported me during the early stages.

Thanks go to Jim Hall, Florian Pappenberger, Keith Beven, Ezio Todini and Clive Anderson for discussing their work with me. Håvard Rue and Douglas Nychka have also been a valuable source of expertise.

I would like to thank the LISFLOOD-FP team at the University of Bristol, Neil Hunter, Matt Horritt, Matt Wilson and Tim Fewtrell, for all their help.

This work was made possible by the generous funding of the NERC and the University of Bristol.

My friends have been a constant support throughout my PhD. I would like to thank Marina, Matt, Huw, Ian, Martin, Geoff and Chris for making our office a fun environment to work in.

I would like to thank my parents Garth and Lissie, and my siblings Rebekah, Sarah and Tim, for their encouragement, and my nephews Olly and Jude for being a constant source of amusement.

Finally, I am indebted to my wife Anna, without whom I would have surely never finished this.

Declaration

I, the author, declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Bristol. The work is original except where indicated by special reference in the text and no part of the dissertation has been submitted for any other degree.

The views expressed in the dissertation are those of the author and in no way represent those of the University of Bristol.

The dissertation has not been presented to any other University for examination either in the United Kingdom or overseas.

Simon Peter Barratt Woodhead

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This introduction sets the scene for the analysis that is carried out in the remaining chapters, namely the development of a Bayesian framework for the calibration of flood inundation simulators using an observation of flood extent. We begin with an assessment of the continuing global and national flood hazard, and the need for calibrated flood inundation simulators. This is followed by a description of two methods for flood inundation prediction: the current practice adopted by the Environment Agency in the UK, and generalised likelihood uncertainty estimation (GLUE) which has been ubiquitous in recent research. The shortcomings of these two approaches and their consequences for flood management are used to argue for a new paradigm for calibration and calibrated prediction. We consider the features required of our framework, and justify the need for a Bayesian approach and use of observations of flood extent. At the end of the chapter we give an overview of the rest of the thesis.

1.1 Flood Hazard Prediction and Uncertainty

The Oxford English Dictionary (1989) defines a flood to be

> An overflowing or irruption of a great body of water over land not usually submerged; an inundation, a deluge.

Examples of this phenomenon are found in confined regions after sudden extreme rainfall, on coasts during storms and in river basins after sustained heavy rainfall. The complexity of flood hydraulics and strong dependence on topography make

prediction very difficult (Anderson *et al.*, 1996).

According to the World Disasters Report (2001) floods affect on average 140 million people worldwide every year, more than all the other natural hazards combined. Drowning, forceful destruction of property and sediment transport are all part of the multifaceted flood hazard. The transported sediment may consist of sewage, pesticides or other chemicals, spreading disease and causing further destruction to property. The impact of flooding differs between developed and developing countries: the financial impact is greater in developed countries where high monetary value is attached to buildings, but the number of flood related deaths is higher in developing countries because of the absence of flood management.

In England and Wales four to five million people and 1.9 million homes are estimated to be at risk from flooding (Harman *et al.*, 2002). The total value of properties at risk exceeds £200 billion. The average annual economic damage from flooding and coastal erosion is over £1 billion per year but without any flood defences is predicted to be over £3.5 billion per year (Halcrow Group Ltd, 2001). One recent example of flooding occurred on the 16th August 2004 when floods caused widespread devastation to Boscastle in Cornwall. No lives were lost but 100 homes were affected, vehicles were swept away by walls of water and much infrastructure was damaged (Living with the risk, booklet).

The Foresight Programme's Future Flooding report (Foresight, 2004) was commissioned by the Office of National Statistics to provide a vision of flood and coastal erosion risk between 2030 and 2100. Such long term predictions are necessary because decisions made today may have a profound impact on flood risk in the future. The focus of the report is risk-based decision making, where the risk of a particular outcome is defined to be the product of its probability and consequence.

The report begins by examining how flooding and coastal erosion might develop under the baseline assumption that flood risk management remains unchanged. Coastal grazing marshes and other similar environmental habitats are threatened, and by the 2080s the average annual damage from coastal erosion will increase by

3 to 9 times, although this is still far less than current losses from flooding, and the number of people at high risk from river or coastal flooding will increase from the current figure of 1.6 million to between 2.3 and 3.6 million.

The main factors affecting flood risk were identified to be climate change, through increasing precipitation and sea-level rise; urbanisation and rural land management by increasing run-off; environmental regulations by limiting maintenance of flood defences and flood risk management along rivers, estuaries and coasts; and growing national wealth by increasing the value of property and assets at risk.

The Future Flooding report considers 120 responses with respect to their potential for reducing future flood risk. Considering the impact of each of these responses in terms of future flood risk provides a common approach for comparing a variety of very different options. Risk analysis means investigating the possible ways a response could influence the future and attaching probabilities to these future scenarios (Ministry of Agriculture, Fisheries and Food, 2000). The decision maker would then select the response which maximises the expected risk reduction less the expected cost. The cost of a response relates to its environmental, social and economic sustainability.

Isolated responses were unable to adequately reduce risk and meet the sustainability criteria, although catchment-wide storage, land-use planning and realigning coastal defenses scored highly. Instead an integrated portfolio of responses was proposed, which was found to reduce the risks of river and coastal flooding for the worse-case scenario from £20 billion per year down to £2 billion per year in the 2080s. This figure is still double the present day annual damage. This integrated response would cost between £22 and £75 billion by 2080 and to meet the sustainability criteria must be implemented sensitively. The task of flood risk management is made significantly easier if it is combined with efforts to reduce climate change. It was found that reducing greenhouse emissions alone could reduce the annual damage by £6 billion per year by the 2080s (Foresight, 2004).

The Future Flooding report concludes that it is not enough to maintain current levels of flood risk, rather it is essential they are reduced. This is due to the expectations of society and the economic benefits of reducing flood risk which will be much greater than the costs. The report acknowledges that risk is inherently uncertain and that we need to reduce uncertainty in our predictions of risk.

The Government's current strategy for flood and coastal erosion management in England is outlined in the report "Making space for water" (Department for Environment, Food and Rural Affairs, 2005). The Government is taking a holistic, risk-driven approach which accounts for all sources of flooding and integrates flood and coastal risk management with Government policies. The strategy is to reduce the threat to people and property whilst delivering environmental, social and economic benefits consistent with the Government's sustainable development principles. A key component of the strategy is adaptability to climate change and to ensure that decisions are increasing risk driven. For the latter of these they identify that it is essential to include better data on the consequences of flooding. The Government spent £600 million on flood risk management in 2005-6 but proposed projects still have to be prioritised and this is accomplished by risk-based decision making. The Government's commitment to flood risk management strategies is a condition of the Association of British Insurer's commitment to cover most properties at risk. The "Making space for water" policy falls short of enforcing Flood Risk Assessment in the planning process, but it will be strongly encouraged. In 2003/4 local planning authorities approved 12% of the planning proposals that were objected to by the Environment Agency (A better place?, booklet).

The Department for Environment, Food and Rural Affairs (DEFRA) has responsibility for the implementation of the Government's policy for flood and coastal defence in England and manages flood emergencies. The Environment Agency (EA) supervises the implementation of this policy which is the joint responsibility of the operating authorities: the EA, Internal Drainage Boards and local authorities. The EA is also responsible for flood forecasting and warning, and increasing public awareness of flood risk. DEFRA has produced a series of Flood and

Coastal Defence Project Appraisal Guidance reports for the operating authorities. The fourth of these guides (Ministry of Agriculture, Fisheries and Food, 2000) is a guide on approaches to risk and once again highlights the Government's strategy of risk-based decision making.

One way the flood risk can be represented is as a map of the probability of flood inundation for a particular design event, for example the 1 in 100 years level flood (i.e. a flood with a probability of 0.01 of occurring in any given year). For most rivers it is unlikely a 1 in 100 years level flood will have been observed within the history of current monitoring practice, so direct statistical analysis of flood extent is not possible. However, in England and Wales gauging stations continuously measure river elevation over time which can be used as input conditions for a physical or numerical simulator to predict flood extent (Hall and Anderson, 2002). Flood modelling is particularly challenging because the low gradients indicative of floodplains mean the flood extent is very sensitive to small perturbations in water surface elevation, so accurate simulators and accurate topographic data are required.

A number of flood inundation simulators have been developed, each with a different emphasis on process representation, computational efficiency and inclusion of high resolution topographic data. The best simulator for a particular application depends on the type of flood event and the data available for prediction, no simulator is optimal for all events. Most simulators if run twice with the same input will produce the same output, we say they are *deterministic*. Some inputs to flood inundation simulators are unknown and must be estimated, we call them *calibration inputs*. For a given calibration input value the simulator output is certain, but we are uncertain about how the output relates to reality. By comparing the output of the simulator, run at various values of the calibration inputs, to the observed data, we learn how the values of the calibration inputs relate to how well the simulator output represents the observed data, this is called *calibration*.

The calibration inputs are assumed to be stationary between the event we calibrate on and the event we want to predict. Even so, the values of the calibration

inputs which correspond to the simulation that is closest to the observed data will differ between events (Romanowicz and Beven, 2003). This is due in part to the observation error but largely because the simulator does not reproduce reality perfectly. The inability of the simulator to reproduce reality is called the *simulator inadequacy*. Therefore the calibration inputs, although seemingly physical quantities, must take account of unrepresented processes which may differ between events. For the calibration inputs to be stationary the processes they explicitly and implicitly account for must be stationary between events.

1.2 Current Practice in Flood Inundation Prediction

The 1991 Water Resources Act required the EA to provide flood maps showing estimates of 1 in 100 and 1 in 1000 year level floods which are now available on their website (Fleming, 2002). Figure 1.1 shows a flood map for the River Thames near Buscot. Flood maps raise public awareness by providing localised predictions of flood risk. However, where observations of large floods are unavailable, the flood maps are typically obtained from a single run of a deterministic simulator where the values of the calibration inputs driving the simulator are chosen by calibration on a more probable event (Bates *et al.*, 1998) or assigned on the basis of judgement. The optimum values of the calibration inputs will not in general be the same for the event we calibrate on and the event we wish to predict. Furthermore, the single run of the simulator is presented with no measure of how close this may be to reality.

Flood maps could be improved by taking the weighted average of the simulator outputs when different calibration inputs are used, where the weights are determined by calibration (Aronica *et al.*, 2002; Bates *et al.*, 2004). At the same time we can quantify the inadequacy of the simulator in predicting real flood events.

In the absence of formal statistical guidelines for calibration of flood inundation simulators using observations of flood extent, many non-probabilistic methods have been developed. Of these the generalised likelihood uncertainty estimation

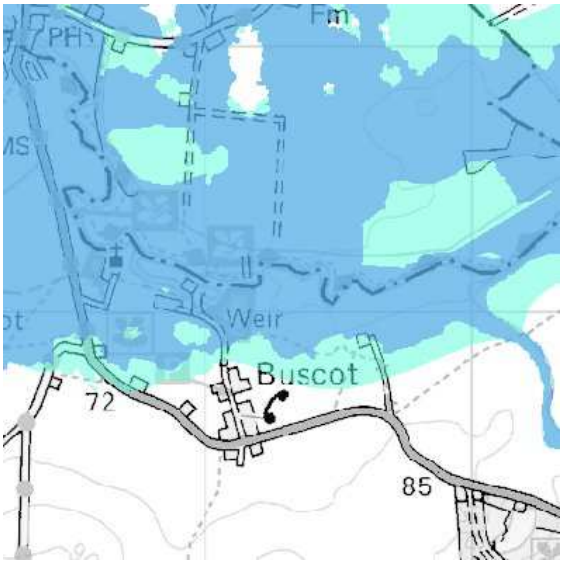


Figure 1.1: Flood map for the River Thames near Buscot. The predicted inundated area in the 1 in 100 and 1 in 1000 year flood events are dark blue and light blue respectively.

(GLUE) method introduced by Beven and Binley (1992) has attracted a lot of interest, and there is now a considerable body of work developing or using the GLUE methodology (Aronica *et al.*, 1998, 2002; Blazkova *et al.*, 2002). For the calibration event the simulator is run at many values of the calibration inputs, the resulting outputs are compared to the observed data by means of a *generalised likelihood* which is a function chosen subjectively to measure goodness-of-fit. For the prediction event the simulator is run at the same values of the calibration inputs. Routinely a sample from a uniform distribution with a user-defined range is used for the values of the calibration inputs at which the simulator is run. This is a deficiency of GLUE because this will not, in general, adequately capture the user's subjective knowledge about the values of the calibration inputs. The weighted mean, where the weights are the generalised likelihood values from calibration, is claimed to provide an estimate of the probability of flooding for each pixel. Although there are some similarities between GLUE and Bayesian statistics, the relaxation of the conditions a likelihood function must satisfy, the so-called generalised likelihood, means probabilistic inference is not possible using this method. This holds for any method that fails to satisfy the conditions of probability and

therefore we propose that it is necessary to develop a probabilistic approach to calibration. A probabilistic method is preferable because probability is coherent.

1.3 The Need for an Alternative Calibration Methodology

Flood maps can be helpful in many areas of flood management. For a particular region many flood defence measures may be possible, and flood maps depicting the impact of each defence measure will help in the identification of the optimum. Flood maps can be used to assess the flood risk associated with new construction for planning applications, for calculating insurance premiums for houses in high risk areas, and to make public warnings more localised. However, for a flood map to be useful it must provide reliable information. It is impossible to be certain about the flood extent in a future event, there are many uncertainties in the modelling process that should be quantified in our prediction. Presenting a single simulator output as a certain flood map could result in non-optimal choice of flood defence, incorrect planning permission decisions and insurance premiums, and poor flood warnings. The ramification will be a downturn in public opinion of flood maps, which will be difficult to rectify even when flood maps improve. It is therefore very important that a statistical method is developed to produce accurate maps of the probability of flooding.

Romanowicz and Beven (2003) have shown that, as a consequence of the errors that arise in predicting flood inundation and errors in observation, different values of the calibration inputs may lead to equally good results in the calibration event but give different results for the prediction event, in particular the optimum value is not the same. The EA flood maps are typically the result of a single run of a flood inundation simulator, so the uncertainty about the value of the calibration inputs and the simulator inadequacy is not quantified. The flood maps produced using the GLUE approach attempt to account for uncertainty about the value of the calibration inputs by taking the weighted average of a number of simulator runs. The probabilities presented are not true probabilities because the generalised

likelihood used is not a likelihood in a formal sense. Even if a proper likelihood is used, the flood maps show the probability the flood inundation simulator predicts an area is inundated, not the probability it is actually inundated in a future event (see Section 4.2.3). Claiming that the GLUE flood maps provide the probability of flooding for a future event is equivalent to saying there is no simulator inadequacy. We must account for simulator inadequacy, and this should be done within a formal statistical calibration framework.

Although river stage measurements at regular intervals along the channel may be used as calibration data (see Krzysztofowicz, 2002), we ultimately want to predict flood extent and would therefore need to work out how inadequacy in predicting river stage translates to inadequacy in predicting flood extent. Ideally we would have spatio-temporal data to calibrate flood inundation simulators with, because we want to predict the spatial evolution of flood extent over time. However, there is no such spatio-temporal data currently available, whereas spatial data of flood extent is becoming more readily available because of improvements in segmentation algorithms (Horritt, 1999; Horritt *et al.*, 2001).

The next step is to develop a statistical approach to calibration using observations of flood extent. A future development would be to combine spatial and time-series data calibration methods, so calibration can be performed on multiple sources of data.

1.4 Required Features of a New Calibration Methodology

An optimal framework for uncertainty handling in flood inundation modelling would be strictly probabilistic. The subjective choice of goodness-of-fit measures allowed in less rigorous approaches results in predictions of uncertainty that cannot be interpreted as probabilities. However, it may still be important to be able to incorporate subjective information such as expert beliefs and this can be done in a rigorous way using Bayesian statistics.

The calibration method should account for all sources of uncertainty, implicitly or explicitly, to provide reliable probability flood maps. Unfortunately the rarity of multiple observations of flood extent means we will not be able to validate our probabilistic predictions to test how reliable they are. By identifying the inadequacy of the simulator in representing the real flood extent the areas of the simulator in need of improvement may be identified. The effect of uncertainty about the calibration inputs and the level of simulator inadequacy can be investigated.

The greatest difficulty in developing a calibration method using observations of flood extent is defining an appropriate likelihood, i.e. a statistical model for the observed data given a simulator output. Indeed, this is the very reason that simple non-probabilistic methods have been so popular. The majority of this thesis will be concerned with identifying an appropriate likelihood model, and examining what features, including spatial dependence, blur and heterogeneity, it is necessary to represent.

1.5 Why is this the Next Logical Step?

Interest in environmental modelling has grown as a consequence of concern about the effects of climate change on the frequency of natural hazards. In recognition of the growing interest in environmental statistics the Royal Statistical Society formed the Environmental Statistics Study Group in 1996. Advances in computing mean many types of numerical simulator are now feasible, but there is not an equal advance in data available to validate the predictions, so it is essential that uncertainty in simulator output is quantified. The final report of the Institution of Civil Engineers' presidential commission on floods identified as vital improved procedures for quantifying the uncertainty in flood inundation simulators. The increasing acceptance of non-probabilistic techniques should be challenged by the statistics community because statistical approaches have not yet been exhausted. Although there may be tasks to which other non-probabilistic methods are better suited, it is the opinion of this thesis that a formal statistical framework for

calibration using observations of flood extent is possible, and that it is preferable.

Ideally the framework we develop for calibration will be applicable to a wide range of environmental applications. Flood inundation simulators are a good case to consider because they contain the key ingredients of environmental systems in a comparatively simple way. Simulators typically have very few calibration inputs, the fluid dynamics is well understood, and because the process of interest is on the surface, data can be obtained for calibration.

Until recently flood inundation modelling had been a data poor problem but improvements in remote sensing technology and algorithms for the extraction of flood extent mean data for the calibration of simulators is becoming more readily available (Bates, 2004). Flood extent is an important quantity for flood management and so it is sensible to construct a method for calibration on observations of flood extent. Also the development of a likelihood for the observed flood extent given the simulator output is mathematically interesting. Finally, advances in computing should be acknowledged as making the current research possible.

1.6 Thesis Overview

This chapter began by describing the hazard posed by flooding both globally and in England and Wales. In particular flood risk in England and Wales is predicted to increase over the next century and climate change has been identified as a powerful driver (Foresight, 2004). The non-probabilistic methods popular in the flood modelling research community were argued against on the grounds that the relaxation of probabilistic conditions may lead to arbitrary predictions. A Bayesian approach is favoured because it allows the incorporation of subjective expert beliefs and integration out of all uncertainties.

In Chapter 2 we present the hydrological background. We begin with hydraulic modelling and introduce the storage cell code LISFLOOD-FP, then the input and observed data appropriate to flood inundation modelling is described. At the end of the chapter we provide the details of the Buscot dataset that is used throughout the thesis.

In Chapter 3 we present the statistical background. We are going to develop a Bayesian framework so we give an introduction to Bayesian statistics, Markov chain Monte Carlo (MCMC), and directed acyclic graphs (DAGs).

In Chapter 4 we review the problem of handling uncertainty in computer codes, with emphasis on flood inundation simulators. We use the generic classification of uncertainties from Kennedy and O'Hagan (2001) to classify the uncertainties in hydraulic modelling. Then we describe some current methods for handling uncertainty. At the end of the chapter we give an example of GLUE applied to the Buscot dataset.

Chapters 2, 3 and 4 provide the background for the work in the rest of the thesis. If the reader feels they do not require this background information they may go directly to Chapter 5, where the original work begins.

Our Bayesian framework for handling uncertainty in flood inundation simulators is described in Chapter 5. We illustrate our framework using a DAG. To demonstrate the framework we assume a binary channel (BC) model for the likelihood. As will become clear this simple model is unrealistic but allows the probability of flooding in a future event to be calculated analytically. In applying the framework to the Buscot dataset we observe the inadequacy of the BC model as a likelihood model. In Chapters 6, 7 and 8 we investigate properties of alternative likelihood models.

In Chapter 6 we show that the Ising model is the only model for binary images with interactions between nearest neighbours and homogeneous parameters. Then we extend the Ising model to regression on a binary image (the simulator output). The normalising constant is notoriously difficult to calculate, so we review importance sampling, bridge sampling and path sampling methods for approximation. Then we look at novel applications of path sampling to paths between model parameterisations and between different binary images. We also extend the idea of path sampling to area sampling by integrating over areas rather than paths. This extension suggests an additive approximation if covariance is ignored. We consider a number of approximations in order to estimate the normalising constant more

efficiently. Unfortunately, we are unable to identify a method which is sufficiently accurate and efficient, so the Ising model cannot be used as the likelihood.

In Chapter 7 we extend the BC model to represent heterogeneity and call this the heterogeneous binary channel (HBC) model. By extending the BC model we aim to develop a practical likelihood model. We show how calibration and calibrated prediction can be done with this model and propose an MCMC algorithm to obtain an approximate sample from the posterior. We explore the effects of forcing regression to be positive by constructing a similar model for which this is the case, the positive heterogeneous binary channel (PHBC) model. The properties of both models are shown using a one-dimensional toy dataset. We then apply the HBC model to the Buscot dataset. For many values of the HBC model parameters the Markov chain convergence is very slow, we show how within-model sampling (WMS) might be used to improve this. In the HBC model there is no explicit link between the parameters corresponding to negative and positive simulation values. In Chapter 8 we consider an alternative model for which this link exists.

Chapter 8 begins with a description of conditional autoregressive (CAR) models. We extend the hidden CAR (HCAR) model of Weir and Pettitt (1999) to regression on a binary image and describe how to calibrate and make calibrated predictions using this model as the likelihood. We review block-circulant matrices and their relevance in making the method practical, and an MCMC algorithm is proposed. The method is applied to the Buscot dataset and then we suggest various ways in which the mixing of the Markov chain can be improved. We find the limit of the HCAR model as the parameters approach a particular boundary of the parameter space, we call this the hidden intrinsic autoregressive (HIAR) model. We show that calibration is possible using this model but not calibrated prediction. We look at two extensions of the HCAR model. First, the heterogeneous HCAR (HHCAR) model, which represents heterogeneity, and second the continuous HCAR (CHCAR) model, which uses continuous simulation values.

Finally, in Chapter 9 we present conclusions about our Bayesian framework and the likelihood models we developed, and suggest future work in these areas.

Chapter 2

Hydrological Background

We begin this chapter with a review of hydraulic modelling. Then we describe the data required to run a flood inundation simulator and the data required to calibrate one. Finally we give an overview of the Buscot dataset that will be used many times throughout the thesis.

2.1 Hydraulic Modelling

Starting with a summary of flow processes we describe how flow may be represented using the equations of motion with suitable boundary conditions. We classify numerical simulators according to the dimensionality of the flow processes represented, and justify the use of LISFLOOD-FP for the examples in this thesis.

2.1.1 Flow Processes in Floods

The regular flow of a river defines the channel that is carved out of the landscape. When extreme flows occur the river level may exceed that of the river bank causing the river to spill onto the floodplain (Knight and Shiono, 1996). A flood is defined as a large, low amplitude wave flowing over complex geometry (Bates *et al.*, 2005). The flow conveyance in flood may be very different from normal flow as new pathways become available (Bates and De Roo, 2000). The size of the flood wave is important in the selection of an appropriate simulator as in larger river basins the length may exceed 10^3 km, have an amplitude of around 10 m and take months to travel through the system (Bates *et al.*, 2005). As the wave propagates downstream

it slows and flattens out (attenuates) due to friction.

We now present a brief overview of flow processes in compound channels. During normal flow, shear layers form between the main flow and slower moving regions called *dead zones* (Hankin *et al.*, 2001). The *primary velocity field* is the velocity profile on cross-sections perpendicular to the main flow. Turbulence and interactions with dead zones cause circulatory motions in the primary velocity field called *secondary circulations*. Turbulent eddies are generally created at the scale of the flow geometry, each eddy breaks down into a number of smaller eddies and in doing so dissipates some kinetic energy as heat. This process is repeated until kinetic energy is completely dissipated by the smallest eddies at the Kolmogorov length scale which may be 10^{-2} mm.

During a flood, vortices with vertically aligned axes transfer momentum between the slow floodplain flow and fast channel flow (Knight and Shiono, 1996), and for meandering channels, water that spills onto the floodplain may travel over the floodplain and only return to the channel further down the reach (Sellin and Willets, 1996). The impact of these processes is greatest when the flow on the floodplain is shallow, as the depth increases the channel and topography begin to act as a single channel unit (Bates *et al.*, 2005; Knight and Shiono, 1996). The floodplain flow away from the channel is characterised by relatively rapid horizontal fluctuation, it is imperative that this behaviour is captured in any flood inundation simulator. Interaction with vegetation becomes more important for overbank flow particularly when the floodplain is used as an additional means of conveyance.

Processes resulting from interaction with the catchment are generally ignored but occasionally it may be necessary to account for some of them, including evapotranspiration losses, direct precipitation and bank-storage effects (Bates *et al.*, 2005).

All of the processes described in this section can be represented using the equations of motion, called the *Navier-Stokes equations*.

2.1.2 The Equations of Motion

Fluids are discrete, they are composed of a number of molecules such that at any point they are either there or not (Paterson, 1997). For example the density of a fluid takes a large positive value where a molecule is but is elsewhere zero, it is a discontinuous function. It is very difficult to work with discontinuous functions, for which even differentiation is not possible. Therefore we are forced to make our first assumption, that the fluid is a continuum. This widely used assumption is a good approximation in most cases, however an example where it is not applicable is shock waves, where there are discontinuities in the fluid.

The equations of motion are derived from the laws of conservation of mass and momentum that state that these quantities cannot be created or destroyed. We will show the equations of motion by applying these laws to a small fluid parcel with volume V and surface S.

By the law of conservation of mass the rate of increase of mass in the fluid parcel V must be equal to the rate mass enters V from the outside (Julien, 2002).

$$\int_V \frac{\partial \rho}{\partial t} \, \mathrm{d}V = -\int_S \rho \boldsymbol{v} \cdot \boldsymbol{n} \, \mathrm{d}S \tag{2.1}$$

where ρ is the density, $\boldsymbol{v}$ is the fluid velocity and $\boldsymbol{n}$ is the outward pointing unit normal of the surface S. The surface integral can be replaced by a volume integral using the divergence theorem, on rearranging we find

$$\int_V \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) \; \mathrm{d}V = 0.$$

Furthermore V is arbitrary, so by the Dubois-Reymond lemma (Paterson, 1997) the integrand must equate to zero,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0, \tag{2.2}$$

which is the *continuity equation*. The differential of the density with respect to time is only possible because of the continuum assumption. The continuity equation provides a link between the velocity components.

By the law of conservation of momentum the rate of increase in momentum in a fluid parcel V is the sum of three components: the rate of momentum inflow

through S, the forces acting inside V, and the forces acting on S (Paterson, 1997). For the ith component

$$\frac{d}{dt}\int_V \rho v_i \,\mathrm{d}V = -\int_S \rho v_i \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}S + \int_V F_i\,\mathrm{d}V + \int_S \sigma_{i1}n_1 + \sigma_{i2}n_2 + \sigma_{i3}n_3\,\mathrm{d}S$$

where F_i is the ith component of the body force per unit volume and σ_{ij} is the ith component of the force per unit area for an area with normal in the jth direction.

The tangential stresses are shear stresses so we write $\sigma_{ij} = \tau_{ij}$ for $i \neq j$. Also the normal stresses are the sum of pressure effects and deformation shear stresses $\sigma_{ii} = -p + \tau_{ii}$ (Julien, 2002).

Converting the surface integrals to volume integrals using the divergence theorem and equating the integrand to zero by the Dubois-Reymond lemma, we find

$$\frac{\partial}{\partial t}(\rho v_i) + \sum_{j=1}^{3}\frac{\partial}{\partial x_j}(\rho v_i v_j) = F_i - \frac{\partial p}{\partial x_i} + \sum_{j=1}^{3}\frac{\partial \tau_{ij}}{\partial x_j}.$$

This is the *equation of motion*. The left hand side can be rearranged using the continuity equation (2.2) to show that it is just the density multiplied by the acceleration of a particle following the fluid (Paterson, 1997),

$$\rho\frac{Dv_i}{Dt} = \rho\frac{\partial v_i}{\partial t} + \rho\boldsymbol{v}\cdot\nabla v_i.$$

2.1.3 Conditions for Flood Modelling

The equations of motion can be applied to a wide range of applications, from the ripples in a glass of milk to ocean waves. To make use of the equations in a particular application boundary conditions must be specified.

For most liquids it is appropriate to assume that the density does not change following the fluid, $D\rho/Dt = 0$, we say the fluid is *incompressible*. In this case the continuity equation (2.2) becomes

$$\nabla\cdot\boldsymbol{v} = 0. \tag{2.3}$$

The shearing stresses τ_{ij} are the product of the rate at which the layers are sheared, $\partial v_i/\partial x_j + \partial v_j/\partial x_i$, and the strength of the bonds between the layers, μ, called the *coefficient of viscosity*. The equation of motion becomes

$$\rho\frac{D\boldsymbol{v}}{Dt} = \boldsymbol{F} - \nabla p + \mu\nabla^2\boldsymbol{v} \tag{2.4}$$

which is the *Navier-Stokes equation* for an incompressible fluid.

The bed and free water surface provide two boundary conditions for river basin applications. The bed is assumed to be solid so the normal velocity is zero and we also assume that water molecules next to the bed "stick" to this surface, the so-called *no-slip condition*, so horizontal velocities are zero. Water particles are unable to cross the free water surface so the normal velocity here is also zero.

For shallow flows on floodplains with high relative roughness, friction is likely to be the dominant component of $\boldsymbol{F}$. For large scale floodplain flows other effects may need to be accounted for: Coriolis effects may be included in the force per volume vector $\boldsymbol{F}$; we may be unable to assume that density is constant over horizontal translation; we may need to account for wind shear stresses at the water surface; and atmospheric pressure may vary over the water surface (Lane, 1998).

We have now presented the equations of motion and the boundary conditions required for flood modelling. However, to solve these equations numerically is computationally infeasible, it would require a discretization of the flow with cell spacing significantly shorter than the length of the smallest eddies (typically about 10^{-2} mm) and a time step that is shorter than the lifespan of these smallest eddies. In the next section we will discuss the types of numerical simulator that have been developed and the various assumptions that they encode.

2.1.4 Reynolds Averaging

In most cases it is unnecessary to know instantaneous velocities, and we can simply model their effect on the mean flow. *Reynolds averaging* splits the instantaneous velocity, v_i, into time averaged mean, $\overline{v}_i$, and fluctuation, v_i', such that the time average of v_i' is zero. On substituting $v_i = \overline{v}_i + v_i'$ into the equations of motion and time averaging the continuity equation (2.3) becomes

$$\nabla \cdot \overline{\boldsymbol{v}} = 0. \tag{2.5}$$

and the ith component of the Navier-Stokes equation (2.4) becomes

$$\frac{\partial \overline{v}_i}{\partial t} + \sum_{j=1}^{3} \overline{v}_j \frac{\partial \overline{v}_i}{\partial x_j} = \frac{F_i}{\rho} - \frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \frac{1}{\rho}\sum_{j=1}^{3} \frac{\partial}{\partial x_j}\left(\mu \frac{\partial \overline{v}_i}{\partial x_j} - \rho \overline{v_i' v_j'}\right) \tag{2.6}$$

called the *Reynolds-averaged Navier-Stokes equation*. The *Reynolds shear stresses* $\rho\overline{v_i'v_j'}$ measure the retardation on the mean flow due to turbulence. There are now more unknowns than equations, to resolve this dilemma we formulate models for the Reynolds shear stresses. Many of these models rely on the Boussinesq assumption

$$\rho\overline{v_i'v_j'} = \nu_{ij}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)$$

where the *eddy viscosity coefficient*, ν_{ij}, is typically much larger that the laminar counterpart μ (Rodi, 1980). The eddy viscosity must itself be modelled: zero-equation models implicitly assume turbulence is dissipated where it is generated; one-equation models account for transport of the turbulent velocity scale; and two-equation models in addition account for transport of the turbulent length scale (Rodi, 1980). Alternatively, the Reynolds shear stresses can be calculated directly from their transport equations, which does not rely on the Boussinesq assumption, but as yet this has only been applied to steady flows through channels of simple geometry.

2.1.5 Numerical Simulators

Reality cannot be perfectly revealed by a finite number of processes using a finite sample of data, rather "all models are wrong, but some are useful" Box (1976). Because it is impossible to apply the Navier-Stokes equations directly, it is the task of the modeller to ascertain which processes must be included and which may be neglected. A simulator is judged by its ability to fulfil the job for which it was designed. Adopting the simulator classification from Pender (2006) we now present a summary of the various flood inundation simulators.

Three-Dimensional Simulators (3D)

A three-dimensional solution to the Navier-Stokes (or Reynolds averaged Navier-Stokes) equations is required when three-dimensional processes dominate the behaviour of the feature we want to model. Examples include sediment transport, flow-vegetation interaction and the interaction at the channel-floodplain interface (Bates *et al.*, 2005). The sophistication of the solution depends on the grid scale

Class	Description	Application	Examples	Inputs	Outputs	Computation Time
0D	Direct interpolation of gauged elevations	Broad-scale and benchmark	ArcGIS, Delta mapper	DEM, upstream and downstream water levels	Flood extent and depths from intersecting planar water surface with DEM	Seconds
1D	Solution of the one-dimensional Saint Venant equations	Reaches of the order of 10s or 100s of km depending on catchment size	HEC-RAS, Infoworks RS (ISIS), MIKE 11	Surveyed cross-sections, upstream discharge hydrographs and downstream stage hydrographs	Water depth and average velocity at each cross-section, inundation extent by intersecting predicted water depths with DEM, and downstream outflow hydrograph	Minutes
1D+	1D plus a storage cell approach to the simulation of floodplain flow	Reaches of the order of 10s to 100s of km depending on catchment size, also broad-scale if used with sparse cross-section data	HEC-RAS, Infoworks RS (ISIS), MIKE 11	As for 1D models	As for 1D models	Minutes to hours
2D-	2D minus the law of conservation of momentum for the floodplain flow	Broad-scale modelling or urban inundation depending on cell dimensions	LISFLOOD-FP, JFLOW	DEM, upstream discharge hydrographs and downstream stage hydrographs	Inundation extent, water depths and downstream outflow hydrograph	Hours

Table 2.1: Classification of flood inundation simulators. Based on Table 3 from Pender (2006). Zero-dimensional to two-dimensional minus simulators.

Class	Description	Application	Examples	Inputs	Outputs	Computation Time
2D	Solution of the two-dimensional shallow wave equations	Reaches of the order of 10s km, also broad-scale if applied with very course grids	TUFLOW, MIKE 21 and TELEMAC	DEM, upstream discharge hydrographs and downstream stage hydrographs	Inundation extent, water depths, depth-averaged velocities at each computational node, and downstream outflow hydrograph	Hours to days
2D+	2D plus a solution for vertical velocities using continuity only	Coastal modelling applications where 3D velocity profiles are important, also applied to reach scale river modelling problems in research	TELEMAC 3D	DEM, upstream discharge hydrographs, inlet velocity distribution, and downstream stage hydrographs	Inundation extent, water depths, velocities for each computational cell, and downstream outflow hydrograph	Days
3D	Solution of the three-dimensional Reynolds averaged Navier Stokes equations	Local predictions of three-dimensional velocity fields in main channels and floodplains	CFX, FLUENT and PHEONIX	DEM, upstream discharge hydrographs, inlet velocity, turbulent kinetic energy distribution, and downstream stage hydrographs	Inundation extent, water depths, velocities and turbulent kinetic energy for each computational cell, and downstream outflow hydrograph	Days

Table 2.2: Classification of flood inundation simulators. Based on Table 3 from Pender (2006). Two-dimensional to three-dimensional simulators.

and the turbulence closure used. This turbulence closure ranges from *large eddy simulation* (LES) to depth-averaged RANS equations adopting the Boussinesq assumption with a zero-equation model for the eddy viscosity coefficient. All solutions are implemented using a numerical tool such as finite volumes, finite differences or finite elements. The solution sophistication and domain size and discretization determine the computational cost; a modeller must balance these factors with their computational budget.

Stoesser *et al.* (2003) successfully applied a three-dimensional solution of the RANS equations, adopting the Boussinesq assumption with a two-equation model for the eddy viscosity coefficient, to a tangible compound channel flow. However, more complex solutions such as LES have only been applied to channels with regular geometry (Thomas and Williams, 1995).

Examples of three-dimensional codes are CFX, FLUENT and PHEONIX (Pender, 2006).

Two-Dimensional Plus Simulators (2D+)

A *shallow water application* is one in which the horizontal scale is at least 10 times larger than the vertical scale, in this case we may assume that the pressure gradient in the vertical direction is balanced by gravity, we say we are in *hydrostatic equilibrium*. The vertical component of the Navier-Stokes equations (2.4) is replaced by the *hydrostatic distribution*

$$\frac{\partial p}{\partial z} = -\rho g.$$

If the density is constant then the hydrostatic distribution implies the pressure is linear in z, so the horizontal pressure gradients $\partial p/\partial x$ and $\partial p/\partial y$, that appear in the first and second components of the Navier-Stokes equations (2.4), are independent of z. Therefore horizontal flow is independent of height and (by incompressibility) the vertical velocity is linear in depth.

Although strictly three-dimensional, codes based on the Navier-Stokes equations with hydrostatic pressure distribution are referred to as 2D+ codes by Pender (2006) because the vertical velocities are found from continuity only.

One example of a two-dimensional plus code is TELEMAC 3D.

Two-Dimensional Simulators (2D)

Depending on the feature we want to reproduce it may be unnecessary to employ three-dimensional codes which can be computationally expensive. In shallow water flows the horizontal variation in velocities is much greater than the vertical variation, suggesting that depth averaged velocities may be adequate.

The two most common two-dimensional approaches are derived by integrating the RANS equations (2.5) and (2.6) over depth: the *Saint Venant equations* assume a hydrostatic pressure distribution and the *Boussinesq equations* do not (Hervouet and Van Haren, 1996). The Saint Venant continuity and ith momentum equations are

$$\frac{\partial h}{\partial t} + \overline{\boldsymbol{v}}_d \cdot \nabla h + h \nabla \cdot \overline{\boldsymbol{v}}_d = 0$$

and

$$\frac{\partial \overline{\boldsymbol{v}}_d}{\partial t} + \overline{\boldsymbol{v}}_d \cdot \nabla \overline{v}_{d,i} = -g \frac{\partial h}{\partial x_i} - g \frac{\partial z_b}{\partial x_i} + \nabla \cdot (\nu \nabla \overline{v}_{d,i}) + S_i$$

where $\overline{\boldsymbol{v}}_d = (\overline{v}_{d,1}, \overline{v}_{d,2})$ are the depth averaged velocities, z_b is the bed elevation, h is the water depth, and S_1 and S_2 are the source terms.

Codes based on the Saint Venant equations and Boussinesq equations are the two-dimensional equivalents of two-dimensional plus and three-dimensional codes respectively. The equations are solved numerically using a discretization of the flow and employing a turbulence scheme for the eddy viscosity coefficient ν (see Bates *et al.*, 2005, for details).

Examples of two-dimensional codes are TUFLOW, MIKE 21, TELEMAC and DIVAST (Pender, 2006).

One-Dimensional Simulators (1D)

When we are only interested in the attenuation and translation of the flood wave the dominant variation is in the streamwise direction, so cross-stream and vertical variations may be ignored. This assumption is valid for floodplains that are less than three times wider than the main channel (Pender, 2006).

The *one-dimensional Saint Venant equations* are derived by applying the laws

of conservation of mass and momentum to two cross-sections δx apart,

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = q \tag{2.7}$$

and

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + gA\left(\frac{\partial h}{\partial x} + S_f - S_b\right) = 0 \tag{2.8}$$

where A is the cross-section area, Q is the flow discharge, S_f and S_b are the friction slope and bed slope respectively, and q is the lateral flow per unit length. To solve these equations the river is represented as a number of irregularly spaced cross-sections. All flow is in the streamwise direction in one-dimensional simulators, but by splitting the cross-section into a series of panels and then modelling the shear between these panels, some cross-sectional conveyance can be accounted for (Knight and Shiono, 1996).

Further simplifications of the Saint Venant equations are possible by ignoring certain terms in the momentum equation (2.8). The *diffusive wave model* assumes that inertia can be neglected so the momentum equation becomes $\frac{\partial h}{\partial x} + S_f - S_b = 0$, but there is little computational gain in not computing the inertia terms. In the *kinematic wave model* the momentum equation becomes $S_f - S_b = 0$, which equates momentum of an unsteady flow to that of a steady uniform flow, with the consequence that the flood wave is not attenuated in channels of uniform geometry and disturbances cannot affect upstream flow (Bates, 2005).

Examples of one-dimensional codes are MIKE 11, HEC-RAS and Infoworks RS (ISIS) (Pender, 2006).

One-Dimensional Plus and Two-Dimensional Minus Simulators (1D+/2D-)

One-dimensional simulators cannot represent lateral flow or account for variations in topography between the subjectively chosen cross-sections. Although two- and three-dimensional simulators overcome these issues they do so at great computational expense (Bates and De Roo, 2000). For this reason hybrid simulators have been developed which treat the channel flow as one-dimensional and the floodplain flow as two-dimensional.

The channel flow is normally represented using the one-dimensional Saint Venant equations and the floodplain flow using a two-dimensional storage cell method (Cunge *et al.*, 1980). Flows between the storage cells and the channel or other storage cells are defined by weir flow based discharge relationships (Pender, 2006).

In *one-dimensional plus* (1D+) simulators the floodplain is split into a number of user defined polygonal storage cells with horizontal water levels. Examples are MIKE 11, HEC-RAS and Infoworks (RS) (Pender, 2006).

To avoid the need to subjectively define storage cells and to make use of high resolution topographic data (see Section 2.2.3) *two-dimensional minus* simulators have been developed in which the floodplain is discretized as a grid of square cells. An example is LISFLOOD-FP (Bates and De Roo, 2000).

In LISFLOOD-FP the channel flow is represented using a kinematic or diffusive wave approximation to the Saint Venant equations, see Equations (2.7) and (2.8), with friction slope

$$S_f = \frac{n_c^2 P^{4/3} Q^2}{A^{10/3}}$$

where n_c is Manning's friction coefficient for the channel and P is the wetted perimeter. If the channel is assumed to be wide and shallow the wetted perimeter can be approximated by the channel width.

The floodplain flows are described in terms of continuity and momentum equations, discretized over a grid of square cells (Bates *et al.*, 2004). Let $N_{ij} = \{(i-1,j),(i+1,j),(i,j-1),(i,j+1)\}$ be the set of neighbours of cell (i,j) and suppose $(k,l) \in N_{ij}$, then the flow from (i,j) to (k,l) is defined to be proportional to the difference between the free surface heights

$$Q_{(i,j),(k,l)} = h_f^{5/3} d^{1/2} n_f^{-1} (h_{i,j} - h_{k,l}) / |h_{i,j} - h_{k,l}|^{1/2} \tag{2.9}$$

where $h_{i,j}$ is the height of the free water surface, the flow depth h_f is the difference between the highest free water surface in the two cells and the highest bed elevation, n_f is Manning's friction coefficient for the floodplain, d is the cell dimension, and $Q_{(i,j),(k,l)}$ describes the volumetric flow rate between cells. The water depths

are calculated using the continuity equation

$$\frac{\partial h_{i,j}}{\partial t} = -\frac{1}{d^2} \sum_{(k,l)\in N_{ij}} Q_{(i,j),(k,l)} \tag{2.10}$$

The flows between floodplain and channel cells are also calculated using equation (2.9), then the channel is updated by equating this flow with q in equation (2.7), and the floodplain is updated using equation (2.10) (Bates *et al.*, 2005). An important difference between two-dimensional minus and two-dimensional simulators is that in two-dimensional minus simulators momentum transfer between the channel and the floodplain is not represented.

There are two approaches to discretizing the channel (Bates *et al.*, 2004). If the channel width is approximately equal to the grid cell size then we can define a series of cells to contain the channel and a bankfull depth for each of these cells. When bankfull depth is exceeded the flow between the channel and floodplain is calculated as described above. If the channel width is small compared with the grid cell size this scheme will neglect the potential for storage adjacent to the channel but within the channel cell. In this case Horritt and Bates (2001) propose an alternative scheme called the *near channel floodplain storage* (NCFS) model. The channel no longer occupies any floodplain cells but now specifies an additional pathway between the cells it passes through. Two water depths are associated with the floodplain pixels the channel passes through. Flow between the channel and floodplain pixels lying on the channel is handled using a Manning type flow equation.

To solve the equations numerically a time step is specified. If the time step is too long the solution oscillates, so to prevent this a flow limiter is imposed

$$Q^{\star}_{(i,j),(k,l)} = \begin{cases} Q_{(i,j),(k,l)} & \text{if } |Q_{(i,j),(k,l)}| < |h_{i,j} - h_{k,l}|d^2/(4\Delta t) \\ (h_{i,j} - h_{k,l})d^2/(4\Delta t) & \text{otherwise.} \end{cases}$$

However, when it is used the Manning's friction coefficient n is ignored and the grid size and time step become important. Hunter *et al.* (2005) resolve this issue by calculating the optimum time step at each iteration, but throughout this thesis this adaptive time step is not used. Werner and Lambert (in press) have shown

that LISFLOOD-FP underpredicts inundation extent, flood depth, wave volume and travel time when used without first calibrating. However, after calibration LISFLOOD-FP has been shown to be as good as or better than two-dimensional simulators (Horritt and Bates, 2002).

Zero-Dimensional Simulators (0D)

When the flood wave is long compared to the reach, so within the reach it is relatively flat, we may fit a plane to gauged water surface elevations, and compare this to the topography to obtain flood depths (Werner, 2001). The success of this method depends on the number of gauges and the accuracy of the topography. When many gauges are present we can use a series of planes to improve the surface approximation. Although this method does not conserve mass and can show hydraulically unconnected regions as flooded, its simplicity means it is often used as a benchmark for other simulators. Examples are ArcGIS and Delta mapper (Pender, 2006).

2.1.6 Simulator Choice

The flood inundation simulator used to demonstrate the calibration methodology developed in this thesis should: be capable of representing floodplain flow because we want to calibrate on an observation of flood extent; be computationally inexpensive because we want to generate a large ensemble of results; have few calibration parameters to minimise the size of the ensemble required; be convenient because the methodology we develop will be generic so the simulator adopted is not critical.

Numerical solutions using raster grids are computationally cheaper than those using unstructured meshes, but a resolution capable of representing the channel processes effectively will be too fine on the floodplain. The natural progression is to decouple the channel and floodplain flow, as in one-dimensional plus and two-dimensional minus simulators. The representation of the floodplain is better in two-dimensional minus simulators and they are the simplest simulators capable of dynamic flooding. We will use LISFLOOD-FP as it is well established and has

been developed in Bristol, so source code access and expertise are available. Furthermore, in prediction LISFLOOD-FP fairs poorly compared to other simulators unless the unknown parameters are first calibrated. It is therefore essential that LISFLOOD-FP be used together with a formal calibration methodology.

2.2 Data Requirements for Prediction

To run the simulator we need to specify the boundary conditions, initial conditions, topography, and friction.

2.2.1 Boundary Condition Data

The boundary condition data consists of values for each simulator dependent variable on the boundary at each time step (Bates *et al.*, 2005).

In LISFLOOD-FP boundary conditions must be defined for the one-dimensional channel flow and the storage cell floodplain flow. In the channel, if a kinematic wave is used the upstream inflow, Q_{in}, must be known for all time, and this is usually taken from river gauging station measurements or set constant for steady flow. If a diffusive wave is used, in addition we need the downstream outflow, Q_{out}. Lateral flow q (see Equation (2.7)) would typically be set to zero in a one-dimensional code, but represents the effect of channel-floodplain interaction on the channel flow in LISFLOOD-FP (see Section 2.1.5).

On the floodplain the values of the free water height or the flow discharge on the domain boundary must be specified for each time step. Normally the zero flux condition is assumed so $h_{i,j} = 0.0$ and $Q_{(i,j),(k,l)} = 0.0$ for all cells (i, j) on the domain boundary. However, this implies that water can only enter and leave the domain within the channel, and can result in an unrealistic backward flooding as water arrives in the downstream region quicker than the channel can transport it out of the domain. This is simply resolved by fixing the free water height to some nonzero value for the floodplain cells on the boundary around where the channel exits the domain. Point sources are allowed by specifying the free water height or the flow discharge over time for non-boundary cells.

2.2.2 Initial Condition Data

The initial condition data consists of values for every simulator dependent variable at time $t = 0$ (Bates *et al.*, 2005). Normally these values are unknown, so for uniform flows arbitrary initial values are specified and the simulator is run until steady state is obtained, for unsteady flows the corresponding steady state results are used as the initial values (Bates, 2005).

In LISFLOOD-FP initial conditions must be defined for the one-dimensional channel flow and the storage cell floodplain flow. For steady flows we assume the initial floodplain flow depth is 0 m, the channel flow depth is 1.75 m and the channel discharge is 0 m^3s^{-1}.

2.2.3 Topography

For one-dimensional simulators it is ideal for the river cross-sections to be measured by field survey, because this is the most accurate form of topography with the norm of the error being only a few millimetres (Bates, 2005). However, field surveys are very expensive and only provide a series of vertical planar measurements that must be interpolated for use with higher-dimensional codes. The interpolated topography neglects variation between cross-sections, and consequently will be different if the cross-sections are measured at different positions along the channel.

For shallow water flows it is essential that the topography is represented accurately over the floodplain to facilitate the modelling of the rapid fluctuation of the flood boundary. Such large scale maps are provided by satellite and airborne sensors and are becoming readily available (Bates *et al.*, 2005). Airborne sensors are far more accurate and of these the light detection and ranging (LiDAR) technique, which measures the distance between the aircraft and the ground using pulses of laser energy, has proved particularly popular. The Environment Agency in the UK is using a LiDAR system to capture the topography of river basins in England and Wales to aid the assessment of flood risk. At an operating altitude of about 800 metres the width of the scan is about 600 metres and measurements are made at about 2 metre intervals giving very high resolution topographic data.

Vegetation is partially penetrated by the laser pulse and so the last signal received by the aircraft will hopefully give the ground level. Unfortunately LiDAR does not penetrate the water surface so bathymetry must be obtained by field survey.

In LISFLOOD-FP the channel is discretized as a series of rectangular cross-sections where the width and bankfull depth are taken from the bathymetry, and the floodplain is discretized as a raster grid.

2.2.4 Friction

The unknown parameters of hydraulic simulators are the *lumped friction coefficient* (e.g. Manning's n) and, if turbulence is modelled using the Boussinesq assumption, the eddy viscosity ν (Bates *et al.*, 2005). The eddy viscosity only appears in two- and three-dimensional models and is rarely treated as an unknown in practice, usually being modelled using transport equations (see Section 2.1.4).

The lumped friction coefficient combines: skin friction arising from interaction with the channel bed; form friction caused by meanders and changes to the cross-sectional area; vegetative resistance that dominates floodplain flow; shear between channel and floodplain flow; turbulence that is not explicitly represented; and acceleration and deceleration. The resistances are lumped together because although some can be modelled directly (e.g. skin friction and vegetative resistance) most cannot. The lumped friction coefficient is normally taken to be some standard resistance coefficient such as *Manning's coefficient of roughness*, n, derived from uniform flow theory.

The processes represented in the lumped friction coefficient n depend on the simulator dimensionality and discretization. For example the flow geometry representation is better in two-dimensional simulators than one-dimensional simulators; consequently the form friction contribution to the lumped friction coefficient is closer to the true physical form friction in two-dimensional simulators, whereas in one-dimensional simulators the poor geometry representation must be compensated for by the form friction contribution. As the simulator dimensionality decreases or the discretization becomes coarser the lumped friction coefficient must account for more unrepresented processes, and the simulator results become more

sensitive to this value. Although the lumped friction coefficient is often referred to as Manning's n, because the value n takes is dependent on the simulator dimensionality and discretization, it is nonsensical to compare n between different simulators, in particular to Manning's empirical formula for open channel flow. We prefer to think of this as a convention adopted (maybe erroneously) for lucidity.

In LISFLOOD-FP Manning's channel friction n_c can be different for every cross-section and Manning's floodplain friction n_f can be different for every floodplain pixel, although in general they are both fixed.

2.3 Data Requirements for Calibration

To run hydraulic simulators we must specify values for the unknown parameters (friction coefficients and infrequently the eddy viscosity). As the friction coefficient combines many sources of hydraulic resistance and compensates for unrepresented processes and discretization (see Section 2.2.4), the value of the parameter is meaningless outside the context of the present simulator. In particular, Manning's n is used in various simulators but the physical value of Manning's n will not, in general, be the value which results in the best simulator output, the so-called *true value*. Furthermore, the true Manning's n value will be different for different simulators. The *best simulator output* is that which is closest to the truth. To learn about the true values of the unknown parameters we compare the simulator output, for various values of the unknown parameters, to an observation of the truth. Calibration is discussed fully in Chapter 4.

Time series of water depth and discharge from river gauging stations have been used to test wave routing in hydraulic simulators (Cunge *et al.*, 1980; Horritt and Bates, 2002). In the UK the distribution of national gauging stations relates to flood risk but not directly to hydraulic simulator calibration (Bates, 2005). With a typical separation of 10–40 km there will be few stations within a simulator domain. Measurements are made at least hourly, the stage is accurate but the discharge has a 5% error for in-channel flows and a 20% error for out-of-channel flows. Such data cannot directly test the ability of a hydraulic simulator to reproduce flood

depths over the whole domain, indeed it has been shown that different friction values can yield the same time series but different flood extents (Romanowicz and Beven, 2003; Romanowicz *et al.*, 1996).

Point scale data includes point measurements of velocity and water level measured during the flood, and maximum water levels identified from high water marks or deposits of material at maximum inundation (Bates *et al.*, 2005). However, Lane *et al.* (1999) warn against using point scale data for calibration because they are unreconcilable with simulator variables which are normally averaged over space and time.

Instantaneous observations of flood extent can be obtained by ground survey but are becoming more readily available through the use of airborne and satellite imaging (Bates *et al.*, 2005). Such images provide data for the whole spatial domain, and in shallow floodplains the flood extent is very sensitive to small changes in water depth so hydraulic simulators must accurately reproduce flood depths to reproduce the flood extent.

Synthetic aperture radar (SAR) uses the Doppler effect to simulate a larger aperture radar, microwaves penetrate the cloud cover and, by using different frequencies, give a picture of the canopy (Horritt, 1999). Flood extent can be extracted from the resulting noisy image using an *active contour region* or *snake* (Horritt, 1999). A snake is a closed curve with an energy functional dependent on the snake geometry and the properties of the image. The functional is defined such that the energy is minimised when the snake lies on the flood boundary, so identifying the flood extent becomes an energy minimisation problem. Figure 2.1 shows a SAR image of a 3 km by 3 km subregion of the River Thames between Buscot and Standlake overlaid with shorelines derived using the snake algorithm and from aerial photographs (Horritt *et al.*, 2001). The error in shoreline delineation from aerial photographs is less than 20 metres, so inconsistencies between the shorelines are due to errors in the SAR or snake algorithm. Top-right flooded vegetation appears dry in the SAR image, and dry islands appear wet because sparsely vegetated areas have similar backscatter to water, otherwise the two shorelines agree reasonably

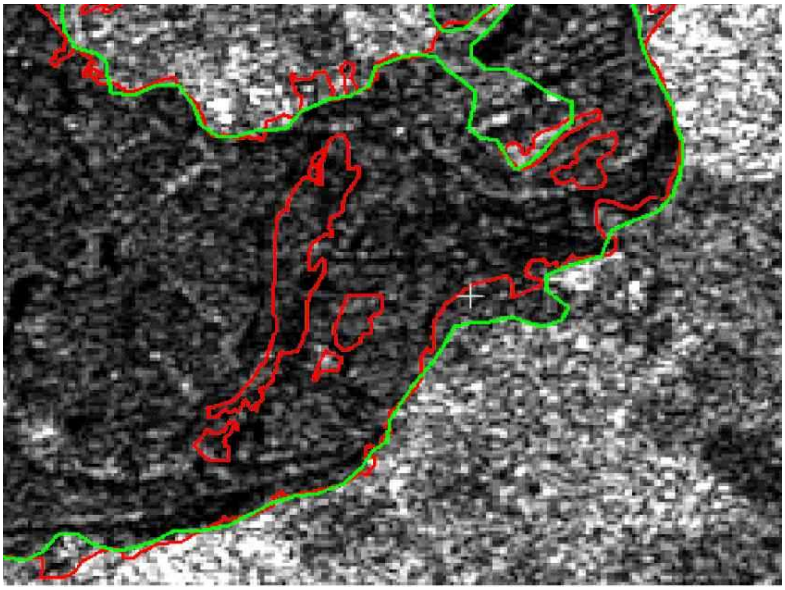


Figure 2.1: SAR image overlaid with shorelines derived using the snake algorithm (green) and from aerial photographs (red), for a 3 km by 3 km subregion of the River Thames between Buscot and Standlake. Reprinted with kind permission of Horritt *et al.* (2001).

well.

Satellite overpass times are of the order of days so it is rare to obtain multiple observations of flood extent for the same event, which we could use to test flood wave propagation. In valley filling events in which large changes in water depth result in small changes to flood extent, the simulator will not need to accurately represent flow depth to reproduce the observed flood extent, and so this data does not help constrain the simulator (Mason *et al.*, 2003).

Ideally the data used for calibration will be spatio-temporal, but until this becomes a reality it will be necessary to combine time series with spatial data in order to fully constrain hydraulic simulators. However, before this can be done a formal calibration framework should be developed for observations of flood extent which until now have only been treated with non-probabilistic methods.

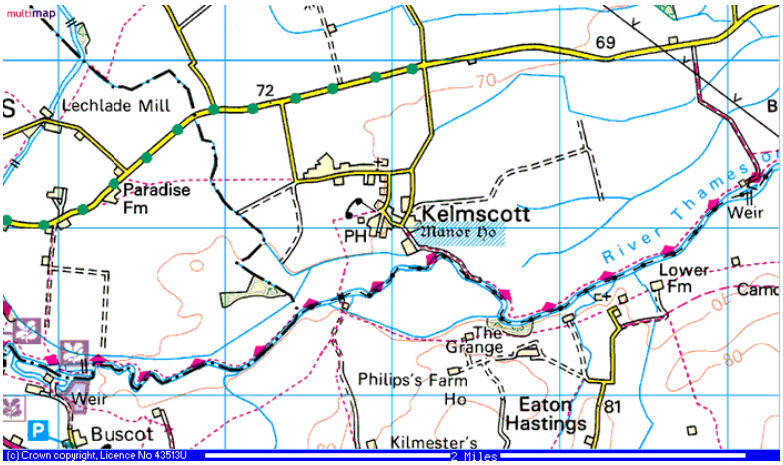


Figure 2.2: Image reproduced with kind permission of Ordnance Survey and Ordnance Survey of Northern Ireland.

2.4 Buscot Dataset

The test site we use throughout the thesis is located on the upper river Thames in Oxfordshire, UK, see Figure 2.2. The 4 km long reach is almost entirely agricultural, to the South the flow is restricted by high land but there is an extensive floodplain to the North (Horritt and Bates, 2001). The bankfull discharge is 40 m^3s^{-1} and the river drains a 1000 km^2 catchment (Aronica *et al.*, 2002).

The catchment is bounded upstream by a weir at Buscot. Topographic data is provided by a 50 m resolution airborne stereophotogrammetric digital elevation model (DEM) with a vertical accuracy of ±25 cm and 48 by 76 cells (Aronica *et al.*, 2002). Channel position can be discretized from ordnance survey 1:10000 series maps and cross-sections can be found from ground surveys.

In December 1992 a 1 in 5 year flood event coincided with an overpass of the ERS-1 satellite. The SAR image was taken 20 hours after the peak discharge of 76 m^3s^{-1}, but the hydrograph was very broad so the discharge was still 73 m^3s^{-1}. The SAR image has a resolution of 12.5 m and was processed with the snake algorithm (Horritt *et al.*, 2001) to form a map of flood extent with boundaries accurate to ±50 m.

A dynamic simulation was deemed unnecessary because the reach is short so flow reacts quickly to any changes to the inflow, and the hydrograph changes

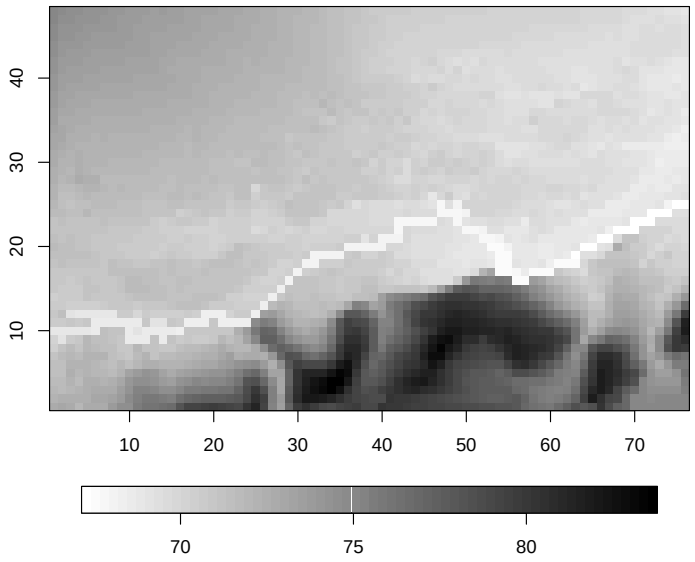


Figure 2.3: Digital elevation model for the Buscot dataset. The channel has been added manually.

slowly. Therefore a kinematic wave was used for the channel flow. The DEM was modified to include a dyke which runs for 500 m along the North side of the channel upstream (Horritt and Bates, 2001), and the channel cross-sections are all set to be 20 m wide and 2 m deep, see Figure 2.3. The boundary conditions are: for the channel a constant inflow of 73 m^3s^{-1}, and for the floodplain a fixed water surface elevation on the East side to allow water to flow out of the domain without first returning to the channel, and zero flux conditions on the other three sides. The initial conditions are: no water on the floodplain, 2 m deep in the channel, and zero outflow. The unknown parameters are Manning's channel friction n_c and floodplain friction n_f. LISFLOOD-FP was run for 500 values of n_c and n_f sampled uniformly between 0.01 and 0.05 m^3s^{-1} and 0.02 and 0.10 m^3s^{-1} respectively, the total computation time was 35 hours (Horritt and Bates, 2001).

The simulator output takes the form of water depths on a 48 by 76 raster grid whereas the flood extent is represented as a binary valued 192 by 304 raster grid. We resolve the discrepancy in resolutions by reprojecting the flood extent onto a 48 by 76 raster grid; each 50 m resolution cell corresponds to 16 12.5 m resolution

cells, if the average over these cells is greater than 0.5 we take the value 1 otherwise -1. Water depths give some indication about *how wet* a cell is but not *how dry*, in Section 8.9 we discuss a method for using this data directly and the problems with doing so, elsewhere in the thesis we threshold the simulator outputs at 0 m water depth to obtain binary array representations of flood extent.

In this chapter we have given an overview of the hydrological background. We justified the use of LISFLOOD-FP and described the Buscot dataset that will be used to demonstrate our calibration framework. In the next chapter we will describe the statistical background.

Chapter 3

Statistical Background

In the last chapter we gave an overview of the simulator, the need for calibration, and the data we will use to calibrate the simulator. In this chapter we give an overview of Bayesian statistics, the Markov chain Monte Carlo (MCMC) method for generating a sample from the unnormalised posterior, and directed acyclic graphs (DAGs) for illustrating hierarchical models.

3.1 Bayesian Statistics

Statistical inference is the science of making conclusions about a population from samples from that population. Let X be a random variable corresponding to some property of a sample from the population. We specify a probability model $p(x|\theta)$ for X, then for an observed sample $X = x$ we can make inference about a population characteristic θ. How we make inference about θ depends on the approach to inference that we adopt. Throughout this discussion we will adopt the notation of continuous random variables, the discrete case is derived similarly. We will use the terms density and distribution interchangeably.

There are two main approaches to statistical inference which differ in their treatment of θ. In the *classical* or *frequentist* approach probability is defined to be the long run proportion of times an event occurs, so θ is treated as an unknown constant (see Rice, 1995). In the *Bayesian* approach probability is defined as a measure of an individuals belief, so θ is treated as a random variable (see Gelman *et al.*, 2004). The consequence is that frequentist inference is based on $p(x|\theta)$ whereas

Bayesian inference is based on $p(\theta|x)$. The interplay of Bayesian and frequentist statistics is discussed in Bayarri and Berger (2004).

We often have a belief about an experiment that is not captured in the data. Treating θ as a random variable allows us to specify a *prior* distribution for θ, $p(\theta)$, which encodes our subjective beliefs about the value of θ before any data has been observed. An example from O'Hagan (1994) will make this clearer. Suppose we look out of the window and see a big brown thing with smaller brown things coming out of that and little green things on them. Is the thing we observe a tree or a postman? Let A be the event we observe such an object, B_1 be the event the object is a tree, and B_2 be the event the object is a postman. Clearly we would reject the hypothesis that the object is a postman because $P(A|B_1) > P(A|B_2)$. Is the thing we observe a tree or a fake tree? Let B_3 be the event the object is a fake tree, then $P(A|B_1) = P(A|B_3)$ so they are equally likely but surely we would reject the idea it is a fake anyway. We need to include our prior belief, $P(B_1) > P(B_3)$, in making our decision.

The presence of a prior distribution leads naturally to inference using the *posterior distribution* through Bayes' theorem,

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{\int p(x|\theta)p(\theta)\,\mathrm{d}\theta} \tag{3.1}$$

where the denominator is $p(x)$. We can rewrite Equation (3.1) as

$$\text{posterior} \propto \text{likelihood} \times \text{prior}.$$

In frequentist inference the value of θ which *maximises* the likelihood is important, in Bayesian inference we take the *weighted average* of the likelihood where the weights used are taken from the prior distribution. One of the main advantages of the Bayesian approach is that the entire inference is contained in the posterior distribution.

We can also view the posterior as the prior updated by the likelihood. Thinking of it in these terms motivates *sequential updating*. Consider two independent variables X and Y from $p(x|\theta)$ and $p(y|\theta)$. Suppose we observe x then the posterior

$p(\theta|x)$ becomes the prior before observing y,

$$p(\theta|x, y) \propto p(y|\theta)p(x|\theta)p(\theta).$$

Note that the result will be the same regardless of the order of the observations, and would also be the same if we had updated simultaneously on (x, y) because $p(y|\theta)p(x|\theta) = p(x, y|\theta)$.

So far we have treated θ as a scalar, but the algebra for the multivariate case $\boldsymbol{\theta}$ is the same. To make posterior inference about a single term θ_i we integrate out the other parameters from the posterior density

$$p(\theta_i|x) = \int p(\boldsymbol{\theta}|x)\, \mathrm{d}\boldsymbol{\theta}_{-i}, \tag{3.2}$$

where $\boldsymbol{\theta}_{-i} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}_{i+1}, \ldots, \boldsymbol{\theta}_n)$ for some integer n.

The ability to specify subjective beliefs through the prior distribution is both the appeal of the Bayesian method and the feature most susceptible to misuse. We now briefly discuss some of the issues involved in specifying a prior.

The computational difficulty in integrating out $\boldsymbol{\theta}_{-i}$ in Equation (3.2) means, in general, the posterior cannot be found exactly and we must use some approximate method (see Section 3.2). However, when the likelihood is a member of an exponential family, we are able to identify a prior for which the posterior is in the same family of distributions as the prior. These are called *conjugate priors*. For example let $X_1, X_2, \ldots, X_n$ be independent and identically distributed $\mathcal{N}(\theta, \kappa^{-1})$, where the precision κ is known. Suppose our prior is $\theta \sim \mathcal{N}(a, b^{-1})$, then

$$\theta|x_1, \ldots, x_n \sim \mathcal{N}\left(\frac{ba + n\kappa\overline{x}}{b + n\kappa}, \frac{1}{b + n\kappa}\right). \tag{3.3}$$

Although very convenient, conjugate priors are not appropriate if they do not represent our prior belief, even if they do exist.

As the sample size n increases the prior becomes less important, see Equation (3.3). However, this does not mean Bayesian inference is only appropriate when a lot of data is available. When there is not a lot of data available it is essential to include expert subjective beliefs when making decisions.

Consider what happens in Equation (3.3) as $b \to 0$. The posterior becomes $\mathcal{N}(\overline{x}, (n\kappa)^{-1})$ but the prior becomes $p(\theta) \propto 1.0$ for $\theta \in \mathbb{R}$ which cannot be normalised, we say it is an *improper prior*. Although the use of improper priors is a contentious issue, if we take b to be arbitrarily close to 0.0 we find the resulting posterior is arbitrarily close to the one obtained using an improper prior, which in some sense justifies their use.

Representing ignorance is not as trivial as taking $p(\theta) \propto 1.0$ because priors are not in general invariant to transformations. Bertrand's paradox demonstrates this very effectively (see for example Kac and Ulam, 1968). Consider the probability that a chord of a circle drawn at random is longer than the side of an inscribed equilateral triangle. If one end of the chord is fixed and we consider the angle that the chord makes with the tangent to be $\mathcal{U}[0, \pi]$ then the probability will be 1/3, but if we assume the midpoint of the chord is picked randomly within the circle then the chord is only longer if the midpoint lies within a circle of half the radius so the probability is 1/4. We shall see some more examples in Chapters 7 and 8.

The main objection to Bayesian inference by proponents of frequentist inference is the fact that the results will depend on the prior which is subjective. The Bayesian counter to this argument is best summarised by de Finetti (1974) who argues that probability does not exist in any objective sense and can only be thought of as an individual's bet. The main advantage of the Bayesian approach is that probabilistic statements can be made about θ because we treat it as a random quantity. This is well illustrated by comparing (frequentist) confidence intervals with (Bayesian) credible intervals. The region C_α is a $100(1-\alpha)\%$ *credible interval* if

$$\int_{C_\alpha(x)} p(\theta|x)\, \mathrm{d}\theta = 1 - \alpha.$$

A $100(1-\alpha)\%$ confidence interval does not mean θ lies in this interval with probability $(1-\alpha)$ because θ is fixed – it is either in the interval or not. The correct interpretation is that if many samples are made from the distribution then in the long run, $100(1-\alpha)\%$ of the confidence intervals calculated using these samples

will encompass the true θ value. A $100(1-\alpha)\%$ credible interval means the posterior probability that θ is in the interval is $(1-\alpha)$. Credible intervals provide the information we want to know and their interpretation is more intuitive than confidence intervals.

As for confidence intervals, credible intervals are not unique, therefore highest posterior density regions are defined. Let $C_\alpha(x) = \{\theta : p(\theta|x) > \delta\}$ then choose δ such that

$$\int_{C_\alpha(x)} p(\theta|x)\,\mathrm{d}\theta = 1-\alpha,$$

then C_α is the $100(1-\alpha)\%$ *highest posterior density credible region*.

Berger and Wolpert (1988) argue that frequentism should be rejected for not satisfying the likelihood principle, which states that if two experiments yield likelihood functions that are proportional to one another then the same inference must be made from these two experiments (see O'Hagan, 1994). The fact that the likelihood principle is satisfied by Bayesian inference is obvious from Equation (3.1). The likelihood appears in the numerator and denominator, so multiplying the likelihood by a constant makes no difference to the posterior. The reason frequentist inference does not satisfy the likelihood principle is because frequentist inference is based not only on the value observed but on the distribution $p(x|\theta)$ at unobserved values. The likelihood principle requires inference to only be based on the values of x that are observed. For example the concept of unbiasedness does not satisfy the likelihood principle, the bias of an estimator $\hat{\theta}(x)$

$$\text{bias}(x) = \mathrm{E}\left(\hat{\theta}(x)|\theta\right) - \theta$$

depends on $p(x|\theta)$ for all x through the expectation.

Except in some special cases (e.g. conjugate priors) the evaluation of the normalising constant $p(x) = \int p(x|\theta)p(\theta)\,\mathrm{d}\theta$ cannot be avoided or done analytically so we must make use of approximate methods. The (re)discovery of Markov chain Monte Carlo (MCMC) by the Bayesian community in the 1980s has proved to be a large factor in making Bayesian statistics more generally applicable. In the next section we discuss MCMC.

3.2 Markov Chain Monte Carlo

The general situation we consider in this section is when we have an unnormalised density π^u and we want to make inference using the normalised density π, e.g. $P(\boldsymbol{\theta} \in C) = \int_C \pi(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$. This situation is common in Bayesian inference because we are typically not able to integrate out $\boldsymbol{\theta}_{-i}$, see Section 3.1. This is also true of distributions defined in terms of their full conditionals such as the Ising model (see Besag, 1974, and Chapter 6). Other approaches to this problem include rejection sampling and importance sampling (see Robert and Casella, 2004, pages 90–106), but for rejection sampling a distribution h must be identified such that π^u/h is bounded and the efficiency of the algorithm is strongly dependent on the h chosen. If π^u/h is not bounded then expectations can be calculated using importance sampling but it is not possible to form a sample from π.

Markov chain Monte Carlo works by constructing a time-homogenous discrete time Markov chain with stationary distribution π (Gilks *et al.*, 1996). We form a realisation $\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(K)}\}$ and treat it as a random sample from π. Note it is not a random sample but the empirical distribution estimates the target distribution (Green, 2001). Approximate expectations and probabilities are

$$\mathrm{E}\left(f(\boldsymbol{\theta})\right) = \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} \approx \frac{1}{K}\sum_{k=1}^{K} f(\boldsymbol{\theta}^{(k)}) \quad \text{and}$$

$$\int_C \pi(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} \approx \frac{1}{K}\sum_{k=1}^{K} \mathbf{1}[\boldsymbol{\theta}^{(k)} \in C]$$

for any region C.

Let the Markov chain *transition probability* be written $P(d\boldsymbol{\theta}'|\boldsymbol{\theta})$, then to construct a Markov chain with stationary distribution π we require

$$\int_{\boldsymbol{\theta}\in\Theta} \pi(d\boldsymbol{\theta})P(d\boldsymbol{\theta}'|\boldsymbol{\theta}) = \pi(d\boldsymbol{\theta}'), \tag{3.4}$$

where Θ is the state space for $\boldsymbol{\theta}$, i.e. if the current distribution is π then one step later we are still in π, we say π is *invariant* for the transition kernel P. If this is satisfied by the transition probabilities we *hope* that a realisation from the Markov chain, $\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(K)}\}$, will approximate a sample from π as K increases (we

will discuss convergence a little later). A sufficient but not necessary condition for π to be invariant for transition kernel P is that we have *detailed balance*

$$\pi(\boldsymbol{\theta})P(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}')P(\boldsymbol{\theta}|\boldsymbol{\theta}')$$

for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$. We say the Markov chain is *reversible with respect to* π. It is simple to prove that Equation (3.4) is satisfied so π is the stationary distribution of the chain. Most MCMC methods are developed on the basis of detailed balance because it is a lot easier to work with than invariance (see Green, 2001).

We will now discuss some of the main recipes for MCMC. Many methods can be seen as special cases of the Metropolis-Hastings sampler, which was introduced by Hastings (1970) as a generalisation of the Metropolis method (Metropolis *et al.*, 1953).

In the Metropolis-Hastings method a candidate value $\boldsymbol{\theta}'$ is proposed from an *arbitrary* density $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$. This proposal is accepted as the next state of the chain with probability

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = \min\left\{1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right\}.$$

This acceptance probability has been chosen so the Markov chain is reversible with respect to π, it is not unique but is optimal in the sense that it maximises the probability of acceptance. Peskun's theorem (Peskun, 1973) says that changing a reversible Markov chain sampler to increase the probability of acceptance cannot be bad and we expect it to reduce asymptotic variance. There is no need to calculate the unknown normalising constant because in the ratio it cancels, $\pi(\boldsymbol{\theta}')/\pi(\boldsymbol{\theta}) = \pi^u(\boldsymbol{\theta}')/\pi^u(\boldsymbol{\theta})$.

We do not need to update all components of $\boldsymbol{\theta}$ simultaneously. Let A be a subset of indices and $\boldsymbol{\theta}_A = \{\theta_i | i \in A\}$, then the proposal distribution could be

$$q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = q_A(\boldsymbol{\theta}'_A|\boldsymbol{\theta})\mathbf{1}[\boldsymbol{\theta}'_{-A} = \boldsymbol{\theta}_{-A}].$$

Although single component updates are simplest, updating more than one component at a time sometimes reduces the time to convergence. The order in which the updates are made at each iteration does not need to be fixed, and furthermore, not all components need to be updated at each iteration.

There are many interesting special cases of the Metropolis-Hastings sampler, which include: *independence Metropolis-Hastings* when we ignore the current state of the chain $q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = q(\boldsymbol{\theta}')$; *Metropolis method* when the proposal is symmetric $q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}')$, in this case the proposal ratio cancels; and *random walk Metropolis* when $q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = q(|\boldsymbol{\theta}' - \boldsymbol{\theta}|)$ and $q(\cdot)$ is symmetric about 0. The *Gibbs sampler* for $\boldsymbol{\theta}_A$ is a special case of the Metropolis-Hastings sampler when the proposal distribution is taken to be the full conditional distribution, $q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}'_A|\boldsymbol{\theta}_{-A})\mathbf{1}[\boldsymbol{\theta}'_{-A} = \boldsymbol{\theta}_{-A}]$ (see Geman and Geman, 1984). The Gibbs sampler is in some sense automatic because there is no possibility of tuning the proposal distribution and proposals are always accepted $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = 1$. The Gibbs sampler is of particular interest in applications in which the full conditionals have a simple form, such as in spatial statistics where the joint distribution may be defined in terms of the full conditionals, (see for example Besag *et al.*, 1995).

Only the stationary distribution of the chain is of interest, the first iterations will be affected by the initial value $\boldsymbol{\theta}^{(0)}$ and should be removed. We call this period before the chain converges the *burn-in period*.

The kernel P is *ϕ-irreducible* if there exists a probability distribution, ϕ, on Θ such that for all $A \subseteq \Theta$

$$\phi(A) > 0 \Rightarrow P(\tau_A < \infty | \boldsymbol{\theta}^{(0)} = \boldsymbol{\theta}) = 1$$

for π-almost all $\boldsymbol{\theta} \in \Theta$ where $\tau_A = \min\{k : \boldsymbol{\theta}^{(k)} \in A\}$. If irreducible for any ϕ then it is π-irreducible, the weaker condition allows checking of fewer sets.

If the Markov chain $\{\boldsymbol{\theta}^{(k)}\}$ with transition kernel P is ϕ-irreducible and invariant, then the sample expectation converges to the population expectation for π-almost all $\boldsymbol{\theta}^{(0)}$.

Let $\{A_0, A_1, \ldots, A_{m-1}\}$ be a collection of subsets such that $P(A_{i+1 \mod m}|\boldsymbol{\theta}) = 1$ for all $\boldsymbol{\theta} \in A_i$ and all i, this is called an *m-cycle*. A chain is *aperiodic* if the largest m for which a m-cycle exists is 1.

If the Markov chain $\{\boldsymbol{\theta}^{(k)}\}$ is ϕ-irreducible, invariant and aperiodic then the distribution of $\{\boldsymbol{\theta}^{(k)}\}$ converges to π for π-almost all $\boldsymbol{\theta}^{(0)}$.

3.3 Introduction to Directed Acyclic Graphs

Graphical models (GMs) have been used in a number of contexts, including machine learning, market research, speech cognition, information theory, pattern recognition, and engineering (see Best and Green, 2005). The key themes running through all of these are uncertainty and complexity. GMs break complex systems down into smaller parts, the parts are connected by probability theory which provides a consistent model and a means of interfacing with data (see Jordan, 1999). For a comprehensive reference on the theory of GMs see Lauritzen (1996).

Common to all GMs are: nodes representing random variables, and edges or arrows between variables encoding conditional independence assumptions. The two main classes of GMs are *directed acyclic graphs* (DAGs) and *conditional independence graphs* (CIGs). DAGs (also known as *Bayesian* or *belief networks*) are commonly used in the statistics community where there is some directional dependence to encode. The term acyclic refers to the fact that there can be no directed loops. *Condition independence graphs* (also known as *undirected graphs*, *Markov fandom fields* or *Markov networks*) are often used in spatial statistics where the dependence between variables has no clear direction (see Møller, 2003; Rue and Held, 2005). It is possible to form a certain combination of these two classes, where there may be directional dependence between sets of variables but within the sets there may be undirected dependence.

Directed acyclic graphs have three components:

1. Nodes representing random variables.

2. Arrows between nodes encoding conditional independence assumptions. (If a variable is not modelled directly on another, there will be no arrow between them.)

3. Conditional distributions defined at each node.

If there is an arrow from node A to node B we say node A is the *parent* of B, and B is the *child* of A. If a node has no parents it is called a *founder node*,

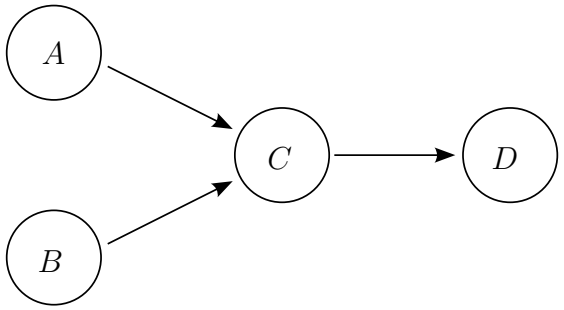


Figure 3.1: Example directed acyclic graph.

and requires a marginal distribution to be specified rather than a conditional one. Given conditional (or marginal) distributions at each node the joint distribution is completely and uniquely specified; let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be the variables in the DAG and $pa(i)$ be the set of indices of the parents of x_i, then

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i | \boldsymbol{x}_{pa(i)}),$$

is the joint distribution of $\boldsymbol{x}$. For example, the joint distribution for the DAG in Figure 3.1 would be

$$\begin{aligned} P(A, B, C, D) &= P(D|A, B, C)P(C|A, B)P(B|A)P(A) \\ &= P(D|C)P(C|A, B)P(B)P(A). \end{aligned}$$

GMs provide a compact visual representation of the model structure, and reduce the complexity in defining joint distributions for high dimensional problems. Best and Green (2005) describe a paternity experiment in which a direct specification of the joint distribution requires ≈ 2000 million numbers to be specified (and we would have to check the probabilities sum to 1). However, when a DAG is used only 1347 numbers need to be specified, making the problem more manageable. This is because we only need to specify the values of the distribution at each node conditional on all possible values of the corresponding parent nodes.

Inference in GMs is very simple if the variables of interest are descendants of the observed variables, called *top-down reasoning*. For example, suppose A and B in Figure 3.1 are observed and we want to make inference about C and D, then

$$P(C, D|A, B) = P(D|C)P(C|A, B).$$

However, if the variables of interest are ancestors of the observed variables, inference, called *bottom-up reasoning*, requires recourse to Bayes' theorem. For our example suppose D is observed and we want to make inference about A, B and C, and all variables are continuous, then

$$P(A, B, C|D) = \frac{P(D|C)P(C|A, B)P(A)P(B)}{P(D)},$$

where

$$P(D) = \iiint P(D|C)P(C|A, B)P(A)P(B)\,\mathrm{d}A\,\mathrm{d}B\,\mathrm{d}C.$$

In general the denominator, $P(D)$, is very difficult to calculate, making exact computation of the posterior density $P(A, B, C|D)$ infeasible. However, Markov chain Monte Carlo (MCMC) can be used to generate a sample from the joint posterior distribution $P(A, B, C|D)$, and this requires only the unnormalised density $P(D|C)P(C|A, B)P(A)P(B)$. Marginal distributions, for example

$$P(A|D) = \iint P(A, B, C|D)\,\mathrm{d}B\,\mathrm{d}C,$$

have integrals in the numerator and denominator which may both be difficult to calculate. Let $\{(A^{(k)}, B^{(k)}, C^{(k)})|k = 1, \ldots, K\}$ be a sample from the joint posterior $P(A, B, C|D)$ generated by MCMC, then if we "throw away" the B and C values we obtain a sample from the marginal distribution $P(A|D)$, $\{A^{(k)}|k = 1, \ldots, K\}$. All MCMC methods that update subsets of variables require full conditional distributions, which makes them particularly well suited to DAGs because the conditional distribution of a node given all others is dependent only on its children, parents and the other parents of its children.

In DAGs all variables are conditionally independent of their non-descendants given their parents. Conditional independence assumptions encoded by DAGs are well described by the *Bayes Ball* algorithm from Ross Shachter (see Shachter, 1998). Nodes A and B are conditionally dependent given the set of observed nodes, if a ball can travel along the graph from A to B where the allowable movements of the ball are shown in Figure 3.2.

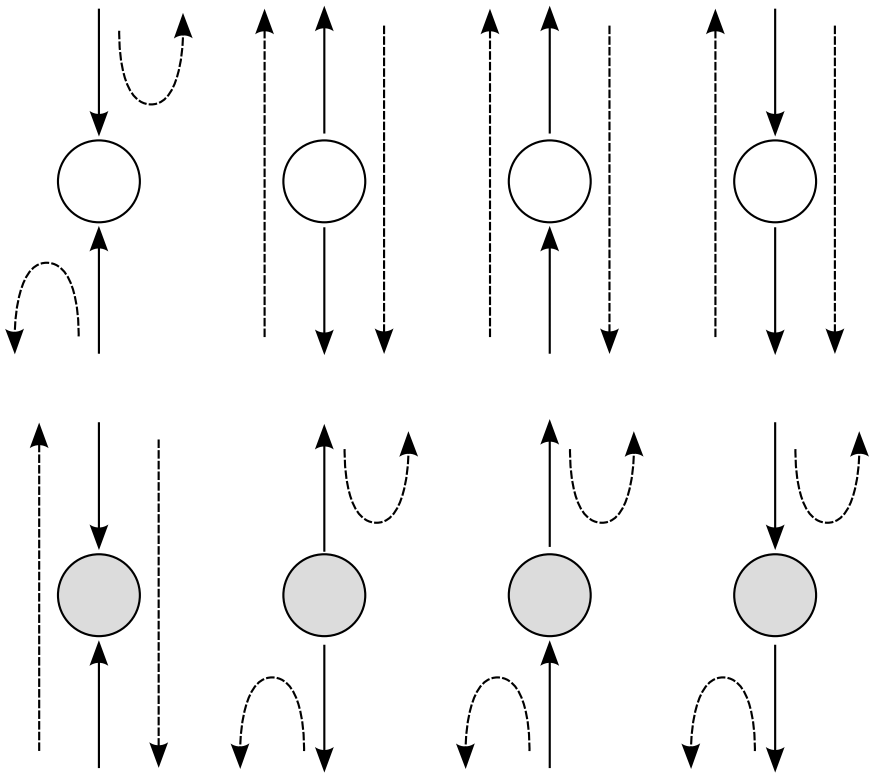


Figure 3.2: Rules for the Bayes ball algorithm from Ross Shachter. The white nodes correspond to unobserved variables and the grey nodes to observed variables. The solid arrows show the connections to the neighbouring nodes along the path of the ball. The dashed lines indicate whether the ball can pass through the node or whether it is "bounced".

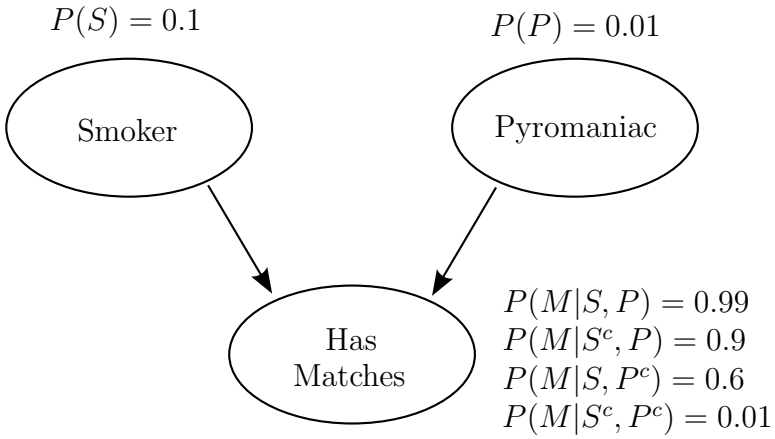


Figure 3.3: An example of Berkson's paradox. The superscript c indicates the complement of the event, e.g. S^c is the event that the person does not smoke.

The least intuitive relationship is shown in the first column of Figure 3.2. Marginally, parents with a common child are independent, but they become conditionally dependent if the child is observed. This is known as *Berkson's paradox* (or *explaining away*, see Murphy, 2001). As an example let S be the event that someone smokes, let P be the event that the person is a pyromaniac, and let M be the event they have matches. Let the joint distribution be defined by the DAG in Figure 3.3. Then looking at the population of people who were found to have matches, we find $P(S|M) = 0.780$ and $P(S|M, P) = 0.109$. We conclude that, within the population of people with matches, being a pyromaniac makes you less likely to be a smoker.

Let us illustrate the concepts discussed above using an example borrowed from Best and Green (2005). Suppose we have one fair coin A and one biased coin B, such that the probability of getting a head $P(B = H) = 0.8$. We pick a coin at random and toss it 6 times. Suppose we get 6 heads, then what is the chance we get a head on the next toss? Figure 3.4 shows how we could represent this using a DAG.

If we know which coin has been chosen then the chance of getting a head on the next throw is independent of whether the first 6 tosses were all heads. However, when the coin is unknown the two events are dependent because getting all heads on the first 6 throws informs us about the probability that we have chosen coin A

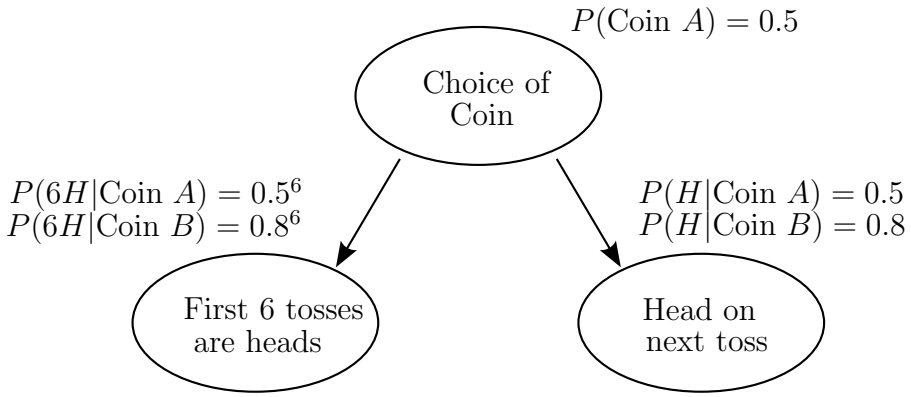


Figure 3.4: An example illustrating the various features of directed acyclic graphs.

or coin B. This can be read straight from Figure 3.4 using the Bayes Ball rules laid down in Figure 3.2.

The efficiency of using graphical models, instead of defining the joint density directly, can be seen by noticing that we would need to define $2^3 = 8$ joint probability values if we attacked the problem directly, but using graphical models we have been able to define the system with only 5.

Suppose we obtain all heads on the first 6 throws. We can make inference about the probability of getting a head on the next throw by first bottom-up reasoning using Bayes theorem to find the posterior for the coin choice,

$$P(A|6H) = \frac{P(6H|A)P(A)}{P(6H|A)P(A) + P(6H|B)P(B)} = 0.056,$$

and $P(B|6H) = 0.944$, and then top-down reasoning,

$$P(H|6H) = P(H|A)P(A|6H) + P(H|B)P(B|6H) = 0.783,$$

where we have used the fact, encoded by the DAG, that given the choice of coin the probability of getting a head on the next throw is independent of whether the first 6 throws were heads, i.e. $P(H|A, 6H) = P(H|A)$ and $P(H|B, 6H) = P(H|B)$.

In Chapter 5 we will use a DAG to illustrate our Bayesian framework for calibration of flood inundation simulators conditioned on an observation of flood extent.

In this chapter we have reviewed the statistical background for the thesis. In the next chapter we will classify the uncertainties in flood inundation prediction and review methods for calibration and calibrated prediction.

Chapter 4

Handling Uncertainty in Flood Inundation Simulators

We begin this chapter by classifying the uncertainties in hydraulic modelling. We claim that most uncertainties are due to lack of knowledge rather than intrinsic randomness. This cannot be represented by frequentist statistics, so we need to use Bayesian statistics. We introduce calibration and calibrated prediction, and review two methods, one Bayesian and one non-probabilistic, that are indicative of the methods currently available. We end the chapter with the way forward for calibration and calibrated prediction.

4.1 Classifying Uncertainties in Hydraulic Modelling

In his paper on the role of statistics in science, Box (1976) explains that "all models are wrong, but some are useful". There are many different sources of uncertainty that contribute to a simulator being wrong and it is important to identify which sources are accounted for by a given uncertainty handling approach (see Section 4.2 for a review of approaches). Kennedy and O'Hagan (2001) present a classification of uncertainties that is appropriate for all computer codes of complex physical systems. In this section we describe this classification and say how it relates to the specific problem of flood inundation modelling.

However, before we describe how we might classify the sources of uncertainty it is worth noting that there are only two types of uncertainty. The uncertainty

in repeated events because of intrinsic randomness and unpredictability is called *aleatory uncertainty*, and the uncertainty in unrepeatable events due to a lack of knowledge is called *epistemic uncertainty* (O'Hagan, 2004a,b). Many of the sources of uncertainty in computer code approximations to physical systems are epistemic and this has an important bearing on the method used to handle uncertainty. In frequentist statistics the probability of an event is defined to be the long run proportion of times it occurs, therefore it is only appropriate for aleatory uncertainties. On the other hand, Bayesian statistics, as defined in Section 3.1, can quantify both aleatory and epistemic uncertainties through probabilities (O'Hagan, 2004a) and is therefore more appropriate as a tool for handling uncertainty in computer codes of physical systems.

The simulator inputs can be divided into calibration inputs $\boldsymbol{\theta}$ and variable inputs $\boldsymbol{x}$. For our purposes, the calibration inputs are the channel friction, θ_c, and floodplain friction, θ_f, and the variable input which changes between the calibration and prediction events is the inflow discharge. We assume the topography is measured without error and is constant between events. For a given parameter set $(\boldsymbol{x}, \boldsymbol{\theta})$ the deterministic output of the flood inundation simulator, $\eta(\boldsymbol{x}, \boldsymbol{\theta}) \in \{-1, 1\}^n$, is a binary array in which pixels take the value 1 if wet and -1 if dry. The true flood extent we are attempting to predict is denoted $\boldsymbol{\xi} \in \{-1, 1\}^n$, and the observation of this flood extent is denoted $\boldsymbol{\zeta} \in \{-1, 1\}^n$. For the calibration event, $\boldsymbol{x}$, the simulator is run for a sample $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(K)}$ of calibration input values to obtain $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(K)}$ where $\boldsymbol{y}^{(i)} = \boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta}^{(i)})$. For the same set of calibration input values the simulator is run for the event we want to predict, $\boldsymbol{x}'$, to obtain $\boldsymbol{y}'^{(1)}, \ldots, \boldsymbol{y}'^{(K)}$ where $\boldsymbol{y}'^{(i)} = \boldsymbol{\eta}(\boldsymbol{x}', \boldsymbol{\theta}^{(i)})$. We want to make inference about the flood extent in the future given our simulator and an observation of a past event.

4.1.1 Parametric Uncertainty

Parametric uncertainty is the uncertainty associated with the unknown input parameters, $\boldsymbol{\theta}$, called the *calibration inputs*. For the Buscot application of LISFLOOD-FP (see Section 2.4) there are two unknown parameters: one for friction in the channel, θ_c, and one for friction on the floodplain, θ_f. More generally, using

LISFLOOD-FP we can assign a different friction parameter for each pixel in the floodplain, and more complex codes may also have unknown input parameters that characterise the higher-order processes represented. However, for simplicity we here use only the two parameter case.

To make predictions about a future event by calibrating the simulator on an observed event, the unknown parameters are assumed to be the same for calibration and prediction events. This is a very strong assumption for the flood inundation problem because the calibration event will typically be much smaller in magnitude than the prediction event, and when an area becomes flooded the effect of friction changes (Romanowicz and Beven, 2003). However, we are calibrating on only one observation and therefore there is no possibility of interpolation between events of different magnitudes. For the Buscot example the same 500 simulations are used for the calibration and prediction events so this issue will not arise, but in practice the validity of this assumption must be checked for each case considered.

In Bayesian statistics the assumption of a "true" distribution for an unobserved event, $p(\boldsymbol{\xi})$, implies that there is a "pseudotrue" parameter value $\boldsymbol{\theta}^\star$, such that, under relatively weak conditions, $p(\boldsymbol{\theta}|\boldsymbol{\xi})$ converges to $\boldsymbol{\theta}^\star$ as information about $\boldsymbol{\theta}$ contained in $\boldsymbol{\xi}$ increases (Spiegelhalter *et al.*, 2002). In practice it may be difficult to distinguish between different parameter values with the limited data available, in statistics this is called *nonidentifiability of parameters* and in hydrology this is called *equifinality* (Beven, 2006). Rather confusingly equifinality is said to be a lumping together of "nonidentifiability", "nonuniqueness" and "instability" (Ebel and Loague, 2006). We have used quotation marks to differentiate these hydrological terms from their statistical namesakes which have subtly different definitions. In hydrology, "identifiability" requires there to be a unique model parameterisation in which all the parameters are meaningful. "Uniqueness" requires that only one set of parameters can be estimated from the observed data and that this set of parameters represent the behaviour in the event we want to predict. "Stability" requires that small changes in the observed data do not significantly change the estimated parameter set values, and, conversely, that small changes to the

parameter set values do not significantly change the simulator output.

Beven (2006) argues that the potential for multiple acceptable values should be a feature of any uncertainty handling approach. This view is now widely accepted throughout hydroinformatics (Wagener and Gupta, 2005). In the Bayesian context, Swartz *et al.* (2004) argue that issues of nonidentifiability should be rectified by using a prior that is informative about the nonidentifiability. We prefer the stance of Lindley (1971), that nonidentifiability causes no problem for the Bayesian approach, simply integrate the posterior as required. Allowing the posterior to be flatter means that in prediction we will be averaging over a greater number of simulations. This seems preferable because the lack of observed data requires us to assume $\boldsymbol{\theta}$ is stationary between the event we are calibrating on and the event we want to predict although this will rarely be the case (see Section 5.1). Typically any particular $\boldsymbol{\theta}$ only provides acceptable performance for a small range of $\boldsymbol{x}$.

4.1.2 Parametric Variability

When some of the variable inputs $\boldsymbol{x}$ are not fixed but are allowed to vary according to some joint distribution the resulting additional uncertainty on the prediction of the process $\boldsymbol{z}$ is called *parametric variability*. We may leave some inputs unspecified because we cannot measure them or because we are interested in how uncertainty on the inputs propagates to uncertainty on the predictions, this is called *uncertainty analysis* (UA). For the Buscot application of LISFLOOD-FP the variable inputs are the inflow hydrograph and the topography, both of which are treated as error free for our purposes. Wilson and Atkinson (2005) investigate the effect of topographic uncertainty on inundation predictions, and Pappenberger, Matgen, and Beven (Pappenberger *et al.*) investigate the effect of rating curve uncertainty on inundation predictions.

Now we have discussed the uncertainties associated with variable inputs, $\boldsymbol{x}$, and calibration inputs, $\boldsymbol{\theta}$, it is worth clarifying the differences between these two types of input. Between the event we calibrate on and the event we want to predict we expect the calibration inputs to be the same and the variable inputs to be different. The variable inputs relate directly to physical quantities that can be measured,

but the calibration inputs take account of unrepresented processes in the simulator so do not relate to measurable physical quantities. If variable inputs cannot be measured then prediction cannot be carried out because we cannot calibrate these inputs, they are not stationary between events. However, if the measurement of a variable input is not very accurate we can assign it a prior distribution.

4.1.3 Residual Variability

The real world process that we are trying to predict is conditioned by the variable inputs of the simulator, $\boldsymbol{x}$. There will not be a unique real process satisfying these conditions, the variation given these conditions is called *residual variability*. We are combining two sources of uncertainty here: first the stochasticity of nature and second the effect of the variable inputs not fully conditioning the real process. Therefore residual variability may be reduced by identifying more conditions in the simulator. For example for flood inundation simulators we may find that including higher-order processes such as full three-dimensional solutions to the Navier-Stokes equations helps constrain the space of possible real process values. Kennedy and O'Hagan (2001) define the *true process value* to be the mean averaged over residual variability.

4.1.4 Simulator Inadequacy

Even if we are certain about the values of the simulator inputs the simulator predictions will not be perfect. *Simulator inadequacy* is defined by Kennedy and O'Hagan (2001) as the discrepancy between the true mean value of the real world process and the output of the simulator run at the true value of the calibration inputs, $\boldsymbol{\theta}^{\star}$. For the flood inundation application we can see that simulator inadequacy should increase as the dimensionality of the processes represented in the simulator decreases.

4.1.5 Code Uncertainty

Although the computer code is deterministic we do not know the value of the output until the code has been run, and as each run may be very computer intensive

we may only be able to run the code for a limited number of input configurations. The uncertainty about the code output is called *code uncertainty* and is zero where the code has been run. Kennedy and O'Hagan (2001) propose the use of a statistical emulator for the code output to interpolate between the points at which the code has been run. Although LISFLOOD-FP can be slow when high resolution data are used, in Chapter 5 we develop a method that works with a computationally feasible set of runs of the code and requires no emulator.

4.1.6 Observation Error

Observation error is that associated with the measurement of the real world process. For the flood inundation problem we are using an observation of flood extent taken from synthetic aperture radar (SAR) using an active region segmentation algorithm (Horritt, 1999). The errors in the SAR are of two types: errors near the flood boundary and field misclassifications (see Section 2.3). In practice, observation error cannot be separated from residual variability unless we have repeated observations where all conditions are the same, even those which are not recognised by the variable inputs (Kennedy and O'Hagan, 2001).

Now we have classified the sources of uncertainty in complex codes of physical systems, in the next section we consider a number of methods for handling uncertainty in these codes.

4.2 Calibration and Calibrated Prediction

In this section we review methods for calibration and calibrated prediction.

4.2.1 Handling Uncertainty

Without an observation of a past event, uncertainty handling is normally limited to studying the relationship between the calibration and variable inputs, $\boldsymbol{\theta}$ and $\boldsymbol{x}$, and the simulator output, $\boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta})$. Uncalibrated predictions of the true value of the real process, $\boldsymbol{\xi}$, are only possible if we model the relationship between $\boldsymbol{\xi}$ and

the simulator output $\boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta})$, i.e. the simulator inadequacy, and specify values or distributions for the unknown calibration inputs. Traditional use of simulators assumes that variable inputs, $\boldsymbol{x}$, are measured without error, unknown calibration inputs, $\boldsymbol{\theta}$, can be specified without error, and there is no inadequacy in the resulting simulator output $\boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta})$, so this is a prediction of the real process.

Sensitivity analysis (SA) is the study of how changes in individual input parameters, x_i or θ_i, affect the simulator output $\boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta})$, the aim being to identify those parameters to which the simulator is particularly sensitive or insensitive (Saltelli *et al.*, 2000). Local SA amounts to finding partial derivatives of the simulator output with respect to the input, and in global SA the input is varied over a range (Kennedy *et al.*, 2002).

Uncertainty analysis (UA) is the study of how uncertainty on one or more inputs translates to uncertainty in the simulator output. In the standard Monte Carlo approach a probability distribution is defined on the inputs, then a sample is drawn from this distribution and the simulator is run for each sample point. The outputs form a sample from the simulator output distribution (Kennedy and O'Hagan, 2001). UA accounts for parametric uncertainty and variability.

Now suppose we have an observation $\boldsymbol{z}$ of $\boldsymbol{\xi}$, then we can make inference about the values of the unknown calibration inputs $\boldsymbol{\theta}$ and the simulator inadequacy by comparing the simulator output with $\boldsymbol{z}$. Using what we have learnt we can use the simulator to predict a future event with variable inputs $\boldsymbol{x}'$ (see Chapter 5 for details of the Bayesian approach).

Calibration is the act of making inference about the values of the calibration inputs $\boldsymbol{\theta}$ on the basis of how well the corresponding simulator output $\boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta})$ fits the observed data $\boldsymbol{z}$. Traditionally, calibration amounts to identifying the best fitting value and using this for future predictions without any quantification of parametric uncertainty or simulator inadequacy (Kennedy and O'Hagan, 2001). In a Bayesian sense calibration means updating the prior for the calibration inputs as a result of comparing the simulator output with observations (Campbell, 2002). A major difficulty with calibration is in the specification of the measure of fit which should,

from the Bayesian point of view, be the likelihood of the data. Calibration accounts for observation error, residual variation and simulator inadequacy through the measure of fit (Kennedy and O'Hagan, 2001).

As discussed in Section 2.1, simulators are truncations of reality so the true values of the calibration inputs (e.g. Manning's n) will not be physically meaningful. Whilst the measurement error can be quantified for measured physical parameters, the uncertainty due to unrepresented processes in the simulator cannot be quantified. If the latter error is large it is better to calibrate the input if the input is stationary between the event we are calibrating on and the event we want to predict. The measured value and measurement error can be used to inform the prior distribution of the calibration input. The uncertainty in a calibrated input includes implicitly the uncertainty due to unrepresented processes and observation error (Campbell, 2002); the larger the difference between the value of a calibrated input and the measured value the more significant the observation error or lack of process representation.

A *calibrated prediction* is a prediction with a quantification of uncertainty taken from calibration on a past event. The uncertainty comes from the uncertainty in the calibrated inputs and from the simulator inadequacy. Calibration inputs are assumed stationary between the event we calibrate on and the event we wish to predict. For this to be true all the processes accounted for explicitly or implicitly by the calibrated input must be stationary (Romanowicz and Beven, 2003). This is rarely the case so it is important to include the uncertainty about the calibration inputs in prediction, rather than just taking the best fitting value. Some methods account for the uncertainty in the calibrated inputs but not the simulator inadequacy, and so the resulting predictions correspond to the simulator output and not the true value of the real process (e.g. generalised likelihood uncertainty estimation).

In the following two sections we consider two of the main approaches to calibration that are indicative of the numerous approaches available. The first is a formal Bayesian methodology and the second a non-probabilistic approach.

4.2.2 Bayesian Analysis of Computer Code Output

The fundamental idea behind Bayesian analysis of computer code output (BACCO) is to build a statistical emulator of the computer code output (see O'Hagan, 2004a). Using a statistical emulator of the computer code output means fewer simulator runs are required for analysis of the code (e.g. UA and SA), which makes the method very useful in situations where the simulator takes a long time to run.

The aim of the BACCO method is to describe the accuracy of computer codes statistically and even correct for error in the simulator through the *simulator inadequacy* function. O'Hagan (2004a) argues that the inputs and simulator inadequacy are epistemic uncertainties (see Section 4.1) and therefore require a Bayesian, rather than frequentist, treatment.

The use of emulators of computer code output was first devised in the design and analysis of computer experiments (DACE) work summarised in Sacks *et al.* (1989). The simulator output $\eta(x)$ is modelled as a random function, the prior for $\eta(x)$ is a Gaussian process, and then updating using runs of the simulator we obtain a posterior for $\eta(x)$. The resulting statistical emulator can predict $\eta(x)$ at untried x values. Statistical analysis of computer code outputs (SACCO) generalised DACE to interpolation, sensitivity analysis, uncertainty analysis, calibration and simulator uncertainty (Kennedy *et al.*, 2002). BACCO brings together the ideas of SACCO in a unified framework.

In BACCO the random functions are typically Gaussian processes. A random function $f : \mathbb{R}^n \to \mathbb{R}$ is a *Gaussian process* if, for all $k \in \mathbb{N}$ and $\boldsymbol{x}_i \in \mathbb{R}^n$ for $i = 1, \ldots, k$, the joint distribution of $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_k)$ is multivariate normal. For example, for all $\boldsymbol{x} \in \mathbb{R}^n$, $f(\boldsymbol{x})$ is normal (Hankin, 2005). The error in the emulator is zero where the simulator has been run and elsewhere the covariance is chosen to ensure that $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ are close if $\boldsymbol{x}$ and $\boldsymbol{x}'$ are. It is this assumption of smoothness encoded in the emulator that allows sensitivity and uncertainty analysis to be performed with far fewer runs (Oakley and O'Hagan, 2002, 2004). Also UA and SA methods using a statistical emulator account for code uncertainty.

One problem with Gaussian processes is that uncertainty increases rapidly if we try to extrapolate outside the range of the data (O'Hagan, 2004a).

Kennedy and O'Hagan (2001) introduces the BACCO approach to calibration which is the first attempt to take account of all sources of uncertainty explicitly. Given observations of the real process the difference between the simulator output and reality is modelled by a Gaussian process called the *simulator inadequacy function*. We perform Bayesian calibration to learn about the calibration inputs and the simulator inadequacy function. Consequently, the simulator output can be corrected using the simulator inadequacy function, this can then be used to improve predictions or to inform simulator development (O'Hagan, 2004a).

By augmenting the variable inputs $\boldsymbol{x}$ with a parameter which indexes the pixels, the simulator output can be written as a scalar $\eta(\boldsymbol{x}, \boldsymbol{\theta})$. The true value of the real process is written $\xi(\boldsymbol{x})$ and we make N simulator runs to obtain $\boldsymbol{y} = (y_1, \ldots, y_N)$ where $y_i = \eta(\boldsymbol{x}_i, \boldsymbol{\theta}_i)$. The calibration data consist of n observations $\boldsymbol{z} = (z_1, \ldots, z_n)$, where z_i is an observation of $\xi(\boldsymbol{x}_i^\star)$ for known variable inputs $\boldsymbol{x}_i^\star$ which need not be the same as the points where the simulator is run.

Kennedy and O'Hagan (2001) suggest the following model

$$z_i = \xi(\boldsymbol{x}_i) + e_i = \rho\eta(\boldsymbol{x}_i, \boldsymbol{\theta}) + \delta(\boldsymbol{x}_i) + e_i \tag{4.1}$$

where $e_i \sim \mathcal{N}(0, \lambda)$ is the observation error and residual variation for the ith observation, ρ is an unknown regression parameter, and $\delta(\cdot)$ is the simulator inadequacy function. Calibrated predictions are made using the marginal posterior for reality $\xi(\boldsymbol{x})$ given the simulator runs $\boldsymbol{y}$ and observed data $\boldsymbol{z}$, this is obtained by integrating the posterior with respect to all parameters and therefore takes account of all uncertainties. However, this is rarely practical so the hyperparameters ρ, λ and the parameters of the covariance for the Gaussian processes for $\eta(\cdot, \cdot)$ and $\delta(\cdot)$ are fixed. Consequently, this approach is not fully Bayesian and does not fully account for observation error, simulator inadequacy and code uncertainty.

In the BACCO approach to calibration we *simultaneously* fit a statistical emulator to the data and learn about the calibration inputs and the simulator inadequacy function. The simulator inadequacy function is essential for calibrated prediction

but causes the true value of the calibration inputs to be less physically meaningful because the majority of the fitting is done by this simulator inadequacy function (Kennedy *et al.*, 2002). The fundamental assumption of BACCO is that the simulator output is a smooth continuous function of its inputs, whilst a binary indicator of inundation is not a smooth function of position, water depth is and it may be possible to model this quantity using BACCO. More realistic simulator inadequacy functions need to be developed, for example Goldstein and Rougier (2004) look at the relationship between simulator output and reality. Computationally this approach to calibration is very demanding, requiring the inversion of a variance matrix with dimension given by the number of simulator runs (O'Hagan, 2004a).

We have described BACCO in depth because it is well established and encapsulates most of the features present in all Bayesian approaches to calibration and calibrated prediction. However, there are many other Bayesian methods and we now summarise a few.

Bates *et al.* (2003) apply Bayesian calibration to obtain the posterior for the calibration inputs, the corresponding uncertainty on the simulator output is found by uncertainty analysis. Simulator inadequacy is not accounted for and no attempt is made to predict the true value of the real process.

Bayes linear methods use expectation rather than probability as a primitive and provide a way of tackling problems when standard Bayesian analysis is prohibitively complex (see Goldstein, 1995, for an introduction). Our prior beliefs are quantified via expectations, variances and covariances which are *adjusted* given data. Bayes linear calibration is described in Craig *et al.* (1996) and calibrated prediction in Craig *et al.* (2001) and Goldstein and Rougier (2004).

A *Bayesian forecasting system* (BFS) for short-term probabilistic river stage forecasts is described by Krzysztofowicz (2002) using a deterministic hydraulic simulator with a probabilistic precipitation forecast input. The effects of precipitation uncertainty and hydrologic uncertainty on the river stage are quantified separately, then they are integrated together using Bayesian theory.

Bayesian total error analysis (BATEA) for environmental simulators is a method

for learning about the hydrologic calibration inputs and variable inputs from observations of the simulator output and the variable inputs (Kavetski *et al.*, 2002). The observations are assumed to be independent given the true value of the variable inputs. There is no obvious advantage over using the observation of the variable input to specify the prior and then updating this for the posterior.

4.2.3 Generalised Likelihood Uncertainty Estimation

Beven and Binley (1992) proposed *generalised likelihood uncertainty estimation* (GLUE) as an alternative to the traditional search for an optimum parameter set, after identifying in Binley and Beven (1991) that the optimum is rarely the same between calibration and prediction events, but the response surfaces may be similar. (In the Bayesian setting the response surface is the posterior for the calibration inputs.) The philosophy underpinning GLUE is *equifinality*: multiple simulator structures and parameter sets may be equally acceptable as simulators of reality (Beven, 2006). GLUE is described as a way of refining hypotheses about simulator structure and parameter sets by associating generalised likelihood values and rejecting *non-behavioural simulators*. A *behavioural simulator* is one that agrees with the observed data to a degree specified by the modeller, and the *generalised likelihood* replaces the standard likelihood and is not required to satisfy the conditions of probabilistic inference.

We assume variable inputs $\boldsymbol{x}$ are fixed, although GLUE can be extended to account for parametric variability (Kennedy and O'Hagan, 2001). A prior is specified for the calibration inputs, $p(\boldsymbol{\theta})$, and is usually assumed to be uniform on a feasible region but not required to be so. The simulator is run for a sample from the prior, $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(M)}$, to obtain a simulator output sample, $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(M)}$, where $\boldsymbol{y}^{(i)} = \boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta}^{(i)})$. A generalised likelihood, $p^{\star}(\boldsymbol{z}|\boldsymbol{\theta})$, is defined which must increase monotonically as the similarity between the simulator output, $\boldsymbol{y} = \boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta})$, and observed data, $\boldsymbol{z}$, increases. Unlike the standard likelihood the generalised likelihood is not required to be proportional to the conditional distribution of the observed data given the simulator output, $p(\boldsymbol{z}|\boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta}))$. We will elaborate on the implications for probabilistic inference shortly. Non-behavioural simulations are

rejected if the generalised likelihood is less than some user-defined threshold p', so

$$q^\star(\boldsymbol{z}|\boldsymbol{\theta}) = \begin{cases} p^\star(\boldsymbol{z}|\boldsymbol{\theta}) & \text{if } p^\star(\boldsymbol{z}|\boldsymbol{\theta}) > p' \\ 0 & \text{otherwise.} \end{cases}$$

The (generalised) posterior for the calibration inputs is

$$p^\star(\boldsymbol{\theta}|\boldsymbol{z}) \propto q^\star(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}), \tag{4.2}$$

and for the prediction event with variable inputs $\boldsymbol{x}'$ the simulator output $\boldsymbol{y}' = \boldsymbol{\eta}(\boldsymbol{x}', \boldsymbol{\theta})$ is weighted by $p^\star(\boldsymbol{y}'|\boldsymbol{z}) = p^\star(\boldsymbol{\theta}|\boldsymbol{z})$.

GLUE is exactly a Bayesian analysis if the generalised likelihood is a standard likelihood and the rejection of non-behavioural simulators step is removed (see for example Romanowicz *et al.*, 1996). In this case Equation (4.2) is the standard Bayesian formula for the posterior given the likelihood and prior.

The generalised likelihood is a feature of GLUE and is argued for in preference to the Bayesian approach on the basis of equifinality and ease of specification (Beven, 2006). However, equifinality is not a new concept to statistics where it is known as unidentifiability, and is not outside the capabilities of Bayesian statistics where it is trivially possible for the posterior $p(\boldsymbol{\theta}|\boldsymbol{z})$ to take the same value for different values of the calibration inputs $\boldsymbol{\theta}$. Bayesian inference does encode the existence of a true value of the calibration inputs (Spiegelhalter *et al.*, 2002), but this does not prevent unidentifiability. Whilst it is true that likelihood specification can be very tricky, Mantovan and Todini (in press) have shown that using a generalised likelihood leads to incoherence. A consequence of incoherence is that the addition of more data does not improve the value of the analysis.

There is typically no clear demarcation between behavioural and non-behavioural simulations. The threshold p' is usually selected based on the assumption that errors in simulation are similar to errors in observation, but this often results in all simulations being rejected so no prediction can be made. Beven (2006) interprets this total rejection as indicative of conceptual, structural or data errors, but it may equally be the result of setting the threshold too high. Removing any simulations means we are not fully representing parameter uncertainty. We

would rather provide probabilistic predictions to flood engineers on which they may make decisions about simulator adequacy.

Unlike the more rigorous approaches to calibration, GLUE has been applied to flood inundation modelling. Romanowicz *et al.* (1996) use continuous observations of water level for cross-sections to calibrate and make calibrated predictions. They explicitly include simulator inadequacy and use a proper likelihood so the analysis is strictly Bayesian. The simulator inadequacy is assumed to be the same for calibration and prediction. Romanowicz and Beven (2003) calibrate on inundation widths extracted from an observation of maximum inundation extent for cross-sections. The generalised likelihood for a cross-section is 1 if the predicted width is within 30 m of the observed width, and decreases to 0 away from this region.

Aronica *et al.* (2002), Bates *et al.* (2004), Hunter *et al.* (2005) and (Pappenberger *et al.*, 2005) calibrate directly on observations of flood extent in the form of binary images, rather than transform this data to inundation widths. Table 4.1 shows the possible combinations of simulator output and observed data for pixel i. The generalised likelihood is based on a function of the total number of true-negatives, false-negatives, true-positives and false-positives, called a *skill score* (see Jolliffe and Stephenson, 2003, page 8). Let $n_{s,t} = \sum_{i=1}^{n} \mathbf{1}[y_i = s]\mathbf{1}[z_i = t]$ where $s, t \in \{-1, 1\}$, then the most frequently used skill score is

$$p^\star(\boldsymbol{z}|\boldsymbol{\theta}) = \frac{n_{1,1}}{n_{1,1} + n_{1,-1} + n_{-1,1}}. \tag{4.3}$$

Many others are discussed in Hunter *et al.* (2005). GLUE provides so-called *maps of flood probability*, for each pixel

$$p^\star(\xi'_i = 1|\boldsymbol{z}) = \int \mathbf{1}[y'_i = 1]p^\star(\boldsymbol{\theta}|\boldsymbol{z})\,\mathrm{d}\boldsymbol{\theta} \tag{4.4}$$

where $y'_i = \eta_i(\boldsymbol{x}', \boldsymbol{\theta})$ (Aronica *et al.*, 2002). However, even if a proper likelihood is used so $p^\star(\boldsymbol{\theta}|\boldsymbol{z})$ is a proper posterior, Equation (4.4) approximates $p(y'_i = 1|\boldsymbol{z})$ because it does not take account of simulator inadequacy. In equating future predictions with reality, Equation (4.4) implies that there is no simulator inadequacy after calibration, which is not true. Therefore when comparing our method to GLUE it will be appropriate to compare our inference about $\boldsymbol{y}'$ to these maps of flood probability.

y_i	z_i	
	-1	1
-1	true-negative	false-negative
1	false-positive	true-positive

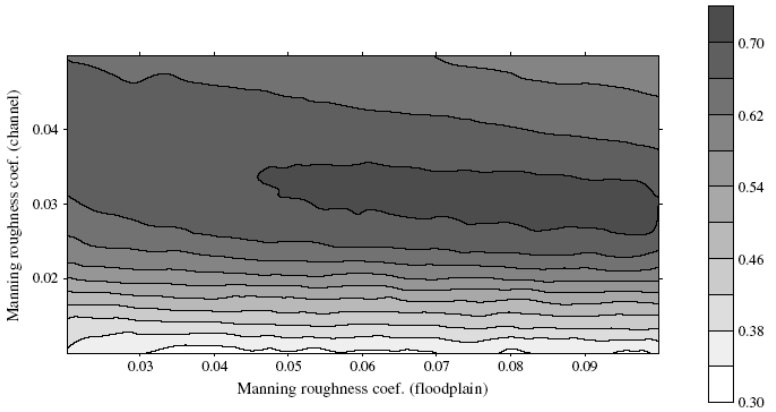
Table 4.1: Binary cross-classifications for simulator output y_i and observed data z_i for pixel i.

Aronica *et al.* (2002) apply this methodology to the Buscot dataset described in Section 2.4. Figure 4.1 show the results of the analysis, the response surface shows an insensitivity to floodplain friction, θ_f, and the maps of flooding probability with and without non-behavioural simulations show how uncertainty on calibration inputs induces uncertainty on simulator output.

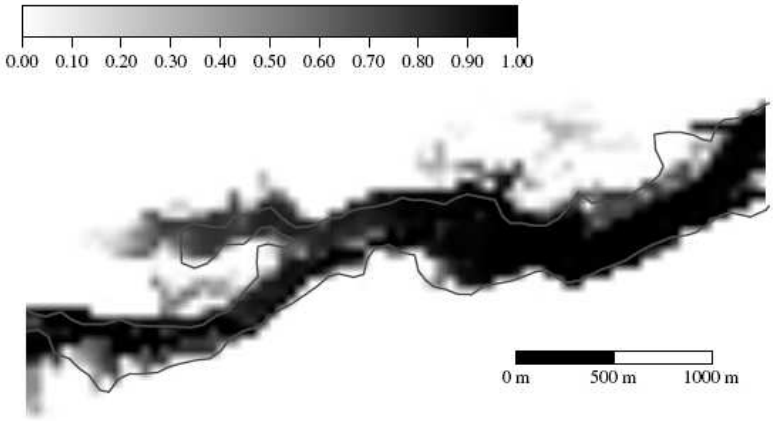
There are numerous other non-probabilistic approaches to calibration and calibrated prediction but we have focused on GLUE because it has been applied to flood inundation simulation. Alternatives include *multi-objective calibration* (Gupta *et al.*, 1998; Yapo *et al.*, 1998). This requires the simultaneous optimisation of a number of objective functions with respect to the calibration parameters, the set of solutions is called the *Pareto set*. Yapo *et al.* (1998) claim that because individual objectives relate to different things they cannot be combined to form an overall objective. However, in the Bayesian approach this can all be included in the prior and likelihood.

4.2.4 The Way Forward for Calibration and Calibrated Prediction

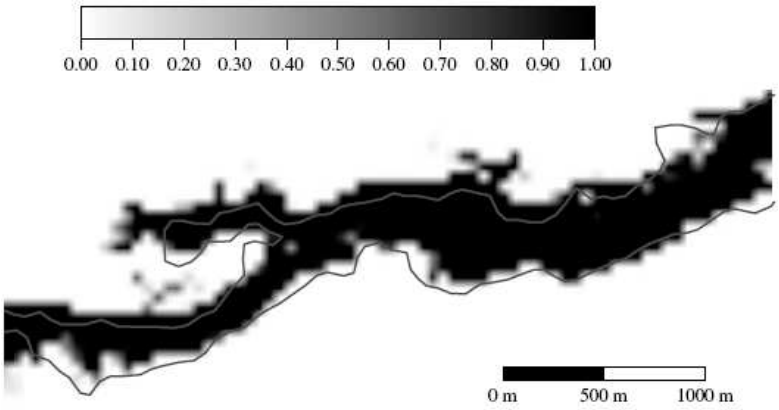
For any calibration procedure it is necessary to define a function which judges how well the simulator reproduces the observed data. In Bayesian calibration the function is the likelihood of the observed data given the simulator output, in GLUE it is the generalised likelihood, and in multi-objective calibration it is the set of objective functions. The specification of this function can be very difficult if the Bayesian paradigm is adopted and the laws of probability must be satisfied. It is for this reason that less rigorous approaches, such as GLUE, have proved so popular. The propriety of this function determines the success of the calibration



(a) Response surface.



(b) Map of flood probability, $p^\star(\xi_i' = 1|\boldsymbol{z})$.



(c) Map of flood probability, $p^\star(\xi_i' = 1|\boldsymbol{z})$, when non-behavioural simulations, characterised by $p^\star(\boldsymbol{z}|\boldsymbol{\theta}) < 0.7$, are removed.

Figure 4.1: Results of GLUE analysis for the Buscot dataset using the skill score from Equation (4.3), shown with and without non-behavioural simulations. Images reproduced with kind permission of Aronica *et al.* (2002).

procedure. The need for a better model for simulator inadequacy is acknowledged by Kavetski *et al.* (2002) for BATEA, and by O'Hagan (2004a) for BACCO.

Although we attribute the popularity of non-probabilistic methods to their accessibility, not all information in hydraulic applications is probabilistic and therefore recourse should be made to non-probabilistic approaches (Hall and Anderson, 2002; Hall, 2003). The important point is that non-probabilistic approaches should still have a rigorous grounding in mathematics. Alternative uncertainty methods developed through various weakening of Kolmogorov's axioms of probability include Chequet's theory of capacities, random set theory, evidence theory, fuzzy set theory, possibility theory and Walley's theory of imprecise probabilities (see Hall, 2003, for a review). For example Ben-Haim (2001) describes information gap theory for handling ignorance which is not probabilistic. Methods which cannot be described within any uncertainty framework are unlikely to be useful.

Although the value of each of these methods is appreciated, we are of the opinion that the Bayesian paradigm has not been exhausted in the flood inundation context.

In this chapter we have classified flood inundation simulator uncertainties and reviewed calibration methods. In the next chapter we will describe our Bayesian framework for calibration and calibrated prediction. This will be illustrated using a directed acyclic graph (DAG) which allows the problem to be broken down into smaller, more manageable, parts. Thus the specification of an appropriate likelihood model can be focused on.

Chapter 5

Bayesian Framework for Calibration of Flood Inundation Simulators Conditioned on an Observation of Flood Extent

In this chapter we introduce a hierarchical model for Bayesian calibration of flood inundation simulators conditioned on an observation of flood extent. We start by illustrating our hierarchical model using a directed acyclic graph. We show how the tasks of calibration and calibrated prediction can be carried out in relation to the specified hierarchical model. The most problematic issue in the framework is the specification of the likelihood of the observed flood extent given a simulation of flood extent. For the purpose of demonstration we propose a very simple likelihood model and work through an example. In the following chapters more complex likelihood models, which better represent the data, will be developed.

5.1 Directed Acyclic Graph for Flood Inundation Simulator Calibration

In Section 4.2 we reviewed methods for handling uncertainty in complex computer models, including sensitivity analysis, uncertainty analysis, calibration and calibrated prediction. Methods have been developed by environmental scientists interested in quantifying the uncertainty in their simulators and also by applied statisticians. Bayesian methods are prominent in both communities, combining

probabilistic rigour with the ability to use subjective beliefs via a prior distribution.

Bayesian analysis of computer code output (BACCO) accounts for all sources of uncertainty explicitly and, in this sense, is the most developed uncertainty handling methodology. However, fundamental to BACCO is the use of Gaussian processes, in emulating the complex computer code output and for simulator inadequacy (see Section 4.2.2). In Section 4.2.4 we identified the need for a realistic likelihood model for the observed data given the simulator output, it therefore seems inappropriate to restrict our research to Gaussian processes. Also BACCO is not fully Bayesian because the hyperparameters are set to their posterior means for prediction, whereas for full Bayesian analysis they should be integrated out (see Kennedy and O'Hagan, 2001).

Rather than subscribe to an existing calibration methodology we will present our own hierarchical model for Bayesian calibration and calibrated predication.

The simulator inputs can be divided into calibration inputs, $\boldsymbol{\theta}$, and variable inputs, $\boldsymbol{x}$. For our purposes, the calibration inputs are the channel friction, θ_c, and floodplain friction, θ_f, and the variable input which changes between the calibration and prediction events is the inflow discharge. We assume the topography is measured without error and is constant between events. For a given parameter set $(\boldsymbol{x}, \boldsymbol{\theta})$ the deterministic output of the flood inundation simulator, $\boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta}) \in \{-1, 1\}^n$, is a binary array of size $n = r \times c$ where r is the number of rows and c is the number of columns. Pixels take the value 1 if wet and -1 if dry. Pixel i is located in row $i \bmod r$ and column $\lfloor i/r \rfloor$, where $\lfloor u \rfloor$ is the largest integer not greater than u, and rows and columns are numbered from 0. The true flood extent on the binary array is denoted $\boldsymbol{\xi} \in \{-1, 1\}^n$ (see Section 4.1.3 for the definition of a true process value), and the observation of this flood extent is denoted $\boldsymbol{z} \in \{-1, 1\}^n$. For the calibration event, $\boldsymbol{x}$, the simulator is run for a sample $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(K)}$ of calibration input values to obtain $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(K)}$ where $\boldsymbol{y}^{(i)} = \boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta}^{(i)})$. For the same set of calibration input values the simulator is run for the event we want to predict, $\boldsymbol{x}'$, to obtain $\boldsymbol{y}'^{(1)}, \ldots, \boldsymbol{y}'^{(K)}$ where

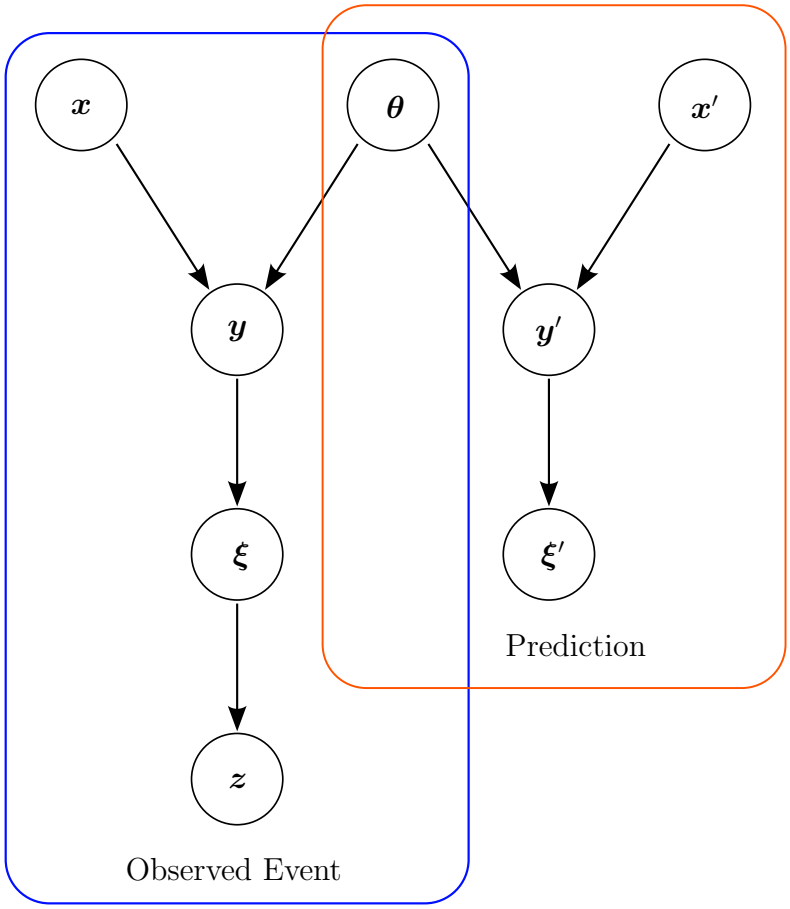


Figure 5.1: A directed acyclic graph (DAG) for Bayesian calibration of flood inundation simulators conditioned on an observation of flood extent.

$\boldsymbol{y}'^{(i)} = \boldsymbol{\eta}(\boldsymbol{x}', \boldsymbol{\theta}^{(i)})$. We want to make inference about the flood extent in the future given our simulator and an observation of a past event.

Figure 5.1 shows our DAG for calibration and calibrated prediction. It encodes the uncertainties and dependencies in calibrating flood inundation simulators on an observation of flood extent, $\boldsymbol{z}$, and making calibrated predictions of the true flood extent in a future event, $\boldsymbol{\xi}'$. We have endeavoured to account for the uncertainties described in Section 4.1. We learn about the parametric uncertainty associated with the unknown calibration inputs by assigning a prior $p(\boldsymbol{\theta})$. Then, given the observation, $\boldsymbol{z}$, we can calculate the posterior $p(\boldsymbol{\theta}|\boldsymbol{z})$. If the values of the variable inputs are not fully known we can express the parametric variability through a prior, $p(\boldsymbol{x})$. The simulator inadequacy is encoded in the likelihoods, $p(\boldsymbol{\xi}|\boldsymbol{y})$ and $p(\boldsymbol{\xi}'|\boldsymbol{y}')$. To build a statistical emulator of the flood inundation simulator we can

express the code uncertainty in $p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})$. The residual variability and observation error are included in $p(\boldsymbol{z}|\boldsymbol{\xi})$.

We have assumed that the calibration inputs, $\boldsymbol{\theta}$, are stationary between the calibration and prediction events, $\boldsymbol{x}$ and $\boldsymbol{x}'$. However, the calibration inputs are the floodplain friction, θ_f, and channel friction, θ_c, which we do not expect to be stationary between events of different magnitudes because the frictional properties of land change when it is inundated. Without further observations or information about how friction changes between events of different magnitude, we must assume stationarity, but this remains a concern. If we had more time we might have elaborated the model (e.g. by allowing $\boldsymbol{\theta}$ to vary spatially) to get round this problem.

Bayesian calibration amounts to finding the posterior for the calibration inputs which is done by bottom-up reasoning,

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{z}) &\propto \sum_{\boldsymbol{\xi}} \sum_{\boldsymbol{y}} \int p(\boldsymbol{z},\boldsymbol{\xi},\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{x}, \\ &= \sum_{\boldsymbol{\xi}} \sum_{\boldsymbol{y}} \int p(\boldsymbol{z}|\boldsymbol{\xi})p(\boldsymbol{\xi}|\boldsymbol{y})p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{x})p(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{x}, \end{aligned} \tag{5.1}$$

using the DAG. Calibrated predictions based on the posterior for the calibration parameters is done by top-down reasoning,

$$\begin{aligned} p(\boldsymbol{\xi}'|\boldsymbol{z}) &\propto \sum_{\boldsymbol{\xi}} \sum_{\boldsymbol{y}} \sum_{\boldsymbol{y}'} \iiint p(\boldsymbol{z},\boldsymbol{\xi},\boldsymbol{\xi}',\boldsymbol{y},\boldsymbol{y}',\boldsymbol{x},\boldsymbol{x}',\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{x}'\,\mathrm{d}\boldsymbol{\theta}, \\ &= \sum_{\boldsymbol{y}'} \iint p(\boldsymbol{\xi}'|\boldsymbol{y}')p(\boldsymbol{y}'|\boldsymbol{x}',\boldsymbol{\theta})p(\boldsymbol{x}')p(\boldsymbol{\theta}|\boldsymbol{z})\,\mathrm{d}\boldsymbol{x}'\,\mathrm{d}\boldsymbol{\theta}, \end{aligned} \tag{5.2}$$

using the DAG.

We do not attempt to emulate the flood simulator output and therefore remove the code uncertainty component from the DAG. This does not necessitate a change to the graph because DAGs can encode deterministic relationships. We take

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = \begin{cases} 1 & \text{if } \boldsymbol{y} = \boldsymbol{\eta}\,(\boldsymbol{x},\boldsymbol{\theta}), \\ 0 & \text{otherwise,} \end{cases}$$

and similarly for $\boldsymbol{y}'$. We retain the $\boldsymbol{y}$ and $\boldsymbol{y}'$ nodes to make it clear that the true flood extent, $\boldsymbol{\xi}$, is modelled as dependent on the simulator output, $\boldsymbol{y}$, and not the inputs, $\boldsymbol{x}$ and $\boldsymbol{\theta}$.

Parametric variability may play a significant role in calibration and calibrated prediction but it has been researched extensively for environmental applications (see Kavetski *et al.*, 2002, for a review of approaches) and would add significantly to the computation expense of our method. Therefore, in order to focus on other components of our framework we will not consider parametric variability and will remove the $\boldsymbol{x}$ and $\boldsymbol{x}'$ nodes from the DAG.

We cannot calculate the marginal posterior distributions, $p(\boldsymbol{\theta}|\boldsymbol{z})$ and $p(\boldsymbol{\xi}'|\boldsymbol{z})$, because we are not able to perform the summations and integrals in Equations (5.1) and (5.2). Instead we use MCMC to generate an estimate sample from the marginal posterior, $p(\boldsymbol{\theta}|\boldsymbol{z})$, using the unnormalised density (see Section 3.2).

In MCMC algorithms the parameters are updated many times, but each time $\boldsymbol{\theta}$ changes we must run the flood inundation simulator to obtain $\boldsymbol{y} = \boldsymbol{\eta}(\boldsymbol{x}, \boldsymbol{\theta})$ and $\boldsymbol{y}' = \boldsymbol{\eta}(\boldsymbol{x}', \boldsymbol{\theta})$, because we are not using an emulator. This is impractical because each simulation will take at least a few minutes if not hours or days depending on the scale of the problem.

Instead of updating $\boldsymbol{\theta}$ in the MCMC algorithm, we discretize $\boldsymbol{\theta}$ by taking a sample $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(M)}$ from the prior, $p(\boldsymbol{\theta})$, and running the simulator to obtain $\boldsymbol{y}^{(m)} = \boldsymbol{\eta}\left(\boldsymbol{x}, \boldsymbol{\theta}^{(m)}\right)$ and $\boldsymbol{y}'^{(m)} = \boldsymbol{\eta}\left(\boldsymbol{x}', \boldsymbol{\theta}^{(m)}\right)$ for $m = 1, \ldots, M$. MCMC methods in which the index m is updated are feasible because the flood inundation simulator outputs have been stored and can be reused.

We will assume there is no observation error, so $\boldsymbol{z} = \boldsymbol{\xi}$ and $\boldsymbol{z}' = \boldsymbol{\xi}'$, and we remove the nodes corresponding to the true flood extents, $\boldsymbol{\xi}$ and $\boldsymbol{\xi}'$, from the DAG. The flood extent is delineated from SAR imagery using a region growing algorithm (see Horritt, 1999; Horritt and Bates, 2002, and Section 2.3). The error in shoreline delineation is relatively small but occasionally fields are misclassified as wet because sparsely vegetated areas have similar backscatter to water. This error can be removed manually by comparing the shoreline to the topography. Another way to interpret this assumption is that we are lumping together observation error, residual variability and simulator inadequacy, and our calibrated prediction will be of a future observation rather than a future truth. If it is acceptable to base

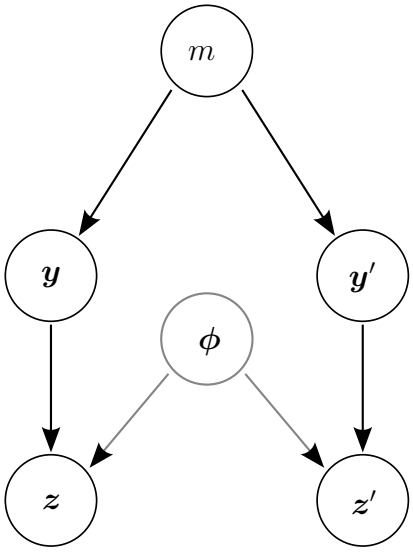


Figure 5.2: Revised DAG for the Bayesian analysis of flood inundation simulators conditioned on an observation of flood extent

decisions on observations of flood extent then predictions of a future observation seem justified. However, lumping together observation error, residual variability and simulator inadequacy limits the attraction of the framework because they are quite different sources of uncertainty. With further observations future research might consider separating these sources of uncertainty again, but for now this assumption allows us to focus on one aspect of the framework.

We need to define the likelihood of the observed data given the simulator output for the calibration and prediction events, $p(\boldsymbol{z}|\boldsymbol{y})$ and $p(\boldsymbol{z}'|\boldsymbol{y}')$. Let $\boldsymbol{\phi}$ be the vector of likelihood parameters, then assume the same distribution for $p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{\phi})$ and $p(\boldsymbol{z}'|\boldsymbol{y}', \boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is stationary between events. Now by calibrating on the observed data, $\boldsymbol{z}$, we learn about not only the parametric uncertainty, $p(\boldsymbol{\theta}|\boldsymbol{z})$, but also the simulator inadequacy, $p(\boldsymbol{\phi}|\boldsymbol{z})$.

Our revised hierarchical model is illustrated in Figure 5.2. To complete the model we must specify conditional (or marginal) distributions at each node. For the simulation index, m, of the sample $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(M)}$ from $p(\boldsymbol{\theta})$, we take a discrete uniform prior, $p(m) = 1/M$ (note that our prior knowledge is reflected in the values of the sample through the prior on $\boldsymbol{\theta}$). The nodes corresponding to $\boldsymbol{y}$ and $\boldsymbol{y}'$ are only included for completeness because $\boldsymbol{y}|m$ and $\boldsymbol{y}'|m$ are deterministic, the

distributions can be written

$$p(\boldsymbol{y}|m) = \begin{cases} 1 & \text{if } \boldsymbol{y} = \boldsymbol{\eta}\left(\boldsymbol{x}, \boldsymbol{\theta}^{(m)}\right) \\ 0 & \text{otherwise,} \end{cases} \tag{5.3}$$

and

$$p(\boldsymbol{y}'|m) = \begin{cases} 1 & \text{if } \boldsymbol{y}' = \boldsymbol{\eta}\left(\boldsymbol{x}', \boldsymbol{\theta}^{(m)}\right) \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

The remaining distributions $p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{\phi})$, $p(\boldsymbol{z}'|\boldsymbol{y}', \boldsymbol{\phi})$ and $p(\boldsymbol{\phi})$ will prove somewhat harder to specify. The prior $p(\boldsymbol{\phi})$ depends on the likelihood we chose for $p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{\phi})$ and $p(\boldsymbol{z}'|\boldsymbol{y}', \boldsymbol{\phi})$.

The likelihood is a model for a binary array, $\boldsymbol{z}$, conditional on the value of another binary array, $\boldsymbol{y}$. We expect the error in predicting the value of pixel i will be related to the error in predicting the value of the pixels in some neighbourhood of pixel i, therefore the likelihood should account for spatial dependence. We expect the error in predicting the value of pixel i will be greater at the flood boundary than in the channel or on the floodplain away from the flood boundary, therefore the likelihood should account for heterogeneity. Finally, we expect the observed value of pixel i will be related, not only to pixel i in the simulator output, but also to the neighbours of pixel i in the simulator output, therefore the likelihood should account for blur.

The equations for calibration and calibrated prediction become

$$p(m|\boldsymbol{z}) \propto \int p(\boldsymbol{z}|\boldsymbol{y}^{(m)}, \boldsymbol{\phi}) p(\boldsymbol{\phi}) \, \mathrm{d}\boldsymbol{\phi}, \qquad \text{and}$$

$$p(\boldsymbol{z}'|\boldsymbol{z}) \propto \sum_{m=1}^{M} \int p(\boldsymbol{z}'|\boldsymbol{y}'^{(m)}, \boldsymbol{\phi}) p(\boldsymbol{\phi}, m|\boldsymbol{z}) \, \mathrm{d}\boldsymbol{\phi}.$$

5.2 The Binary Channel Model

The observed data, $\boldsymbol{z}$, and the simulator output, $\boldsymbol{y}$, are binary arrays, so for the likelihood we need a model for a binary array conditional on the value of another binary array. The likelihood should include spatial dependence, heterogeneity and blur and this makes the likelihood specification a non-trivial problem. The development of a suitable likelihood model will form a major part of this thesis.

In this section we introduce the binary channel (BC) model, which is a very simple model for binary regression. It does not account for spatial dependence, heterogeneity or blur. However, the posteriors for calibration, $p(\boldsymbol{\theta}|\boldsymbol{z})$, and calibrated prediction, $p(z_i' = 1|\boldsymbol{z})$, can be found analytically if we use the BC model as the likelihood. We will demonstrate our Bayesian framework using the BC model and then explain why it is inadequate for our purposes. The likelihood models discussed in future chapters can all be motivated as extensions of this simple model.

The BC model is motivated by the transmission of a binary digit over a communication channel in which there may be some interference. Suppose y_i is the binary digit input and z_i the binary digit output, then

$$p(z_i = 1|y_i = 1, \alpha) = \alpha, \tag{5.5}$$

$$p(z_i = -1|y_i = -1, \beta) = \beta, \tag{5.6}$$

and z_i are conditionally independent given $\boldsymbol{y}$, for $i = 1, \ldots, n$. Comparing to the DAG in Figure 5.2 we see that $\boldsymbol{\phi} = (\alpha, \beta)$. To complete our Bayesian framework we need to define a prior for $p(\boldsymbol{\phi}) = p(\alpha, \beta)$. Both of the hyperparameters, α and β, are constrained to lie in $[0, 1]$, and because of the structure of the BC model we will find it convenient to take $\alpha \sim \text{beta}(a, b)$ and $\beta \sim \text{beta}(c, d)$ independently, where a, b, c, d are known constants. It is this combination of likelihood and prior that means the marginal posteriors can be found analytically.

Let $n_{r,s}^{(m)} = \sum_{i=1}^{n} \mathbf{1}[z_i = r, y_i^{(m)} = s]$ for $r, s \in \{-1, 1\}$ and $\text{B}(s, t) = \Gamma(s)\Gamma(t)/\Gamma(s+t)$ be the Beta function for real numbers s and t. Then the posterior for the parameters m, α and β given $\boldsymbol{z}$ is

$$\begin{aligned} p(m, \alpha, \beta|\boldsymbol{z}) &\propto p(\boldsymbol{z}|\boldsymbol{y}^{(m)}, \alpha, \beta)p(\alpha)p(\beta)p(m), \\ &= \alpha^{n_{1,1}^{(m)}}(1-\alpha)^{n_{-1,1}^{(m)}}\beta^{n_{-1,-1}^{(m)}}(1-\beta)^{n_{1,-1}^{(m)}}\frac{\alpha^{a-1}(1-\alpha)^{b-1}}{\text{B}(a,b)}\frac{\beta^{c-1}(1-\beta)^{d-1}}{\text{B}(c,d)}, \\ &\propto \alpha^{n_{1,1}^{(m)}+a-1}(1-\alpha)^{n_{-1,1}^{(m)}+b-1}\beta^{n_{-1,-1}^{(m)}+c-1}(1-\beta)^{n_{1,-1}^{(m)}+d-1}. \end{aligned}$$

The normalising constant can be calculated analytically, let $q(m, \alpha, \beta|\boldsymbol{z})$ be the

unnormalised density, then the normalising constant is

$$\sum_{m=1}^{M}\int_0^1\int_0^1 q(m,\alpha,\beta|\boldsymbol{z})\mathrm{d}\alpha\mathrm{d}\beta = \sum_{m=1}^{M}\mathrm{B}\left(n_{1,1}^{(m)}+a, n_{-1,1}^{(m)}+b\right)\mathrm{B}\left(n_{-1,-1}^{(m)}+c, n_{1,-1}^{(m)}+d\right).$$

To find the marginal posterior for the simulation index, m, we integrate the joint posterior with respect to the likelihood parameters, α and β,

$$\begin{aligned} p(m|\boldsymbol{z}) &= \int_0^1\int_0^1 p(m,\alpha,\beta|\boldsymbol{z})\,\mathrm{d}\alpha\,\mathrm{d}\beta, \\ &= \frac{\mathrm{B}\left(n_{1,1}^{(m)}+a, n_{-1,1}^{(m)}+b\right)\mathrm{B}\left(n_{-1,-1}^{(m)}+c, n_{1,-1}^{(m)}+d\right)}{\sum_{m^\star=1}^{M}\mathrm{B}\left(n_{1,1}^{(m^\star)}+a, n_{-1,1}^{(m^\star)}+b\right)\mathrm{B}\left(n_{-1,-1}^{(m^\star)}+c, n_{1,-1}^{(m^\star)}+d\right)}. \end{aligned} \tag{5.7}$$

The marginal posterior for α and β is found similarly,

$$\begin{aligned} p(\alpha,\beta|\boldsymbol{z}) &= \sum_{m=1}^{M} p(m,\alpha,\beta|\boldsymbol{z}), \\ &= \frac{\sum_{m=1}^{M}\alpha^{n_{1,1}^{(m)}+a-1}(1-\alpha)^{n_{-1,1}^{(m)}+b-1}\beta^{n_{-1,-1}^{(m)}+c-1}(1-\beta)^{n_{1,-1}^{(m)}+d-1}}{\sum_{m=1}^{M}\mathrm{B}\left(n_{1,1}^{(m)}+a, n_{-1,1}^{(m)}+b\right)\mathrm{B}\left(n_{-1,-1}^{(m)}+c, n_{1,-1}^{(m)}+d\right)}. \end{aligned}$$

When the arguments are large the Beta function, $\mathrm{B}(\cdot,\cdot)$, will be very small. In our case the arguments relate to the size of the binary array so will be very large. Therefore computation of the joint and marginal posteriors is complicated because the computer may be unable to represent the small value of the Beta function, so will equate it to zero. This is called *underflow*.

The marginal posterior for the simulation index, m, can be found by computing the ratio $p(m|\boldsymbol{z})/p(m^\star|\boldsymbol{z})$ for $m = 1,\ldots,M$ and some reference simulation $m^\star$. This ratio can be computed because terms cancel in the ratio of the Beta functions. To see this, replace the Beta functions by their Gamma function representation, then $p(m|\boldsymbol{z})/p(m^\star|\boldsymbol{z})$ is the product of ratios of Gamma functions with arguments separated by an integer. Let x be a real number and n be a positive integer, then using $\Gamma(x) = (x-1)\Gamma(x-1)$ we find

$$\frac{\Gamma(x+n)}{\Gamma(x)} = (x+n-1)(x+n-2)\cdots(x+1)x.$$

We cannot compute the joint posterior, $p(m,\alpha,\beta|\boldsymbol{z})$, or the marginal posterior, $p(\alpha,\beta|\boldsymbol{z})$, but conditional on $\boldsymbol{z}$ and m

$$p(\alpha,\beta|\boldsymbol{z},m) = \frac{\alpha^{n_{1,1}^{(m)}+a-1}(1-\alpha)^{n_{-1,1}^{(m)}+b-1}}{\mathrm{B}\left(n_{1,1}^{(m)}+a, n_{-1,1}^{(m)}+b\right)}\frac{\beta^{n_{-1,-1}^{(m)}+c-1}(1-\beta)^{n_{1,-1}^{(m)}+d-1}}{\mathrm{B}\left(n_{-1,-1}^{(m)}+c, n_{1,-1}^{(m)}+d\right)},$$

so $p(\alpha, \beta|\boldsymbol{z}, m) = p(\alpha|\boldsymbol{z}, m)p(\beta|\boldsymbol{z}, m)$, and

$$\alpha|\boldsymbol{z}, m \sim \text{beta}\Big(n_{1,1}^{(m)} + a, n_{-1,1}^{(m)} + b\Big) \quad \text{and}$$
$$\beta|\boldsymbol{z}, m \sim \text{beta}\Big(n_{-1,-1}^{(m)} + c, n_{1,-1}^{(m)} + d\Big).$$

It makes sense that α and β are independent given m because the number of positives and the number of negatives are fixed given m, and α and β correspond to positives and negatives respectively. The expectations and variances can be found from standard formulae and because of independence they are very informative about $p(\alpha, \beta|\boldsymbol{z}, m)$. The distribution $p(\alpha, \beta|\boldsymbol{z}, m)$ is of interest because the overlap for different m indicates whether a simulation index update in a simple MCMC algorithm could be accepted.

Using the identities for conditional expectations and variances we can calculate the marginal posterior expectation and variance of α and β, for example

$$\begin{aligned}
\mathrm{E}\left(\alpha|\boldsymbol{z}\right) &= \sum_{m=1}^{M} \mathrm{E}\left(\alpha|\boldsymbol{z}, m\right) p(m|\boldsymbol{z}) \\
\mathrm{Var}\left(\alpha|\boldsymbol{z}\right) &= \sum_{m=1}^{M} \mathrm{Var}\left(\alpha|\boldsymbol{z}, m\right) p(m|\boldsymbol{z}) \\
&\quad + \sum_{m=1}^{M} \mathrm{E}\left(\alpha|\boldsymbol{z}, m\right)^2 p(m|\boldsymbol{z}) - \left(\sum_{m=1}^{M} \mathrm{E}\left(\alpha|\boldsymbol{z}, m\right) p(m|\boldsymbol{z})\right)^2.
\end{aligned}$$

It is typically difficult to calculate the probability of flooding for each pixel because we need to integrate out all other parameters, but using the BC model

z_i	$y_i^{(m)}$		
	-1	1	
-1	$n^{(m)}_{-1,-1}$	$n^{(m)}_{-1,1}$	$n_{-1,\cdot}$
1	$n^{(m)}_{1,-1}$	$n^{(m)}_{1,1}$	$n_{1,\cdot}$
	$n^{(m)}_{\cdot,-1}$	$n^{(m)}_{\cdot,1}$	n

Table 5.1: Cross-classification counts for the observed data, $\boldsymbol{z}$, and a simulation, $\boldsymbol{y}^{(m)}$, where, for example, $n_{-1,\cdot} = n^{(m)}_{-1,-1} + n^{(m)}_{-1,1}$.

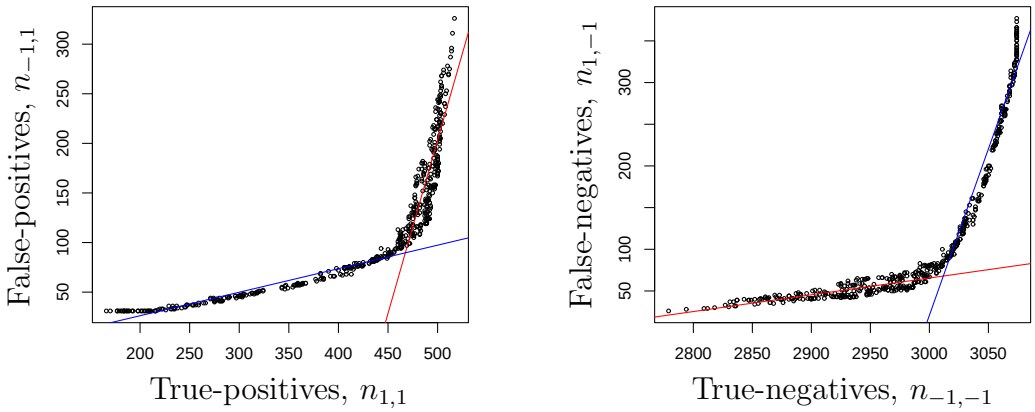
with beta priors this is simple,

$$
\begin{aligned}
&p(z'_i = 1|\boldsymbol{z}) \\
&\quad= \sum_{m=1}^{M} \int_0^1 \int_0^1 p(z'_i = 1|\boldsymbol{y}'^{(m)}, \alpha, \beta) p(m, \alpha, \beta|\boldsymbol{z}) \,\mathrm{d}\alpha \,\mathrm{d}\beta, \\
&\quad= \sum_{m=1}^{M} \int_0^1 \int_0^1 \Big(\alpha \mathbf{1}[y'^{(m)}_i = 1] + (1-\beta)\mathbf{1}[y'^{(m)}_i = -1]\Big) p(m|\boldsymbol{z}) p(\alpha, \beta|m, \boldsymbol{z}) \,\mathrm{d}\alpha \,\mathrm{d}\beta \\
&\quad= \sum_{m=1}^{M} \Big(\mathrm{E}\left(\alpha|m, \boldsymbol{z}\right) \mathbf{1}[y'^{(m)}_i = 1] + \mathrm{E}\left(1-\beta|m, \boldsymbol{z}\right) \mathbf{1}[y'^{(m)}_i = -1]\Big) p(m|\boldsymbol{z}) \\
&\quad= \sum_{m=1}^{M} \Bigg(\frac{n^{(m)}_{1,1} + a}{n^{(m)}_{1,1} + a + n^{(m)}_{-1,1} + b} \mathbf{1}[y'^{(m)}_i = 1] \\
&\qquad\qquad + \left(1 - \frac{n^{(m)}_{-1,-1} + c}{n^{(m)}_{-1,-1} + c + n^{(m)}_{1,-1} + d}\right) \mathbf{1}[y'^{(m)}_i = -1]\Bigg) p(m|\boldsymbol{z}). \qquad (5.8)
\end{aligned}
$$

Table 5.1 shows the cross-classification counts for the observed data, $\boldsymbol{z}$, and a simulation, $\boldsymbol{y}^{(m)}$. We have only one observation so the number of pixels observed wet, $n_{1,\cdot}$, and observed dry, $n_{-1,\cdot}$, do not change. The only way the counts $n^{(m)}_{-1,-1}$, $n^{(m)}_{-1,1}$, $n^{(m)}_{1,-1}$ and $n^{(m)}_{1,1}$ can change is if the simulation index, m, changes.

The values of α and β determine how false-positives and false-negatives are penalised in the BC model. Suppose there are more true-positives in simulation $\boldsymbol{y}^{(2)}$ than simulation $\boldsymbol{y}^{(1)}$, $n^{(2)}_{1,1} > n^{(1)}_{1,1}$, then we expect there to be more false-positives in $\boldsymbol{y}^{(2)}$ than $\boldsymbol{y}^{(1)}$, $n^{(2)}_{-1,1} > n^{(1)}_{-1,1}$ since this is true empirically for the M runs, see Figure 5.3. Let the ratio of the increase in false-positives to the increase in true-positives be s, so $n^{(2)}_{-1,1} - n^{(1)}_{-1,1} = s(n^{(2)}_{1,1} - n^{(1)}_{1,1})$, and suppose α and β are fixed, then

$$
\frac{p(\boldsymbol{y}^{(2)}|\boldsymbol{z})}{p(\boldsymbol{y}^{(1)}|\boldsymbol{z})} = \left(\frac{\alpha}{1-\beta}\right)^{n^{(2)}_{1,1} - n^{(1)}_{1,1}} \left(\frac{1-\alpha}{\beta}\right)^{s(n^{(2)}_{1,1} - n^{(1)}_{1,1})}. \qquad (5.9)
$$



(a) False-positives versus true-positives. Red line gradient is 3.495 and blue line gradient is 0.2371.

(b) False-negatives versus true-negatives. Red line gradient is 0.2011 and blue line gradient is 3.993.

Figure 5.3: Plots showing the relationship between falses and trues for the Buscot dataset.

We want to identify the range of s for which $p(\boldsymbol{y}^{(2)}|\boldsymbol{z})/p(\boldsymbol{y}^{(1)}|\boldsymbol{z}) < 1$ for a given α and β. From Equation (5.9) we find

$$\frac{p(\boldsymbol{y}^{(2)}|\boldsymbol{z})}{p(\boldsymbol{y}^{(1)}|\boldsymbol{z})} < 1 \text{ if } \begin{cases} s > s^\star \text{ and } \beta > 1-\alpha \\ s < s^\star \text{ and } \beta < 1-\alpha, \end{cases}$$

where

$$s^\star = \frac{\log(\alpha) - \log(1-\beta)}{\log(\beta) - \log(1-\alpha)}. \tag{5.10}$$

For the special case $\alpha = \beta > 0.5$, the posterior ratio $p(\boldsymbol{y}^{(2)}|\boldsymbol{z})/p(\boldsymbol{y}^{(1)}|\boldsymbol{z}) < 1$ if $s > 1$. As an example, take $\alpha = 0.9$ and $\beta = 0.5$ then $s^\star = 0.365$, so if the ratio of the increase in false-positives to the increase in true-positives, s, is greater than 0.365 then the posterior probability of $\boldsymbol{y}^{(2)}$ is less than that of $\boldsymbol{y}^{(1)}$.

For the Buscot dataset (see Section 2.4) the rate of increase of false-positives with true-positives, see Figure 5.3(a), changes significantly at a point corresponding to the optimum simulation, either side of this point the rate is almost constant. This is because past the optimum simulation it is probable that an increase in the number of positives, $n_{\cdot,1}$, will result in more false-positives, $n_{-1,1}$, than true-positives, $n_{1,1}$.

m	$n_{1,1}^{(m)}$	$n_{-1,1}^{(m)}$	$n_{-1,-1}^{(m)}$	$n_{1,-1}^{(m)}$
110	482	108	2997	61
91	497	156	2949	46
349	288	45	3060	255

Table 5.2: Cross-classification counts for the simulations shown in Figure 5.4.

5.3 Buscot Example

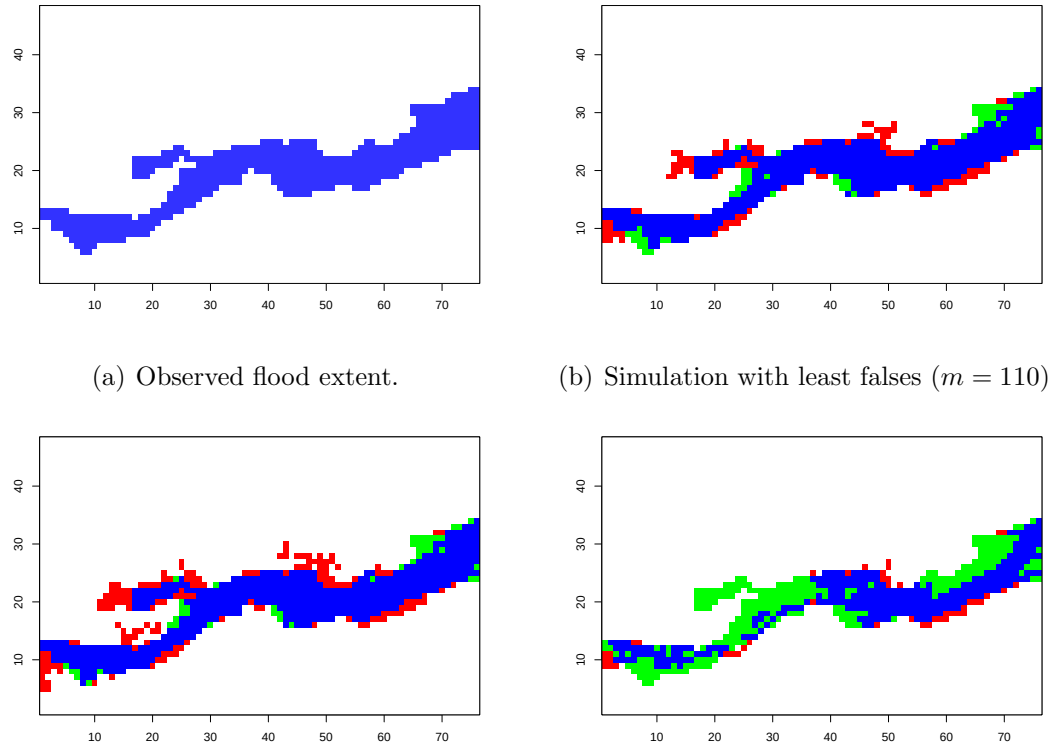
In this section we use the Buscot dataset introduced in Section 2.4 to illustrate our Bayesian framework for calibration and calibrated prediction using a BC model with beta priors. There is only one observation of flood extent, $\boldsymbol{z}$, so we are unable to validate calibrated predictions of an independent event. Instead we use the same event for calibration and prediction, $\boldsymbol{x}' = \boldsymbol{x}$ and $\boldsymbol{y}'^{(m)} = \boldsymbol{y}^{(m)}$ for $m = 1, \ldots, M$. Consequently we are unable to test the stationarity of the calibration inputs, $\boldsymbol{\theta}$, and likelihood parameters, $\boldsymbol{\phi} = (\alpha, \beta)$, between events of different magnitudes.

Figure 5.4 shows the observed data and three simulations from the Buscot dataset. We chose the simulation with the least falses, $m = 110$, one with many false-positives, $m = 91$, and one with many false-negatives, $m = 349$. In the examples which follow the results corresponding to these simulations will be labelled. The cross-classification counts for these simulations are given in Table 5.2.

5.3.1 Example Using $\alpha, \beta \sim \mathcal{U}[0, 1]$

Figure 5.5 shows the results of calibration and calibrated prediction using $\alpha, \beta \sim$ beta$(1, 1) \equiv \mathcal{U}[0, 1]$. The marginal posterior for the simulation index, $p(m|\boldsymbol{z})$, is nonnegligible for very few values of m, and of these values it is much larger for the marginal posterior mode, say $m^\star$, than the others, $p(m^\star|\boldsymbol{z}) \gg p(m|\boldsymbol{z})$ for $m \neq m^\star$, see Figure 5.5(b). In other words, the marginal posterior discriminates a lot between simulations because falses are heavily penalised. Consequently our calibrated prediction is dominated by a single simulation, $\boldsymbol{y}'^{(m^\star)}$.

For the simulation corresponding to the marginal posterior mode, $\boldsymbol{y}^{(m^\star)}$, the number of falses is small but is not the minimum, i.e. $m^\star \neq 110$. Figure 5.5(d) shows the marginal posterior expectations for α and β (grey cross) which give an



(a) Observed flood extent.

(b) Simulation with least falses ($m = 110$).

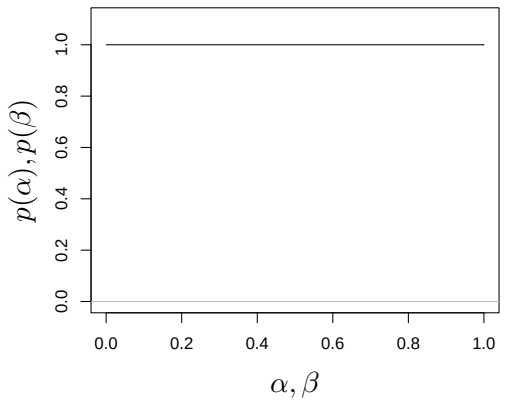
(c) Simulation with many false-positives ($m = 91$).

(d) Simulation with many false-negatives ($m = 349$).

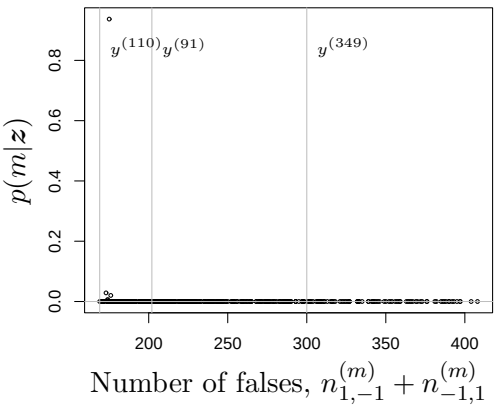
Figure 5.4: Observed data and three simulations from the Buscot dataset. For Figures 5.4(b) to 5.4(d) true-negatives are white, false-negatives are green, true-positives are blue, and false-positives are red.

insight into why $m^\star \neq 110$. From Figure 5.5(d) $\mathrm{E}(\alpha|\boldsymbol{z}) < \mathrm{E}(\beta|\boldsymbol{z})$, taking these values for α and β in the BC model we find true-negatives are rewarded more than true-positives, but false-negatives are penalised more than false-positives ($1 - \mathrm{E}(\beta|\boldsymbol{z}) < 1 - \mathrm{E}(\alpha|\boldsymbol{z})$). The effect of $\alpha \neq \beta$ was investigated at the end of Section 5.2. For our example $\alpha \doteq \mathrm{E}(\alpha|\boldsymbol{z}) = 0.798$ and $\beta \doteq \mathrm{E}(\beta|\boldsymbol{z}) = 0.983$, on substitution into Equation (5.10) we find that the increase in false-positives from $\boldsymbol{y}^{(1)}$ to $\boldsymbol{y}^{(2)}$ would need to be more than 2.432 times the increase in true-positives to obtain $p(\boldsymbol{y}^{(2)}|\boldsymbol{z}) < p(\boldsymbol{y}^{(1)}|\boldsymbol{z})$.

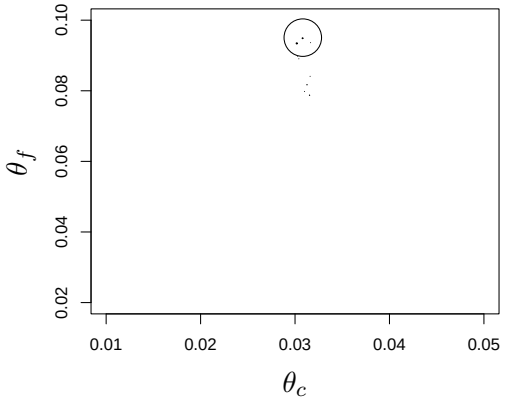
The BC model parameters α and β encode the uncertainty around the flood extent boundary and away from it, although in reality this uncertainty is very different. The number of trues is much greater than the number of falses for all simulations because falses typically only occur around the flood extent boundary.



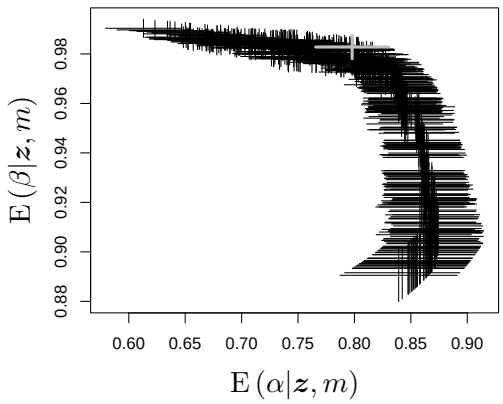
(a) Prior.



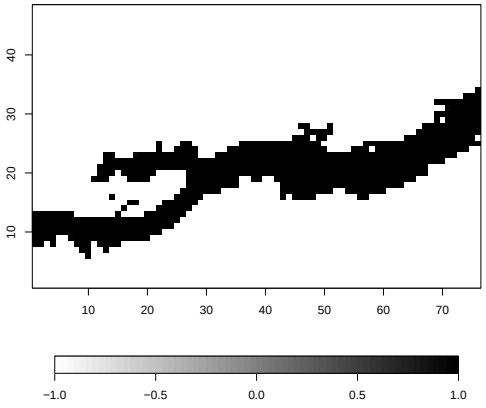
(b) Marginal posterior for m.



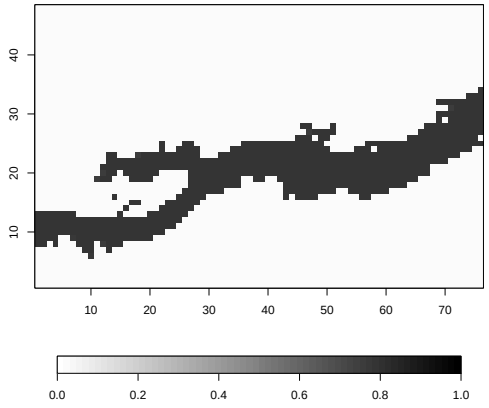
(c) Posterior for the calibration inputs, $p(\boldsymbol{\theta}|\boldsymbol{z})$.



(d) Expectations for α and β given $\boldsymbol{z}$ and m.

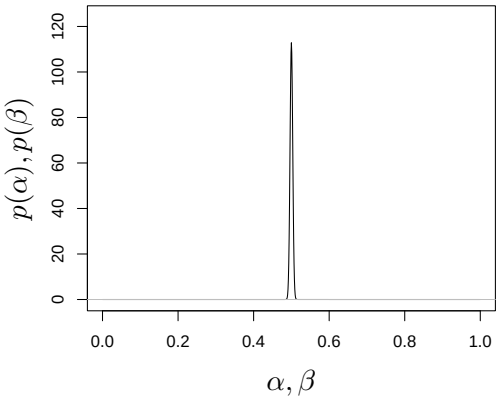


(e) $\mathrm{E}(\boldsymbol{y}'|\boldsymbol{z})$.

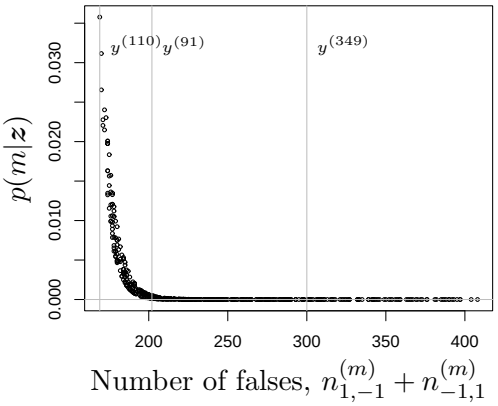


(f) $p(z_i' = 1|\boldsymbol{z})$.

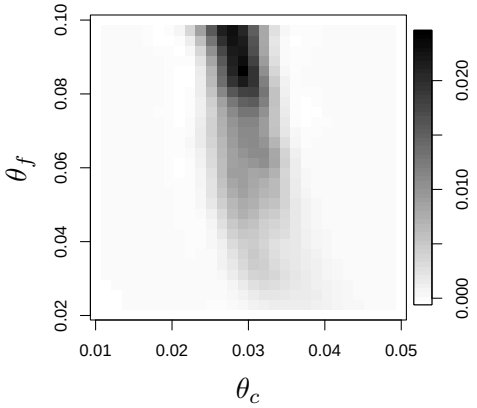
Figure 5.5: Results of calibration and calibrated prediction using the BC model with priors $\alpha, \beta \sim \mathrm{beta}(1,1) \equiv \mathcal{U}[0,1]$. In Figure 5.5(c) the posterior for $\boldsymbol{\theta}$ is represented by circles centred at $\boldsymbol{\theta}^{(m)}$ with radius proportional to $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$. In Figure 5.5(d) the black crosses are centred at $(\mathrm{E}(\alpha|\boldsymbol{z}, m), \mathrm{E}(\beta|\boldsymbol{z}, m))$ with horizontal and vertical bars of length $4\mathrm{Sd}(\alpha|\boldsymbol{z}, m)$ and $4\mathrm{Sd}(\beta|\boldsymbol{z}, m)$. The grey cross is centred at $(\mathrm{E}(\alpha|\boldsymbol{z}), \mathrm{E}(\beta|\boldsymbol{z}))$ with horizontal and vertical bars of length $4\mathrm{Sd}(\alpha|\boldsymbol{z})$ and $4\mathrm{Sd}(\beta|\boldsymbol{z})$.



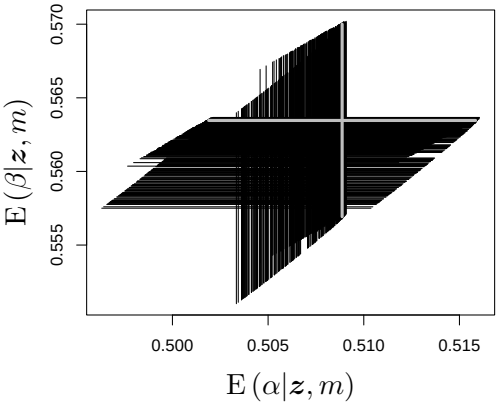
(a) Prior.



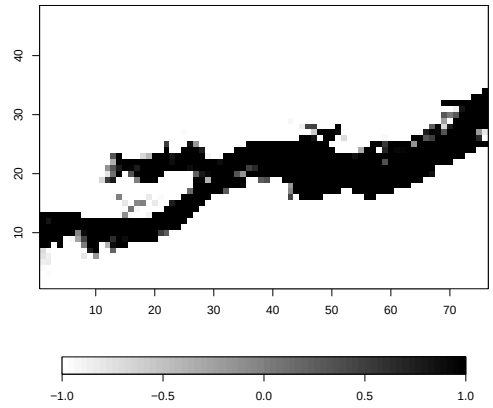
(b) Marginal posterior for m.



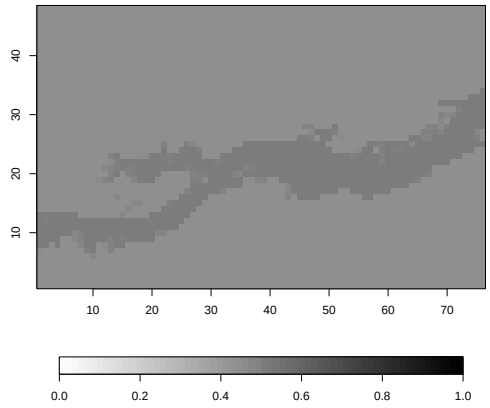
(c) Posterior for $\boldsymbol{\theta}$.



(d) Expectations for α and β given $\boldsymbol{z}$ and m.



(e) $\mathrm{E}(\boldsymbol{y}'|\boldsymbol{z})$.



(f) $p(z'_i = 1|\boldsymbol{z})$.

Figure 5.6: Results of calibration and calibrated prediction using the BC model with priors $\alpha, \beta \sim \text{beta}(10000, 10000)$. In Figure 5.6(c) $p(\boldsymbol{\theta}|\boldsymbol{z})$ is approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. In Figure 5.6(d) the black crosses are centred at $(\mathrm{E}(\alpha|\boldsymbol{z}, m), \mathrm{E}(\beta|\boldsymbol{z}, m))$ with horizontal and vertical bars of length $4\mathrm{Sd}(\alpha|\boldsymbol{z}, m)$ and $4\mathrm{Sd}(\beta|\boldsymbol{z}, m)$. The grey cross is centred at $(\mathrm{E}(\alpha|\boldsymbol{z}), \mathrm{E}(\beta|\boldsymbol{z}))$ with horizontal and vertical bars of length $4\mathrm{Sd}(\alpha|\boldsymbol{z})$ and $4\mathrm{Sd}(\beta|\boldsymbol{z})$.

For these reasons the marginal posterior for the BC model parameters, $p(\alpha, \beta|\boldsymbol{z})$, is only nonnegligible for very large values of α and β. Consequently falses are very heavily penalised. The reason $\mathrm{E}(\beta|\boldsymbol{z}) > \mathrm{E}(\alpha|\boldsymbol{z})$ is that the ratio of true-negatives to false-negatives is larger than the ratio of true-positives to false-positives.

Figure 5.5(e) shows $\mathrm{E}(\boldsymbol{y}'|\boldsymbol{z})$ which is included for comparison to GLUE where it is described as a map of flood probability (compare to Figure 4.1). The calibrated prediction, $p(z_i' = 1|\boldsymbol{z})$ for $i = 1, \ldots, n$, is shown in Figure 5.5(f). This is our prediction of the probability of flooding in a future event having calibrated the simulator and likelihood using an observation of flood extent. Because the BC model is homogeneous and $p(m^\star|\boldsymbol{z}) \gg p(m|\boldsymbol{z})$ for $m \neq m^\star$, the probability of flooding seemingly only takes two values, in particular the uncertainty is no larger near the boundary.

For a general prior $p(\alpha, \beta)$ the posterior $p(m, \alpha, \beta|\boldsymbol{z})$ will not be available analytically, but we may be able to generate a sample from the posterior using MCMC. Figure 5.5(d) gives an insight into the potential of MCMC for generating a sample from the posterior. Consider the following Metropolis-Hastings update for m holding α and β fixed: propose m' from a discrete uniform on $1, \ldots, m-1, m+1, \ldots, M$ and accept this proposal with probability

$$\min\left(1, \frac{p(\alpha, \beta|\boldsymbol{z}, m')p(m'|\boldsymbol{z})}{p(\alpha, \beta|\boldsymbol{z}, m)p(m|\boldsymbol{z})}\right).$$

Suppose $p(\alpha, \beta|\boldsymbol{z}, m)$ is large then from Figure 5.5(d) $p(\alpha, \beta|\boldsymbol{z}, m')/p(\alpha, \beta|\boldsymbol{z}, m)$ will probably be very small. Also Figure 5.5(d) suggests that a more intelligent proposal, $q(m'|m)$, would increase the probability of acceptance. For the BC model with beta priors this issue does not arise, but it motivates some of the problems we will encounter when we come to consider more complicated likelihoods.

For calibration we expect many simulation indexes to have nonnegligible marginal posteriors because the simulations are very similar. For calibrated prediction we expect more than one simulation to be important because in different parts of the array different simulations may be closer to the observed data. We have assumed stationarity of the calibration inputs, $\boldsymbol{\theta}$, and the likelihood parameters,

$\boldsymbol{\phi} = (\alpha, \beta)$, between the events of interest to make calibration and calibrated prediction feasible, but in practice they will not be stationary so there is a danger of over-fitting the model to the calibration event.

5.3.2 Example Using $\alpha, \beta \sim \text{beta}(10000, 10000)$

Figure 5.6 shows the results of calibration and calibrated prediction using $\alpha, \beta \sim \text{beta}(10000, 10000)$. This choice of prior ensures that α and β are close to 0.5 (see Figure 5.6(a)), and therefore falses cannot be so heavily penalised. For example, suppose $\boldsymbol{y}^{(2)}$ can be obtained from $\boldsymbol{y}^{(1)}$ by changing one true-positive to a false-negative, then taking $\alpha = \beta = 0.6$ we find $p(\boldsymbol{y}^{(2)}|\boldsymbol{z})/p(\boldsymbol{y}^{(1)}|\boldsymbol{z}) = (1-\beta)/\alpha = 0.667$. In Figure 5.6(b) we see that the marginal posterior, $p(m|\boldsymbol{z})$, is nonnegligible for many simulation indexes, and this transforms into a flatter posterior for the calibration inputs, $p(\boldsymbol{\theta}|\boldsymbol{z})$ (see Figure 5.6(c)). The marginal posterior expectations for the BC model parameters are closer than in the first example, now $\text{E}(\alpha|\boldsymbol{z}) = 0.509$ and $\text{E}(\beta|\boldsymbol{z}) = 0.563$. Substituting $\alpha = \text{E}(\alpha|\boldsymbol{z}) = 0.509$ and $\beta = \text{E}(\beta|\boldsymbol{z}) = 0.563$ into Equation 5.10 we find $s^\star = 1.114$, so false-negatives are penalised only a little more than false-positives. Accordingly the posterior mode for the simulation index corresponds to the simulation with the least falses, $m^\star = 110$. The calibrated prediction is very uncertain over the whole array because α and β are constrained to be close to 0.5 (see Figure 5.6(f)). This is undesirable because we are very certain about the prediction within the channel and on the floodplain away from the flood boundary.

In conclusion, the BC model is useful for illustrating our Bayesian framework for calibration and calibrated prediction because it is so simple, but we are unable to obtain the desired calibration and calibrated prediction results simultaneously using this simple model. In Chapters 6, 7 and 8 we consider various extensions of the BC model which represent spatial dependence, heterogeneity and blur.

In this chapter we have described a Bayesian framework for the calibration of flood inundation simulators. To illustrate the framework we used the BC model for the likelihood, which lead to analytical results for calibration and calibrated

prediction. However, the BC model does not represent the data accurately because it does not account for blur, spatial dependence or heterogeneity. In the next chapter we consider the Ising model for the likelihood, which accounts for blur and spatial dependence.

Chapter 6

The Ising Model

We begin by introducing the Ising model for spatially distributed binary valued variables. Then we extend this model to regression on a binary image (the simulator output), which improves on the binary channel model by representing blur and spatial dependence. We describe how calibration and calibrated prediction is performed using the Ising model and in doing so identify that an intractable normalising constant must be estimated. We review importance, bridge and path sampling methods for the estimation of normalising constants. Then we thoroughly investigate the potential for path sampling in our application: testing the accuracy against exact computations; extending the methodology to paths between images and model parameterisations; and introducing a method for sampling over areas. When none of these strategies prove to be computationally efficient enough we discuss numerous approximations to the path sampling identity, including Tukey's transformation for additivity.

6.1 The Ising Model

The Ising model was devised in 1924 by Ernst Ising as a model for ferromagnetism (Ising, 1925). The classical construction is in terms of joint probabilities but here we present a conditional probability approach which Besag (1974) argues is a more natural way to define the Ising model. In conditional probability approaches to spatial processes, there are strong constraints on the structure of the conditional probabilities to guarantee that a joint probability exists. Fortuitously it is these

very constraints that mean the Ising model is necessarily generated when the variables are binary valued, the set of sites is a regular lattice, the only interactions are between nearest neighbour pairs, and the parameters are homogeneous.

We start with the class of *pixel-based models*, in which the set of pixel sites, Λ, and the set of values that a pixel can take, χ, are quite general. The sites, although quite general, are fixed and it is only the value of the pixels we are interested in modelling. For convenience, the "distribution of a pixel" will be taken to mean the distribution of the value of that pixel.

It is easier to consider the distribution of a pixel given all other pixels than the joint distribution of all the pixels; this is what makes the conditional probability approach so appealing (Hurn *et al.*, 2003).

A *random field* is a collection of random variables $\boldsymbol{z} = \{z_i \in \chi : i \in \Lambda\}$. We define a binary relation on Λ, denoted by $\sim$. It is required to be symmetric, and if $i \sim j$ we say i and j are neighbours. The random field $\boldsymbol{z}$ is a *Markov random field* if the distribution of one pixel given all others depends only on its neighbours

$$p\left(z_i|\boldsymbol{z}_{-i}\right) = p\left(z_i|\boldsymbol{z}_{\partial i}\right),$$

where ∂i denotes the neighbours of i and because the neighbourhood relation is symmetric $j \in \partial i \Leftrightarrow i \in \partial j$.

The specification of a neighbourhood for each site $i \in \Lambda$ defines a class of valid stochastic schemes (Besag, 1974). We must identify this class to ensure that the full conditionals we define will give rise to a legitimate joint density. Given the full conditionals for each pixel, we need only consider the density relative to some reference configuration $\boldsymbol{z}^\star$ because the joint density must sum to 1. Assuming that $p\left(\boldsymbol{z}\right) > 0.0$ for all realisations $\boldsymbol{z}$, this ratio can be written

$$\frac{p\left(\boldsymbol{z}\right)}{p\left(\boldsymbol{z}^\star\right)} = \prod_{i=1}^{n} \frac{p\left(z_i|z_1,\ldots,z_{i-1},z_{i+1}^\star,\ldots,z_n^\star\right)}{p\left(z_i^\star|z_1,\ldots,z_{i-1},z_{i+1}^\star,\ldots,z_n^\star\right)},$$

(see Section 8.2.1 for a proof). The labelling of the sites is arbitrary so there are many alternative such factorisations of $p\left(\boldsymbol{z}\right)/p\left(\boldsymbol{z}^\star\right)$. Clearly the value of $p\left(\boldsymbol{z}\right)/p\left(\boldsymbol{z}^\star\right)$ should be invariant to which factorisation is used, which puts severe restrictions on the functional form of the full conditionals. Also the joint density $p(\boldsymbol{z})$

should be invariant to the reference configuration, $\boldsymbol{z}^{\star}$. The Hammersley-Clifford theorem determines the form that the full conditionals must take to respect these consistency conditions (Besag, 1974).

Theorem (Hammersley-Clifford). *Let $p(\cdot)$ be a distribution with $p(\boldsymbol{z}) > 0$, $\forall \boldsymbol{z} \in \chi^{|\Lambda|}$. Define a clique to be a subset of sites in which all members are neighbours of all others. Then $\boldsymbol{z}$ is a Markov random field if, and only if, the joint density takes the form*

$$p(\boldsymbol{z}) = \frac{1}{Z} \exp\left(\sum_{C \in \mathcal{C}} \Phi_C(\boldsymbol{z}_C)\right) \tag{6.1}$$

where $\mathcal{C}$ is the set of all cliques, the potential functions, *$\{\Phi_C\}$, may be chosen arbitrarily, and*

$$Z = \sum_{\boldsymbol{z}} \exp\left(\sum_{C \in \mathcal{C}} \Phi_C(\boldsymbol{z}_C)\right) \tag{6.2}$$

is the normalising constant.

A simple proof of this theorem can be found in Besag (1974). Z is also known as the *partition function*. This theorem, in addition to giving the most general form for the full conditionals, suggests a way of defining them implicitly through the potential functions.

If a clique contains a large number of sites it may be hard to define the potential function. Besag (1972) treats a subclass of problems, called *auto-models*, in which the cliques may contain at most two sites and the conditional probability associated with each site comes from the exponential family. When the pixels are binary valued, $\boldsymbol{z} \in \{-1, 1\}^{|\Lambda|}$, the model is called the *auto-logistic model*.

Assuming pairwise only interactions, the potential for higher order cliques is 0 and the cliques of interest are individual sites and neighbour pairs. Let $\boldsymbol{\mu} = \left(\mu_1, \ldots, \mu_{|\Lambda|}\right)$ and $\Delta = (\delta_{ij})$ be a vector and a matrix of real parameters, then define

$$\Phi_i(z_i \,|\boldsymbol{\mu}) = \mu_i z_i, \tag{6.3}$$

to be the potential function for individual sites, i, and

$$\Phi_{i \sim j}(\{z_i, z_j\} \,|\Delta) = \delta_{ij} \mathbf{1}[z_i = z_j] \tag{6.4}$$

to be the potential function for neighbour pairs, $i \sim j$. Increasing μ_i increases the probability that $z_i = 1$, and increasing δ_{ij} increases the probability that $z_i = z_j$. On substituting these potential functions into Equation (6.1) we find the unnormalised density for the auto-logistic model is

$$q(\boldsymbol{z}\,|\boldsymbol{\mu}, \Delta) = \exp\left(\sum_{i\in\Lambda} \mu_i z_i + \sum_{i\sim j} \delta_{ij} \mathbf{1}\left[z_i = z_j\right]\right), \tag{6.5}$$

where $\sum_{i\sim j}$ is the sum over all neighbour pairs. However, there are many different but equivalent parameterisations of the auto-logistic model.

Auto-models are particularly appropriate for nearest-neighbour lattice based processes, such as we have here, where it is natural to consider the cliques to be either individual sites, or pairs of North–South or East–West nearest-neighbours. Now $i \sim j$ says i and j are North–South or East–West nearest-neighbours. The *Ising model* is the simplest form of the auto-logistic model in which the set of sites, Λ, is a $r \times c$ regular lattice, $\mu_i = \mu$ for all sites i, and $\delta_{ij} = \delta$ for all pairs of North–South or East–West nearest-neighbours $i \sim j$. Denote the number of sites $(r \times c)$ by n, then the unnormalised density for the Ising model is

$$q(\boldsymbol{z}|\mu, \delta) = \exp\left(\mu \sum_{i=1}^{n} z_i + \delta \sum_{i\sim j} \mathbf{1}\left[z_i = z_j\right]\right), \tag{6.6}$$

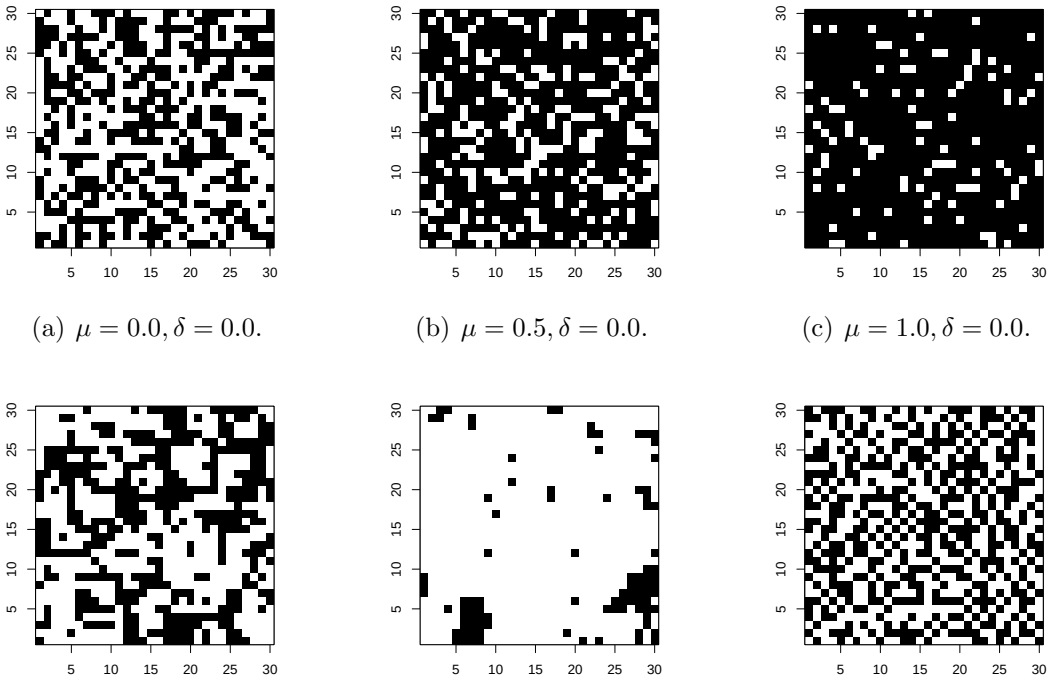
where the *trend parameter* μ controls the overall level of the image, and the *clustering parameter* δ controls the clustering of like-coloured sites. Conveniently, $\mu = 0$ corresponds to no bias for one pixel value over another, and $\delta = 0$ to independence of pixels.

An important feature of the conditional probability approach to spatial processes is that the full conditionals are simple to obtain for Gibbs sampling Markov chain Monte Carlo. For the Ising model the full conditional for z_i is

$$p(z_i = 1|\boldsymbol{z}_{-i}, \mu, \delta) = \left(1 + \exp\left(-2\mu - \delta \sum_{j\in\partial i} z_j\right)\right)^{-1} \tag{6.7}$$

where ∂i is the set of four nearest neighbours of the site i.

Figure 6.1 shows a selection of realisations from the Ising model on a 30×30 lattice for different values of the trend and clustering parameters. The realisations



(a) $\mu = 0.0, \delta = 0.0$. (b) $\mu = 0.5, \delta = 0.0$. (c) $\mu = 1.0, \delta = 0.0$.

(d) $\mu = 0.0, \delta = 0.5$. (e) $\mu = 0.0, \delta = 1.0$. (f) $\mu = 0.5, \delta = -0.5$.

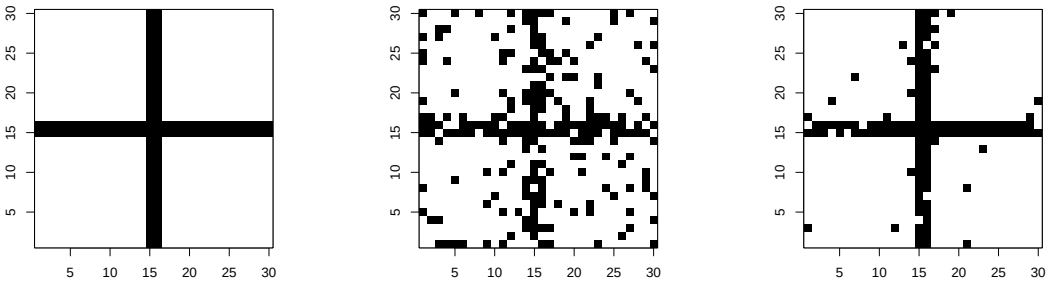
Figure 6.1: Realisations from the Ising model on a 30×30 lattice, using various values of the trend and clustering parameters. Black and white correspond to pixel values of 1 and -1 respectively.

were obtained using Gibbs sampling MCMC with the full conditionals from Equation (6.7). The whole image was updated by sequentially updating each pixel, this was repeated 100000 times to remove the effect of the initial conditions.

We have purposely focused on the Ising model because of its simple interpretation and mathematics. For the flood inundation problem we expect heterogeneity to be a necessary feature of the likelihood (see Section 5.3), but this would dramatically increase the computational expense of posterior inference, and as we shall see the simple Ising model is already too computationally demanding.

6.2 The Ising Model with Regression on a Binary Image

For the Bayesian framework described in Chapter 5 we need a model for $p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{\phi})$, where the observed data, $\boldsymbol{z}$, and the simulator output, $\boldsymbol{y}$, are binary images. In



(a) The covariate image $\boldsymbol{y}$. (b) $\mu = 0.0, \delta = 0.0, \gamma = 0.5$. (c) $\mu = 0.0, \delta = 0.0, \gamma = 1.0$.

Figure 6.2: Data simulated from the Ising model with regression on the 30×30 binary image $\boldsymbol{y}$ shown in Figure 6.2(a). Black and white correspond to pixel values of 1 and -1 respectively.

order to be used in this context the Ising model needs to be extended to represent regression on a binary image. This can be done by augmenting the single site potential function in Equation (6.3) with a term expressing the dependence of $\boldsymbol{z}$ on $\boldsymbol{y}$,

$$\Phi_i\left(z_i \,|\boldsymbol{y}, \mu, \gamma\right) = \mu z_i + \gamma \sum_{k \in \nu_i} \mathbf{1}\left[z_i = y_k\right]$$

where ν_i is the set of sites of covariates of z_i in $\boldsymbol{y}$ and γ is the *regression parameter*. We take ν_i to be the site i and its four nearest neighbours. Using this together with Equations (6.4) and (6.1), we find the unnormalised density for the *Ising model with regression on a binary image* is

$$q\left(\boldsymbol{z} \,|\boldsymbol{y}, \boldsymbol{\phi}\right) = \exp\left(\mu \sum_{i=1}^{n} z_i + \delta \sum_{i \sim j} \mathbf{1}\left[z_i = z_j\right] + \gamma \sum_{i=1}^{n} \sum_{k \in \nu_i} \mathbf{1}\left[z_i = y_k\right]\right), \qquad (6.8)$$

where $\boldsymbol{\phi} = (\mu, \delta, \gamma)$.

This is not the most general formulation of the Ising model for binary $\boldsymbol{z}$ given binary $\boldsymbol{y}$, because the term expressing dependence of $\boldsymbol{z}$ on $\boldsymbol{y}$ need not be constrained to take this form. If we consider the wider class of auto-logistic models the potential functions may be different for each clique, and if we consider the even wider class of Markov random fields we may define more cliques.

Our model (see Equation 6.8) is related to a joint Ising model for $\boldsymbol{y}$ and $\boldsymbol{z}$, for which we would have to define potential functions for the single sites and neighbour pairs in $\boldsymbol{y}$ and $\boldsymbol{z}$, and for the cliques between sites in $\boldsymbol{y}$ and sites in $\boldsymbol{z}$. In an Ising

model for the distribution of $\boldsymbol{z}$ given $\boldsymbol{y}$, the potential functions for single sites and neighbour pairs in $\boldsymbol{y}$ are not required, and the potential function for the cliques between sites in $\boldsymbol{y}$ and sites in $\boldsymbol{z}$ becomes part of the potential functions for cliques in $\boldsymbol{z}$.

In Figure 6.2 an example $\boldsymbol{y}$ is given together with realisations of the model when $\gamma = 0.5$ and 1.0. The regression parameter γ determines the dependence of the observed data $\boldsymbol{z}$ on the simulator output $\boldsymbol{y}$ and $\gamma = 0$ means $\boldsymbol{z}$ is independent of $\boldsymbol{y}$.

To find the normalised density $p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{\phi})$ from Equation (6.8) we must calculate the normalising constant, $Z(\boldsymbol{y}, \boldsymbol{\phi})$, which is the sum over 2^n terms (being the number of possible configurations of a binary image of size n),

$$\sum_{\boldsymbol{z}\in\{-1,1\}^n} \exp\left(\mu \sum_{i=1}^{n} z_i + \delta \sum_{i\sim j} \mathbf{1}\left[z_i = z_j\right] + \gamma \sum_{i=1}^{n} \sum_{k\in\nu_i} \mathbf{1}\left[z_i = y_k\right]\right).$$

For the Buscot dataset $n = 48 \times 76 = 3648$ and there is no way this summation can be computed directly when $\delta \neq 0$ (see Section 6.4.1). The calculation of the normalising constant of the Ising model is a well known problem and much research has gone into either avoiding the need for or approximating this quantity. In the next section we discuss the implications of this problem for calibration and calibrated prediction.

6.2.1 Posterior, Calibration and Calibrated Prediction

The posterior distribution for the simulation index parameter m and the Ising model parameters $\boldsymbol{\phi}$ is

$$\begin{aligned} p(m, \boldsymbol{\phi}|\boldsymbol{z}) &\propto \sum_{\boldsymbol{y}} p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{\phi}) p(\boldsymbol{y}|m) p(\boldsymbol{\phi}) p(m) \\ &\propto \frac{q(\boldsymbol{z}|\boldsymbol{y}^{(m)}, \boldsymbol{\phi})}{Z(\boldsymbol{y}^{(m)}, \boldsymbol{\phi})} p(\boldsymbol{\phi}), \end{aligned} \tag{6.9}$$

where $p(m) = 1/M$, $p(\boldsymbol{y}|m) = \mathbf{1}[\boldsymbol{y} = \boldsymbol{y}^{(m)}]$ and $p(\boldsymbol{y}'|m) = \mathbf{1}[\boldsymbol{y}' = \boldsymbol{y}'^{(m)}]$ as in Section 5.1. Note the presence of the (likelihood) normalising constant which does not cancel in the ratio

$$\frac{p(m', \boldsymbol{\phi}'|\boldsymbol{z})}{p(m, \boldsymbol{\phi}|\boldsymbol{z})} = \frac{q(\boldsymbol{z}|\boldsymbol{y}^{(m')}, \boldsymbol{\phi}')}{q(\boldsymbol{z}|\boldsymbol{y}^{(m)}, \boldsymbol{\phi})} \frac{Z(\boldsymbol{y}^{(m)}, \boldsymbol{\phi})}{Z(\boldsymbol{y}^{(m')}, \boldsymbol{\phi}')} \frac{p(\boldsymbol{\phi}')}{p(\boldsymbol{\phi})}, \tag{6.10}$$

so we cannot calculate the posterior for m and $\boldsymbol{\phi}$ exactly. We will see when we come to Section 6.3 that the ratio of normalising constants can be estimated more directly than the normalising constant itself.

The marginal posterior for m is obtained by integrating over $\boldsymbol{\phi}$,

$$p(m|\boldsymbol{z}) \propto \int \frac{q(\boldsymbol{z}|\boldsymbol{y}^{(m)}, \boldsymbol{\phi})}{Z(\boldsymbol{y}^{(m)}, \boldsymbol{\phi})} p(\boldsymbol{\phi}) \, \mathrm{d}\boldsymbol{\phi}, \tag{6.11}$$

in this case the normalising constant is again present but taking the ratio does not lead to a ratio of normalising constants. This means we will have to estimate the normalising constant and not just the ratio.

The calibrated predictions are

$$\begin{aligned} p(z'_i = 1|\boldsymbol{z}) &= \sum_{m=1}^{M} \int p(z'_i = 1|\boldsymbol{y}'^{(m)}, \boldsymbol{\phi}) p(m, \boldsymbol{\phi}|\boldsymbol{z}) \, \mathrm{d}\boldsymbol{\phi} \\ &\propto \sum_{m=1}^{M} \int \frac{\sum_{\boldsymbol{z}'_{-i}} q(z'_i = 1, \boldsymbol{z}'_{-i}|\boldsymbol{y}'^{(m)}, \boldsymbol{\phi})}{Z(\boldsymbol{y}'^{(m)}, \boldsymbol{\phi})} \frac{q(\boldsymbol{z}|\boldsymbol{y}^{(m)}, \boldsymbol{\phi})}{Z(\boldsymbol{y}^{(m)}, \boldsymbol{\phi})} p(\boldsymbol{\phi}) \, \mathrm{d}\boldsymbol{\phi}. \end{aligned} \tag{6.12}$$

Whilst some simplification is possible by noting that $Z(\boldsymbol{y}'^{(m)}, \boldsymbol{\phi}) = \sum_{\boldsymbol{z}'_{-i}} q(z'_i = 1, \boldsymbol{z}'_{-i}|\boldsymbol{y}'^{(m)}, \boldsymbol{\phi}) + \sum_{\boldsymbol{z}'_{-i}} q(z'_i = -1, \boldsymbol{z}'_{-i}|\boldsymbol{y}'^{(m)}, \boldsymbol{\phi})$, we still need to evaluate the normalising constant and a sum over 2^{n-1} terms.

Møller *et al.* (2004) present a method for avoiding the calculation of the normalising constant using an auxiliary variable method in MCMC. An auxiliary variable is introduced on the same state space as the image $\boldsymbol{z}$ with a certain conditional density. The task is then to find the posterior distribution of the parameters and the auxiliary variable by Metropolis-Hastings MCMC, where the Hastings ratio contains the ratio of two (different) normalising constants. The trick of this approach is to choose the proposal distribution to be equal to the likelihood. This introduces two more normalising constants into the Hastings ratio, which cancel with the previous two. This simpler Hastings ratio makes the Metropolis-Hastings MCMC algorithm possible.

Aside from the mixing of the algorithm being strongly dependent on the density used for the auxiliary variable, the efficiency of this method unfortunately relies heavily on the ability to perform perfect sampling from the proposal distribution. Coupling from the past algorithms exist for the auto-logistic model (see Propp

and Wilson, 1996), but we were not able to sample efficiently from the Ising model with regression on an external field because of heterogeneity.

None of the methods presented here offer an efficient way of avoiding the calculation of the normalising constant for the flood inundation problem.

6.3 Approximating the Normalising Constant

Having failed to identify a way in which to avoid the calculation of the normalising constant, we will now describe ways in which it can be approximated. Although methods based on analytic approximation and numerical integration are possible the most widely used method in statistics, because of its general applicability, is Monte Carlo simulation. Gelman and Meng (1998) give a thorough exposition of this subject, illustrating the relationships between the various methods. We shall only give a brief review here.

When the normalising constant is not tractable it may still be possible to simulate from the model (as is the case for the Ising model). Expectations can be approximated using these model simulations and this is exploited in the Monte Carlo methods we now present.

6.3.1 Importance Sampling

Consider a density $p(\boldsymbol{z}|\omega)$ indexed by a scalar parameter ω, where the term "density" is used for both continuous and discrete distributions. We are concerned with situations where the density is not known exactly but can be expressed in terms of an easily-computed unnormalised density, $q(\boldsymbol{z}|\omega)$, and an unknown normalising constant $Z(\omega)$,

$$p(\boldsymbol{z}|\omega) = \frac{q(\boldsymbol{z}|\omega)}{Z(\omega)}.$$

If an approximate density $\tilde{p}(\cdot|\omega)$ can be found for $p(\cdot|\omega)$, then using the identity

$$Z(\omega) = \mathrm{E}_{\tilde{p}}\left(\frac{q(\boldsymbol{z}|\omega)}{\tilde{p}(\boldsymbol{z}|\omega)}\right),$$

where $\mathrm{E}_{\tilde{p}}(g(\boldsymbol{z})) = \sum_{\boldsymbol{z}} g(\boldsymbol{z})\,\tilde{p}(\boldsymbol{z}|\omega)$, the *Importance Sampling* estimator of $Z(\omega)$ is

$$\frac{1}{n}\sum_{i=1}^{n}\frac{q(\boldsymbol{z}_i|\omega)}{\tilde{p}(\boldsymbol{z}_i|\omega)},$$

where $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ is a sample from $\tilde{p}(\cdot)$. This simple method is only efficient when $\tilde{p}(\cdot)$ is a good approximation to $p(\cdot)$.

When comparing parameters ω_0 and ω_1, it is sufficient to calculate the likelihood ratio

$$\frac{p(\boldsymbol{z}|\omega_1)}{p(\boldsymbol{z}|\omega_0)} = \frac{q(\boldsymbol{z}|\omega_1)}{q(\boldsymbol{z}|\omega_0)} \frac{Z(\omega_0)}{Z(\omega_1)},$$

so it is sufficient to calculate the ratio of normalising constants (or equivalently the difference of the logarithms). The importance sampling estimate of this ratio is based on the identity

$$\frac{Z(\omega_1)}{Z(\omega_0)} = \mathrm{E}_{\omega_0}\left(\frac{q(\boldsymbol{z}|\omega_1)}{q(\boldsymbol{z}|\omega_0)}\right) \tag{6.13}$$

where the expectation $\mathrm{E}_{\omega_0}(\cdot)$ is with respect to $p(\boldsymbol{z}|\omega_0)$. Suppose the components of $\boldsymbol{z}$ are discrete then

$$\mathrm{E}_{\omega_0}\left(\frac{q(\boldsymbol{z}|\omega_1)}{q(\boldsymbol{z}|\omega_0)}\right) = \sum_{\boldsymbol{z}} \frac{q(\boldsymbol{z}|\omega_1)}{q(\boldsymbol{z}|\omega_0)} p(\boldsymbol{z}|\omega_0) = \sum_{\boldsymbol{z}} \frac{q(\boldsymbol{z}|\omega_1)}{Z(\omega_0)} = \frac{Z(\omega_1)}{Z(\omega_0)}.$$

The samples are only taken from one of the unnormalised densities so the efficiency of this estimator depends on the amount of overlap between the two. If the densities are not heavily overlapping the values of $\boldsymbol{z}$ obtained will not represent a good sample from $q(\boldsymbol{z}|\omega_1)$ in which case the approximation will be poor.

6.3.2 Bridge Sampling

Bridge Sampling was introduced by Meng and Wong (1996) to allow draws to be taken from both unnormalised densities, whilst another density serves as a bridge to connect the two samples. The fundamental identity for Bridge Sampling is

$$\frac{Z(\omega_1)}{Z(\omega_0)} = \frac{\mathrm{E}_{\omega_0}(q(\boldsymbol{z}|\omega_1)\,\alpha(\boldsymbol{z}))}{\mathrm{E}_{\omega_1}(q(\boldsymbol{z}|\omega_0)\,\alpha(\boldsymbol{z}))}, \tag{6.14}$$

where $\alpha(\cdot)$ is a function satisfying

$$0 < \left| \sum_{\boldsymbol{z} \in \Omega_0 \cap \Omega_1} \alpha(\boldsymbol{z})\, p(\boldsymbol{z}|\omega_0)\, p(\boldsymbol{z}|\omega_1) \right| < \infty,$$

and Ω_t is the support for $p(\boldsymbol{z}|\omega_t)$, for $t = 0, 1$. Let $\boldsymbol{z}_{0,1}, \ldots, \boldsymbol{z}_{0,n_0}$ and $\boldsymbol{z}_{1,1}, \ldots, \boldsymbol{z}_{1,n_1}$ be random samples from $p(\boldsymbol{z}|\omega_0)$ and $p(\boldsymbol{z}|\omega_1)$ respectively. Then the *Bridge Sampling* estimate of $Z(\omega_1)/Z(\omega_0)$ is

$$\frac{Z(\omega_1)}{Z(\omega_0)} \approx \frac{\frac{1}{n_0}\sum_{i=1}^{n_0} q(\boldsymbol{z}_{0,i}|\boldsymbol{\omega}_1)\,\alpha(\boldsymbol{z}_{0,i})}{\frac{1}{n_1}\sum_{i=1}^{n_1} q(\boldsymbol{z}_{1,i}|\boldsymbol{\omega}_0)\,\alpha(\boldsymbol{z}_{1,i})}.$$

The effect of the bridge is to reduce the amount of overlap needed between the two unnormalised densities. To see the "bridging" element of this method more clearly, assume $q_b(\boldsymbol{z})$ is a (bridge) density that lies between $q(\boldsymbol{z}|\omega_0)$ and $q(\boldsymbol{z}|\omega_1)$, i.e. has support $\Omega_{\omega_0} \cap \Omega_{\omega_1}$, and let

$$\alpha(\boldsymbol{z}) = \frac{q_b(\boldsymbol{z})}{q(\boldsymbol{z}|\omega_0)\, q(\boldsymbol{z}|\omega_1)}.$$

Substituting this into Equation (6.14) and then using the importance sampling identity (6.13) we find

$$\frac{\mathrm{E}_{\omega_0}\left(q_b(\boldsymbol{z})/q(\boldsymbol{z}|\omega_0)\right)}{\mathrm{E}_{\omega_1}\left(q_b(\boldsymbol{z})/q(\boldsymbol{z}|\omega_1)\right)} = \frac{Z_b/Z(\omega_0)}{Z_b/Z(\omega_1)} = \frac{Z(\omega_1)}{Z(\omega_0)}$$

where Z_b is the normalising constant corresponding to the unnormalised density $q_b(\boldsymbol{z})$. The above equation shows how bridge sampling can be seen as a way of carrying out importance sampling with respect to some arbitrary density and then combining the results.

6.3.3 Path Sampling

A natural extension to Bridge sampling is to use multiple bridges, and taking this to the limit we may consider infinitely many continuously connected bridges linking the two densities. In doing so we arrive at the fundamental identity for *path sampling*, (see Gelman and Meng, 1998, for a proof). We shall present a derivation of the path sampling identity from first principles, because it is informative to see how the algorithm is constructed. This derivation follows Gelman and Meng (1998).

Let $Z(\boldsymbol{\omega}) = \sum_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{\omega})$ where $\boldsymbol{\omega}$ is a continuous d-dimensional vector parameter. Suppose we are interested in the ratio $Z(\boldsymbol{\omega}_1)/Z(\boldsymbol{\omega}_0)$ for given vectors $\boldsymbol{\omega}_0$ and $\boldsymbol{\omega}_1$. The first step requires the construction of a path between $\boldsymbol{\omega}_0$ and $\boldsymbol{\omega}_1$. Define a vector function $\boldsymbol{\omega}(t) = \{\omega_1(t), \ldots, \omega_d(t)\}$ with $t \in [0,1]$ and endpoints $\boldsymbol{\omega}(0) = \boldsymbol{\omega}_0$ and $\boldsymbol{\omega}(1) = \boldsymbol{\omega}_1$. Take the logarithm of the normalising constant and

then differentiate with respect to t,

$$\begin{aligned}
\frac{d}{dt}\log Z\left(\boldsymbol{\omega}\left(t\right)\right) &= \frac{1}{Z\left(\boldsymbol{\omega}\left(t\right)\right)}\sum_{\boldsymbol{z}}\frac{d}{dt}q\left(\boldsymbol{z}|\boldsymbol{\omega}\left(t\right)\right) \\
&= \frac{1}{Z\left(\boldsymbol{\omega}\left(t\right)\right)}\sum_{\boldsymbol{z}}\sum_{k=1}^{d}\omega_k'\left(t\right)\frac{\partial}{\partial\omega_k}q\left(\boldsymbol{z}|\boldsymbol{\omega}\left(t\right)\right) \\
&= \sum_{\boldsymbol{z}}\sum_{k=1}^{d}\omega_k'\left(t\right)\frac{\partial}{\partial\omega_k}\left(\log q\left(\boldsymbol{z}|\boldsymbol{\omega}\left(t\right)\right)\right)p\left(\boldsymbol{z}|\boldsymbol{\omega}\left(t\right)\right) \\
&= \mathrm{E}_{\boldsymbol{\omega}(t)}\left(\sum_{k=1}^{d}\omega_k'\left(t\right)\frac{\partial}{\partial\omega_k}\log q\left(\boldsymbol{z}|\boldsymbol{\omega}\left(t\right)\right)\right) \qquad (6.15)
\end{aligned}$$

where $\mathrm{E}_{\boldsymbol{\omega}(t)}\left(\cdot\right)$ is the expectation with respect to the density $p\left(\boldsymbol{z}|\boldsymbol{\omega}\left(t\right)\right)$. Let $\Theta(\boldsymbol{\omega}) = \log Z(\boldsymbol{\omega})$, then integrating from 0 to 1 yields

$$\Theta(\boldsymbol{\omega}_1) - \Theta(\boldsymbol{\omega}_0) = \int_0^1 \mathrm{E}_{\boldsymbol{\omega}(t)}\left(\sum_{k=1}^{d}\omega_k'\left(t\right)\frac{\partial}{\partial\omega_k}\log q\left(\boldsymbol{z}|\boldsymbol{\omega}\left(t\right)\right)\right)\,\mathrm{d}t. \qquad (6.16)$$

This is the most general representation of the path sampling algorithm, it includes thermodynamic integration (see for example Frenjel, 1986) as a special case. A simple unbiased estimator is

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{d}\omega_k'\left(t_i\right)\frac{\partial}{\partial\omega_k}\log q\left(\boldsymbol{z}_i|\boldsymbol{\omega}\left(t_i\right)\right)$$

where $t_i \sim \mathcal{U}[0,1]$ and $\boldsymbol{z}_i \sim p(\boldsymbol{z}|\boldsymbol{\theta}(t_i))$, so $(\boldsymbol{z}_i, t_i)$ is a sample from the joint distribution $p(\boldsymbol{z}, t) = p(\boldsymbol{z}|\boldsymbol{\theta}(t))p(t)$ where $p(t)$ is uniform on $[0,1]$. Alternatively, the integral in Equation (6.16) can be evaluated numerically, for example using Simpson's rule, where the expectation is approximated by the sample mean.

Path sampling is limited to calculating the ratio of normalising constants. When the absolute value is required, the normalising constant will have to be known for some reference parameter $\boldsymbol{\omega}^\star$. Then taking $\boldsymbol{\omega}_0 = \boldsymbol{\omega}^\star$ in Equation (6.16), the absolute value can be found for all other parameters $\boldsymbol{\omega}_1$. The question is "for what parameters is the normalising constant known exactly?".

One method, appropriate for the auto-logistic model, is presented in Pettitt *et al.* (2003). They begin by showing how the normalising constant can be calculated exactly if cylindrical boundary conditions are assumed, so pixels in the last column

are neighbours with pixels in the first column. To approximate the normalising constant on a lattice they introduce an auxiliary variable ω_c with the property that the boundary conditions are cylindrical when $\omega_c = 0.0$ and lattice when $\omega_c = 1.0$. Path sampling over this auxiliary variable from 0.0 to 1.0 and observing that the normalising constant is known exactly when $\omega_c = 0.0$, the absolute normalising constant can be predicted for lattice boundary conditions. Computational restrictions mean that the cylinder normalising constant can only be calculated when either the number of rows or columns is less than about 10.

Friel and Pettitt (2004) extend this model to larger lattices following a similar auxiliary variable method to that above. The large lattice is split up into a number of more manageable sub-lattices, for which the cylinder normalising constant can be calculated. An auxiliary variable represents the connection between the sub-lattices, and by path sampling along this parameter as before we can obtain a prediction of the absolute normalising constant for the large lattice. We present our own auxiliary variable methods in Sections 6.4.4 and 6.4.6 for paths between parameterisations and paths between binary images.

6.4 Path Sampling for the Ising Model with Regression on a Binary Image

In this section we consider how path sampling can be utilised for the Ising model with regression on a binary image.

6.4.1 Exact Computation of the Normalising Constant

In Sections 6.1 and 6.2 we presented symmetric parameterisations of the Ising model and the Ising model with regression on an image. The argument for adopting these parameterisations is that the log normalising constant is symmetric about $\mu = 0$, $\Theta(-\mu, \delta, \gamma, \boldsymbol{y}) = \Theta(\mu, \delta, \gamma, \boldsymbol{y})$, so we only need to calculate it for $\mu \geqslant 0$. Furthermore, the interpretation of the parameters is transparent: $\mu = 0$ corresponds to no bias towards positives or negatives; $\delta = 0$ corresponds to independence of pixels; and $\gamma = 0$ means $\boldsymbol{z}$ is independent of $\boldsymbol{y}$. However, we decided to test the

functionality of a different parameterisation that avoids the use of indicator functions, and it was within the framework of this alternative model that we tested the accuracy of path sampling (see Section 6.4.2). Note that it is also possible to calculate the normalising constant exactly for the symmetric model when the clustering parameter is zero, $\delta = 0$, and therefore this is not an advantage of the non-symmetric specification.

Suppose the simulator output is $\boldsymbol{s} \in \{0,1\}^n$ and the observed data is $\boldsymbol{r} \in \{0,1\}^n$ and let α, β, ψ denote the trend, clustering and regression parameters. Then the unnormalised density for the non-symmetric Ising model with regression on a binary image is

$$q\left(\boldsymbol{r}|\boldsymbol{s},\alpha,\beta,\psi\right) = \exp\left(\alpha\sum_{i=1}^{n} r_i + \beta\sum_{i\sim j} r_i r_j + \psi\sum_{i=1}^{n}\sum_{k\in\nu_i} r_i s_k\right).$$

When $\beta = 0.0$ the pixels are independent, and the normalising constant $Z(\boldsymbol{s},\alpha,\beta = 0,\psi)$ is

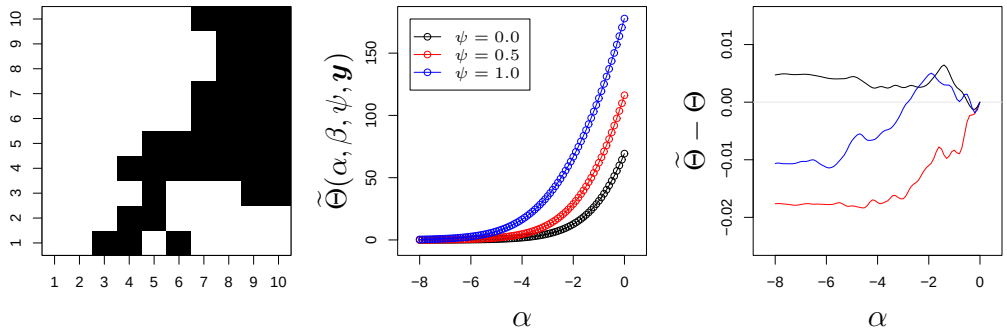
$$\sum_{\boldsymbol{r}\in\{0,1\}^n}\prod_{i=1}^{n}\exp\left(r_i\left(\alpha+\psi\sum_{k\in\nu_i} s_k\right)\right)$$

and from $\sum_{\boldsymbol{r}}\prod_i a_i^{r_i} = \prod_i\sum_{r_i} a_i^{r_i}$ we find

$$\begin{aligned} Z\left(\boldsymbol{s},\alpha,\beta=0,\psi\right) &= \prod_{i=1}^{n}\sum_{r_i\in\{0,1\}}\exp\left(r_i\left(\alpha+\psi\sum_{k\in\nu_i} s_k\right)\right) \\ &= \prod_{i=1}^{n}\left(\exp\left(\alpha+\psi\sum_{k\in\nu_i} s_k\right)+1\right). \end{aligned} \tag{6.17}$$

Suppose we want to compute the log normalising constant $\Theta\left(\boldsymbol{s},\alpha,\beta^\star,\psi\right)$ where $\beta^\star \neq 0.0$. Path sampling from $\beta = 0.0$ to $\beta = \beta^\star$, keeping the other parameters fixed provides the difference in log normalising constants. Then, using Equation (6.17) to calculate the normalising constant when $\beta = 0.0$, we can calculate $\Theta\left(\alpha,\beta^\star,\psi,\boldsymbol{s}\right)$.

We have now presented a way of predicting the absolute value of the normalising constant. The remaining part of the chapter is only concerned with the approximation of normalising constant ratios or equivalently the difference of their logarithms.



(a) The covariate image $\boldsymbol{s}$. (b) Log normalising constant. (c) Error.

Figure 6.3: Results of path sampling along the α coordinate, together with the error in the approximation. Figure 6.3(a) shows the binary image $\boldsymbol{s}$, where black and white correspond to 1 and 0 respectively. In Figure 6.3(b) the path sampling estimates are shown as circles and the exact values as lines.

6.4.2 A Test of Path Sampling Estimate Accuracy

Equation (6.17) provides a means of evaluating the performance of the path sampling estimate (PSE), which will be important when the Ising model is used within our calibration framework.

With $\beta = 0.0$, we can path sample along α or ψ or, indeed, any path in the $\alpha - \psi$ plane, and compare the outcomes to the exact analytical result of Equation (6.17). This exercise will not tell us anything about the performance of the PSE when $\beta \neq 0.0$.

We test the PSE along the α coordinate between $\alpha_0 = 0.0$ and $\alpha_1 = -8.0$ with $\beta = 0.0$ and $\psi = 0.0, 0.5, 1.0$. For the binary image $\boldsymbol{s}$ we use a 10×10 subregion of a LISFLOOD-FP output for the Buscot dataset (see Section 2.4 and Figure 6.3(a)). To find the absolute log normalising constant rather than the ratio, the value at the lower limit of the integration must be known, we calculated the value exactly at $\alpha_0 = 0.0$ using Equation (6.17). Figure 6.3 summarises the results of the analysis.

An error in the prediction of the log normalising constant, $\xi = \widetilde{\Theta} - \Theta$, leads to a multiplicative factor of $\exp(-\xi)$ in the corresponding likelihood prediction. If $|\xi| \ll 1.0$ then $\exp(-\xi) \approx 1 - \xi$ and the error in the likelihood prediction is about $100|\xi|\%$. From Figure 6.3(c) $\xi < 0.02$ so the error in the likelihood prediction will be less than 2%. Note that the error does not increase as the regression, ψ, on $\boldsymbol{s}$

increases.

6.4.3 Paths Between Parameterisations

Motivated by the auxiliary variable method of Friel and Pettitt (2004) we have developed a method for path sampling between parameterisations. Define an additional parameter ε with the property that the parameterisation is asymmetric when $\varepsilon = 0.0$ and symmetric when $\varepsilon = 1.0$. To respect the different binary representation between the two parameterisations, we have $z_i = 2r_i - 1$ and $y_i = 2s_i - 1$. The unnormalised density of this hybrid Ising model is

$$\begin{aligned} &q\left(\boldsymbol{z}|\boldsymbol{y}, \alpha, \beta, \psi, \mu, \delta, \gamma, \varepsilon\right) \\ &\quad = \exp\left((1-\varepsilon)\left(\mu\sum_{i=1}^{n} z_i + \delta\sum_{i\sim j}\mathbf{1}\left[z_i = z_j\right] + \gamma\sum_{i=1}^{n}\sum_{k\in\nu_i}\mathbf{1}\left[z_i = y_k\right]\right)\right. \\ &\left.+\varepsilon\left(\alpha\sum_{i=1}^{n}\frac{z_i+1}{2} + \beta\sum_{i\sim j}\left(\frac{z_i+1}{2}\right)\left(\frac{z_j+1}{2}\right) + \psi\sum_{i=1}^{n}\sum_{k\in\nu_i}\left(\frac{z_i+1}{2}\right)\left(\frac{y_k+1}{2}\right)\right)\right). \end{aligned}$$

Let the path be $\boldsymbol{\omega}^{\varepsilon}(t) = (\alpha, \beta, \psi, \mu, \delta, \gamma, \boldsymbol{y}, \varepsilon = t)$, then using the path sampling identity from Equation (6.16) we find

$$\begin{aligned} \log\left[\frac{Z\left(\varepsilon=1\right)}{Z\left(\varepsilon=0\right)}\right] = \int_0^1 \mathrm{E}_{\boldsymbol{\omega}^{\varepsilon}(t)}&\left(\sum_{i=1}^{n}\left(\alpha\left(\frac{z_i+1}{2}\right) - \mu z_i\right)\right. \\ &+\sum_{i\sim j}\left(\beta\left(\frac{z_i+1}{2}\right)\left(\frac{z_j+1}{2}\right) - \delta\mathbf{1}\left[z_i = z_j\right]\right) \\ &\left.+\sum_{i=1}^{n}\sum_{k\in\nu_i}\left(\psi\left(\frac{z_i+1}{2}\right)\left(\frac{y_k+1}{2}\right) - \gamma\mathbf{1}\left[z_i = y_k\right]\right)\right)\,\mathrm{d}t. \end{aligned}$$

Although this operation is not trivial and may prove computer intensive, it only needs to be done once. When we have a prediction for the log normalising constant in the desired parameterisation, this can be used as the start point for subsequent path sampling.

6.4.4 Paths Over the Continuous Parameters μ, δ and γ

In this section we describe path sampling over the continuous parameters μ, δ and γ, the discussion of path sampling between simulator outputs will be taken up in Section 6.4.6.

The normalising constant is only known exactly when $\delta = 0$, in which case

$$Z(\boldsymbol{y}, \mu, \delta = 0, \gamma) = \prod_{i=1}^{n} \left(\exp\left(-\mu + \gamma \sum_{i \in \nu_i} \mathbf{1}[y_k = -1] \right) + \exp\left(\mu + \gamma \sum_{i \in \nu_i} \mathbf{1}[y_k = 1] \right) \right),$$

following a similar argument to that in Section 6.4.1.

For the Ising model with regression on a binary image the most general path over the continuous parameters can be written $\boldsymbol{\omega}(t) = (\mu(t), \delta(t), \gamma(t), \boldsymbol{y})$, where $\boldsymbol{y}$ is included so $\boldsymbol{\omega}(t)$ fully parameterises the model.

Suppose we wish to integrate between δ_0 and δ_1 whilst keeping the other parameters fixed, then the path would be $\boldsymbol{\omega}^{\delta}(t) = (\mu, (\delta_1 - \delta_0)t + \delta_0, \gamma, \boldsymbol{y})$. In the sum over parameters in Equation (6.16), the only nonzero term will be that containing the derivative with respect to δ,

$$\frac{\partial}{\partial \delta} \log\left(q\left(\boldsymbol{z} | \boldsymbol{\omega}^{\delta}(t) \right) \right) = \sum_{i \sim j} \mathbf{1}[z_i = z_j],$$

and the path sampling identity becomes

$$\log\left[\frac{Z\left(\boldsymbol{\omega}_1^{\delta}\right)}{Z\left(\boldsymbol{\omega}_0^{\delta}\right)} \right] = (\delta_1 - \delta_0) \int_0^1 \mathrm{E}_{\boldsymbol{\omega}^{\delta}(t)} \left(\sum_{i \sim j} \mathbf{1}[z_i = z_j] \right) \mathrm{d}t \tag{6.18}$$

where $\boldsymbol{\omega}_t^{\delta} = (\mu, \delta_t, \gamma, \boldsymbol{y})$ for $t = 0, 1$.

The results for μ and γ are derived in the same way. Let $\boldsymbol{\omega}^{\mu}(t) = ((\mu_1 - \mu_0)t + \mu_0, \delta, \gamma, \boldsymbol{y})$ and $\boldsymbol{\omega}^{\gamma}(t) = (\mu, \delta, (\gamma_1 - \gamma_0)t + \gamma_0, \boldsymbol{y})$ be the paths, and let $\boldsymbol{\omega}_t^{\mu}$ and $\boldsymbol{\omega}_t^{\gamma}$, for $t = 0, 1$, be the end points of the integration then the path sampling identities are

$$\log\left[\frac{Z\left(\boldsymbol{\omega}_1^{\mu}\right)}{Z\left(\boldsymbol{\omega}_0^{\mu}\right)} \right] = (\mu_1 - \mu_0) \int_0^1 \mathrm{E}_{\boldsymbol{\omega}^{\mu}(t)} \left(\sum_{i=1}^{n} z_i \right) \mathrm{d}t \tag{6.19}$$

and

$$\log\left[\frac{Z\left(\boldsymbol{\omega}_1^{\gamma}\right)}{Z\left(\boldsymbol{\omega}_0^{\gamma}\right)} \right] = (\gamma_1 - \gamma_0) \int_0^1 \mathrm{E}_{\boldsymbol{\omega}^{\gamma}(t)} \left(\sum_{i=1}^{n} \sum_{k \in \nu_i} \mathbf{1}[z_i = y_k] \right) \mathrm{d}t. \tag{6.20}$$

6.4.5 Robustness of Path Sampling Estimates Along μ, δ and γ

When the clustering parameter, δ, is not zero there is no simple analytical result available with which to test the accuracy of path sampling estimates (PSEs), but

we can test the robustness of PSEs, i.e. the variability of the results.

In this section we test the robustness of the PSEs along the μ, δ and γ parameter coordinates. We restrict the analysis to the set of parameters that produces output that we consider to be appropriate for the flood inundation problem. By examining the realisations from the Ising model for various values of the parameters, we choose to focus on $\{(\mu, \delta, \gamma) \,|\mu \in [-1, 1], \delta \in [0, 1], \gamma \in [0, 1]\}$.

When a parameter is not being sampled its value will be set according to $\mu = -0.5, \delta = 0.25$ or $\gamma = 0.5$. To test the effect of the image $\boldsymbol{y}$ on the robustness we identified a 10×10 subregion of the Buscot floodplain for which the simulator outputs are very variable, and we selected five simulations which characterise this variability.

The calculation of the PSE is done in two steps, we describe the method for μ but it is the same for δ and γ. The range of integration, which for μ is $[-1, 1]$, is split into intervals of width 0.1. The first step is to estimate the expectation for $\mu \in \{-1.0, -0.9, \ldots, 0.9, 1.0\}$ using Gibbs sampling MCMC; we use 50000 iterations for each calculation. The second step is to estimate the integral. The approximate expectations are smoothed using a spline smoothing routine and then the area under the graph is calculated using Simpson's rule. The iterative nature of this scheme means that the PSE is obtained at each step along the path.

The results of the analyses are shown in Figures 6.4 and 6.5. The runs are so close it is impossible to distinguish between them so we have plotted the difference separately. The greatest magnitude difference is less than 0.04 and the magnitude does not appear to be related to the binary image $\boldsymbol{y}$. Finally, it is interesting to note that $\widetilde{\Theta}(\delta)$ and $\widetilde{\Theta}(\gamma)$ are quite linear in their arguments, whereas $\widetilde{\Theta}(\mu)$ is not.

This analysis highlighted a major problem with the PSE, that of computation time. For example, the path sampling estimate along μ required 21 expectations to be approximated, each of which took 52 seconds on a Pentium 4 2GHz processor with 512MB of RAM.

For calibration and calibrated prediction we need to be able to generate a sample from $p(m, \boldsymbol{\phi}|\boldsymbol{z})$, see Equation (6.9). If the unnormalised density, $q(m, \boldsymbol{\phi}|\boldsymbol{z})$, is

known then a sample can be generated using MCMC, but the (likelihood) normalising constant $Z(\boldsymbol{y}^{(m)}, \boldsymbol{\phi})$ is present in this unnormalised density. Therefore to use MCMC we must evaluate the ratio of normalising constants (see Equation (6.10)), every time μ, δ, γ or m changes. For the continuous parameters we can use the path sampling methods described in Section 6.4.4 to approximate this ratio, and in Section 6.4.6 we develop a method for approximating this ratio when the simulation index m changes. As a conservative estimate of the computation time, suppose 50000 iterations are required, each of the four parameters are updated on 50% of the iterations, and the path sampling estimate takes one minute to compute, then the sample would take almost ten weeks to generate. For the Buscot dataset the image is $48 \times 76 = 3648$ pixels so will take considerably longer.

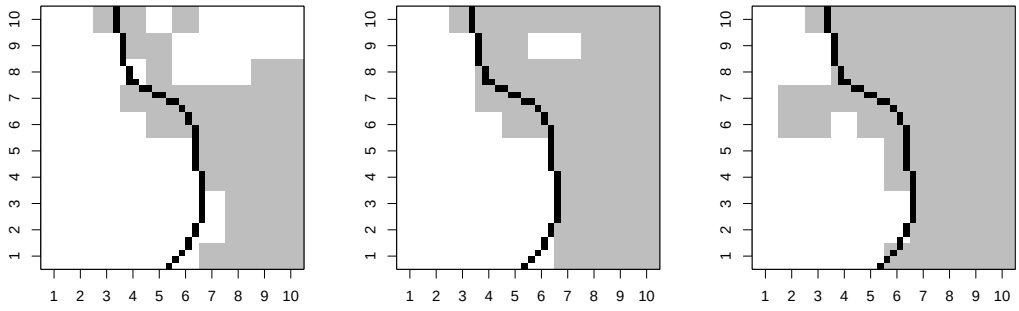
An alternative is to estimate the normalising constant offline for each m and for μ, δ and γ on a grid which encompasses the values of interest. Suppose the grid is defined by the range of parameters given previously with spacing of 0.1, this gives $500 \times 21 \times 11 \times 11 \approx 1$ million parameter sets. For a particular image, $\boldsymbol{y}$, an efficient path sampling algorithm is to start with

$$\{(\mu = -1.0, \delta = 0.0, \gamma, \boldsymbol{y}) | \gamma \in \{0.0, 0.1, \ldots, 0.9, 1.0\}\}$$

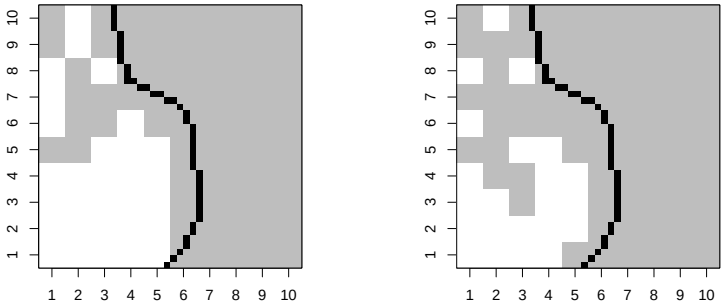
and integrate along μ between -1.0 and 1.0; then start at

$$\{(\mu, \delta = 0.0, \gamma, \boldsymbol{y}) | \mu \in \{-1.0, -0.9, \ldots, 0.9, 1.0\}, \gamma \in \{0.0, 0.1, \ldots, 0.9, 1.0\}\}$$

integrate along δ from 0.0 to 1.0. This would require 11 path samples along μ and 21×11 along δ, where the latter are shorter, taking only 28 seconds on the machine described above. The total time to estimate the normalising constant on the grid for every image will be over 40 days. Again this is for the 10×10 binary image. During the MCMC run either the continuous parameters must be discretized according to the grid or the values of the normalising constant can be interpolated from our grid of estimates. We have had to perform a number of one dimensional integrals to obtain the grid of estimates, in Section 6.4.7 we develop an extension to path sampling over areas.



(a) $\boldsymbol{y}^{(1)}$. (b) $\boldsymbol{y}^{(2)}$. (c) $\boldsymbol{y}^{(3)}$.



(d) $\boldsymbol{y}^{(4)}$. (e) $\boldsymbol{y}^{(5)}$.

Figure 6.4: Simulations used in the test of robustness. Grey and white correspond to pixel values of 1 and -1 respectively. The observed flood boundary is shown in black.

6.4.6 Paths Between Images

In this section we present a new method for path sampling between binary images, motivated by the auxiliary variable method Pettitt *et al.* (2003) used for path sampling between different lattice boundary conditions. We describe the method for path sampling between two binary images but it can be extended to more than two images.

The path sampling formulation that we outlined in Section 6.3.3 works for any path in the continuous parameter space, but we cannot integrate over the binary image $\boldsymbol{y}$ from $\boldsymbol{y}^{(0)}$ to $\boldsymbol{y}^{(1)}$. Therefore to construct a path from $\boldsymbol{y}^{(0)}$ to $\boldsymbol{y}^{(1)}$ they must both be present in the unnormalised density, let $\varepsilon \in [0, 1]$ be an auxiliary

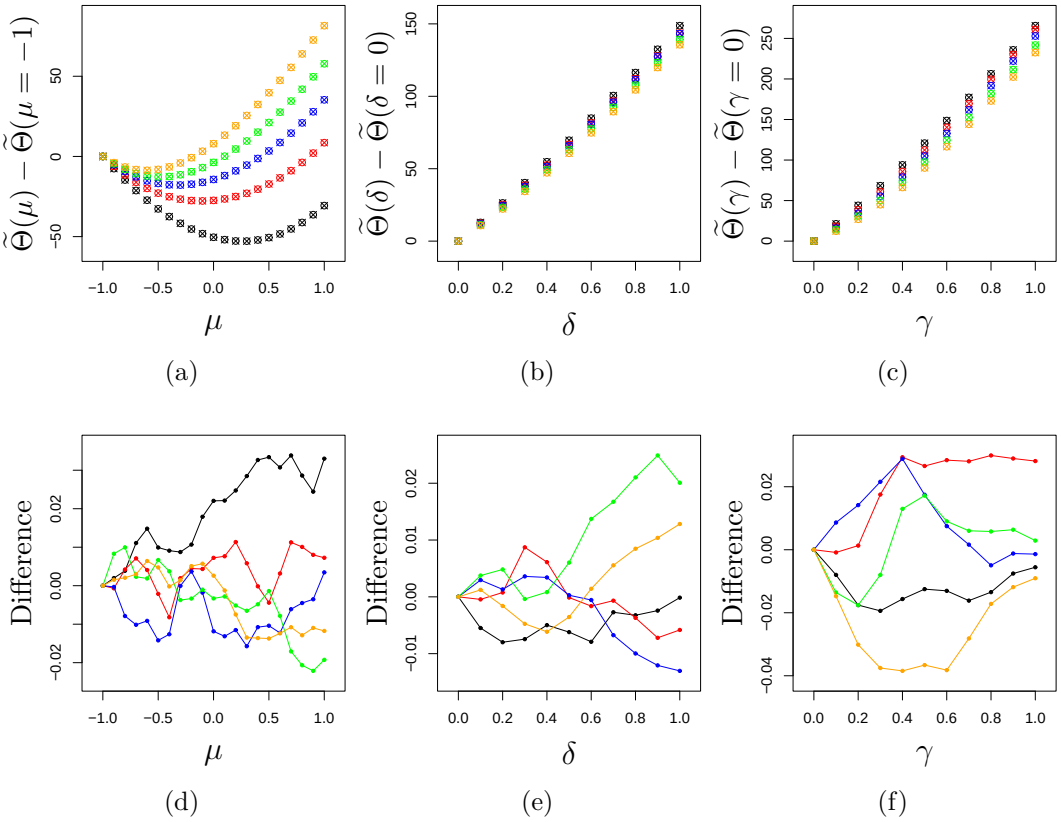


Figure 6.5: Two estimates of the difference between log normalising constants, one shown as circles and the other by crosses. Also the difference between these estimates. The colours correspond to simulations: $\boldsymbol{y}^{(1)}$ is black, $\boldsymbol{y}^{(2)}$ is red, $\boldsymbol{y}^{(3)}$ is blue, $\boldsymbol{y}^{(4)}$ is green, and $\boldsymbol{y}^{(5)}$ is orange.

variable then

$$q(\boldsymbol{z}|\boldsymbol{y}^{(0)}, \boldsymbol{y}^{(1)}, \boldsymbol{\phi}, \varepsilon) = \exp\left(\mu \sum_{i=1}^{n} z_i + \delta \sum_{i \sim j} \mathbf{1}\left[z_i = z_j\right] + (1-\varepsilon)\left(\gamma \sum_{i=1}^{n} \sum_{k \in \nu_i} \mathbf{1}\left[z_i = y_k^{(0)}\right]\right) + \varepsilon\left(\gamma \sum_{i=1}^{n} \sum_{k \in \nu_i} \mathbf{1}\left[z_i = y_k^{(1)}\right]\right)\right),$$

and $\boldsymbol{\omega}_{\varepsilon}(t) = (\boldsymbol{\phi}, (\varepsilon_1 - \varepsilon_0)\, t + \varepsilon_0, \boldsymbol{y}^{(0)}, \boldsymbol{y}^{(1)})$, where $\varepsilon_0 = 0$ and $\varepsilon_1 = 1$, is the desired path. The path sampling identity is

$$\log\left[\frac{Z\left(\boldsymbol{\omega}_1^{\varepsilon}\right)}{Z\left(\boldsymbol{\omega}_0^{\varepsilon}\right)}\right] = (\varepsilon_1 - \varepsilon_0) \int_0^1 \mathrm{E}_{\boldsymbol{\omega}_{\varepsilon}(t)}\left(\sum_{i=1}^{n} \sum_{k \in \nu_i} \gamma\left(\mathbf{1}\left[z_i = y_k^{(1)}\right] - \mathbf{1}\left[z_i = y_k^{(0)}\right]\right)\right) \mathrm{d}t.$$

Note that when $0.0 < \varepsilon < 1.0$ there is a contribution from both $\boldsymbol{y}^{(0)}$ and $\boldsymbol{y}^{(1)}$, this

is of no practical interest, we only make inference about $\boldsymbol{z}$ when $\varepsilon = 0.0$ or 1.0.

We now consider a simple test of this method. Suppose for $\boldsymbol{y}^{(0)}$ with $\delta = 0.25$ and $\gamma = 0.5$ we wish to know the difference of log normalising constants from $\mu_0 = -0.5$ to $\mu_1 = 0.5$, that is

$$\Theta\left(\boldsymbol{y}^{(0)}, \boldsymbol{y}^{(1)}, \mu_1 = 0.5, \delta, \gamma, \varepsilon = 0\right) - \Theta\left(\boldsymbol{y}^{(0)}, \boldsymbol{y}^{(1)}, \mu_0 = -0.5, \delta, \gamma, \varepsilon = 0\right).$$

We could use Equation (6.19) and integrate over μ between $\mu_0 = -0.5$ and $\mu_1 = 0.5$ or we could take a longer route from $\varepsilon_0 = 0.0$ to $\varepsilon_1 = 1.0$, then $\mu_0 = -0.5$ to $\mu_1 = 0.5$ (with $\varepsilon = 1.0$) and then $\varepsilon_0 = 1.0$ to $\varepsilon_1 = 0.0$. Because both paths should take us to the same destination we can test the effectiveness of path sampling between images by comparing the results.

The PSEs along these two paths are illustrated in Figure 6.6. The kinks in the longer path occur when the parameter we are path sampling over changes, but it does appear to arrive at the correct value. The exact values were

$$\widetilde{\Theta}\left(\mu_1 = 0.5, \varepsilon = 0.0\right) - \widetilde{\Theta}\left(\mu_0 = -0.5, \varepsilon = 0.0\right) = -17.9353$$

and

$$\begin{aligned}\widetilde{\Theta}\left(\mu = -0.5, \varepsilon_1 = 1.0\right) - \widetilde{\Theta}\left(\mu = -0.5, \varepsilon_0 = 0.0\right) \\ + \widetilde{\Theta}\left(\mu_1 = 0.5, \varepsilon = 1.0\right) - \widetilde{\Theta}\left(\mu_0 = -0.5, \varepsilon = 1.0\right) \\ + \widetilde{\Theta}\left(\mu = 0.5, \varepsilon_1 = 0.0\right) - \widetilde{\Theta}\left(\mu = 0.5, \varepsilon_0 = 1.0\right) = -17.9114.\end{aligned}$$

The difference between the two estimates is 0.0239 which is within the range of differences in the robustness analysis of Section 6.4.5. This suggests this method may give good results.

6.4.7 From Paths to Higher Dimensions

We have not exploited the full functionality of path sampling which would allow arbitrary paths to be taken over the continuous parameter space. There are two reasons for this. Firstly, we need the value of the log normalising constant at all points in the continuous parameter space (discretized into a 0.1 grid) and the most

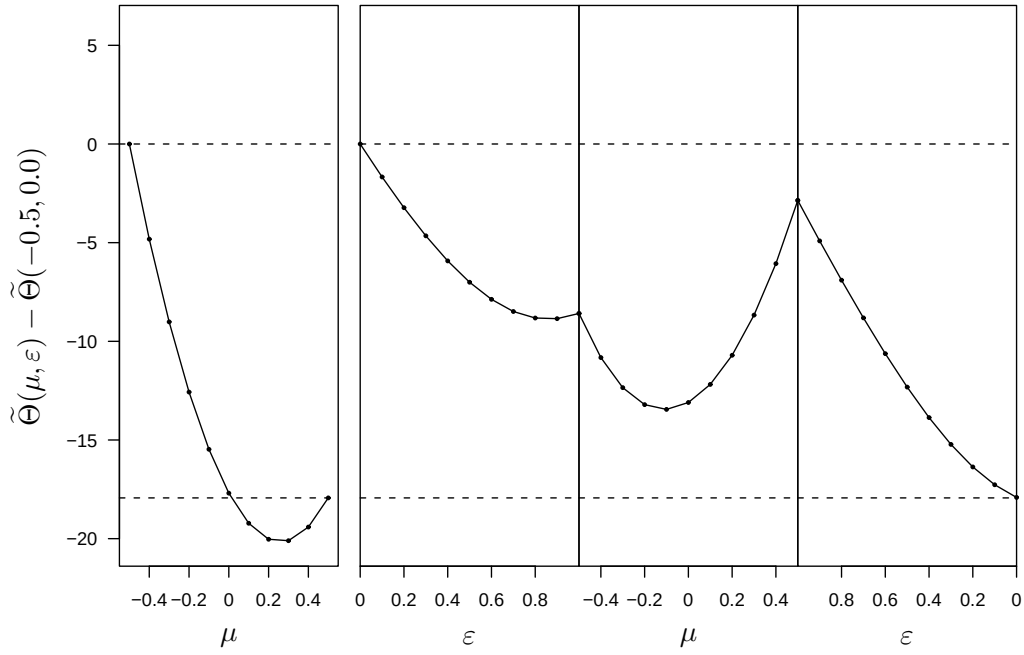


Figure 6.6: Comparison of two path sampling paths: one direct over $\mu \in [-0.5, 0.5]$ and one along $\varepsilon \in [0, 1]$ then $\mu \in [-0.5, 0.5]$ then $\varepsilon \in [1, 0]$.

efficient way to do this is to sample along the coordinates. Secondly, the expectation we need to evaluate becomes significantly more complicated when anything other than componentwise paths are used, making it more computationally expensive. Although a search for more optimal paths may be fruitless, because we are trying to calculate the values over a large number of parameter sets in a high dimensional space, it is worth considering an extension to path sampling that allows for integration over more than one dimension.

We will introduce the idea for the Ising model in Equation (6.6). Suppose we need $\Theta(\mu, \delta) = \log(Z(\mu, \delta))$ for $\mu \in [-1, 1]$ and $\delta \in [0, 1]$. This can be estimated by the PSE along μ from -1 to 1 with δ set at $\{0.0, 0.1, \ldots, 1.0\}$ (see Section 6.4.5). However, it can also be estimated by a certain integral over the area $\{(\mu, \delta) | \mu \in [-1, 1], \delta \in [0, 1]\}$.

We start the derivation of the area based method with the log normalising constant for the symmetric Ising model,

$$\Theta(\mu, \delta) = \log\left(\sum_{\boldsymbol{z}} \exp\left(\mu \sum_{i=1}^{n} z_i + \delta \sum_{i \sim j} \mathbf{1}[z_i = z_j]\right)\right),$$

differentiating with respect to μ and δ we obtain

$$\begin{aligned}
\frac{\partial^2 \Theta}{\partial \mu \, \partial \delta} &= \frac{\sum_{\boldsymbol{z}} \sum_{i=1}^{n} z_i \sum_{i \sim j} \mathbf{1}\left[z_i = z_j\right] q\left(\boldsymbol{z}|\mu, \delta\right)}{\sum_{\boldsymbol{z}} q\left(\boldsymbol{z}|\mu, \delta\right)} \\
&\quad - \frac{\left(\sum_{\boldsymbol{z}} \sum_{i=1}^{n} z_i q\left(\boldsymbol{z}|\mu, \delta\right)\right)\left(\sum_{\boldsymbol{z}} \sum_{i \sim j} \mathbf{1}\left[z_i = z_j\right] q\left(\boldsymbol{z}|\mu, \delta\right)\right)}{\left(\sum_{\boldsymbol{z}} q\left(\boldsymbol{z}|\mu, \delta\right)\right)^2} \\
&= \mathrm{E}_{\mu,\delta}\left(\sum_{i=1}^{n} z_i \sum_{i \sim j} \mathbf{1}\left[z_i = z_j\right]\right) - \mathrm{E}_{\mu,\delta}\left(\sum_{i=1}^{n} z_i\right) \mathrm{E}_{\mu,\delta}\left(\sum_{i \sim j} \mathbf{1}\left[z_i = z_j\right]\right) \\
&= \mathrm{Cov}_{\mu,\delta}\left(\sum_{i=1}^{n} z_i, \sum_{i \sim j} \mathbf{1}\left[z_i = z_j\right]\right), \qquad (6.21)
\end{aligned}$$

where $\mathrm{E}_{\mu,\delta}\left(\cdot\right)$ and $\mathrm{Cov}_{\mu,\delta}\left(\cdot,\cdot\right)$ are the expectation and covariance with respect to the density $p\left(\boldsymbol{z}|\mu,\delta\right)$. To find the difference in log normalising constants between (μ_0, δ_0) and (μ_1, δ_1), we integrate both sides with respect to δ and μ and substitute for $\Theta(\mu_1, \delta_0)$ and $\Theta(\mu_0, \delta_1)$ using the path sampling identities

$$\begin{aligned}
\Theta(\mu_1, \delta_0) &= \Theta(\mu_0, \delta_0) + \int_{\mu_0}^{\mu_1} \frac{\partial \Theta}{\partial \mu}(\mu, \delta_0) \, \mathrm{d}\mu \quad \text{and} \\
\Theta(\mu_0, \delta_1) &= \Theta(\mu_0, \delta_0) + \int_{\delta_0}^{\delta_1} \frac{\partial \Theta}{\partial \delta}(\mu_0, \delta) \, \mathrm{d}\delta
\end{aligned}$$

then

$$\begin{aligned}
\Theta\left(\mu_1, \delta_1\right) &= \Theta\left(\mu_0, \delta_0\right) + \int_{\mu_0}^{\mu_1} \frac{\partial \Theta}{\partial \mu}\left(\mu, \delta_0\right) \, \mathrm{d}\mu + \int_{\delta_0}^{\delta_1} \frac{\partial \Theta}{\partial \delta}\left(\mu_0, \delta\right) \, \mathrm{d}\delta \\
&\quad + \int_{\mu_0}^{\mu_1} \int_{\delta_0}^{\delta_1} \frac{\partial^2 \Theta}{\partial \mu \partial \delta}\left(\mu, \delta\right) \, \mathrm{d}\mu \, \mathrm{d}\delta \\
&= \Theta\left(\mu_0, \delta_0\right) + \int_{\mu_0}^{\mu_1} \mathrm{E}_{\mu,\delta_0}\left(\sum_{i=1}^{n} z_i\right) \, \mathrm{d}\mu + \int_{\delta_0}^{\delta_1} \mathrm{E}_{\mu_0,\delta}\left(\sum_{i \sim j} \mathbf{1}\left[z_i = z_j\right]\right) \, \mathrm{d}\delta \\
&\quad + \int_{\mu_0}^{\mu_1} \int_{\delta_0}^{\delta_1} \mathrm{Cov}_{\mu,\delta}\left(\sum_{i=1}^{n} z_i, \sum_{i \sim j} \mathbf{1}\left[z_i = z_j\right]\right) \, \mathrm{d}\mu \, \mathrm{d}\delta. \qquad (6.22)
\end{aligned}$$

Unfortunately the area sampling estimate (ASE) is more computationally intensive than the PSE for $\Theta(\mu, \delta)$ over $\mu \in [-1, 1]$ and $\delta \in [0, 1]$, where we calculate at 0.1 spacings along these coordinates. The PSE requires $11 \times 21 = 231$ expectations to be computed, whereas the ASE requires $11 + 21 = 33$ expectations and $11 \times 21 = 231$ covariances to be computed.

Although this method does not improve on path sampling in terms of computational efficiency it does immediately suggest an additive approximation to path

sampling if we assume that the covariance can be ignored. Approximate path sampling methods are the next logical step because it is not practical to work with the path sampling identity directly.

6.4.8 Approximating the Path Sampling Integral

Whilst path sampling does offer a way of estimating the normalising constant which cannot be computed directly, it is not practical for our problem because we require normalising constant estimates for many parameter combinations. With a discretization spacing of 0.1 along each of the coordinates and 50000 MCMC iterations for each expectation estimate we are getting errors of less than 5% (see Section 6.4.1). It is not possible to reduce the number of iterations or increase the discretization spacing without increasing this error, therefore we are going to approximate the path sampling identity itself by a simple function that may be computed more readily.

For lucidity we refer to our approximations of the path sampling identity as "approximations" and the numerical estimates based on these approximations "estimates". To illustrate the efficacy of the eight approximations which follow we would like to compare them to the true normalising constant, but because this is not known we can only compare estimates based on our approximations against the PSE. We refer to the difference between the estimates based on our approximations and the PSE as "error", and assume the error due to approximation will be greater than the error due to the numerical estimate. We will be using the simple Ising model from Equation (6.6). We are not interested in $|\mu| \gg 0$ which leads to all pixels having the same value in the model output, or in $\delta < 0.0$ which corresponds to negative dependence between pixels. By examining the model output we identify a sensible parameter space for the experiment, $\{(\mu, \delta) \,|\mu \in [-2.0, 2.0] \,, \delta \in [0.0, 1.0]\}$. To estimate the log normalising constant relative to the origin $\mu_0 = 0.0, \delta_0 = 0.5$ we path sample along μ and then δ. We will compare the PSE to our eight approximations on the 5×5 grid described by

$$\{(\mu, \delta) \,|\mu \in \{-2, -1, 0, 1, 2\} \,, \delta \in \{0.0, 0.25, 0.5, 0.75, 1.0\}\} \,.$$

When path sampling along μ a 0.1 spacing is used and along δ we use 0.05. The number of iterations at each step will be 50000. To calculate the PSE at each point in this discretized parameter space requires the estimation of $41 \times 21 = 861$ expectations. The PSE is shown in Figure 6.7(a).

Additive Approximation

As mentioned in Section 6.4.7, if the covariance term in Equation (6.22) is neglected then we need only evaluate two componentwise integrals relative to the origin $(\mu_0 = 0.0, \delta_0 = 0.5)$,

$$\Theta(\mu, \delta) - \Theta(\mu_0, \delta_0) \approx \int_{\mu_0}^{\mu} \frac{\partial \Theta}{\partial \mu} (\mu', \delta_0) \, \mathrm{d}\mu' + \int_{\delta_0}^{\delta} \frac{\partial \Theta}{\partial \delta} (\mu_0, \delta') \, \mathrm{d}\delta'.$$

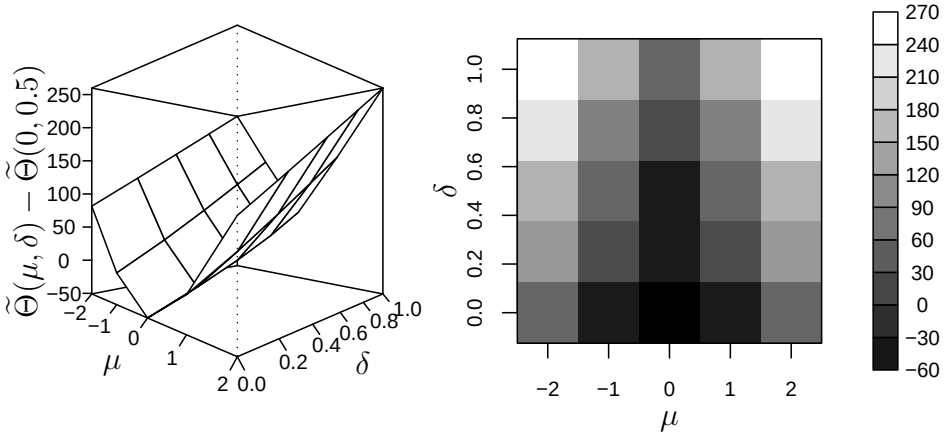
The additive approximation estimate (AAE) requires $41 + 21 = 62$ expectations to be estimated, as opposed to the 861 required for the PSE. The error in the AAE is shown in Figure 6.7(b). The error is zero along the $\mu = 0$ and $\delta = 0.5$ where the AAE and the PSE are equivalent. The fact that the error is up in two corners and down in the other two indicates that although the log normalising constant itself is not additive, there may be a power transform that is. This idea is explored later with Tukey's transformation for additivity.

Additive Linear Approximation

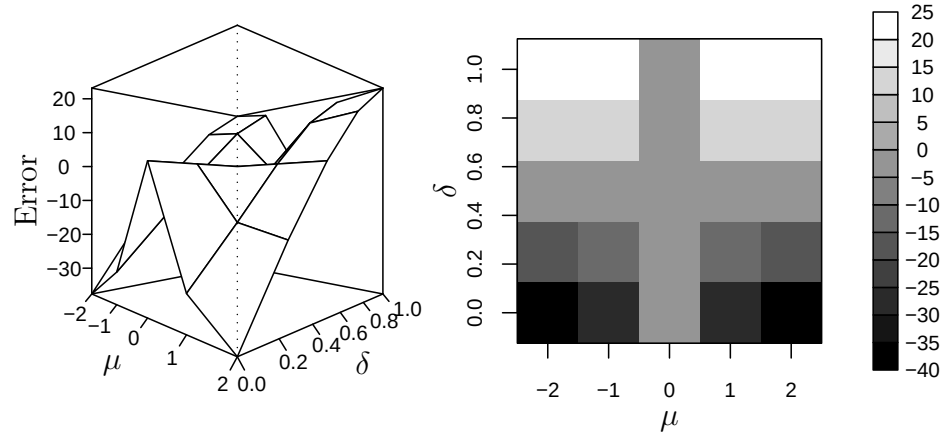
Another sensible approach to approximation is to consider the Taylor series expansion of $\Theta(\mu, \delta)$ and then assume that terms over a certain order can be neglected. The first-order Taylor series is

$$\begin{aligned} \Theta(\mu, \delta) - \Theta(\mu_0, \delta_0) &\approx \frac{\partial \Theta}{\partial \mu} (\mu_0, \delta_0) (\mu - \mu_0) + \frac{\partial \Theta}{\partial \delta} (\mu_0, \delta_0) (\delta - \delta_0) \\ &= \mathrm{E}_{\mu_0, \delta_0} \left(\sum_{i=1}^{n} z_i \right) (\mu - \mu_0) + \mathrm{E}_{\mu_0, \delta_0} \left(\sum_{i \sim j} \mathbf{1} [z_i = z_j] \right) (\delta - \delta_0). \end{aligned}$$

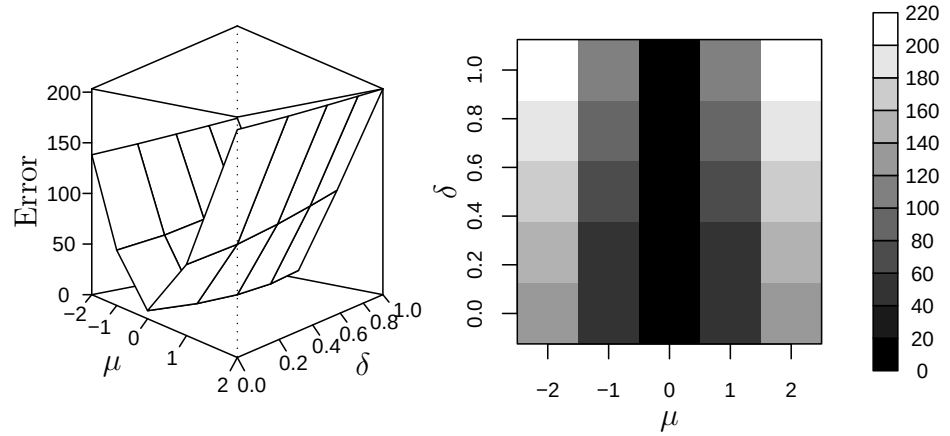
We call it the additive linear approximation because it is additive in the components which in turn are linear approximations to the integrals. For the additive linear approximation estimate (ALAE) we need to estimate only two expectations. The error in the ALAE is shown in Figure 6.7(c). Clearly the ALAE is worse than



(a) Path sampling estimate (PSE) of the log normalising constant.

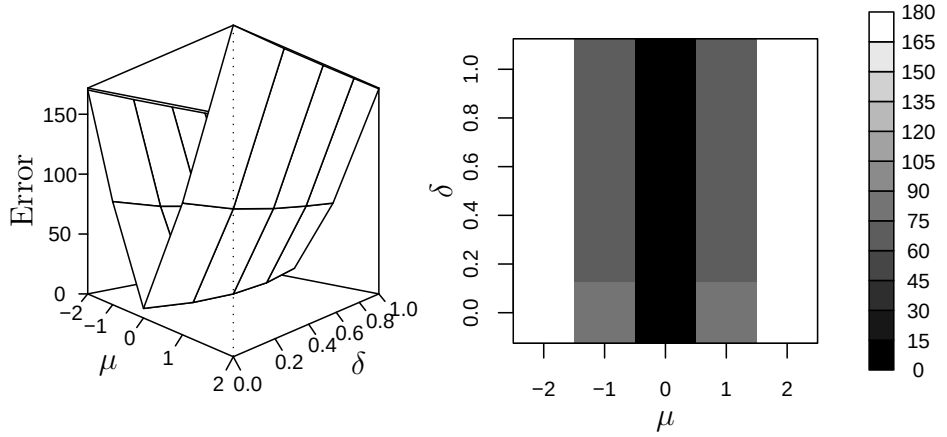


(b) Error in additive approximation estimate (AAE).

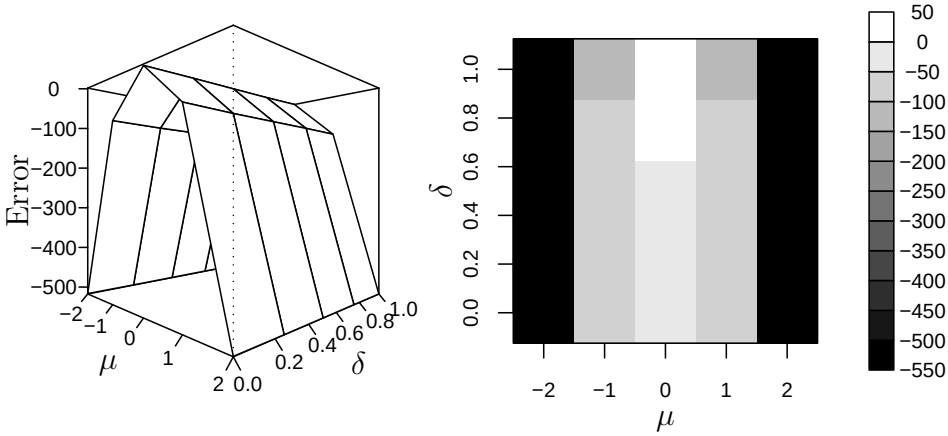


(c) Error in additive linear approximation estimate (ALAE).

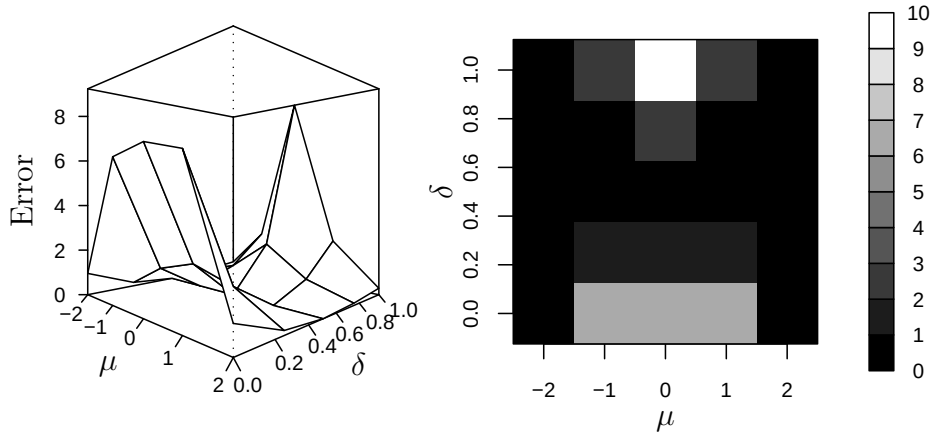
Figure 6.7: Path sampling estimate and approximations for the log normalising constant relative to $(\mu = 0.0, \delta = 0.5)$.



(a) Error in componentwise linear approximation estimate (CLAE).

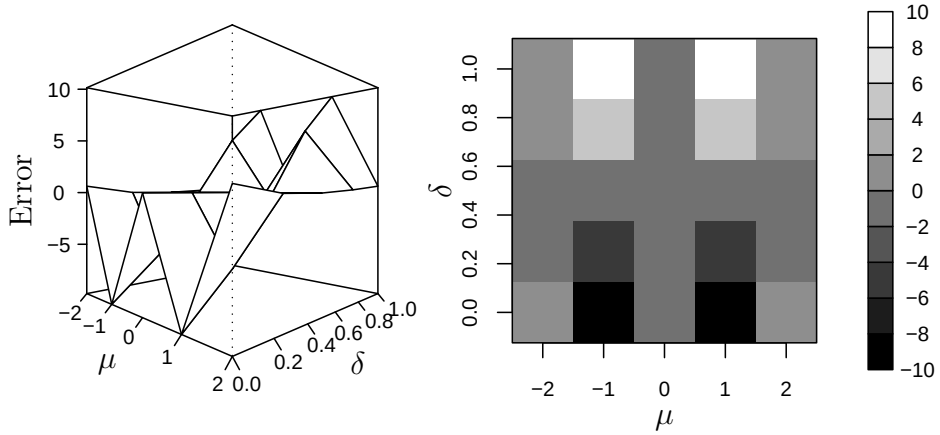


(b) Error in second-order Taylor series approximation estimate (SOTSAE).

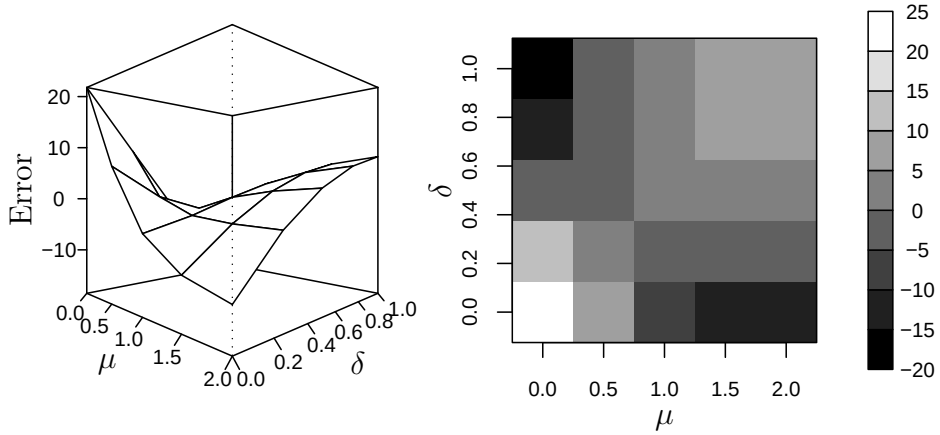


(c) Error in linear approximation along δ only estimate (LAADOE).

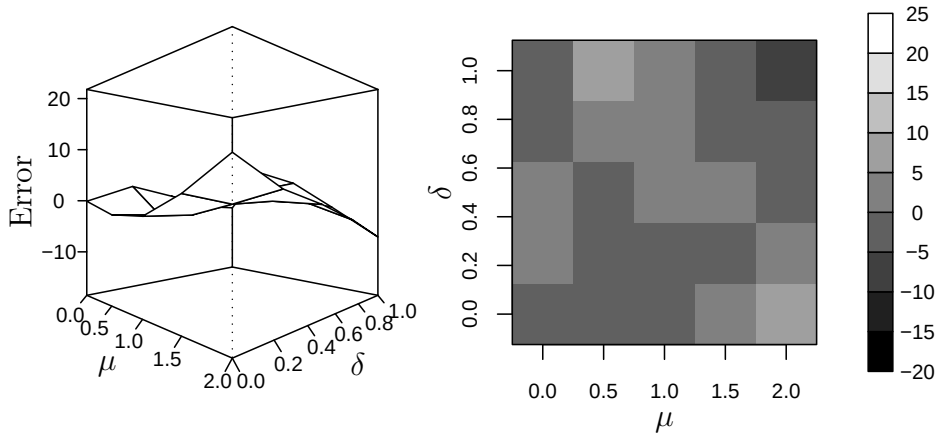
Figure 6.8: Path sampling approximations for the log normalising constant relative to $(\mu = 0.0, \delta = 0.5)$.



(a) Error in hybrid approximation estimate (HAE).



(b) Residuals from fitting the additive linear model $y_{i,j} \sim \kappa + \alpha_i + \beta_j$.



(c) Error in the prediction obtained using Tukey's transformation for additivity.

Figure 6.9: Hybrid and Tukey approximations for the log normalising constant relative to $(\mu = 0.0, \delta = 0.5)$.

the AAE but this is due almost entirely to the linear approximation for the μ term rather than the linear approximation for the δ term, which is not too bad. The reason a linear approximation for the μ term is inappropriate is apparent in Figure 6.7(a).

Componentwise Linear Approximation

An alternative to taking a first-order Taylor series of $\Theta(\mu, \delta)$, is to take a first-order Taylor series of the μ and δ path sampling integrals,

$$\Theta(\mu, \delta) - \Theta(\mu_0, \delta_0) \approx \mathrm{E}_{\mu_0,\delta_0}\left(\sum_{i=1}^{n} z_i\right)(\mu - \mu_0) + \mathrm{E}_{\mu,\delta_0}\left(\sum_{i\sim j} \mathbf{1}\left[z_i = z_j\right]\right)(\delta - \delta_0).$$

The subtle difference is that the second expectation is taken at μ rather than μ_0, that is to say the additive assumption has been dropped. The componentwise linear approximation estimate (CLAE) requires $1 + 41 = 42$ expectations to be estimated. The error in the CLAE is shown in Figure 6.8(a). As expected, whilst the linear approximation of the μ term is still poor, the approximation of the δ term is better.

Second-Order Taylor Series Approximation

One way to improve upon the linear approximation is to use a higher-order Taylor series. Starting with the componentwise linear approximation, if we now include the second term from the Taylor series expansion, the approximation becomes

$$\begin{aligned}\Theta(\mu, \delta) - \Theta(\mu_0, \delta_0) \approx \\ \mathrm{E}_{\mu_0,\delta_0}\left(\sum_{i=1}^{n} z_i\right)(\mu - \mu_0) + \mathrm{Var}_{\mu_0,\delta_0}\left(\sum_{i=1}^{n} z_i\right)\frac{(\mu - \mu_0)^2}{2} \\ + \mathrm{E}_{\mu,\delta_0}\left(\sum_{i\sim j} \mathbf{1}\left[z_i = z_j\right]\right)(\delta - \delta_0) + \mathrm{Var}_{\mu,\delta_0}\left(\sum_{i\sim j} \mathbf{1}\left[z_i = z_j\right]\right)\frac{(\delta - \delta_0)^2}{2}.\end{aligned}$$

For the second-order Taylor series approximation estimate (SOTSAE) we need to estimate $41 + 1 = 42$ expectations and variances. The error in the SOTSAE is shown in Figure 6.8(b). This is worse than the first-order Taylor series approximation because the second term in the Taylor series expansion for the μ term

overcompensates for the error in the linear approximation. Including higher-order terms may help to improve the approximation but will be more computationally demanding.

Linear Approximation Along δ and Path Sampling Estimate Along μ

All efforts at approximating the path sampling integral for μ have proved fruitless, so we will use the PSE for μ and only use the linear approximation for δ,

$$\Theta(\mu,\delta)-\Theta(\mu_0,\delta_0)\approx\int_{\mu_0}^{\mu}\frac{\partial\Theta}{\partial\mu}(\mu',\delta_0)\ \mathrm{d}\mu'+\mathrm{E}_{\mu,\delta_0}\left(\sum_{i\sim j}\mathbf{1}\left[z_i=z_j\right]\right)(\delta-\delta_0).$$

The linear approximation along δ only estimate (LAADOE) requires $41+41=82$ expectations to be estimated, which is still significantly less than the 861 required for PSEs along μ and δ. The error in LAADOE is shown in Figure 6.8(c). This has been the most successful approximation so far, with all the errors being less than 10. However, to put this in perspective an error of 0.1 in the prediction of the log normalising constant corresponds to an error of $> 10\%$ in the posterior density using this normalising constant.

Hybrid of Linear and Additive Estimates for δ and Path Sampling Estimate Along μ

Figure 6.8(c) shows that the linear approximation for the δ term is better when the magnitude of μ is large. This fact can be justified by looking again at Figure 6.7(a), where it can be seen that the relationship of the log normalising constant with δ does become more linear as the magnitude of μ increases. But when $|\mu|$ is small adopting a linear approximation leads to unacceptable errors. The additive approximation suffers the opposite problem, when $|\mu|=0$ there is no error but as $|\mu|$ increases the error increases. This suggests the use of some weighted combination of the additive and linear approximations,

$$\begin{aligned}\Theta(\mu,\delta)-\Theta(\mu_0,\delta_0)\approx\int_{\mu_0}^{\mu}\frac{\partial\Theta}{\partial\mu}(\mu',\delta_0)\ \mathrm{d}\mu'+(1-f(\mu))\int_{\delta_0}^{\delta}\frac{\partial\Theta}{\partial\delta}(\mu_0,\delta')\ \mathrm{d}\delta'\\+f(\mu)\,\mathrm{E}_{\mu_1,\delta_0}\left(\sum_{i\sim j}\mathbf{1}\left[z_i=z_j\right]\right)(\delta-\delta_0)\end{aligned}$$

where $f(\mu)$ is some function taking the values $f(0) = 0$ and $f(\pm 2) = 1$ and $\mu_1 = 2$. The hybrid approximation estimate (HAE) requires $41 + 21 + 1 = 63$ expectations to be estimated. We tried a number of different functions for $f(\cdot)$ but none improved noticeably upon the results of the previous approximation. Figure 6.9(a) shows the error in the HAE when $f(\mu) = |\mu|/2$. The errors we are particularly concerned about are those when $\mu = \pm 1$ and $\delta \neq 0.5$, because the others were effectively constrained to be good.

Tukey's Transformation for Additivity

When investigating the additive approximation to path sampling we mentioned that although data may not be additive itself, we may find that when the data is taken to a certain power the result will be additive.

In the exploratory data analysis of two-way tables, John Tukey developed a method for transforming a two-way array so that it is better approximated by an additive fit. This is known as *Tukey's transformation for additivity*.

When data is placed in increasing order of their effects, and the additive fit is poor, the residuals often exhibit a pronounced pattern. It is this pronounced pattern that may be eliminated by a suitable power transform.

Suppose $\boldsymbol{y}$ is a positive array given by $y_{ij} = \kappa + \alpha_i + \beta_j + \rho\alpha_i\beta_j$ where κ is large compared to α and β. Now consider the binomial expansion of the data raised to the power r,

$$\begin{aligned} y_{ij}^r &= \kappa^r \left(1 + \frac{\alpha_i}{\kappa} + \frac{\beta_j}{\kappa} + \frac{\rho\alpha_i\beta_j}{\kappa}\right)^r \\ &= \kappa^r \left(1 + \frac{r\alpha_i}{\kappa} + \frac{r\beta_j}{\kappa} + \frac{r\rho\alpha_i\beta_j}{\kappa} \right. \\ &\qquad \left. + \frac{r(r-1)}{2}\left(\cdots + \frac{2\alpha_i\beta_j}{\kappa^2} + \ldots\right) + \ldots\right). \end{aligned} \tag{6.23}$$

So to be approximately additive we require

$$\frac{r\rho\alpha_i\beta_j}{\kappa} + \frac{r(r-1)}{2}\frac{2\alpha_i\beta_j}{\kappa^2} = 0$$

which leads to $r = 1 - \rho\kappa$.

To better visualise the pattern the data is placed in increasing order of their effects. From Figure 6.7(a) we see this is true of δ but not of μ. However, this

Error		δ				
		0.00	0.25	0.50	0.75	1.00
μ	0.0	−0.0973	2.9161	0.7799	−1.9296	−2.5182
	0.5	−2.3337	−2.7791	−0.8021	2.7967	6.4979
	1.0	−2.1533	−2.3701	0.0692	2.3598	3.7459
	1.5	1.4835	−0.2291	0.0816	−0.2522	−1.5649
	2.0	6.5234	2.1148	−0.2719	−3.2636	−7.0844

Table 6.1: Error in prediction obtained using Tukey's transformation for additivity.

problem is easily resolved by taking $\mu \geq 0$, where approximations will now be compared at $\{(\mu, \delta) | \mu \in \{0.0, 0.5, 1.0, 1.5, 2.0\}, \delta \in \{0.0, 0.25, 0.5, 0.75, 1.0\}\}$.

Let $\boldsymbol{y}$ be defined by $y_{ij} = \Theta(\mu = i/2, \delta = j/4) - \Theta(\mu = 0.0, \delta = 0.5)$ for $i, j = 0, 1, \ldots, 4$. First we fit the additive linear model $y_{i,j} \sim \kappa + \alpha_i + \beta_j$ to obtain estimates $\hat{\kappa}$, $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$. The residuals $\hat{\varepsilon}_{ij} = y_{ij} - \hat{\kappa} - \hat{\alpha}_i - \hat{\beta}_i$ are shown in Figure 6.9(b). Then we regress the residuals on to the explanatory variables $\hat{\alpha}_i \hat{\beta}_j$. The slope of this regression line is $\hat{\rho}$ and so the estimate of the power transform is $\hat{r} = 1 - \hat{\rho}\hat{\kappa}$. An additive linear model is fitted to the transformed data $y_{i,j}^{\hat{r}} \sim \kappa_r + \alpha_{r,i} + \beta_{r,j}$ to obtain estimates $\hat{\kappa}_r$, $\hat{\boldsymbol{\alpha}}_r$ and $\hat{\boldsymbol{\beta}}_r$. Finally the prediction is $\hat{\boldsymbol{y}} = (\hat{\kappa}_r + \hat{\boldsymbol{\alpha}}_r \hat{\boldsymbol{\beta}}_r^{\mathrm{T}})^{1/\hat{r}}$. Figure 6.9(c) shows the error in this estimate, it substantially improves upon the additive fit of the untransformed data and on all the other path sampling approximations above. However, although the errors are small relative to the other approximations, none are less than 0.05 (see Table 6.1). This is the error in approximating the PSE rather than the log normalising constant, but if we assume they are comparable an error of 0.05 corresponds to a 5% error in posterior inference.

For Tukey's transformation for additivity, if the data contains negative values a constant must be added to the data to make it non-negative. However, it is not necessary for the minimum, c, of the transformed data to be zero. We decided to investigate how c affects the additivity of the transformed data. Figure 6.10 summarises the results for $c \in [0.0, 1.0]$. The maximum R-squared value occurs when $c = 0.29$ but there is little change in the corresponding path sampling approximation (see Table 6.2).

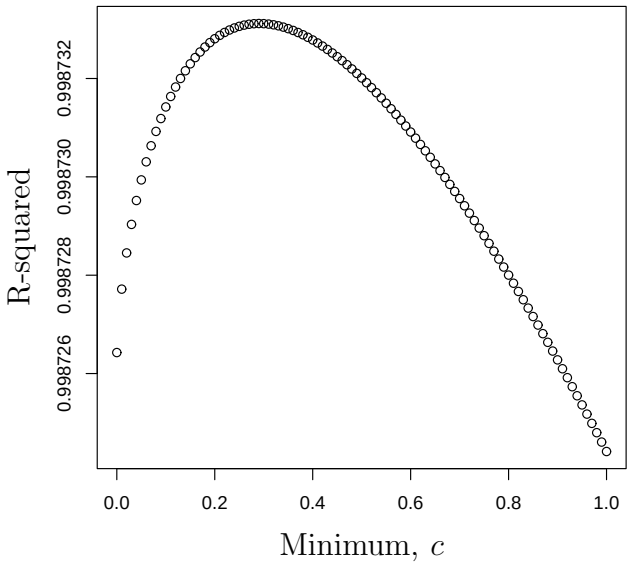


Figure 6.10: R-squared statistic for additive predictions of the transformed data versus the minimum value c.

In this analysis we have used the very data we wish to predict, in practice the row and column effects would be approximated by the additive path sampling integrals (see Equation 6.7(b)), and Tukey's transformation would be calculated for some control data and assumed to be appropriate more generally. We have investigated the effect of these further approximations and found that the results did not worsen significantly.

Although this method has not been successful it has highlighted a very important problem in trying to approximate the log normalising constant. Using this method we were able to find a prediction that fitted the data very well (R-squared = 0.9987), but because we need to take the exponential for posterior inference the prediction is still not good enough.

The normalising constant has proved to be an insurmountable problem in the use of the Ising model for the likelihood of the observed data given the simulator output. Furthermore, to generate sensible results for the flood inundation application the Ising model would need to be heterogeneous, which would add further

Error		δ				
		0.00	0.25	0.50	0.75	1.00
μ	0.0	0.0050	2.7836	0.5880	−2.1546	−2.7653
	0.5	−2.4493	−2.7439	−0.7617	2.8428	6.5502
	1.0	−2.3482	−2.3202	0.1332	2.4319	3.8240
	1.5	1.2472	−0.1703	0.1555	−0.1720	−1.4819
	2.0	6.2611	2.1813	−0.1919	−3.1805	−7.0020

Table 6.2: Error in prediction obtained using Tukey's transformation for additivity with minimum value $c = 0.29$.

computational expense. In the next chapter we discuss an extension of the binary channel model with the emphasis upon developing a model which is tractable.

Chapter 7

The Heterogeneous Binary Channel Model

In Chapter 6 we tried the Ising model for the likelihood of the observed flood extent given a simulation of flood extent. Unfortunately, calibration and calibrated prediction were not possible using the Ising model because of an intractable normalising constant. In this chapter we extend the BC model from Section 5.2 to represent heterogeneity and spatial dependence, the resulting model is called the heterogeneous binary channel (HBC) model. The aim in extending the BC model is to develop a likelihood for which calibration and calibrated prediction are possible. We describe calibration and calibrated prediction using the HBC model and present an MCMC algorithm for estimation. To investigate the effect of allowing $p(z_i \neq y_i|y_i, \boldsymbol{\phi}) > 0.5$, we introduce the positive heterogeneous binary channel (PHBC) model for which this situation is prevented. Heterogeneity is hard to visualise in two dimensions, so a one-dimensional example is used to illustrate the properties of the BC, HBC and PHBC models. Examples are also given using the Buscot dataset, and we discuss how within-model sampling may be used to improve mixing in the MCMC algorithm.

7.1 Introduction

In this chapter we extend the BC model introduced in Chapter 5 to allow for heterogeneity in the regression of $\boldsymbol{z}$ on $\boldsymbol{y}$. The Ising model discussed in Chapter 6 included spatial dependence and blur, but not heterogeneity, and could not be

applied at the scale of real flood events because of the intractable normalising constant. By returning to the BC model to make this extension we hope to develop a model that can be implemented at the scales required.

Heterogeneity is introduced to reduce the effect of local errors on global fit. In order to understand the implications of particular values of α and β in the BC model, assume α and β are fixed, and suppose $\boldsymbol{y}^\star$ can be obtained from $\boldsymbol{y}$ by changing one true-negative to a false-positive, more explicitly $\exists k$ s.t. $z_k = y_k = -1$, $y_k^\star = 1$, and $y_j^\star = y_j$, $\forall j \neq k$. Then, from Equations (5.5) and (5.6), assuming $p(\boldsymbol{y}) = p(\boldsymbol{y}^\star)$

$$\frac{p(\boldsymbol{y}^\star|\boldsymbol{z})}{p(\boldsymbol{y}|\boldsymbol{z})} = \frac{1-\alpha}{\beta}.$$

The nature of the flood inundation problem means the number of trues, $n_{1,1} + n_{-1,-1}$, will typically be much greater than the number of falses, $n_{-1,1} + n_{1,-1}$. This suggests $p(z_i = y_i|y_i)$ should be large, for example $\alpha = \beta = 0.8$ leading to $p(\boldsymbol{y}^\star|\boldsymbol{z})/p(\boldsymbol{y}|\boldsymbol{z}) = 0.25$. *A posteriori* $\boldsymbol{y}$ is four times more probable than $\boldsymbol{y}^\star$, although they differ by only one pixel, and this ratio is independent of the image size. We would favour a larger posterior ratio, say $p(\boldsymbol{y}^\star|\boldsymbol{z})/p(\boldsymbol{y}|\boldsymbol{z}) = 0.9$, but, assuming $\alpha = \beta$, this requires $\alpha = \beta = 0.53$, so given $\boldsymbol{y}$ we are still very uncertain about the value of $\boldsymbol{z}$. By introducing heterogeneity we allow $p(z_i = y_i|y_i, \alpha_i, \beta_i)$ to vary across the floodplain. For example it may be close to 0.5 near the flood boundary and close to 1.0 away from it.

A posteriori $\boldsymbol{y}^\star$ should be less probable than $\boldsymbol{y}$, because the former is obtained from the latter by changing one true-negative to a false-positive. However, *where* this change is made is also important. If the new false-positive is part of a region of false-positives it should be penalised less than if it is entirely isolated from other false-positives. We introduce spatial dependence between the distributed parameters so a block of t false-positives is penalised less than t individual false-positives (similarly false-negatives).

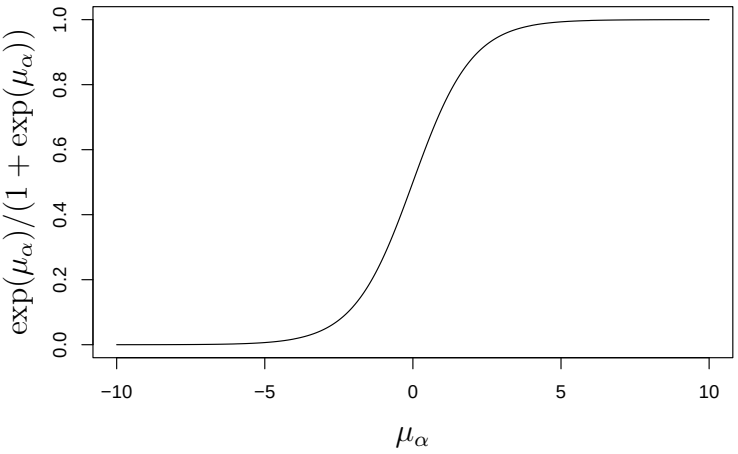


Figure 7.1: The relationship between μ_α and $p(z_i = 1|y_i = 1, \mu_\alpha)$.

7.2 The Heterogeneous Binary Channel Model

In this section we describe the HBC model equations and show how the model may be used within our Bayesian framework for calibration and calibrated prediction.

7.2.1 Likelihood

When the BC model was introduced in Section 5.2, a parameterisation was adopted that meant the posterior distributions could be calculated analytically. However, this parameterisation does not lend itself to a natural heterogeneous extension, and we therefore consider the following alternative parameterisation that makes use of the logistic transform,

$$p(z_i = 1|y_i = 1, \mu_\alpha) = \frac{\exp(\mu_\alpha)}{1 + \exp(\mu_\alpha)} (= \alpha) \quad \text{and} \tag{7.1}$$

$$p(z_i = -1|y_i = -1, \mu_\beta) = \frac{\exp(\mu_\beta)}{1 + \exp(\mu_\beta)} (= \beta) \tag{7.2}$$

where z_i are conditionally independent given y_i, and we take priors $\mu_\alpha \sim \mathcal{N}(\nu_\alpha, \sigma_\alpha^2)$ and $\mu_\beta \sim \mathcal{N}(\nu_\beta, \sigma_\beta^2)$. Figure 7.1 shows how $p(z_i = 1|y_i = 1, \mu_\alpha)$ changes with the value of μ_α.

The corresponding HBC model is

$$p(z_i = 1|y_i = 1, \mu_\alpha, \varepsilon_{\alpha,i}) = \frac{\exp(\mu_\alpha + \varepsilon_{\alpha,i})}{1 + \exp(\mu_\alpha + \varepsilon_{\alpha,i})}$$
$$p(z_i = -1|y_i = -1, \mu_\beta, \varepsilon_{\beta,i}) = \frac{\exp(\mu_\beta + \varepsilon_{\beta,i})}{1 + \exp(\mu_\beta + \varepsilon_{\beta,i})}$$

where z_i are conditionally independent given y_i, and $\mu_\alpha, \mu_\beta \in \mathbb{R}$ and $\boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta \in \mathbb{R}^n$. The likelihood is

$$p(\boldsymbol{z}|\boldsymbol{y}, \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta) = \prod_{i=1}^{n} \left(\frac{\exp\left(\left(\frac{z_i+1}{2}\right)(\mu_\alpha + \varepsilon_{\alpha,i})\right)}{1 + \exp(\mu_\alpha + \varepsilon_{\alpha,i})} \right)^{\frac{y_i+1}{2}} \left(\frac{\exp\left(\left(\frac{1-z_i}{2}\right)(\mu_\beta + \varepsilon_{\beta,i})\right)}{1 + \exp(\mu_\beta + \varepsilon_{\beta,i})} \right)^{\frac{1-y_i}{2}}.$$

7.2.2 Prior Distributions

We now define the conditional (or marginal) distributions for each node of the DAG in Figure 5.2. The node $\boldsymbol{\phi}$ now corresponds to the parameters μ_α, μ_β, $\boldsymbol{\varepsilon}_\alpha$ and $\boldsymbol{\varepsilon}_\beta$, each of which is independent of the others.

The prior for m is still discrete uniform. Given m, $\boldsymbol{y}$ and $\boldsymbol{y}'$ are deterministic, being $\boldsymbol{y}^{(m)}$ and $\boldsymbol{y}'^{(m)}$ respectively, see Equations (5.3) and (5.4).

For the HBC model parameters we take

$$\mu_\alpha \sim \mathcal{N}(\nu_\alpha, \sigma_\alpha^2), \tag{7.3}$$
$$\mu_\beta \sim \mathcal{N}(\nu_\beta, \sigma_\beta^2), \tag{7.4}$$
$$\boldsymbol{\varepsilon}_\alpha \sim \mathcal{MVN}\left(\mathbf{0}, \tau_\alpha^2(\boldsymbol{I} - \lambda_\alpha \boldsymbol{C})^{-1}\right) \quad \text{and} \tag{7.5}$$
$$\boldsymbol{\varepsilon}_\beta \sim \mathcal{MVN}\left(\mathbf{0}, \tau_\beta^2(\boldsymbol{I} - \lambda_\beta \boldsymbol{C})^{-1}\right) \tag{7.6}$$

where $\nu_\alpha, \nu_\beta \in \mathbb{R}$, $\sigma_\alpha, \sigma_\beta, \tau_\alpha, \tau_\beta \in \mathbb{R}_{\geq 0}$, $\lambda_\alpha, \lambda_\beta \in [0, 1)$, and

$$\boldsymbol{C}_{i,j} = \begin{cases} \frac{1}{4} & \text{if } i \text{ and } j \text{ are neighbours} \\ 0 & \text{otherwise,} \end{cases} \tag{7.7}$$

and we assume toroidal boundary conditions, so the East/South neighbours of pixels in the last column/row are pixels in the first column/row. The toroidal assumption means the precision matrix $\boldsymbol{I} - \lambda_\alpha \boldsymbol{C}$ is block-circulant (see Section 8.3).

Block-circulant matrices can be related to the two-dimensional fast Fourier transform for fast matrix inversion and fast multivariate Normal sampling (see Rue and Held, 2005). Although toroidal boundary conditions seem inappropriate for the flood inundation application, any adverse boundary effects can be reduced by adding an artificial frame around the data (see Weir and Pettitt, 1999). For an overview of other approaches to the boundary condition problem see Cressie (1993).

The hyperparameters σ_α, σ_β, τ_α, τ_β, λ_α and λ_β are fixed. If $\tau_\alpha = \tau_\beta = 0$, then $\boldsymbol{\varepsilon}_\alpha = \boldsymbol{\varepsilon}_\beta = \mathbf{0}$ and the HBC model degenerates to the BC model of Equations (7.1) and (7.2).

7.2.3 Posterior, Calibration and Calibrated Prediction

The posterior is

$$p(m, \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta | \boldsymbol{z}) \propto$$
$$\prod_{i=1}^{n} \left(\frac{\exp\left(\left(\frac{z_i+1}{2}\right)(\mu_\alpha + \varepsilon_{\alpha,i})\right)}{1+\exp\left(\mu_\alpha + \varepsilon_{\alpha,i}\right)} \right)^{\frac{y_i^{(m)}+1}{2}} \left(\frac{\exp\left(\left(\frac{1-z_i}{2}\right)(\mu_\beta + \varepsilon_{\beta,i})\right)}{1+\exp\left(\mu_\beta + \varepsilon_{\beta,i}\right)} \right)^{\frac{1-y_i^{(m)}}{2}}$$
$$\exp\left(-\frac{1}{2\sigma_\alpha^2}(\mu_\alpha - \nu_\alpha)^2 - \frac{1}{2\sigma_\beta^2}(\mu_\beta - \nu_\beta)^2 \right.$$
$$\left. -\frac{1}{2\tau_\alpha^2}\boldsymbol{\varepsilon}_\alpha^{\mathrm{T}}(\boldsymbol{I} - \lambda_\alpha \boldsymbol{C})\boldsymbol{\varepsilon}_\alpha - \frac{1}{2\tau_\beta^2}\boldsymbol{\varepsilon}_\beta^{\mathrm{T}}(\boldsymbol{I} - \lambda_\beta \boldsymbol{C})\boldsymbol{\varepsilon}_\beta \right). \quad (7.8)$$

For the BC model of Section 5.2 it was possible to integrate the posterior to find analytical expressions for $p(m|\boldsymbol{z})$ and $p(z_i' = 1|\boldsymbol{z})$. For the HBC model it is not possible to find these quantities analytically, so we will estimate them using MCMC (see Section 3.2).

MCMC is used to generate an estimate sample from the posterior $p(m, \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})$, $\left\{m^{(k)}, \mu_\alpha^{(k)}, \mu_\beta^{(k)}, \boldsymbol{\varepsilon}_\alpha^{(k)}, \boldsymbol{\varepsilon}_\beta^{(k)} | k = 1, \ldots, K\right\}$. Analytically, we would obtain $p(m|\boldsymbol{z})$ from $p(m, \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})$ by integrating out the other parameters, but given a sample from the joint posterior, if we simply discard the values of the other parameters then $\left\{m^{(k)}|k = 1, \ldots, K\right\}$ is a sample from $p(m|\boldsymbol{z})$.

To make calibrated predictions about a future event $\boldsymbol{z}'$ based on simulations of the future event $\boldsymbol{y}'^{(1)}, \boldsymbol{y}'^{(2)}, \ldots, \boldsymbol{y}'^{(M)}$, we let $p(\boldsymbol{z}'|\boldsymbol{y}', \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta)$ be a HBC

model, similar to $p(\boldsymbol{z}|\boldsymbol{y}, \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta)$, and assume the model parameters are the same for each. Then the calibrated predictions are

$$\begin{aligned} p(z'_i = 1|\boldsymbol{z}) &= \sum_{m=1}^{M} \iiiint p(z'_i = 1|\boldsymbol{y}'^{(m)}, \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta) \\ &\qquad \times p(m, \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z}) \, \mathrm{d}\mu_\alpha \, \mathrm{d}\mu_\beta \, \mathrm{d}\boldsymbol{\varepsilon}_\alpha \, \mathrm{d}\boldsymbol{\varepsilon}_\beta \\ &\approx \frac{1}{K} \sum_{k=1}^{K} p(z'_i = 1|\boldsymbol{y}'^{(m^{(k)})}, \mu_\alpha^{(k)}, \mu_\beta^{(k)}, \boldsymbol{\varepsilon}_\alpha^{(k)}, \boldsymbol{\varepsilon}_\beta^{(k)}). \end{aligned} \tag{7.9}$$

7.3 MCMC Algorithm

In this section we describe an MCMC algorithm for sampling from the posterior in Equation (7.8), and then discuss the practical issues in using this algorithm.

Initial values must be defined for the Markov chain, we choose to set the parameters of the HBC model to their distribution means, $\mu_\alpha = \nu_\alpha$, $\mu_\beta = \nu_\beta$ and $\boldsymbol{\varepsilon}_\alpha = \boldsymbol{\varepsilon}_\beta = \mathbf{0}$, and take an arbitrary simulation, $m = 1$.

7.3.1 m Update

In flood inundation applications the simulations $\boldsymbol{y}^{(m)}$, $m = 1, 2, \ldots, M$ can be ordered according to the friction values $\boldsymbol{\theta}^{(m)}$ that we used to generate them, or by some pixel statistic, e.g. the number of positives, $n_{\cdot,1}^{(m)} = \sum_{i=1}^{n} \mathbf{1}[y_i^{(m)} = 1]$. These orderings may be exploited in the proposal distribution to improve mixing. For example for the probability of proposing m' from m we may take $q(m'|m) \propto \left|\boldsymbol{\theta}^{(m)} - \boldsymbol{\theta}^{(m')}\right|^{-1}$.

Propose a new value m' from $q(m'|m)$, then the proposal ratio is $q(m|m')/q(m'|m)$, and the posterior ratio is

$$\begin{aligned} \frac{p(m', \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})}{p(m, \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})} = \prod_{i=1}^{n} &\left(\frac{\exp\left(\left(\frac{z_i+1}{2}\right)(\mu_\alpha + \varepsilon_{\alpha,i})\right)}{1 + \exp(\mu_\alpha + \varepsilon_{\alpha,i})} \right)^{\frac{y_i^{(m')} - y_i^{(m)}}{2}} \\ &\times \left(\frac{\exp\left(\left(\frac{1-z_i}{2}\right)(\mu_\beta + \varepsilon_{\beta,i})\right)}{1 + \exp(\mu_\beta + \varepsilon_{\beta,i})} \right)^{\frac{y_i^{(m)} - y_i^{(m')}}{2}}. \end{aligned} \tag{7.10}$$

The acceptance probability for m' is the minimum of 1.0 and the product of the

posterior ratio and proposal ratio. In the next section we describe an algorithm for fast sampling from an arbitrary discrete distribution, such as $q(m'|m)$.

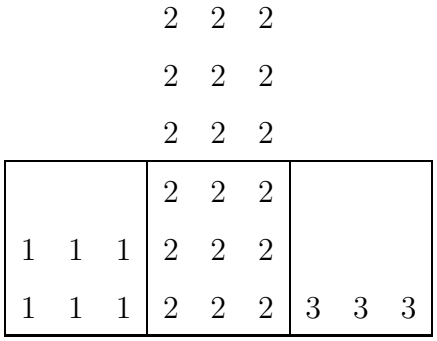
7.3.2 Robin Hood Method for Sampling from a Discrete Distribution

Let X be a discrete random variable taking values in $\{1, 2, \ldots, M\}$. If X has a discrete uniform distribution on $\{1, 2, \ldots, M\}$ and u is a sample from $\mathcal{U}(0, 1]$, then $x = \lceil Mu \rceil$ (the smallest integer greater than Mu) is a sample from the distribution of X.

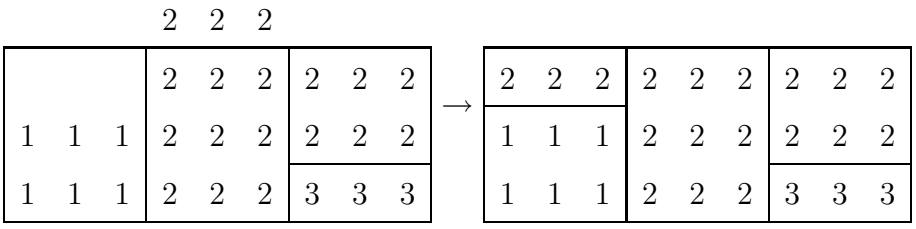
More generally, let $f(x) = P(X = x)$ be the probability mass function of X. The distribution function $F(x) = \sum_{i \leq x} f(x)$ is a step function, and if u is a sample from $\mathcal{U}(0, 1]$ then $x = F^{-1}(u)$ is a sample from X. A very simple practical way of finding a sample from X given u is to identify that $x \in \{1, 2, \ldots, M\}$ for which $F(x - 1) < u \leq F(x)$, but this is very inefficient.

Marsaglia *et al.* (2004) describe a number of methods for fast generation of discrete random variables. The Robin Hood method, originally devised by Walker (1977), requires some preliminary calculations to be done offline and then sampling is very efficient. We describe a simple example of the method from which the extension to the general case is obvious.

Suppose X takes values $1, 2, 3$ with probabilities $2/9, 6/9$ and $1/9$. The target is to form a *square histogram*, which has three equal-width columns and a height of $3/9$. The bottom part of the column belongs to the index $1, 2, 3$ and the top part to the index of the variable that is represented in the top row of that column. Start by forming a standard histogram of the probabilities and superimpose the target square histogram:



The Robin Hood method is iterative, we take from the "richest" and give to the "poorest" until the histogram is square:



The top part of the columns belong to $K = (2, 2, 2)$, and the cumulative division point for each column is $V = (2/9, 3/9 + 3/9 = 6/9, 6/9 + 1/9 = 7/9)$. The rule for generating a random number from this distribution is:

1. $u \sim \mathcal{U}(0, 1]$,
2. $j = \lceil 3u \rceil$; if $u < V_j$ return j, else return K_j.

The vectors K and V can be calculated offline. This can be done for any discrete probability distribution.

For a general discrete random variable X, first initialise $K_i = i$ and $V_i = i/M$ then repeat the following steps $M - 1$ times:

1. Find the largest and smallest probabilities, say $f(j)$ and $f(i)$.
2. Set $K_i = j$, $V_i = (i - 1)/M + f(i)$.
3. Replace $f(j)$ by $f(j) - (1/M - f(i))$ and $f(i)$ by $1/M$.

Then given $u \sim \mathcal{U}(0, 1]$, $j = \lceil Mu \rceil$ and a sample from X is j if $u < V_j$ and K_j otherwise.

For the update of the simulation index K and V must be computed offline for each m, but, crucially, the vectors do not change throughout the MCMC algorithm.

7.3.3 μ_α and μ_β Updates

Propose a new value μ'_α from $\mathcal{U}[\mu_\alpha - f_\alpha, \mu_\alpha + f_\alpha]$, then the proposal ratio will always be 1.0 and the posterior ratio is

$$\frac{p(m, \mu'_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})}{p(m, \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})} = \prod_{i=1}^{n} \left(\exp\left(\left(\frac{z_i + 1}{2} \right) (\mu'_\alpha - \mu_\alpha) \right) \frac{1 + \exp(\mu_\alpha + \varepsilon_{\alpha,i})}{1 + \exp(\mu'_\alpha + \varepsilon_{\alpha,i})} \right)^{\frac{y_i^{(m)}+1}{2}} \exp\left(-\frac{1}{2\sigma_\alpha^2} \left((\mu'_\alpha - \nu_\alpha)^2 - (\mu_\alpha - \nu_\alpha)^2 \right) \right).$$

The acceptance probability is the minimum of 1.0 and the posterior ratio. The update for μ_β is similar.

7.3.4 $\boldsymbol{\varepsilon}_\alpha$ and $\boldsymbol{\varepsilon}_\beta$ Updates

We update $\boldsymbol{\varepsilon}_\alpha$ pixel by pixel. Fortunately the full conditionals take the very simple form

$$\varepsilon_{\alpha,i}|\boldsymbol{\varepsilon}_{\alpha,-i} \sim \mathcal{N}\left(\frac{\lambda_\alpha}{4} \sum_{j \in \delta i} \varepsilon_{\alpha,j}, \tau_\alpha^2 \right)$$

where δi is the set of neighbours of pixel i. For pixel i propose a new value $\varepsilon'_{\alpha,i}$ from $\mathcal{U}[\varepsilon_{\alpha,i} - d_\alpha, \varepsilon_{\alpha,i} + d_\alpha]$ so the proposal ratio is 1.0. Then the posterior ratio is

$$\frac{p(m, \mu_\alpha, \mu_\beta, \varepsilon'_{\alpha,i}, \boldsymbol{\varepsilon}_{\alpha,-i}, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})}{p(m, \mu_\alpha, \mu_\beta, \varepsilon_{\alpha,i}, \boldsymbol{\varepsilon}_{\alpha,-i}, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})} = \left(\exp\left(\left(\frac{z_i + 1}{2} \right) (\varepsilon'_{\alpha,i} - \varepsilon_{\alpha,i}) \right) \frac{1 + \exp(\mu_\alpha + \varepsilon_{\alpha,i})}{1 + \exp(\mu_\alpha + \varepsilon'_{\alpha,i})} \right)^{\frac{y_i^{(m)}+1}{2}} \times \exp\left(-\frac{1}{2\tau_\alpha^2} (\varepsilon'_{\alpha,i} - \varepsilon_{\alpha,i}) \left(\varepsilon'_{\alpha,i} + \varepsilon_{\alpha,i} - \frac{\lambda_\alpha}{2} \sum_{j \in \delta i} \varepsilon_{\alpha,j} \right) \right).$$

The acceptance probability is the minimum of 1.0 and the posterior ratio. The update for $\boldsymbol{\varepsilon}_\beta$ is similar.

7.3.5 Underflow and Overflow

The algorithm is written in the C programming language. To calculate the acceptance probabilities we must take the product of many terms. When this product is very small the computer may equate it to zero, this is called *underflow*. When the product is very large the computer may equate it to infinity, this is called *overflow*.

Computers are far better at dealing with sums, so rather than calculate the products directly we take the logarithm and calculate the sum. For example, the logarithm of Equation (7.10) for the m update is

$$\sum_{i=1}^{n}\left(\frac{y_i^{(m')}-y_i^{(m)}}{2}\right)\Bigg(\left(\frac{z_i+1}{2}\right)(\mu_\alpha+\varepsilon_{\alpha,i})-\left(\frac{1-z_i}{2}\right)(\mu_\beta+\varepsilon_{\beta,i}) \\ -\log(1+\exp(\mu_\alpha+\varepsilon_{\alpha,i}))+\log(1+\exp(\mu_\beta+\varepsilon_{\beta,i}))\Bigg).$$

The terms of the form $\log(1+\exp(x))$ require special treatment because if x is too small or too big the $\exp(\cdot)$ function can underflow or overflow respectively. Note that

$$\lim_{x\to-\infty}\log(1+\exp(x))=0$$

and, from $\log(1+\exp(x))=x+\log(\exp(-x)+1)$, that

$$\lim_{x\to\infty}\log(1+\exp(x))=x.$$

To calculate $\log(1+\exp(x))$ within the code we first check the value of x: if $x<-50$ we return 0.0; if $x>50$ we return x; if $-50\leq x\leq 50$ underflow and overflow are not a problem using double precision on a Pentium 4 2GHz processor with 512MB of RAM, and we evaluate the expression directly.

7.4 Forcing Positive Regression

When we introduced the BC model in Section 5.2 we made no issue of the fact that the parameters α and β can be less than 0.5. It would certainly be very peculiar if either of these parameters were less than 0.5, for example if $\alpha<0.5$ then

$$p(z_i=-1|y_i=1,\alpha)>p(z_i=1|y_i=1,\alpha)$$

for *all* $i \in \{1, \ldots, n\}$. However, although possible, the posterior probability α or β are less than 0.5 is very small because for typical $\boldsymbol{z}$ and $\boldsymbol{y}$ the number of trues is much greater than the number of falses, and therefore the likelihood is much greater for $\alpha, \beta > 0.5$. The only way the posterior would favour values of $\alpha, \beta < 0.5$ is if the prior strongly requires this.

The HBC model differs from the BC model in that $p(z_i = 1|y_i = 1, \mu_\alpha, \varepsilon_{\alpha,i})$ and $p(z_i = -1|y_i = -1, \mu_\beta, \varepsilon_{\beta,i})$ may be different for different i. If there is no spatial dependence, $\lambda_\alpha = \lambda_\beta = 0.0$, then it is entirely feasible, even likely, that posterior values of μ_α and $\varepsilon_{\alpha,i}$ lead to $p(z_i = 1|y_i = 1, \mu_\alpha, \varepsilon_{\alpha,i}) < 0.5$ for some i, similarly for $p(z_i = -1|y_i = -1, \mu_\beta, \varepsilon_{\beta,i})$. In this section we look at a variation of the HBC model in which these probabilities are constrained to be ≥ 0.5, we will call it the *positive heterogeneous binary channel (PHBC) model*. The model equations are

$$p(z_i = 1|y_i = 1, \mu_\alpha, \varepsilon_{\alpha,i}) = \frac{1 + 2\exp(\mu_\alpha + \varepsilon_{\alpha,i})}{2 + 2\exp(\mu_\alpha + \varepsilon_{\alpha,i})}$$
$$p(z_i = -1|y_i = -1, \mu_\beta, \varepsilon_{\beta,i}) = \frac{1 + 2\exp(\mu_\beta + \varepsilon_{\beta,i})}{2 + 2\exp(\mu_\beta + \varepsilon_{\beta,i})}$$

where μ_α, μ_β, $\boldsymbol{\varepsilon}_\alpha$ and $\boldsymbol{\varepsilon}_\beta$ have the same priors as before. There are many constructions that encode this constraint, this one is chosen so $p(z_i = 1|y_i = 1, \mu_\alpha = 0, \varepsilon_{\alpha,i} = 0) = 0.75$.

The posterior distribution is

$$p(m, \mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z}) =$$
$$\prod_{i=1}^{n} \left(\frac{(1 + 2\exp(\mu_\alpha + \varepsilon_{\alpha,i}))^{\frac{z_i+1}{2}}}{2 + 2\exp(\mu_\alpha + \varepsilon_{\alpha,i})} \right)^{\frac{y_i+1}{2}} \left(\frac{(1 + 2\exp(\mu_\beta + \varepsilon_{\beta,i}))^{\frac{1-z_i}{2}}}{2 + 2\exp(\mu_\beta + \varepsilon_{\beta,i})} \right)^{\frac{1-y_i}{2}}$$
$$\exp\left(-\frac{1}{2\sigma_\alpha^2}(\mu_\alpha - \nu_\alpha)^2 - \frac{1}{2\sigma_\beta^2}(\mu_\alpha - \nu_\beta)^2 \right.$$
$$\left. - \frac{1}{2\tau_\alpha^2}\boldsymbol{\varepsilon}_\alpha^{\mathrm{T}}(\boldsymbol{I} - \lambda_\alpha \boldsymbol{C})\boldsymbol{\varepsilon}_\alpha - \frac{1}{2\tau_\beta^2}\boldsymbol{\varepsilon}_\beta^{\mathrm{T}}(\boldsymbol{I} - \lambda_\beta \boldsymbol{C})\boldsymbol{\varepsilon}_\beta \right).$$

As for the HBC model, calibration and calibrated prediction cannot be done analytically and so we use MCMC. The algorithm used is identical to that used for the HBC model (see Section 7.3) except for the likelihood ratio.

7.5 One-Dimensional Toy Example

In this section we demonstrate the impact of different prior assumptions, i.e. different models and different hyperparameter settings, on posterior inference in a toy example.

The heterogeneous characteristics of the HBC model are difficult to visualise using a two-dimensional dataset. We introduce a simple one-dimensional dataset that aims to represent some of the features we expect in two-dimensional flood inundation data.

Figure 7.2 shows the one-dimensional dataset for our toy example. The dataset consists of one observation, $\boldsymbol{z} \in \{-1, 1\}^n$, and 29 simulations, $\boldsymbol{y}^{(m)} \in \{-1, 1\}^n$, $m = 1, \ldots, 29$. Each is $n = 50$ pixels long, and only the central 10 pixels of the observed data are positive. There are no false-negatives. We consider the ways in which to add t false-positives for $t = 1, 2, \ldots, 10$: first as a block away from the boundary,

$$m \in \{2, 4, 7, 10, 13, 16, 19, 22, 25, 28\}$$

second as a block on the boundary,

$$m \in \{1, 3, 6, 9, 12, 15, 18, 21, 24, 27\}$$

and third as t isolated errors,

$$m \in \{2, 5, 8, 11, 14, 17, 20, 23, 26, 29\}.$$

Note that $\boldsymbol{y}^{(2)}$ qualifies for both the individual error and block (of 1) away from the boundary categories.

The two-dimensional HBC and PHBC models (see Sections 7.2 and 7.4) require a small modification for use with one-dimensional data. The two-dimensional toroidal boundary conditions become cyclic boundary conditions in one dimension, and the precision matrix in Equations (7.5) and (7.6) becomes

$$\boldsymbol{C}_{i,j} = \begin{cases} \frac{1}{2} & \text{if } i \text{ and } j \text{ are neighbours} \\ 0 & \text{otherwise.} \end{cases}$$

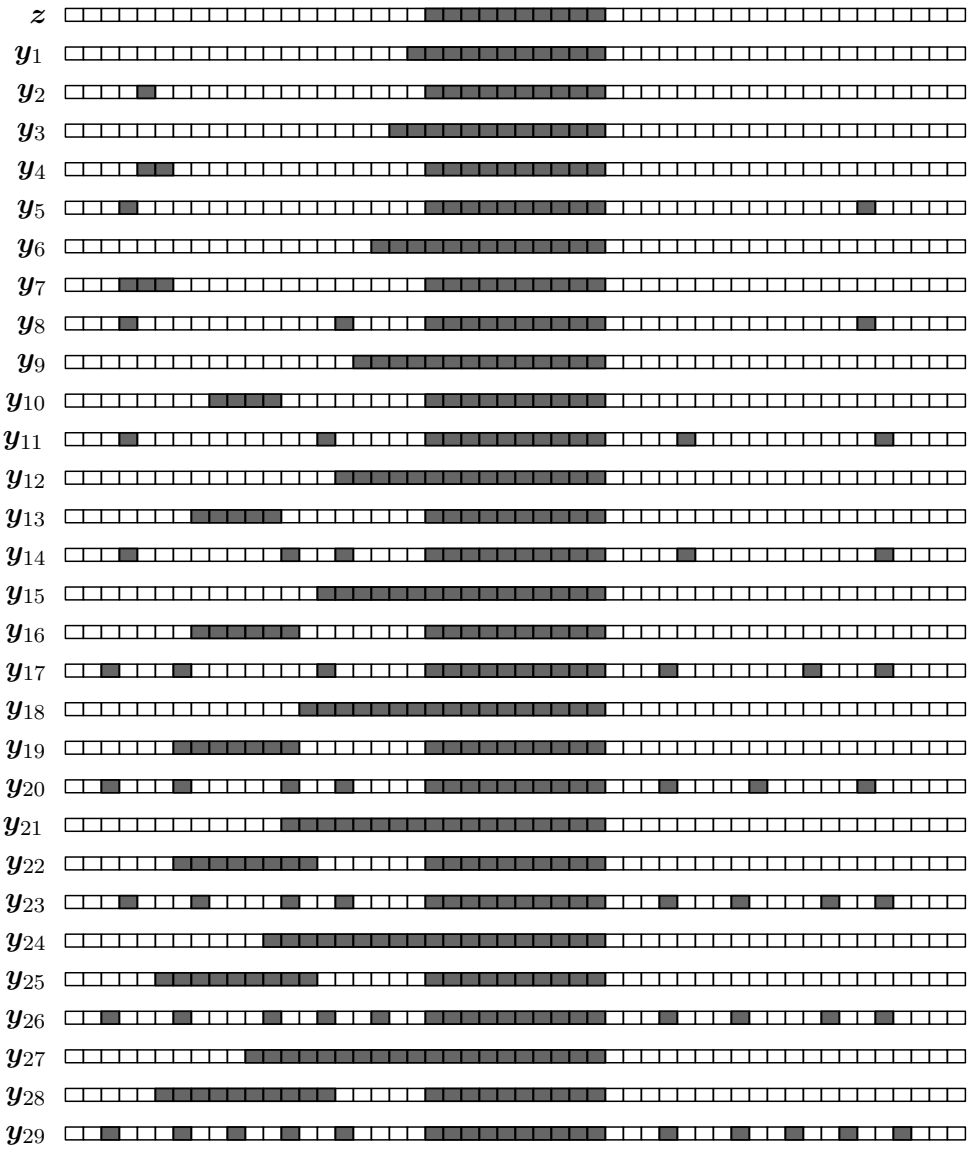


Figure 7.2: One-dimensional toy example for illustrating the characteristics of the HBC and PHBC models. Pixel values of 1 are grey and -1 are white.

In the following discussion we will refer to the two parameterisations of the BC model, introduced in Sections 5.2 and 7.2, as the (α, β) BC model and the (μ_α, μ_β) BC model respectively.

In Section 5.3 we discussed how to make the (α, β) BC model penalise false-positives more than false-negatives (or vice-versa) by choice of the hyperparameters a, b, c, d, but for our toy example there are no false-negatives so in all the examples in this section we take $\nu_\alpha = \nu_\beta = \nu$, $\sigma_\alpha = \sigma_\beta = \sigma$, $\lambda_\alpha = \lambda_\beta = \lambda$ and $\tau_\alpha = \tau_\beta = \tau$.

For comparison to the (α, β) BC model, and for ease of interpretation, we plot the density induced on $\alpha_i = p(z_i = 1|y_i = 1, \mu_\alpha, \varepsilon_{\alpha,i})$ and $\beta_i = p(z_i = -1|y_i = -1, \mu_\beta, \varepsilon_{\beta,i})$ by the priors $\mu_\alpha \sim \mathcal{N}(\nu_\alpha, \sigma_\alpha)$, $\mu_\beta \sim \mathcal{N}(\nu_\beta, \sigma_\beta)$, $\boldsymbol{\varepsilon}_\alpha \sim \mathcal{MVN}(\mathbf{0}, \tau_\alpha(\boldsymbol{I} - \lambda_\alpha \boldsymbol{C})^{-1})$ and $\boldsymbol{\varepsilon}_\beta \sim \mathcal{MVN}(\mathbf{0}, \tau_\beta(\boldsymbol{I} - \lambda_\beta \boldsymbol{C})^{-1})$ in the examples which follow. As $\boldsymbol{C}$ is a block-circulant matrix the marginal variance for $\varepsilon_{\alpha,i} \sim \mathcal{N}(0, s_\alpha^2)$ is τ_α^2 times the mean of the inverse eigenvalues, $s_\alpha^2 = \tau_\alpha^2/n \sum_{i=0}^{n-1}(1 - \lambda_\alpha \cos(2\pi i/n))^{-1}$ (see Moran (1973) and Section 8.3). Let $\psi_i = \mu_\alpha + \varepsilon_{\alpha,i}$ so $\alpha_i = \exp(\psi_i)/(1 + \exp(\psi_i))$, then because μ_α and $\varepsilon_{\alpha,i}$ are independent $\psi_i \sim \mathcal{N}(\nu_\alpha, \sigma_\alpha^2 + s_\alpha^2)$, and by the change of variables formula

$$p(\alpha_i) = \frac{1}{\alpha_i(1-\alpha_i)\sqrt{2\pi(\sigma_\alpha^2 + s_\alpha^2)}} \exp\left(-\frac{1}{2(\sigma_\alpha^2 + s_\alpha^2)}\left(\log\left(\frac{\alpha_i}{1-\alpha_i}\right) - \nu_\alpha\right)^2\right).$$

The prior for β_i is found similarly. (For examples see Figures 7.4(a) and 7.4(b).)

The posteriors for μ_α, μ_β, $\boldsymbol{\varepsilon}_\alpha$ and $\boldsymbol{\varepsilon}_\beta$ given $\boldsymbol{z}$ are not visually very informative, so we choose to plot the densities induced on $\alpha_i = p(z_i' = 1|y_i' = 1, \mu_\alpha, \varepsilon_{\alpha,i})$ and $\beta_i = p(z_i' = -1|y_i' = -1, \mu_\beta, \varepsilon_{\beta,i})$ by the posterior $p(\mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})$. Given the MCMC sample for μ_α and $\varepsilon_{\alpha,i}$, $\{\mu_\alpha^{(k)}, \varepsilon_{\alpha,i}^{(k)} : k = 1, \ldots, K\}$, we calculate $p(z_i' = 1|y_i' = 1, \mu_\alpha^{(k)}, \varepsilon_{\alpha,i}^{(k)})$ for $k = 1, \ldots, K$. We plot the mean $p(z_i' = 1|y_i' = 1, \boldsymbol{z})$, and the upper and lower quartiles to show spread. For the BC model these plots will always consist of horizontal lines, but are included nevertheless for comparison to the HBC model (see Figures 7.3(c) and 7.3(d)).

7.5.1 (μ_α, μ_β) BC Model Example

We discussed the properties of the (α, β) BC model in Section 5.3. Rather than repeat that analysis here, we will consider how the properties relate to the (μ_α, μ_β)

BC model. A small set of summary results for the (μ_α, μ_β) BC model are presented for comparison to the HBC model (see Figure 7.3).

We found it necessary to take priors for α and β that favoured values very close to 0.5, to prevent the posterior for the simulation index $p(m|\boldsymbol{z})$ being zero for almost all m not equal to the posterior mode. In the (μ_α, μ_β) BC model examples which follow we will take $\nu = 0.0$ corresponding to a prior expectation of 0.5 for α and β. The relationship between the standard deviation σ and the spread of α and β about 0.5 is not obvious so we present a summary using different values for σ in Figure 7.3.

The prior distributions induced on $\alpha = p(z_i = 1|y_i = 1, \mu_\alpha)$ and $\beta = p(z_i = -1|y_i = -1, \mu_\beta)$ by $\mu_\alpha \sim \mathcal{N}(0, \sigma^2)$ and $\mu_\beta \sim \mathcal{N}(0, \sigma^2)$ are shown in Figures 7.3(a) and 7.3(b). Note that these prior distributions become bimodal as σ increases. At first bimodal distributions seem inappropriate. However, for the flood inundation problem we are uncertain about the value of the observed data given a simulator output only near the flood boundary. Within the channel and on the floodplain away from the boundary we expect the value of the observed data to be the same as a simulator output, except where the simulator is consistently wrong, where we expect the value of the observed data to be the opposite of a simulator output.

The posterior distributions induced on $\alpha = p(z'_i = 1|y'_i = 1, \mu_\alpha)$ and $\beta = p(z'_i = -1|y'_i = -1, \mu_\beta)$ by $p(\mu_\alpha, \mu_\beta|\boldsymbol{z})$, are shown in Figures 7.3(c) and 7.3(d). With every doubling of σ, $p(z'_i = 1|y'_i = 1, \boldsymbol{z})$ and $p(z'_i = -1|y'_i = -1, \boldsymbol{z})$ increase, although the rate of increase decreases. Also, $p(z'_i = -1|y'_i = -1, \boldsymbol{z}) > p(z'_i = 1|y'_i = 1, \boldsymbol{z})$ because the ratio of true-negatives to false-negatives is greater than the ratio of true-positives to false-positives. The lines are horizontal because the model parameters are homogeneous, but (with reference to the data in Figure 7.2) we would like to allow $p(z'_i = y'_i|y'_i, \boldsymbol{z})$ to be larger in some regions than others, and possibly even < 0.5 in regions of systematic error.

The posterior for the simulation index, $p(m|\boldsymbol{z})$, is plotted versus the number of falses, $n^{(m)}_{-1,1} + n^{(m)}_{1,-1}$, in Figure 7.3(e). As σ increases, falses are penalised more and $p(m|\boldsymbol{z})$ becomes negligible for more m. Note that the different configurations of

falses have no bearing on $p(m|\boldsymbol{z})$.

Figure 7.3(f) shows the calibrated predictions. As σ increases $|p(z_i = 1|\boldsymbol{z}) - 0.5|$ generally increases, expressing more confidence in our simulations. However, $|p(z_5 = 1|\boldsymbol{z}) - 0.5|$ decreases slightly – the BC model does not correct for the systematic error at pixel 5 in the simulations.

We are interested in the two tasks of calibration and calibrated prediction, about which we can form two objectives:

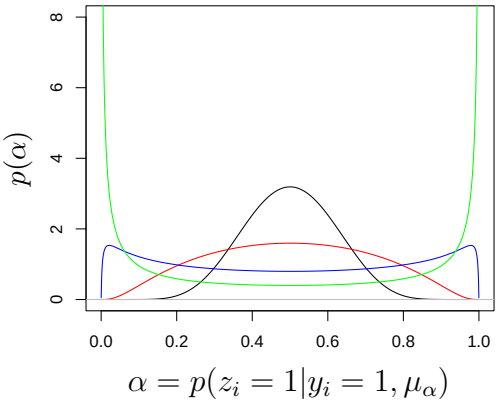
1. Our simulations do not differ substantially, therefore for calibration we do not want falses to be penalised too much otherwise $p(m|\boldsymbol{z})$ will be nonnegligible for very few m.

2. The simulations and observed data are close, therefore for calibrated prediction we do not want $|p(z'_i = 1|\boldsymbol{z}) - 0.5|$ to be close to 0.0 because this suggests that we learn nothing about the true flood from our simulations.

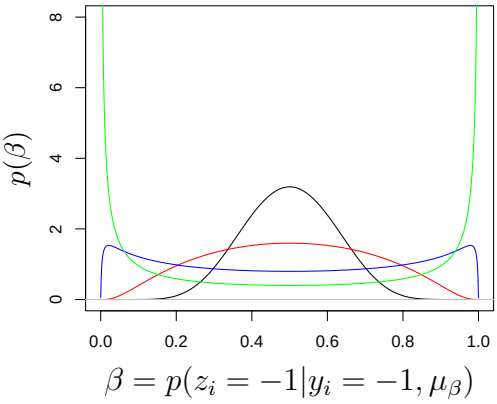
Unfortunately using the BC model we are not able to achieve these two objectives simultaneously. With these objectives in mind we now discuss the HBC model.

7.5.2 HBC Model Examples

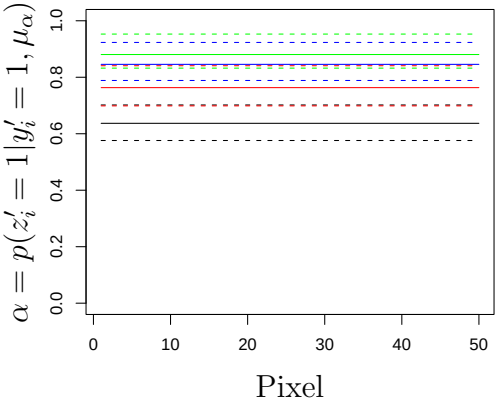
Figure 7.4 illustrates the effect of increasing τ in the HBC model with no dependence, $\lambda = 0.0$. As τ increases $p(m|\boldsymbol{z})$ becomes nonnegligible for more m. Compare this to increasing σ which has the opposite effect, $p(m|\boldsymbol{z})$ becomes zero for more m (see Figure 7.3(e)). At the same time the calibrated prediction $p(z'_i = 1|\boldsymbol{z})$ becomes closer to 0.0 or 1.0, as when we increase σ. In contrast to increasing σ, increasing τ reduces the effect of the false-positives around pixel 5 on the calibrated prediction. The plots of the distributions induced on $\alpha_i = p(z'_i = 1|y'_i = 1, \mu_\alpha, \varepsilon_{\alpha,i})$ and $\beta_i = p(z'_i = -1|y'_i = -1, \mu_\beta, \varepsilon_{\beta,i})$ by $p(\mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})$ highlight the difference from the homogeneous model (see Figures 7.4(c) and 7.4(d)). The model adjusts to each pixel individually so local errors cannot affect the global fit of the model. The problem with treating each pixel independently is that if we fix $\mu_\alpha = \mu_\beta = 0.0$ trues and falses are equally good, and all simulations will be given equal posterior weight, which is clearly not realistic.



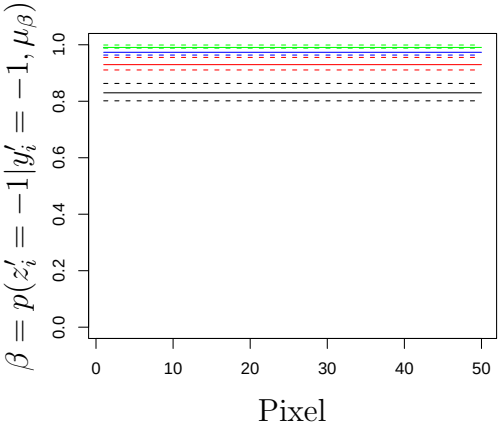
(a) Prior induced on α by $\mu_\alpha \sim \mathcal{N}(\nu_\alpha, \sigma_\alpha^2)$.



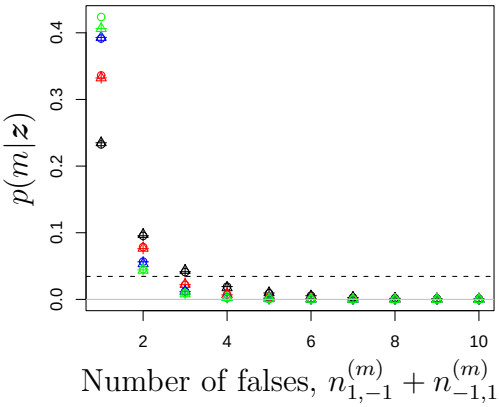
(b) Prior induced on β by $\mu_\beta \sim \mathcal{N}(\nu_\beta, \sigma_\beta^2)$.



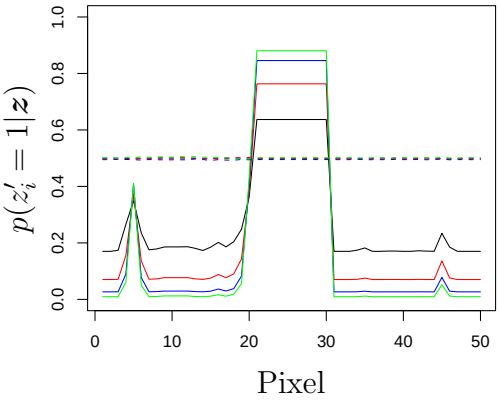
(c) Posterior induced on α by $p(\mu_\alpha|\boldsymbol{z})$. Mean (solid line) and upper and lower quartiles (dashed lines).



(d) Posterior induced on β by $p(\mu_\beta|\boldsymbol{z})$. Mean (solid line) and upper and lower quartiles (dashed lines).



(e) Marginal posterior for m. The symbols indicate the type of error: block on the boundary (circle), isolated block (triangle), and isolated pixels (cross).



(f) Calibrated prediction. For comparison $p(z'_i = 1)$ is shown (dashed lines).

Figure 7.3: Four examples of calibration and calibrated prediction using the (μ_α, μ_β) BC model. The mean $\nu = 0.0$ in all cases and the standard deviation σ is 0.5 (black), 1.0 (red), 2.0 (blue) and 4.0 (green).

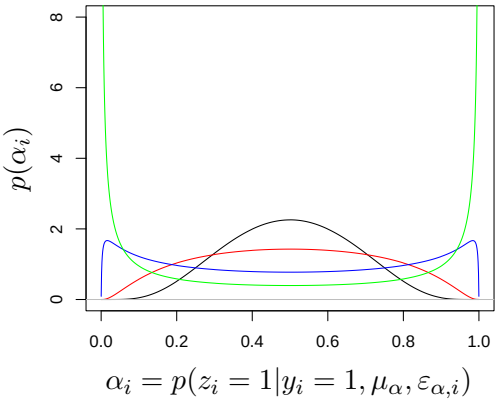
In relation to the flood inundation application, it is likely that the flood extent that we calibrate the model on is less extreme than the one we wish to predict. Therefore the flood extent boundary will be different and, for example, where false-positives occurred in calibration they may not occur in prediction. To reduce these effects we introduce spatial dependence to $\boldsymbol{\varepsilon}_\alpha$ and $\boldsymbol{\varepsilon}_\beta$.

Figure 7.5 shows the effect of spatial dependence in the HBC model. The effect of spatial dependence can be observed in the distributions induced on $\alpha_i = p(z'_i = 1|y'_i = 1, \mu_\alpha, \varepsilon_{\alpha,i})$ and $\beta_i = p(z'_i = -1|y'_i = -1, \mu_\beta, \varepsilon_{\beta,i})$ by $p(\mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta|\boldsymbol{z})$. For example, $p(z'_i = -1|y'_i = 1, \boldsymbol{z})$ is larger at pixel 10, which is the centre of a region of false-positives in 7 simulations, than at pixel 46, which is an isolated false-positive in 5 simulations (see Figure 7.2). The posterior for the simulation indexes $p(m|\boldsymbol{z})$ now depends on the configuration of falses as well as the number of them. For a given number of falses, $p(m|\boldsymbol{z})$ is smallest for the isolated falses configuration, as expected. However, $p(m|\boldsymbol{z})$ is largest for the isolated blocks configuration. We expected blocks of falses on the boundary to be "best" because these are most likely in the flood inundation application, but spatial dependence causes $1 - \alpha_i = p(z_i = -1|y_i = 1, \mu_\alpha, \varepsilon_{\alpha,i})$ to be small near the central 10 observed wet pixels.

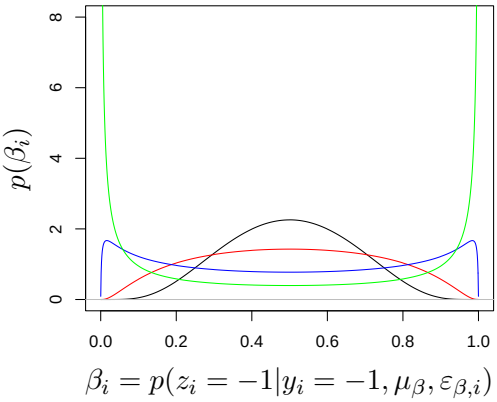
In the examples of the HBC model with and without dependence we see that $p(z' \neq y'_i|y'_i, \boldsymbol{z}) > 0.5$ occurs (for example see Figure 7.5(c)). This is not the same as saying z'_i is independent of y'_i, for which $p(z' \neq y'_i|y'_i, \boldsymbol{z}) = 0.5$, it expresses confidence in the simulator being wrong. This effect can be reduced by taking $\nu > 0.0$ – increasing ν has the dual effect of increasing $p(z'_i = 1|\boldsymbol{z})$ and making $p(m|\boldsymbol{z})$ zero for more m.

7.5.3 PHBC Model Example

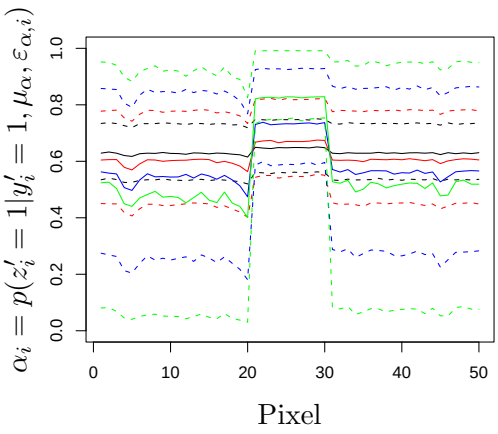
Requiring $\nu > 0.0$ reduces the probability of $p(z'_i \neq y'_i|y'_i, \boldsymbol{z}) > 0.5$ but the possibility remains. Using the positive heterogeneous binary channel (PHBC) model makes it impossible. Figure 7.6 shows a sample of results using the PHBC model. Two issues arise in using the PHBC model: first, the effect of the false-positives around pixel 5, that occur in many of the simulations, cannot be reduced as they



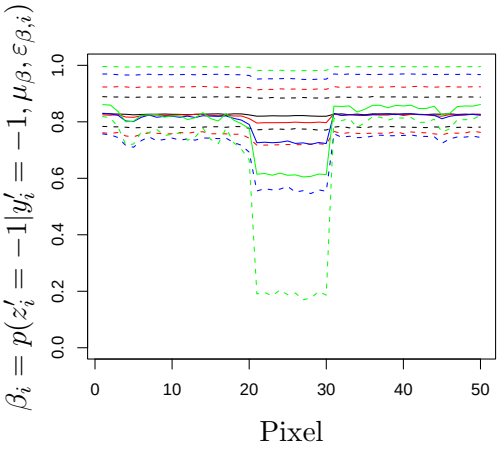
(a) Prior induced on α_i by $\mu_\alpha \sim \mathcal{N}(\nu_\alpha, \sigma_\alpha^2)$ and $\varepsilon_{\alpha,i} \sim \mathcal{N}(0, s_\alpha^2)$ where $s_\alpha^2 = \tau_\alpha^2/n \sum_{i=0}^{n-1}(1 - \lambda_\alpha \cos(2\pi i/n))^{-1}$.



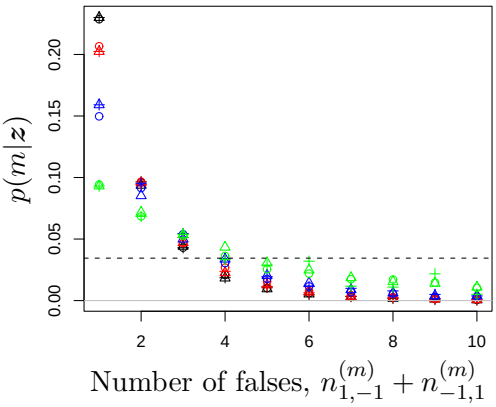
(b) Prior induced on β_i by $\mu_\beta \sim \mathcal{N}(\nu_\beta, \sigma_\beta^2)$ and $\varepsilon_{\beta,i} \sim \mathcal{N}(0, s_\beta^2)$ where $s_\beta^2 = \tau_\beta^2/n \sum_{i=0}^{n-1}(1 - \lambda_\beta \cos(2\pi i/n))^{-1}$.



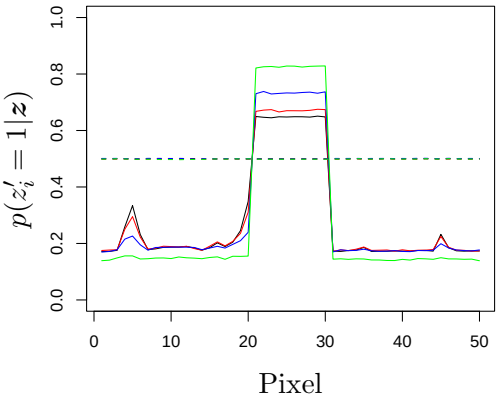
(c) Posterior induced on α_i by $p(\mu_\alpha, \varepsilon_{\alpha,i}|\boldsymbol{z})$. Mean (solid line) and upper and lower quartiles (dashed lines).



(d) Posterior induced on β_i by $p(\mu_\beta, \varepsilon_{\beta,i}|\boldsymbol{z})$. Mean (solid line) and upper and lower quartiles (dashed lines).

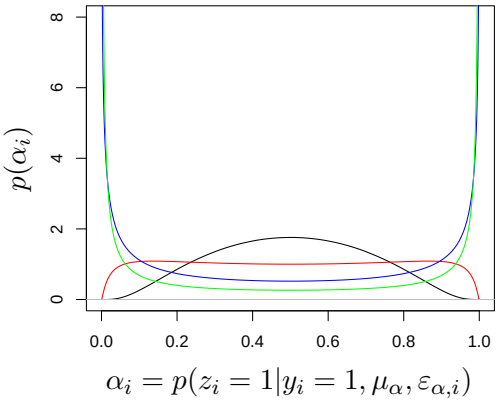


(e) Marginal posterior for m. The symbols indicate the type of error: block on the boundary (circle), isolated block (triangle), and isolated pixels (cross).

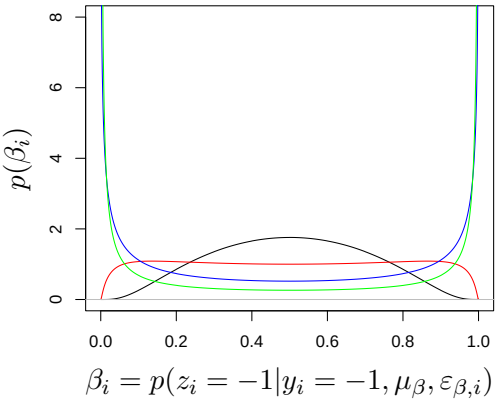


(f) Calibrated prediction. For comparison $p(z_i' = 1)$ is shown (dashed lines).

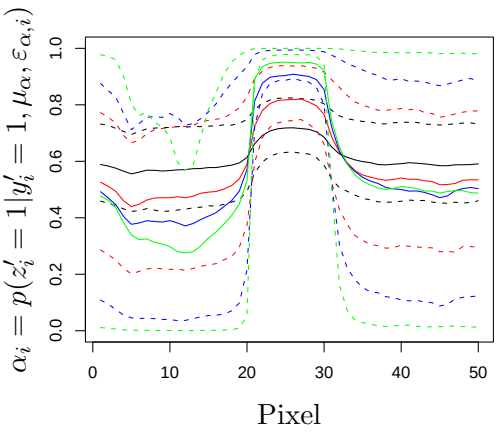
Figure 7.4: Four examples using the HBC model and changing τ. The hyperparameters are $\nu = 0.0$, $\lambda = 0.0$ and $\sigma = 0.5$ in all cases; and τ is 0.5 (black), 1.0 (red), 2.0 (blue) and 4.0 (green).



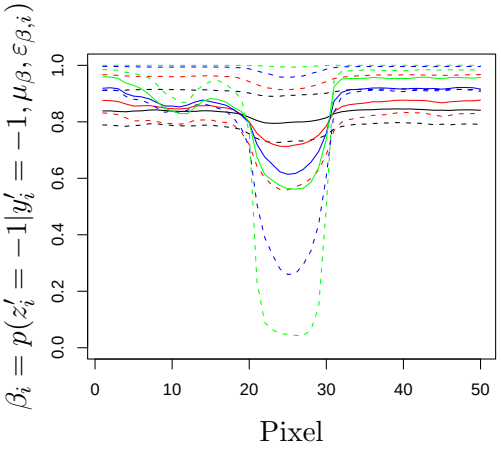
(a) Prior induced on α_i by $\mu_\alpha \sim \mathcal{N}(\nu_\alpha, \sigma_\alpha^2)$ and $\varepsilon_{\alpha,i} \sim \mathcal{N}(0, s_\alpha^2)$ where $s_\alpha^2 = \tau_\alpha^2/n \sum_{i=0}^{n-1}(1 - \lambda_\alpha \cos(2\pi i/n))^{-1}$.



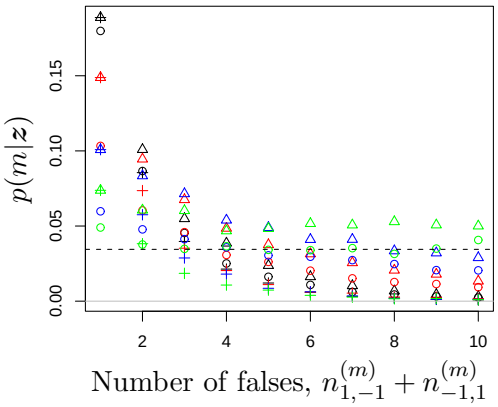
(b) Prior induced on β_i by $\mu_\beta \sim \mathcal{N}(\nu_\beta, \sigma_\beta^2)$ and $\varepsilon_{\beta,i} \sim \mathcal{N}(0, s_\beta^2)$ where $s_\beta^2 = \tau_\beta^2/n \sum_{i=0}^{n-1}(1 - \lambda_\beta \cos(2\pi i/n))^{-1}$.



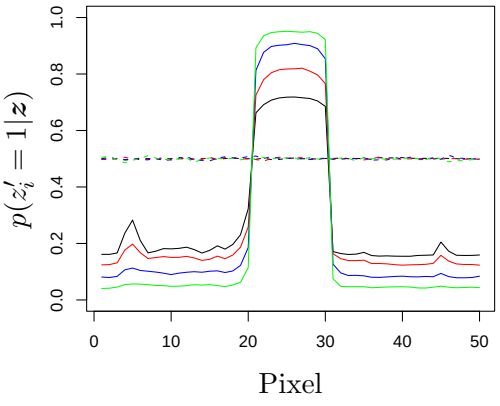
(c) Posterior induced on α_i by $p(\mu_\alpha, \varepsilon_{\alpha,i}|\boldsymbol{z})$. Mean (solid line) and upper and lower quartiles (dashed lines).



(d) Posterior induced on β_i by $p(\mu_\beta, \varepsilon_{\beta,i}|\boldsymbol{z})$. Mean (solid line) and upper and lower quartiles (dashed lines).



(e) Marginal posterior for m. The symbols indicate the type of error: block on the boundary (circle), isolated block (triangle), and isolated pixels (cross).



(f) Calibrated prediction. For comparison $p(z_i' = 1)$ is shown (dashed lines).

Figure 7.5: Four examples using the HBC model and changing τ. The hyperparameters are $\nu = 0.0$, $\lambda = 0.9$ and $\sigma = 0.5$ in all cases; and τ is 0.5 (black), 1.0 (red), 2.0 (blue) and 4.0 (green).

can using the HBC model; and second, $p(m|\boldsymbol{z})$ can only be made nonnegligible for more m by reducing the correlation between $\boldsymbol{y}$ and $\boldsymbol{z}$.

7.5.4 Markov Chain Convergence and Undesirable Model Properties

MCMC relies on the convergence of the Markov chain to the distribution of interest, if the time to convergence is very high then a realisation of the Markov chain will be a poor estimate of a sample from the distribution of interest. For the HBC and PHBC models mixing of the MCMC algorithm is poor for some hyperparameter values. For the continuous parameters $\mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha$ and $\boldsymbol{\varepsilon}_\beta$ mixing can normally be improved by tuning the width of the uniform proposal, but this is not so for the simulation index m.

Suppose we have two simulations $m^{(1)}$ and $m^{(2)}$ such that $p(m^{(1)}|\boldsymbol{z}) = p(m^{(2)}|\boldsymbol{z})$ and

$$p(\mu_\alpha^{(1)}, \mu_\beta^{(1)}, \boldsymbol{\varepsilon}_\alpha^{(1)}, \boldsymbol{\varepsilon}_\beta^{(1)}|m^{(1)}, \boldsymbol{z}) = p(\mu_\alpha^{(2)}, \mu_\beta^{(2)}, \boldsymbol{\varepsilon}_\alpha^{(2)}, \boldsymbol{\varepsilon}_\beta^{(2)}|m^{(2)}, \boldsymbol{z})$$

but

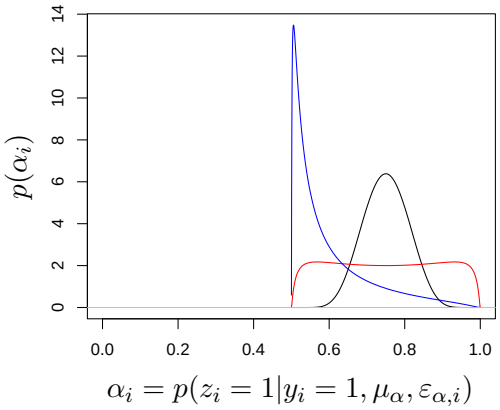
$$p(\mu_\alpha^{(1)}, \mu_\beta^{(1)}, \boldsymbol{\varepsilon}_\alpha^{(1)}, \boldsymbol{\varepsilon}_\beta^{(1)}|m^{(1)}, \boldsymbol{z}) \gg p(\mu_\alpha^{(1)}, \mu_\beta^{(1)}, \boldsymbol{\varepsilon}_\alpha^{(1)}, \boldsymbol{\varepsilon}_\beta^{(1)}|m^{(2)}, \boldsymbol{z})$$

for some parameters $\mu_\alpha^{(1)}, \mu_\beta^{(1)}, \boldsymbol{\varepsilon}_\alpha^{(1)}, \boldsymbol{\varepsilon}_\beta^{(1)}, \mu_\alpha^{(2)}, \mu_\beta^{(2)}, \boldsymbol{\varepsilon}_\alpha^{(2)}, \boldsymbol{\varepsilon}_\beta^{(2)}$. Let the value of the Markov chain at the kth iteration be $m = m^{(1)}$, $\mu_\alpha = \mu_\alpha^{(1)}$, $\mu_\beta = \mu_\beta^{(1)}$, $\boldsymbol{\varepsilon}_\alpha = \boldsymbol{\varepsilon}_\alpha^{(1)}$ and $\boldsymbol{\varepsilon}_\beta = \boldsymbol{\varepsilon}_\beta^{(1)}$, and suppose the proposed value of the simulation index from a discrete uniform distribution on $\{1, \ldots, m^{(1)} - 1, m^{(1)} + 1, \ldots, M\}$ is $m' = m^{(2)}$. Then it is very unlikely that this proposal will be accepted. Accepting that the probabilities will not in general be exactly equal, this example is indicative of the mixing problem for the simulation index update.

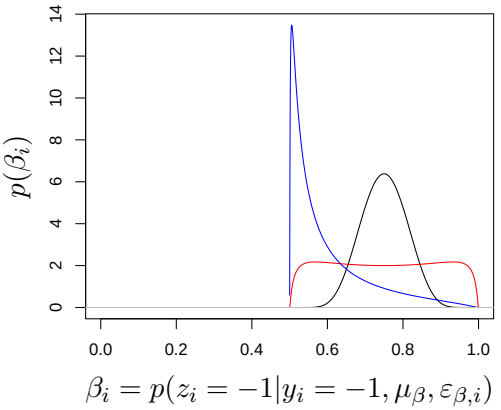
To improve mixing we rejected the discrete uniform proposal for the simulation index in favour of

$$q(m'|m) \propto \left(\sum_{i=1}^{n} \mathbf{1}[y_i^{(m)} \neq y_i^{(m')}] \right)^{-1}$$

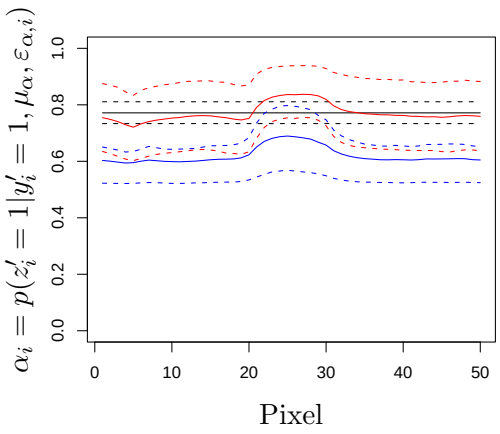
for $m' \in \{1, \ldots, m-1, m+1, \ldots, M\}$. This improved mixing quite considerably for some prior specifications but for others mixing is still poor (see Figure 7.7). In these latter cases it will be necessary to explore other methods for generating



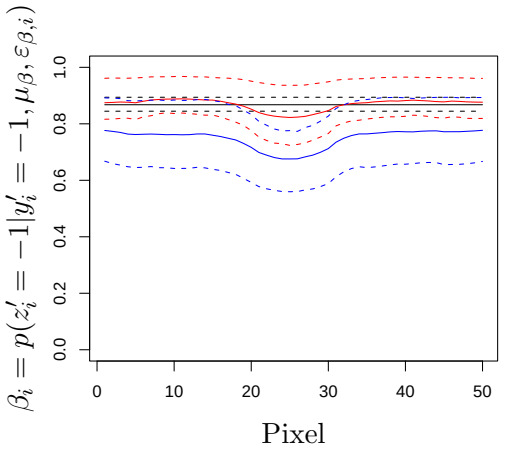
(a) Prior induced on α_i by $\mu_\alpha \sim \mathcal{N}(\nu_\alpha, \sigma_\alpha^2)$ and $\varepsilon_{\alpha,i} \sim \mathcal{N}(0, s_\alpha^2)$ where $s_\alpha^2 = \tau_\alpha^2/n \sum_{i=0}^{n-1}(1 - \lambda_\alpha \cos(2\pi i/n))^{-1}$.



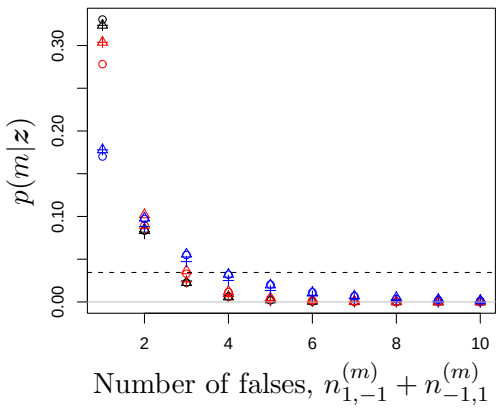
(b) Prior induced on β_i by $\mu_\beta \sim \mathcal{N}(\nu_\beta, \sigma_\beta^2)$ and $\varepsilon_{\beta,i} \sim \mathcal{N}(0, s_\beta^2)$ where $s_\beta^2 = \tau_\beta^2/n \sum_{i=0}^{n-1}(1 - \lambda_\beta \cos(2\pi i/n))^{-1}$.



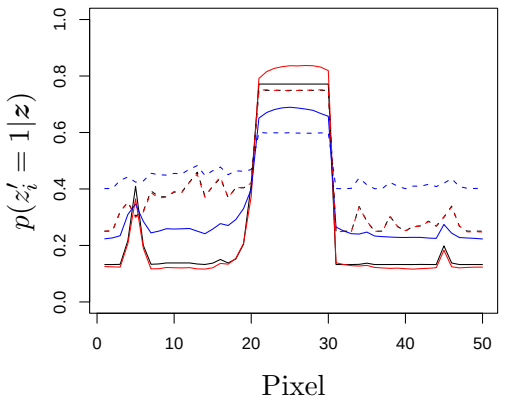
(c) Posterior induced on α_i by $p(\mu_\alpha, \varepsilon_{\alpha,i}|\boldsymbol{z})$. Mean (solid line) and upper and lower quartiles (dashed lines).



(d) Posterior induced on β_i by $p(\mu_\beta, \varepsilon_{\beta,i}|\boldsymbol{z})$. Mean (solid line) and upper and lower quartiles (dashed lines).

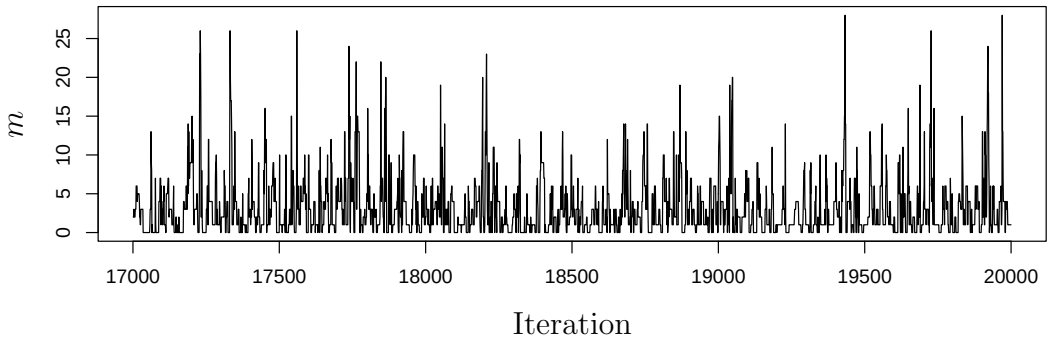


(e) Marginal posterior for m. The symbols indicate the type of error: block on the boundary (circle), isolated block (triangle), and isolated pixels (cross).

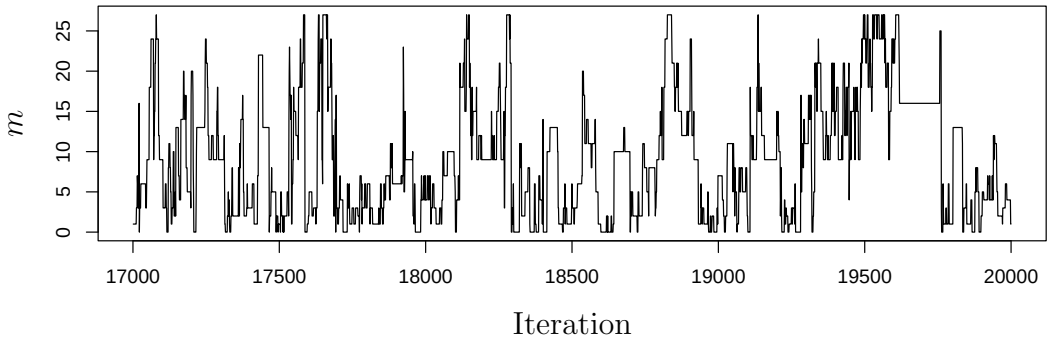


(f) Calibrated prediction. For comparison $p(z_i' = 1)$ is shown (dashed lines).

Figure 7.6: Three examples using the PHBC model: $\nu = 0.0$, $\sigma = 0.5$, $\lambda = \tau = 0.0$ (black); $\nu = 0.0$, $\sigma = 0.5$, $\lambda = 0.9$ and $\tau = 1.0$ (red); and $\nu = -2.0$, $\sigma = 0.5$, $\lambda = 0.9$ and $\tau = 1.0$ (blue).



(a) An example of good mixing when $\nu = \lambda = \tau = 0.0$ and $\sigma = 0.5$.



(b) An example of bad mixing when $\nu = \lambda = 0.0$, $\sigma = 0.5$ and $\tau = 4.0$.

Figure 7.7: Two examples demonstrating the effect of τ on mixing of the simulation index, m. The Markov chain is plotted between iteration 17000 and 20000 in both examples.

posterior samples. One such method is within-model sampling (WMS) which will be discussed in Section 7.7. Here 'model' refers to the value of the simulation index m.

A property common to the BC, HBC and PHBC models is that given simulation index, m, and observed data, $\boldsymbol{z}$, the likelihood parameters relating to positives and negatives are independent,

$$p(\mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta | m, \boldsymbol{z}) = p(\mu_\alpha, \boldsymbol{\varepsilon}_\alpha | m, \boldsymbol{z}) p(\mu_\beta, \boldsymbol{\varepsilon}_\beta | m, \boldsymbol{z}).$$

These parameters are not independent in the marginal posterior because of the sum over the simulations

$$p(\mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta | \boldsymbol{z}) = \sum_{m=1}^{M} p(\mu_\alpha, \boldsymbol{\varepsilon}_\alpha | m, \boldsymbol{z}) p(\mu_\beta, \boldsymbol{\varepsilon}_\beta | m, \boldsymbol{z}) p(m | \boldsymbol{z}).$$

For example consider the distribution of the parameters relating to positives

given simulation, m, and observed data, $\boldsymbol{z}$,

$$p(\mu_\alpha, \boldsymbol{\varepsilon}_\alpha | m, \boldsymbol{z}) \propto \prod_{i=1}^{n} \left(\frac{\exp\left(\left(\frac{z_i+1}{2}\right)(\mu_\alpha + \varepsilon_{\alpha,i})\right)}{1+\exp(\mu_\alpha + \varepsilon_{\alpha,i})} \right)^{\frac{y_i^{(m)}+1}{2}} p(\mu_\alpha)p(\boldsymbol{\varepsilon}_\alpha).$$

This distribution depends on $\{(y_i^{(m)}, z_i) | y_i^{(m)} = 1, i = 1, \ldots, n\}$ and is independent of $\{(y_i^{(m)}, z_i) | y_i^{(m)} = -1, i = 1, \ldots, n\}$. In particular if $y_i^{(m)} = -1 \; \forall i \in \{1, \ldots, n\}$ then $p(\mu_\alpha, \boldsymbol{\varepsilon}_\alpha | m, \boldsymbol{z}) = p(\mu_\alpha)p(\boldsymbol{\varepsilon}_\alpha)$. The marginal posterior $p(\mu_\alpha, \boldsymbol{\varepsilon}_\alpha | \boldsymbol{z})$ only depends on $\{(y_i^{(m)}, z_i) | y_i^{(m)} = -1, i = 1, \ldots, n\}$ through $p(m|\boldsymbol{z})$.

In the HBC and PHBC model equations true-positives are explicitly linked to false-positives and true-negatives are explicitly linked to false-negatives. However, there are no such links between true-positives and false-negatives and between true-negatives and false-positives. Consider a region in which all pixels are all false-negatives in all simulations of the calibration event. Now suppose in all simulations of the event we want to predict these pixels are positive, then our calibrated prediction in this region will be uncertain because it does not take account of any link between false-negatives and true-positives. In Chapter 8 we consider a likelihood model where these links are made.

7.6 Buscot Example

In this section we use the Buscot dataset introduced in Section 2.4 to illustrate our Bayesian framework for calibration and calibrated prediction using the HBC model. The results can be compared to those obtained using GLUE (Section 4.2.3) and the BC model (Section 5.3).

In Section 7.5 we demonstrated the impact of different hyperparameter settings on posterior inference using a one-dimensional example. Therefore in this section we present only two examples indicative of the results possible using the HBC model.

Figures 7.8 and 7.9 show the results of calibration and calibrated prediction using the HBC model with hyperparameters $\nu = 0.0$, $\sigma = 0.014$, $\tau = 1.0$ and two λ values: 0.0 and 0.9. Priors $\mu_\alpha, \mu_\beta \sim \mathcal{N}(0.0, 0.014^2)$ in the (μ_α, μ_β) BC model

correspond to priors $\alpha, \beta \sim \text{beta}(10000, 10000)$ in the (α, β) BC model, for which the results of calibration and calibration prediction are shown in Figure 5.6.

Increasing λ from 0.0 to 0.9 increases the prior probability that α_i and β_i take values away from 0.5 (see Figures 7.8(a) and 7.8(b)). This leads to less uncertainty in the calibrated predictions (see Figures 7.9(e) and 7.9(f)). Also, increasing λ increases spatial dependence (see Figures 7.8(c) and 7.8(d)).

The marginal posterior for the simulation index m (see Figures 7.8(e) and 7.8(f)), and the posterior for the calibration inputs $\boldsymbol{\theta}$ (see Figures 7.9(a) and 7.9(b)), become more peaked when λ changes from 0.0 to 0.9. This is because each term in $\boldsymbol{\varepsilon}_\alpha$ and $\boldsymbol{\varepsilon}_\beta$ must take a similar value to its neighbours – the capacity for adjusting to each pixel has been reduced so falses are inevitably penalised more. Note that if each term in $\boldsymbol{\varepsilon}_\alpha$ and $\boldsymbol{\varepsilon}_\beta$ was required to be identical to its neighbours, the HBC model would degenerate to the BC model.

A peculiar property of the HBC model can be observed in Figures 7.9(c) and 7.9(d). Each image splits roughly into three parts: $p(z'_i = 1 | y'_i = 1, \boldsymbol{z}) \approx 0.5$ in regions dominated by negatives; $p(z'_i = 1 | y'_i = 1, \boldsymbol{z}) > 0.5$ in regions dominated by true-positives; and $p(z'_i = 1 | y'_i = 1, \boldsymbol{z}) < 0.5$ in regions dominated by false-positives. Regions dominated by false-positives occur just outside the observed flood boundary and appear as lighter patches in Figures 7.9(c) and 7.9(d). If these regions are positive in simulations of the event we want to predict, our calibrated prediction will be that the true value is likely to be negative. However, the event we wish to predict is usually greater in magnitude than the event we have calibrated on, so this property of the model is undesirable. This problem is a consequence of overfitting the model to the calibration data.

In summary, the results of calibration and calibrated prediction using the HBC model are a great improvement on those obtained using the BC model in Section 5.3. Fundamentally, it is possible to obtain good results for calibration and calibrated prediction simultaneously. The results may be further improved by increasing τ but in this case Markov chain convergence is poor. In the next section we discuss a method for sampling from the posterior distribution when mixing is

poor.

7.7 Within-Model Sampling

In Section 7.5.4 we discussed the problem of Markov chain convergence for the HBC and PHBC models. We now consider one method for generating posterior samples when mixing is poor.

For the MCMC algorithm outlined in Section 7.3 the aim is to simulate from the joint posterior $p(\boldsymbol{\phi}, m|\boldsymbol{z})$. This is called *across-model sampling* (AMS), where 'model' refers to the value of the simulation index m. When mixing between 'models' is poor we can simulate from $p(\boldsymbol{\phi}|m, \boldsymbol{z})$ for each m and find $p(m|\boldsymbol{z})$ leading to $p(\boldsymbol{\phi}, m|\boldsymbol{z})$, this is called *within-model sampling* (WMS).

The marginal posterior $p(m|\boldsymbol{z})$ can be found from the ratio

$$\frac{p(m|\boldsymbol{z})}{p(m^\star|\boldsymbol{z})} = \frac{p(\boldsymbol{z}|m)p(m)}{p(\boldsymbol{z}|m^\star)p(m^\star)}$$

where $m^\star$ is some reference simulation index, and $p(m) = p(m^\star)$ so we need only calculate

$$p(\boldsymbol{z}|m) = \int p(\boldsymbol{z}|\boldsymbol{\phi}, m)p(\boldsymbol{\phi}|m)\,\mathrm{d}\boldsymbol{\phi}$$

called the *marginal likelihood*.

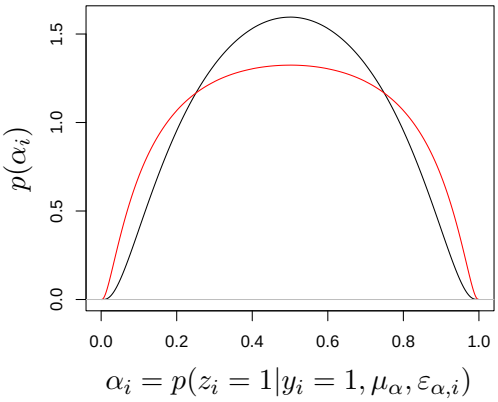
There are a number of ways to approximate the marginal likelihood (see Green, 2003, for a review), we adopt a method based on the identity

$$p(\boldsymbol{z}|m) = \left(\int \frac{p(\boldsymbol{\phi}|m, \boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{\phi}, m)}\,\mathrm{d}\boldsymbol{\phi} \right)^{-1}.$$

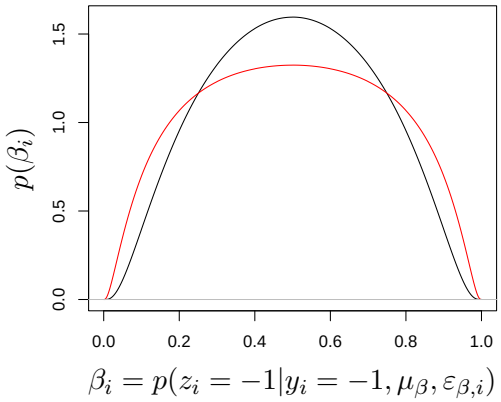
If $\{\boldsymbol{\phi}^{(k)} : k = 1, \ldots, K\}$ is a sample from $p(\boldsymbol{\phi}|m, \boldsymbol{z})$, then

$$p(\boldsymbol{z}|m) \approx \left(\frac{1}{K} \sum_{k=1}^{K} \frac{1}{p(\boldsymbol{z}|\boldsymbol{\phi}^{(k)}, m)} \right)^{-1}. \tag{7.11}$$

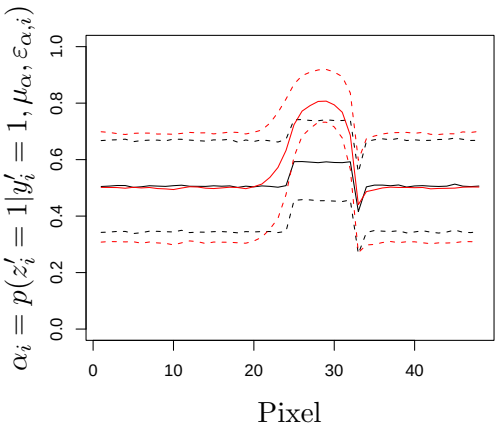
As we need a sample from $p(\boldsymbol{\phi}|m, \boldsymbol{z})$ for each $m \in \{1, 2, \ldots, M\}$, and we must remove burn-in from each sample, WMS is more computationally intensive than AMS. In our case $\boldsymbol{\phi} = (\mu_\alpha, \mu_\beta, \boldsymbol{\varepsilon}_\alpha, \boldsymbol{\varepsilon}_\beta)$ and for each sample $\{\boldsymbol{\phi}^{(k)} : k = 1, \ldots, K\}$



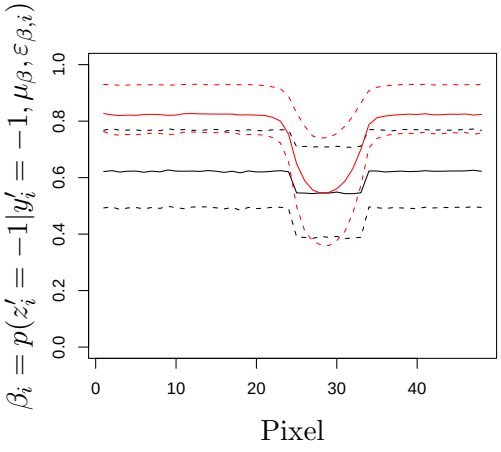
(a) Prior induced on α_i by $\mu_\alpha \sim \mathcal{N}(\nu_\alpha, \sigma_\alpha^2)$ and $\varepsilon_{\alpha,i} \sim \mathcal{N}(0, s_\alpha^2)$ where $s_\alpha^2 = \tau_\alpha^2/(rc)\sum_{i=0}^{c-1}\sum_{j=0}^{r-1}(1 - \lambda_\alpha \cos(2\pi i/c)/2 - \lambda_\alpha \cos(2\pi j/r)/2)^{-1}$.



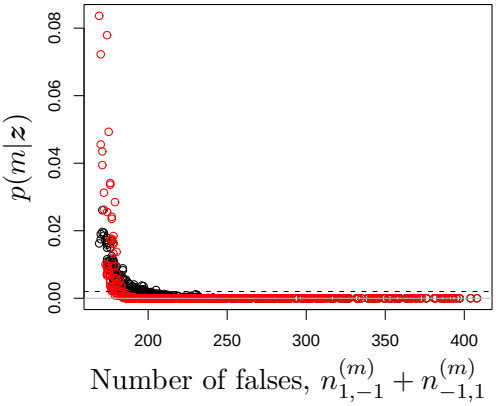
(b) Prior induced on β_i by $\mu_\beta \sim \mathcal{N}(\nu_\beta, \sigma_\beta^2)$ and $\varepsilon_{\beta,i} \sim \mathcal{N}(0, s_\beta^2)$ where $s_\beta^2 = \tau_\beta^2/(rc)\sum_{i=0}^{c-1}\sum_{j=0}^{r-1}(1 - \lambda_\beta \cos(2\pi i/c)/2 - \lambda_\beta \cos(2\pi j/r)/2)^{-1}$.



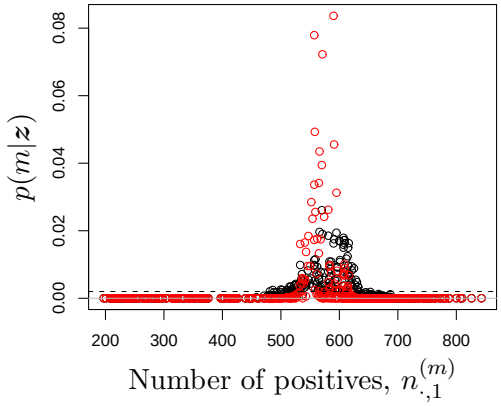
(c) Posterior induced on α_i by $p(\mu_\alpha, \varepsilon_{\alpha,i}|\boldsymbol{z})$. Mean (solid line) and upper and lower quartiles (dashed lines).



(d) Posterior induced on β_i by $p(\mu_\beta, \varepsilon_{\beta,i}|\boldsymbol{z})$. Mean (solid line) and upper and lower quartiles (dashed lines).

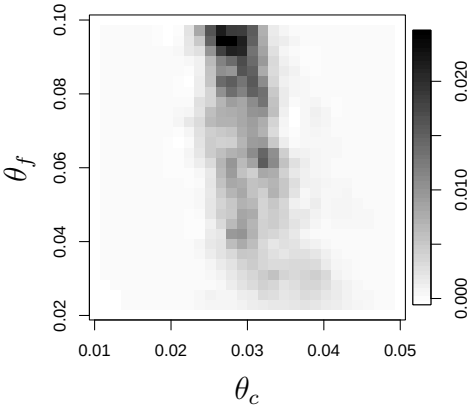


(e) Marginal posterior for m.

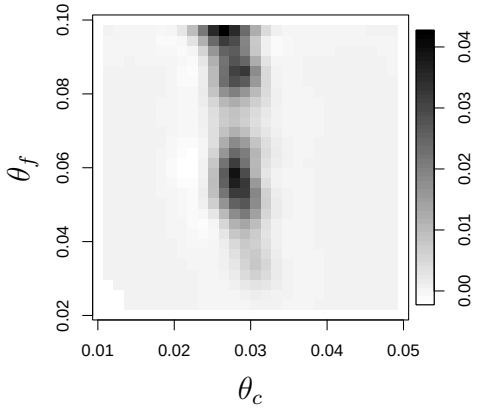


(f) Marginal posterior for m.

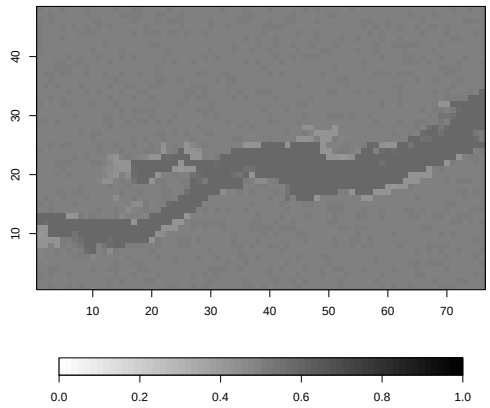
Figure 7.8: Two examples using the HBC model and changing λ. The hyperparameters are $\nu = 0.0$, $\sigma = 0.014$ and $\tau = 1.0$ in both cases; and λ is 0.0 (black) and 0.9 (red).



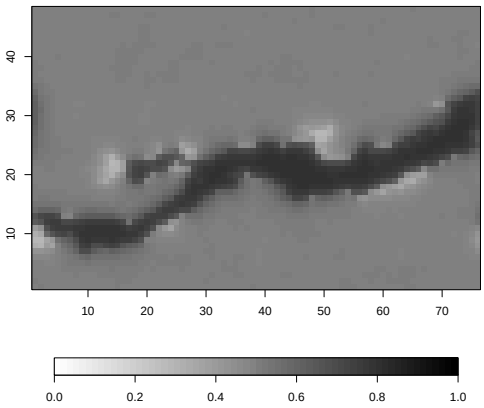
(a) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. $\lambda = 0.0$.



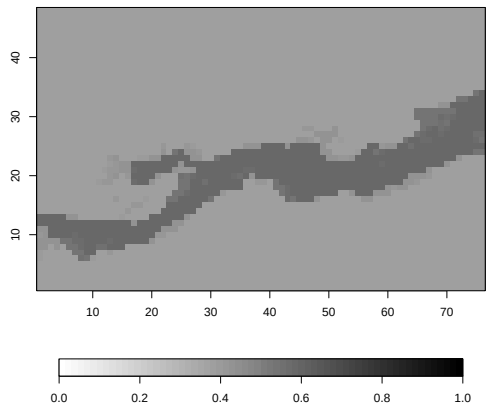
(b) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. $\lambda = 0.9$.



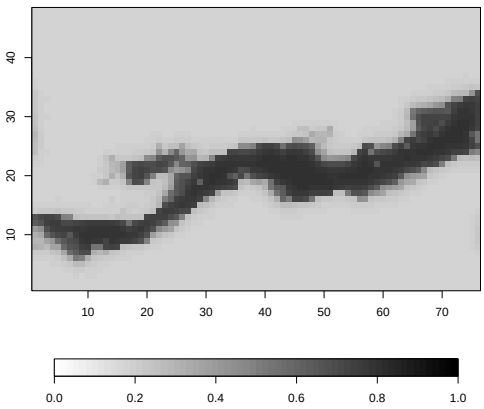
(c) $p(z'_i = 1|y'_i = 1, \boldsymbol{z})$. $\lambda = 0.0$.



(d) $p(z'_i = 1|y'_i = 1, \boldsymbol{z})$. $\lambda = 0.9$.



(e) $p(z'_i = 1|\boldsymbol{z})$. $\lambda = 0.0$.



(f) $p(z'_i = 1|\boldsymbol{z})$. $\lambda = 0.9$.

Figure 7.9: Results of calibration and calibrated prediction for the Buscot dataset using the HBC model with hyperparameters $\nu = 0.0$, $\sigma = 0.014$ and $\tau = 1.0$.

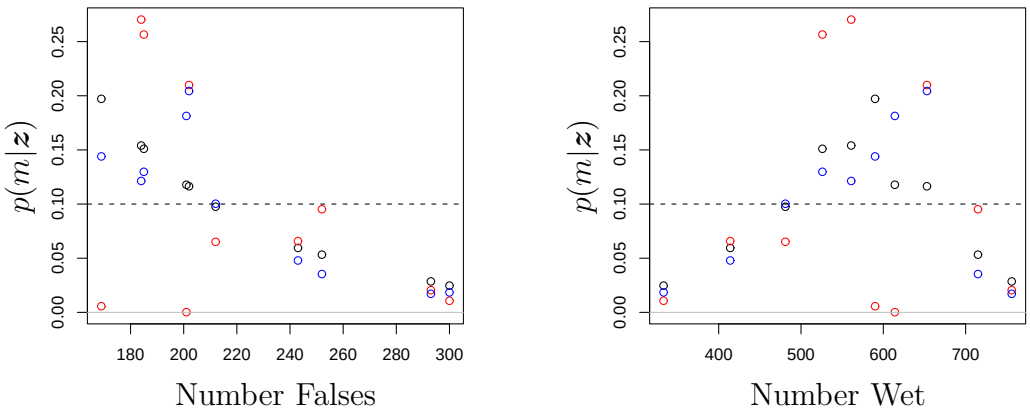
we must calculate

$$p(\boldsymbol{z}|\boldsymbol{\phi}^{(k)}, m) = \prod_{i=1}^{n} \left(\frac{\exp\left(\left(\frac{z_i+1}{2}\right)(\mu_\alpha + \varepsilon_{\alpha,i})\right)}{1+\exp(\mu_\alpha + \varepsilon_{\alpha,i})} \right)^{\frac{y_i+1}{2}} \times \left(\frac{\exp\left(\left(\frac{1-z_i}{2}\right)(\mu_\beta + \varepsilon_{\beta,i})\right)}{1+\exp(\mu_\beta + \varepsilon_{\beta,i})} \right)^{\frac{1-y_i}{2}}$$

for $k = 1, \ldots, K$.

We now consider an example for the (μ_α, μ_β) BC model with $\nu = 0.0$ and $\sigma = 0.0045$. This corresponds to the (α, β) BC model with $a = b = c = d = 100000$, for which $p(m|\boldsymbol{z})$ can be found exactly (see Section 5.2). We select a subset of simulations $m \in \{110, 91, 349, 1, 3, 5, 9, 35, 46, 51\}$ that are representative of the total 500 simulations to reduce computational burden. The results of the analysis are shown in Figure 7.10. The WMS approximation is very poor. We test whether this was the consequence of a few extreme values by removing the 10 largest and 10 smallest $p(\boldsymbol{z}|\boldsymbol{\phi}^{(k)}, m)$ for each m but these results are also very poor. However, the estimator in Equation (7.11) is known to have high variance and be sensitive to very few points in the sample because $p(\boldsymbol{z}|\boldsymbol{\phi}^{(k)}, m)$ is generally very small (see Green, 2003). We will discuss other methods for improving mixing in Section 8.6.

In this chapter we extended the BC model to account for heterogeneity and spatial dependence, and used this extension as the likelihood in our framework for calibration and calibrated prediction. We demonstrated the impact of different prior assumptions on the posterior and identified that mixing is poor for some prior choices. A more fundamental problem with the HBC model is that there are no explicit links between true-positives and false-negatives and between true-negatives and false-positives. In the next chapter we consider a model in which they are linked and we discuss other methods for improving mixing of the Markov chain.



(a) (b)

Figure 7.10: Results of within-model sampling for the (μ_α, μ_β) BC model with $\nu = 0.0$ and $\sigma = 0.0045$. The exact results using the (α, β) BC model with $a = b = c = d = 100000$ are shown by black circles. The WMS approximation using the full sample is shown with red circles, and with the 10 largest and 10 smallest values removed with blue circles.

Chapter 8

The Hidden Conditional Autoregressive Model

In Chapter 7 we used the HBC model for the likelihood of the observed flood extent given a simulation of flood extent, but unfortunately this model does not explicitly link positive and negative simulation values. In this chapter we consider the hidden conditional autoregressive (HCAR) model for the likelihood, which does link positive and negative simulation values. We describe calibration and calibrated prediction using the HCAR model and present an MCMC algorithm for estimation. Examples are given using the Buscot dataset, and we describe various methods to improve mixing in the MCMC algorithm. Three variants of the HCAR model are presented: the hidden intrinsic autoregressive (HIAR) model which is motivated as a limit of the HCAR model; the heterogeneous HCAR model which represents heterogeneity; and the continuous HCAR model which uses continuous valued simulations.

8.1 Introduction

In Chapter 5 we presented our Bayesian framework for calibration and calibrated prediction, and identified that we need to specify an appropriate likelihood model for the observed flood extent given a simulation of flood extent. Subsequently, this specification has formed the main task in this thesis. In Chapter 6 we considered the Ising model which included spatial dependence but was impractical because of the intractable normalising constant. In Chapter 7 we considered the HBC model,

but this model does not explicitly link positive and negative simulation values. For example, fix $\mu_\alpha = \mu_\beta = 0.0$ and $\lambda_\alpha = \lambda_\beta = 0.0$ and suppose we have one simulation for calibration, $\boldsymbol{y}$, and one for prediction, $\boldsymbol{y}'$. For pixel i, suppose in calibration the simulation value is negative, $y_i = -1$, and the observed value is positive, $z_i = 1$, i.e. a false-negative. Then, if in prediction the simulation value is positive, $y'_i = 1$, our calibrated prediction is completely uncertain, $p(z'_i = 1|\boldsymbol{z}) = 0.5$.

In Weir and Pettitt (1999) spatially distributed binary data is treated as the result of thresholding an underlying continuous process, which in turn is modelled as a conditional autoregression (CAR) (see Besag, 1974). Any properties of the binary image, such as blur, spatial dependence and heterogeneity, are represented in the distribution of the underlying continuous process. The advantage of this approach over the Ising model is that there is no need to calculate a complicated normalising constant. In the next section we extend this model to regression on a binary image and parameterise the model so positive and negative simulation values are explicitly linked.

8.2 The Hidden Conditional Autoregressive Model

In this section we extend the approach adopted in Weir and Pettitt (1999) and Pettitt *et al.* (2002) to regression on another image. The observed data $\boldsymbol{z}$ and the simulator output $\boldsymbol{y}^{(m)}$ are binary arrays of size $n = r \times c$, each taking values in $\{-1, 1\}^n$, where -1 indicates a dry pixel and 1 a wet pixel. The hidden continuous process is denoted by $\boldsymbol{\zeta} \in \mathbb{R}^n$, and

$$z_i = \mathbf{1}_{\{-1,1\}}[\zeta_i > 0]$$

for $i = 1, \ldots, n$, where

$$\mathbf{1}_{\{-1,1\}}[\zeta_i > 0] \begin{cases} 1 & \text{if } \zeta_i > 0 \\ -1 & \text{if } \zeta \leq 0, \end{cases}$$

so $p(z_i = 1) = p(\zeta_i > 0)$ and $p(z_i = -1) = p(\zeta_i \leq 0)$.

8.2.1 Conditional Autoregression (CAR)

We model the underlying continuous process $\boldsymbol{\zeta}$ as a *conditional autoregression* (CAR). CARs provide a means of specifying a *Gaussian Markov random field* (GMRF) in terms of full conditionals, and are discussed in Besag (1974). For completeness we show how a unique joint distribution can be defined by specifying the full conditionals only, and work out what constraints must be imposed for this to be true (see Rue and Held, 2005). Let $\boldsymbol{x} \in \mathbb{R}^n$ and suppose

$$x_i|\boldsymbol{x}_{-i} \sim \mathcal{N}\left(\mu_i + \sum_{j=1}^{n} C_{ij}(x_j - \mu_j), \sigma_i^2\right)$$

for $i = 1, \ldots, n$, and some $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and $\{C_{ij} : i, j = 1, \ldots, n\}$. Pixels i and j are neighbours if and only if $C_{ij} \neq 0$. Brook's lemma states that

$$\frac{p(\boldsymbol{x})}{p(\boldsymbol{x}')} = \prod_{i=1}^{n} \frac{p(x_i|x_1, \ldots, x_{i-1}, x'_{i+1}, \ldots, x'_n)}{p(x'_i|x_1, \ldots, x_{i-1}, x'_{i+1}, \ldots, x'_n)} \tag{8.1}$$

provided $p(\boldsymbol{x}) > 0$ and $p(\boldsymbol{x}') > 0$, and this must be invariant to permutations because the labelling is not ordered, so for instance

$$\frac{p(\boldsymbol{x})}{p(\boldsymbol{x}')} = \prod_{i=1}^{n} \frac{p(x_i|x'_1, \ldots, x'_{i-1}, x_{i+1}, \ldots, x_n)}{p(x'_i|x'_1, \ldots, x'_{i-1}, x_{i+1}, \ldots, x_n)}. \tag{8.2}$$

This can be proved by starting with the trivial result

$$p(x_1, \ldots, x_n) = \frac{p(x_n|x_1, \ldots, x_{n-1})}{p(x'_n|x_1, \ldots, x_{n-1})} p(x_1, \ldots, x_{n-1}, x'_n)$$

then replacing $p(x_1, \ldots, x_{n-1}, x'_n)$ by a similar expression and continuing the iteration. This gives Equation (8.1) and the invariance to permutations is obvious from this proof. Equation (8.2) can be found the same way by starting with x'_1 instead of x'_n.

Take $\boldsymbol{x}' = \mathbf{0}$ and $\boldsymbol{\mu} = \mathbf{0}$ then from Equation (8.1)

$$\log \frac{p(\boldsymbol{x})}{p(\mathbf{0})} = -\frac{1}{2} \sum_{i=1}^{n} \left(\frac{x_i}{\sigma_i}\right)^2 + \sum_{i=2}^{n} \sum_{j=1}^{i-1} \frac{C_{ij} x_i x_j}{\sigma_i^2} \tag{8.3}$$

and from Equation (8.2)

$$\log \frac{p(\boldsymbol{x})}{p(\mathbf{0})} = -\frac{1}{2} \sum_{i=1}^{n} \left(\frac{x_i}{\sigma_i}\right)^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{C_{ij} x_i x_j}{\sigma_i^2}. \tag{8.4}$$

Equating the right-hand sides of Equations (8.3) and (8.4) we find

$$\frac{C_{ij}}{\sigma_i^2} = \frac{C_{ji}}{\sigma_j^2}$$

for $i \neq j$. If this condition is met then the (log) joint density is

$$\log p(\boldsymbol{x}) = A - \frac{1}{2}\sum_{i=1}^{n}\left(\frac{x_i}{\sigma_i}\right)^2 + \frac{1}{2}\sum_{i\neq j}\frac{C_{ij}x_i x_j}{\sigma_i^2}$$

where $A \in \mathbb{R}$ is some constant. Therefore $\boldsymbol{x} \sim \mathcal{MVN}(\boldsymbol{0}, \boldsymbol{Q}^{-1})$ provided the precision matrix is positive definite, $\boldsymbol{Q} > 0$, where $Q_{ii} = 1/\sigma_i^2$ and $Q_{ij} = -C_{ij}/\sigma_i^2$ for $i \neq j$. Besag and Kooperberg (1995) prove a sufficient condition for this to be true is that all C_{ij} are nonnegative and that $C_{i+} \leq 1$ for all i with strict inequality for at least one i.

Specifying the model in terms of the precision matrix is particularly convenient because $Q_{ij} \neq 0$ if and only if i and j are neighbours. The variance matrix $\boldsymbol{V} = \boldsymbol{Q}^{-1}$ will in general have the property that V_{ij} is a function of all $\boldsymbol{Q}$.

Throughout this chapter we will specify models through full conditionals for their intuitive appeal. In all cases we have checked that the model satisfies the conditions outlined in this section.

8.2.2 Likelihood

For the underlying continuous process $\boldsymbol{\zeta}$ we define the conditional expectation and variance to be

$$\mathrm{E}\left(\zeta_i|\boldsymbol{\zeta}_{-i}, \boldsymbol{y}, \mu, \rho, \boldsymbol{D}, \boldsymbol{C}\right) = \mu + \rho\left(\boldsymbol{D}\boldsymbol{y}\right)_i + \sum_{j=1}^{n} C_{ij}\left(\zeta_j - \mu - \rho\left(\boldsymbol{D}\boldsymbol{y}\right)_j\right)$$

and

$$\mathrm{Var}\left(\zeta_i|\boldsymbol{\zeta}_{-i}, \boldsymbol{y}, \mu, \rho, \boldsymbol{D}, \boldsymbol{C}\right) = 1.0$$

where $\mu \in \mathbb{R}$ is the *mean parameter*, $\rho \in \mathbb{R}$ the *regression parameter*, $\boldsymbol{D}$ is the *blur matrix*, and $\boldsymbol{C}$ is the *spatial interactions matrix*. The precision matrix is $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{C}$.

We assume toroidal boundary conditions, so the East/South neighbours of pixels

in the last column/row are pixels in the first column/row. This facilitates specification of $\boldsymbol{D}$ and $\boldsymbol{C}$ because we do not need a different treatment for C_{ij} and D_{ij} near the boundary.

We define two binary relations on the set of pixel sites, denoted by $\overset{ew}{\sim}$ and $\overset{ns}{\sim}$. Both relations are required to be symmetric, if $i \overset{ew}{\sim} j$ we say i and j are East–West neighbours, and if $i \overset{ns}{\sim} j$ we say i and j are North–South neighbours. We define the blur matrix as

$$D_{ij} = \begin{cases} c & \text{if } i \overset{ns}{\sim} j \\ d & \text{if } i \overset{ew}{\sim} j \\ 1 - 2c - 2d & \text{if } i = j \\ 0.0 & \text{otherwise,} \end{cases} \tag{8.5}$$

where $c, d \geq 0.0$ and $c + d \leq 0.5$, so $(\boldsymbol{D}\boldsymbol{y})_{ij} \in [-1, 1]$. We define the spatial interactions matrix as

$$C_{ij} = \begin{cases} a & \text{if } i \overset{ns}{\sim} j \\ b & \text{if } i \overset{ew}{\sim} j \\ 0.0 & \text{otherwise,} \end{cases} \tag{8.6}$$

where $|a| + |b| < 0.5$ because of the positive definiteness constraint on the precision matrix $\boldsymbol{Q} > 0$, ($\boldsymbol{Q}\mathbf{1} = \mathbf{0}$ if $|a| + |b| = 0.5$). As a consequence of assuming toroidal boundary conditions $\boldsymbol{Q}$ is a block-circulant matrix. In Section 8.3 we discuss block-circulant matrices in more detail and show that they can be linked to the two-dimensional discrete Fourier transform. The consequence of this is that eigenvalues and eigenvectors can be found easily, determinants are therefore straightforward and matrix inversion is also relatively simple.

From Equations (8.5) and (8.6) we can see that the matrices $\boldsymbol{D}$ and $\boldsymbol{C}$ are determined by the parameters c and d, and a and b respectively. We write $p(\boldsymbol{D}) = p(c, d)$ and $p(\boldsymbol{C}) = p(a, b)$. The vector of likelihood model parameters is $\boldsymbol{\phi} = (\mu, \rho, a, b, c, d)$. Provided $|a| + |b| < 0.5$, the likelihood of the underlying continuous process given a simulation is

$$\boldsymbol{\zeta}|\boldsymbol{y}, \boldsymbol{\phi} \sim \mathcal{MVN}\left(\mu\mathbf{1} + \rho\boldsymbol{D}\boldsymbol{y}, (\boldsymbol{I} - \boldsymbol{C})^{-1}\right)$$

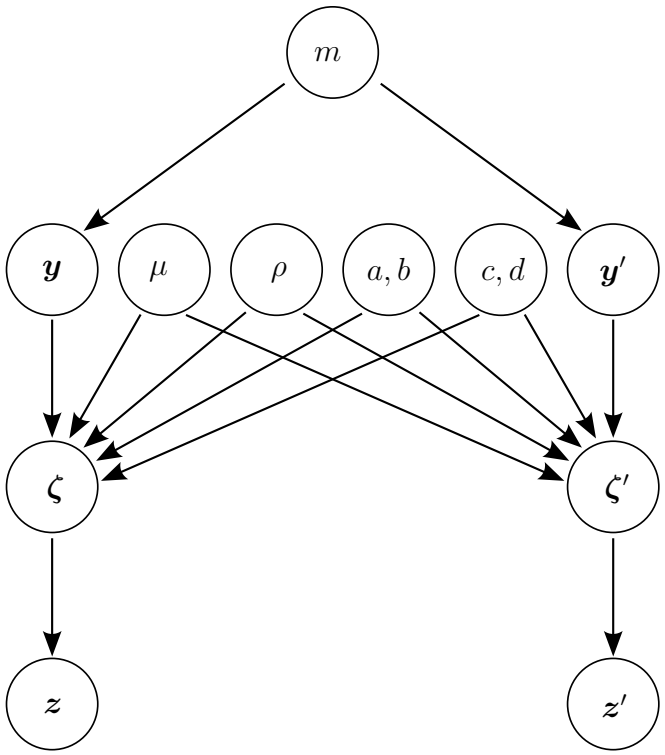


Figure 8.1: DAG for Bayesian calibration of flood inundation simulators conditioned on an observation of flood extent using the HCAR model.

which has density

$$p(\boldsymbol{\zeta}|\boldsymbol{y}, \boldsymbol{\phi}) = (2\pi)^{-\frac{n}{2}} |\boldsymbol{I} - \boldsymbol{C}|^{1/2} \exp\left(-\frac{1}{2}\boldsymbol{v}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{C})\boldsymbol{v}\right) \tag{8.7}$$

where

$$\boldsymbol{v} = \boldsymbol{\zeta} - \mu\mathbf{1} - \rho\boldsymbol{D}\boldsymbol{y}.$$

There is no loss of generality in setting the conditional variance to 1.0, to illustrate this we consider the probit model. Let $x \in \{-1, 1\}$ and $\xi \in \mathbb{R}$, suppose $x = \mathbf{1}_{\{-1,1\}}[\xi > 0]$ and $\xi \sim \mathcal{N}(\mu, \sigma^2)$, then

$$p(x = -1) = p(\xi \leq 0) = \Phi\left(-\frac{\mu}{\sigma}\right) = \Phi\left(-\frac{c\mu}{c\sigma}\right) = p(\xi' \leq 0)$$

where $c \in \mathbb{R}$ is a constant and $\xi' \sim \mathcal{N}(c\mu, c^2\sigma^2)$. So multiplying the mean by c is equivalent to dividing the variance by c^2, making one of the parameters redundant, we set $\sigma^2 = 1.0$.

8.2.3 Prior Distributions

The DAG in Figure 5.2 for Bayesian calibration of flood inundation simulators must be augmented to include nodes for $\boldsymbol{\zeta}$ and $\boldsymbol{\zeta}'$. Figure 8.1 shows the revised DAG. We now define the conditional (or marginal) distributions for each node of the DAG to complete the joint distribution specification.

The prior for m is discrete uniform on $\{1, 2, \ldots, M\}$. Given m, $\boldsymbol{y}$ and $\boldsymbol{y}'$ are deterministic, being $\boldsymbol{y}^{(m)}$ and $\boldsymbol{y}'^{(m)}$ respectively, see Equations (5.3) and (5.4).

For the mean and regression parameters we assume Normal priors, $\mu \sim \mathcal{N}(\nu_\mu, \sigma_\mu^2)$ and $\rho \sim \mathcal{N}(\nu_\rho, \sigma_\rho^2)$, where $\nu_\mu, \nu_\rho \in \mathbb{R}$ and $\sigma_\mu, \sigma_\rho \in \mathbb{R}_{>0}$.

For the blur matrix parameters c and d we assume a Uniform distribution over the feasible parameter space $c \geq 0$, $d \geq 0$ and $c + d \leq 0.5$, $p(c, d) \propto \mathbf{1}[c \geq 0]\mathbf{1}[d \geq 0]\mathbf{1}[c + d \leq 0.5]$.

Although we could take a simple prior for a and b that is uniform over the feasible parameter space $|a| + |b| < 0.5$, we shall see when we come to Section 8.7 that it could be beneficial to prevent the parameters a and b getting too close to the $|a| + |b| = 0.5$ boundary. Let

$$\begin{aligned} \phi_1 &= a - b + 0.5 \quad \text{and} \\ \phi_2 &= a + b + 0.5 \end{aligned}$$

then $|a| + |b| < 0.5$ corresponds to $0.0 < \phi_1, \phi_2 < 1.0$. Suppose $\phi_i \sim \text{beta}(s_i, t_i)$ where $s_i > 0$ and $t_i > 0$ for $i = 1, 2$, and that ϕ_1 and ϕ_2 are independent. Then the joint distribution of a and b is

$$p(a, b) = \frac{2}{\mathrm{B}(s_1, t_1)\mathrm{B}(s_2, t_2)}(0.5 + a - b)^{s_1 - 1}(0.5 - a + b)^{t_1 - 1} \\ (0.5 + a + b)^{s_2 - 1}(0.5 - a - b)^{t_2 - 1}$$

where $\mathrm{B}(s_i, t_i) = \Gamma(s_i)\Gamma(t_i)/\Gamma(s_i + t_i)$ is the beta function. Let A, B, C and D denote the points $(0.5, 0)$, $(0, 0.5)$, $(-0.5, 0)$ and $(0, -0.5)$ respectively. Then $(0.5 - a - b)$, $(0.5 + a - b)$, $(0.5 + a + b)$ and $(0.5 - a + b)$ are $\sqrt{2}$ times the minimum distances from (a, b) to the lines AB, BC, CD and DA respectively. Therefore the density is proportional to the product of the distances to the boundary lines

taken to different powers.

8.2.4 Posterior, Calibration and Calibrated Prediction

The event we wish to predict is linked to the event we are calibrating on by taking

$$\boldsymbol{\zeta}'|\boldsymbol{y}', \boldsymbol{\phi} \sim \mathcal{MVN}(\mu\mathbf{1} + \rho\boldsymbol{D}\boldsymbol{y}', (\boldsymbol{I} - \boldsymbol{C})^{-1})$$

and $z_i' = \mathbf{1}_{\{-1,1\}}[\zeta_i' > 0]$ for $i = 1, \ldots, n$. Then the posterior distribution is

$$\begin{aligned}
p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\boldsymbol{z}) &\propto \sum_{\boldsymbol{y}} p(\boldsymbol{z}|\boldsymbol{\zeta}) p(\boldsymbol{\zeta}|\boldsymbol{y}, \boldsymbol{\phi}) p(\boldsymbol{y}|m) p(m) p(\boldsymbol{\phi}) \\
&= p(\boldsymbol{z}|\boldsymbol{\zeta}) p(\boldsymbol{\zeta}|\boldsymbol{y}^{(m)}, \boldsymbol{\phi}) p(m) p(\mu) p(\rho) p(a, b) p(c, d) \\
&\propto \left(\prod_{i=1}^{n} \mathbf{1}[z_i = \mathbf{1}_{\{-1,1\}}[\zeta_i > 0]] \right) \mathbf{1}[c \geq 0] \mathbf{1}[d \geq 0] \mathbf{1}[c + d \leq 0.5] \\
&\quad \times \exp\left(-\frac{1}{2} \left(\boldsymbol{\zeta} - \mu\mathbf{1} - \rho\boldsymbol{D}\boldsymbol{y}^{(m)} \right)^{\mathrm{T}} (\boldsymbol{I} - \boldsymbol{C}) \left(\boldsymbol{\zeta} - \mu\mathbf{1} - \rho\boldsymbol{D}\boldsymbol{y}^{(m)} \right) \right. \\
&\qquad \left. - \frac{1}{2\sigma_\mu^2}(\mu - \nu_\mu)^2 - \frac{1}{2\sigma_\rho^2}(\rho - \nu_\rho)^2 \right) \\
&\quad \times \frac{2}{\mathrm{B}(s_1, t_1)\mathrm{B}(s_2, t_2)} (0.5 + a - b)^{s_1 - 1} (0.5 - a + b)^{t_1 - 1} \\
&\qquad \times (0.5 + a + b)^{s_2 - 1} (0.5 - a - b)^{t_2 - 1}. \qquad (8.8)
\end{aligned}$$

It is not possible to evaluate the posterior density directly because we do not know the normalising constant. Instead we generate a sample $\{\boldsymbol{\zeta}^{(k)}, \boldsymbol{\phi}^{(k)}, m^{(k)}|k = 1, \ldots, K\}$ using the MCMC algorithm described in Section 8.4.

To find an estimate of the marginal posterior for the simulation index, $p(m|\boldsymbol{z})$, simply disregard the other parameter values then $\{m^{(k)}|k = 1, \ldots, K\}$ is a sample from this distribution.

We can use the posterior distribution $p(m, \boldsymbol{\phi}|\boldsymbol{z})$, obtained by calibration on an observation $\boldsymbol{z}$, to make probabilistic predictions of a future event $\boldsymbol{z}'$. This is called

a calibrated prediction, and is computed as follows

$$\begin{aligned} p(z_i' = 1|\boldsymbol{z}) &= p(\zeta_i' > 0|\boldsymbol{z}) \\ &= \sum_{m=1}^{M} \int p(\zeta_i' > 0|\boldsymbol{\phi}, m) p(m, \boldsymbol{\phi}|\boldsymbol{z}) \, \mathrm{d}\boldsymbol{\phi} \\ &\approx \frac{1}{K} \sum_{k=1}^{K} p(\zeta_i' > 0|\boldsymbol{\phi}^{(k)}, m^{(k)}) \end{aligned} \tag{8.9}$$

where $\{\boldsymbol{\phi}^{(k)}, m^{(k)} | k = 1, \ldots, K\}$ is a sample from the posterior $p(\boldsymbol{\phi}, m|\boldsymbol{z})$, and $\zeta_i'|\boldsymbol{\phi}^{(k)}, m^{(k)}$ is normal with

$$\mathrm{E}\left(\zeta_i'|\boldsymbol{\phi}^{(k)}, m^{(k)}\right) = \mu^{(k)} + \rho^{(k)}(\boldsymbol{D}^{(k)}\boldsymbol{y}'^{(m^{(k)})})_i$$

and variance given by the ith diagonal of $(\boldsymbol{I} - \boldsymbol{C}^{(k)})^{-1}$, which can be calculated easily because $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{C}$ is block-circulant (see Section 8.3).

8.2.5 The HCAR Model as an Extension of the BC Model

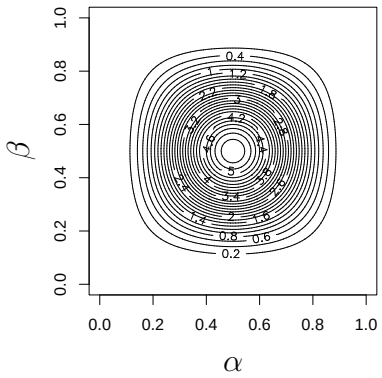
If we assume independence $\boldsymbol{C} = \boldsymbol{0}$ and no blur $\boldsymbol{D} = \boldsymbol{I}$, then $\zeta_i \overset{iid}{\sim} \mathcal{N}(\mu + \rho y_i, 1.0)$ and the HCAR model is exactly a binary channel (BC) model (see Chapter 5) where

$$\begin{aligned} \alpha &= p(z_i = 1|y_i = 1) = p(\zeta_i > 0|y_i = 1) = \Phi(\mu + \rho) \quad \text{and} \\ \beta &= p(z_i = -1|y_i = -1) = p(\zeta_i \leq 0|y_i = -1) = \Phi(\rho - \mu). \end{aligned}$$

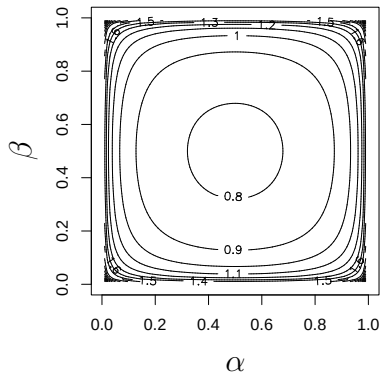
The parameters α and β are more tangible than μ and ρ, and the BC model using these parameters was examined in Chapter 5. Therefore it is beneficial to examine the density induced on (α, β) by the priors $\mu \sim \mathcal{N}(\nu_\mu, \sigma_\mu^2)$ and $\rho \sim \mathcal{N}(\nu_\rho, \sigma_\rho^2)$. Using the change of variables formula (see for example Grimmett and Stirzaker, 2002) we find

$$p(\alpha, \beta) = \frac{\exp\left(-\frac{1}{2\sigma_\mu^2}\left(\frac{\Phi^{-1}(\alpha) - \Phi^{-1}(\beta)}{2} - \nu_\mu\right)^2 - \frac{1}{2\sigma_\rho^2}\left(\frac{\Phi^{-1}(\alpha) + \Phi^{-1}(\beta)}{2} - \nu_\rho\right)^2\right)}{4\pi\sigma_\mu\sigma_\rho\phi(\Phi^{-1}(\alpha))\phi(\Phi^{-1}(\beta))}.$$

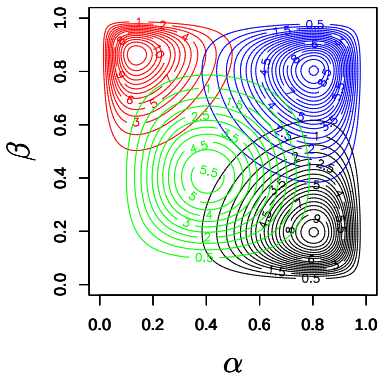
Figures 8.2(a) and 8.2(b) illustrate the effect of $\sigma = \sigma_\mu = \sigma_\rho$ when $\nu_\mu = \nu_\rho = 0.0$. A diffuse prior on μ and ρ does not equate to a diffuse prior on α and β. When $\sigma_\mu = \sigma_\rho = 0.5$, $p(\alpha, \beta) = 1.0$.



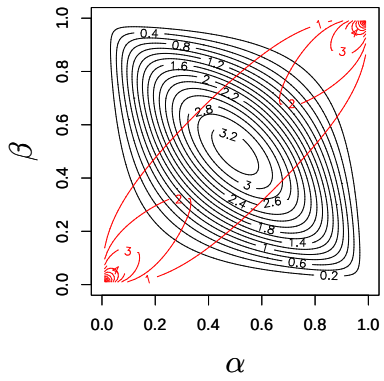
(a) $\nu_\mu = \nu_\rho = 0$, $\sigma_\mu = \sigma_\rho = 0.3$.



(b) $\nu_\mu = \nu_\rho = 0, \sigma_\mu = \sigma_\rho = 0.8$.



(c) $\sigma_\mu = \sigma_\rho = 0.3$: (black) $\nu_\mu = 0.7, \nu_\rho = 0$; (red) $\nu_\mu = -0.9, \nu_\rho = 0$; (blue) $\nu_\mu = 0, \nu_\rho = 0.7$; (green) $\nu_\mu = 0, \nu_\rho = -0.2$.



(d) $\nu_\mu = \nu_\rho = 0$: (black) $\sigma_\mu = 0.5, \sigma_\rho = 0.3$; (red) $\sigma_\mu = 0.3, \sigma_\rho = 0.9$.

Figure 8.2: The density for the binary channel (BC) model parameters $\alpha = p(z_i = 1|y_i = 1)$ and $\beta = p(z_i = -1|y_i = -1)$, corresponding to the HCAR model when $\boldsymbol{C} = \boldsymbol{0}$ and $\boldsymbol{D} = \boldsymbol{I}$ for various priors on μ and ρ.

Figure 8.2(c) shows the effect of ν_μ and ν_ρ when $\sigma_\mu = \sigma_\rho = 0.3$. Bias towards positives or negatives can be controlled with ν_μ; increasing ν_μ increases $p(z_i = 1|y_i = 1)$ and $p(z_i = 1|y_i = -1)$. The dependence on y_i is controlled by ρ; increasing ρ increases $p(z_i = 1|y_i = 1)$ and $p(z_i = -1|y_i = -1)$.

Figure 8.2(d) shows the effect of having different standard deviations. If $\sigma_\mu > \sigma_\rho$ then α and β are negatively correlated, whereas if $\sigma_\mu < \sigma_\rho$ then α and β are positively correlated. In the former case we are expressing that we are more certain about the dependence on y_i than the bias, for the latter the opposite is true.

8.3 Block-Circulant Matrices

The details of this discussion are taken from Rue and Held (2005) which is an excellent monograph on Gaussian Markov random fields. However, the results date back to Moran (1973). We discuss circulant matrices first and then block-circulant matrices.

A matrix $\boldsymbol{G}$ is *circulant* if and only if it can be written

$$\boldsymbol{G} = \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-1} \\ g_{n-1} & g_0 & g_1 & \cdots & g_{n-2} \\ g_{n-2} & g_{n-1} & g_0 & \cdots & g_{n-3} \\ \vdots & \vdots & \vdots & & \vdots \\ g_1 & g_2 & g_3 & \cdots & g_0 \end{pmatrix} = (g_{j-i \bmod n}),$$

where $\boldsymbol{g} = (g_0, g_1, \ldots, g_{n-1})$ is called the *base* of $\boldsymbol{G}$. The eigenvalues λ and (unit) eigenvectors $\boldsymbol{v}$ satisfy $\boldsymbol{G}\boldsymbol{v} = \lambda\boldsymbol{v}$ which defines a set of n linear difference equations with constant coefficients. Solving these we find

$$\lambda_j = \sum_{i=0}^{n-1} g_i \exp\left(-\frac{2\pi\iota ij}{n}\right) \quad \text{and} \tag{8.10}$$

$$\boldsymbol{v}_j = \frac{1}{\sqrt{n}}\left(1, \exp\left(-\frac{2\pi\iota j}{n}\right), \ldots, \exp\left(-\frac{2\pi\iota j(n-1)}{n}\right)\right)^{\mathrm{T}} \tag{8.11}$$

for $j = 0, \ldots, n-1$ where $\iota = \sqrt{-1}$ and the $1/\sqrt{n}$ ensures that $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} = 1$. Let $V = (\boldsymbol{v}_0|\boldsymbol{v}_1|\ldots|\boldsymbol{v}_{n-1})$ be the eigenvector matrix (which is independent of $\boldsymbol{g}$) this is the *discrete Fourier transform matrix*. Let $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{n-1})$ then, from Equations (8.10) and (8.11),

$$\boldsymbol{\Lambda} = \sqrt{n}\,\mathrm{diag}(\boldsymbol{V}\boldsymbol{g}) \tag{8.12}$$

and $\boldsymbol{G} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{H}}$ where $\boldsymbol{V}^{\mathrm{H}} = \boldsymbol{V}^{-1}$ is the conjugate transpose of $\boldsymbol{V}$ (swap rows and columns and negate the imaginary part).

The *discrete Fourier transform* is

$$\mathrm{DFT}(\boldsymbol{s}) = \boldsymbol{V}\boldsymbol{s} = \frac{1}{\sqrt{n}}\begin{pmatrix} \sum_{j=0}^{n-1} s_j \\ \sum_{j=0}^{n-1} s_j \exp\left(-\frac{2\pi\iota j}{n}\right) \\ \vdots \\ \sum_{j=0}^{n-1} s_j \exp\left(-\frac{2\pi\iota j(n-1)}{n}\right) \end{pmatrix}$$

and the *inverse discrete Fourier transform* is

$$\mathrm{IDFT}(\boldsymbol{s}) = \boldsymbol{V}^{\mathrm{H}}\boldsymbol{s} = \frac{1}{\sqrt{n}} \begin{pmatrix} \sum_{j=0}^{n-1} s_j \\ \sum_{j=0}^{n-1} s_j \exp\left(\frac{2\pi\iota j}{n}\right) \\ \vdots \\ \sum_{j=0}^{n-1} s_j \exp\left(\frac{2\pi\iota j(n-1)}{n}\right) \end{pmatrix}.$$

The inverse $\boldsymbol{G}^{-1} = \boldsymbol{V}\boldsymbol{\Lambda}^{-1}\boldsymbol{V}^{\mathrm{H}}$ is also circulant so $\boldsymbol{\Lambda}^{-1} = \sqrt{n}\,\mathrm{diag}(\boldsymbol{V}\boldsymbol{h})$ where $\boldsymbol{h}$ is the base of $\boldsymbol{G}^{-1}$. Furthermore from Equation (8.12), $(\boldsymbol{\Lambda})^{-1} = (\sqrt{n}\,\mathrm{diag}(\boldsymbol{V}\boldsymbol{g}))^{-1}$ so

$$\begin{aligned} \boldsymbol{h} &= \frac{1}{n}\boldsymbol{V}^{\mathrm{H}}(\boldsymbol{V}\boldsymbol{g})^{-1} \\ &= \frac{1}{n}\mathrm{IDFT}(\mathrm{DFT}(\boldsymbol{g}) \circledcirc (-1)) \end{aligned}$$

where $\circledcirc$ denotes elementwise power.

A matrix $\boldsymbol{G}$ is *block-circulant* if and only if it can be written

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{G}_0 & \boldsymbol{G}_1 & \boldsymbol{G}_2 & \dots & \boldsymbol{G}_{N-1} \\ \boldsymbol{G}_{N-1} & \boldsymbol{G}_0 & \boldsymbol{G}_1 & \dots & \boldsymbol{G}_{N-2} \\ \boldsymbol{G}_{N-2} & \boldsymbol{G}_{N-1} & \boldsymbol{G}_0 & \dots & \boldsymbol{G}_{N-3} \\ \vdots & \vdots & \vdots & & \vdots \\ \boldsymbol{G}_1 & \boldsymbol{G}_2 & \boldsymbol{G}_3 & \dots & \boldsymbol{G}_0 \end{pmatrix} = (\boldsymbol{G}_{j-i \bmod N}),$$

where for each i the $n \times n$ matrix $\boldsymbol{G}_i$ is circulant with base $\boldsymbol{g}_i$. The base of $\boldsymbol{G}$ is the $n \times N$ matrix $\boldsymbol{g} = (\boldsymbol{g}_0|\boldsymbol{g}_1|\dots|\boldsymbol{g}_{N-1})$.

Because $\boldsymbol{G}_i$ is circulant, $\boldsymbol{G}_i = \boldsymbol{V}_n\boldsymbol{\Lambda}_i\boldsymbol{V}_n^{\mathrm{H}}$ where $\boldsymbol{\Lambda}_i = \sqrt{n}\,\mathrm{diag}(\boldsymbol{V}_n\boldsymbol{g}_i)$, and therefore

$$\begin{aligned} \boldsymbol{G} &= \begin{pmatrix} \boldsymbol{V}_n & & \\ & \ddots & \\ & & \boldsymbol{V}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{\Lambda}_0 & \dots & \boldsymbol{\Lambda}_{N-1} \\ \vdots & & \vdots \\ \boldsymbol{\Lambda}_1 & \dots & \boldsymbol{\Lambda}_0 \end{pmatrix} \begin{pmatrix} \boldsymbol{V}_n^{\mathrm{H}} & & \\ & \ddots & \\ & & \boldsymbol{V}_n^{\mathrm{H}} \end{pmatrix} \\ &\doteq \boldsymbol{V}_n^N \boldsymbol{\Lambda} (\boldsymbol{V}_n^N)^{\mathrm{H}}. \end{aligned}$$

We want to diagonalize $\boldsymbol{G}$ to find the eigenvalues and eigenvectors, but $\boldsymbol{\Lambda}$ is not diagonal. However, we can permute $\boldsymbol{\Lambda}$ to make it block-diagonal with circulant blocks and then break this down using eigenvalues and eigenvectors. So we construct a permutation matrix $\boldsymbol{P}$ that takes the ith row of block row j to the jth

row of block row i, with $\boldsymbol{P}\boldsymbol{P} = \boldsymbol{I}$. Then

$$\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P} = \begin{pmatrix} \boldsymbol{D}_0 & & & \\ & \boldsymbol{D}_1 & & \\ & & \ddots & \\ & & & \boldsymbol{D}_{n-1} \end{pmatrix} \doteq \boldsymbol{D}$$

where $\boldsymbol{D}_i$ is a circulant matrix with base $\boldsymbol{d}_i$ (the jth element of $\boldsymbol{d}_i$ is the ith diagonal of $\boldsymbol{\Lambda}_j$). $\boldsymbol{D}_i$ is circulant so $\boldsymbol{D}_i = \boldsymbol{V}_N \boldsymbol{\Gamma}_i \boldsymbol{V}_N^{\mathrm{H}}$ where $\boldsymbol{\Gamma}_i = \sqrt{N}\mathrm{diag}(\boldsymbol{V}_N \boldsymbol{d}_i)$, now

$$\begin{aligned} \boldsymbol{G} &= \boldsymbol{V}_n^N \boldsymbol{\Lambda} (\boldsymbol{V}_n^N)^{\mathrm{H}} \\ &= \boldsymbol{V}_n^N \boldsymbol{P}\boldsymbol{D}\boldsymbol{P} (\boldsymbol{V}_n^N)^{\mathrm{H}} \\ &= (\boldsymbol{V}_n^N \boldsymbol{P} \boldsymbol{V}_N^n) \boldsymbol{\Gamma} ((\boldsymbol{V}_N^n)^{\mathrm{H}} \boldsymbol{P} (\boldsymbol{V}_n^N)^{\mathrm{H}}) \end{aligned}$$

where $\boldsymbol{\Gamma} = \mathrm{diag}(\boldsymbol{\Gamma}_0, \ldots, \boldsymbol{\Gamma}_{n-1})$ so $\boldsymbol{\Gamma}$ is diagonal and we have found our eigenvalues and eigenvectors.

$\boldsymbol{V}_n^N \boldsymbol{P} \boldsymbol{V}_N^n$ is the *two-dimensional discrete Fourier transform matrix*. Suppose the eigenvalues are stored in a $n \times N$ matrix $\boldsymbol{\Psi}$ so row i is the diagonal of $\boldsymbol{\Gamma}_i$. The two-dimensional discrete Fourier transform has elements

$$(\mathrm{DFT2}(\boldsymbol{s}))_{ij} = \frac{1}{\sqrt{nN}} \sum_{i'=0}^{n-1} \sum_{j'=0}^{N-1} s_{i'j'} \exp\left(-2\pi\iota (\frac{ii'}{n} + \frac{jj'}{N}) \right)$$

for $i = 0, \ldots, n-1$ and $j = 0, \ldots, N-1$, and the inverse has elements

$$(\mathrm{IDFT2}(\boldsymbol{s}))_{ij} = \frac{1}{\sqrt{nN}} \sum_{i'=0}^{n-1} \sum_{j'=0}^{N-1} s_{i'j'} \exp\left(2\pi\iota (\frac{ii'}{n} + \frac{jj'}{N}) \right).$$

Then

$$\boldsymbol{\Psi} = \sqrt{nN}\mathrm{DFT2}(\boldsymbol{g})$$

contains all the eigenvalues of $\boldsymbol{G}$. The base $\boldsymbol{h}$ of $\boldsymbol{G}^{-1}$ is

$$\boldsymbol{h} = \frac{1}{nN} \mathrm{IDFT2}\left(\mathrm{DFT2}(\boldsymbol{g}) \circledcirc (-1) \right).$$

To sample from $\boldsymbol{x} \sim \mathcal{MVN}(\boldsymbol{0}, \boldsymbol{Q}^{-1})$ where $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$ is block-circulant, simply note that $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{z}$ where $z_i \overset{iid}{\sim} \mathcal{N}(0,1)$ for $i = 1, \ldots, n$. Create a sample

from $\boldsymbol{z}$ then this translation can be done efficiently using the two-dimensional discrete Fourier transform to obtain a sample from the distribution for $\boldsymbol{x}$,

$$\boldsymbol{x} = \mathrm{Re}\left(\mathrm{DFT2}(((\sqrt{nN}\mathrm{DFT2}(\boldsymbol{g})) \circledcirc (-\frac{1}{2})) \odot \boldsymbol{z})\right)$$

where $\odot$ denotes elementwise multiplication.

For our model we need the eigenvalues to calculate the determinant and the diagonal entries of the covariance matrix $(\boldsymbol{I} - \boldsymbol{C})^{-1}$. The form of the precision matrix $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{C}$ is

$$\boldsymbol{I} - \boldsymbol{C} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{B} \\ \boldsymbol{B} & \boldsymbol{A} & \boldsymbol{B} & & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B} & \boldsymbol{A} & & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & & & & & \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} & & \boldsymbol{B} & \boldsymbol{A} \end{pmatrix}$$

where

$$A = \begin{pmatrix} 1 & -a & 0 & & -a \\ -a & 1 & -a & & 0 \\ & & & & \\ -a & 0 & & -a & 1 \end{pmatrix}$$

is a $r \times r$ circulant matrix with base $\boldsymbol{a} = (1, -a, 0, \dots, -a)$,

$$B = \begin{pmatrix} -b & 0 & 0 & & 0 \\ 0 & -b & 0 & & 0 \\ & & & & \\ 0 & 0 & & 0 & -b \end{pmatrix}$$

is a $r \times r$ circulant matrix with base $\boldsymbol{b} = (-b, 0, \dots, 0)$, and $\boldsymbol{0}$ a $r \times r$ matrix with only 0 entries. The base for $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{C}$ is the $r \times c$ matrix

$$\boldsymbol{q} = \begin{pmatrix} 1 & -b & 0 & 0 & -b \\ -a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ & & & & \\ -a & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The eigenvalues are

$$\begin{aligned}\boldsymbol{\Psi}_{ij} &= \sum_{i'=0}^{r-1}\sum_{j'=0}^{c-1} q_{i'j'} \exp\left(-2\pi\iota(\frac{ii'}{r} + \frac{jj'}{c})\right) \\ &= 1 - 2a\cos(\frac{2\pi i}{r}) - 2b\cos(\frac{2\pi j}{c})\end{aligned}$$

for $i = 0, 1, \ldots, r-1$ and $j = 0, 1, \ldots, c-1$.

All the diagonal elements are equal in the inverse so we only need to calculate h_{00},

$$\begin{aligned}h_{00} &= \frac{1}{rc}\left(\text{IDFT2}(\text{DFT2}(\boldsymbol{q}) \oslash (-1))\right)_{00} \\ &= \frac{1}{\sqrt{rc}}(\text{IDFT2}(\boldsymbol{\Psi} \oslash (-1)))_{00} \\ &= \frac{1}{rc}\sum_{i'=0}^{r-1}\sum_{j'=0}^{c-1}(1 - 2a\cos(\frac{2\pi i'}{r}) - 2b\cos(\frac{2\pi j'}{c}))^{-1},\end{aligned}$$

which is the mean of the inverse eigenvalues.

8.4 MCMC Algorithm

In this section we describe an MCMC algorithm for sampling from the posterior in Equation (8.8). Weir and Pettitt (1999) propose an algorithm which uses Metropolis-Hastings updates for each parameter, but we will show that for μ, ρ and ζ_i for $i = 1, \ldots, n$ we can use Gibbs updates.

8.4.1 μ Update

Assuming $p(\boldsymbol{\zeta}, \rho, a, b, c, d, m, \boldsymbol{z}) > 0.0$, the full conditional for μ is

$$\begin{aligned}&p(\mu|\boldsymbol{\zeta}, \rho, a, b, c, d, m, \boldsymbol{z}) \propto p(\boldsymbol{\zeta}|\mu, \rho, a, b, c, d, m)p(\mu) \\ &\propto \exp\left(-\frac{1}{2}(\boldsymbol{\zeta} - \mu\mathbf{1} - \rho\boldsymbol{D}\boldsymbol{y}^{(m)})^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{C})(\boldsymbol{\zeta} - \mu\mathbf{1} - \rho\boldsymbol{D}\boldsymbol{y}^{(m)}) - \frac{1}{2\sigma_\mu^2}(\mu - \nu_\mu)^2\right).\end{aligned}$$

By completing the square for μ we find

$$\mu|\boldsymbol{\zeta}, \rho, a, b, c, d, m, \boldsymbol{z} \sim \mathcal{N}\left(\frac{\sigma_\mu^2\mathbf{1}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{C})\boldsymbol{\zeta} - \rho\sigma_\mu^2\mathbf{1}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{C})\boldsymbol{D}\boldsymbol{y}^{(m)} + \nu_\mu}{\sigma_\mu^2\mathbf{1}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{C})\mathbf{1} + 1}, \frac{\sigma_\mu^2}{\sigma_\mu^2\mathbf{1}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{C})\mathbf{1} + 1}\right).$$

The Gibbs update consists of taking the new value of μ from this full conditional distribution.

8.4.2 ρ Update

Following the method used for μ, provided $p(\boldsymbol{\zeta}, \mu, a, b, c, d, m, \boldsymbol{z}) > 0.0$, the full conditional for ρ is

$$\rho|\boldsymbol{\zeta}, \mu, a, b, c, d, m, \boldsymbol{z} \sim \mathcal{N}\left(\frac{\sigma_\rho^2 \left(\boldsymbol{D}\boldsymbol{y}^{(m)}\right)^{\mathrm{T}} (\boldsymbol{I}-\boldsymbol{C})\boldsymbol{\zeta} - \mu\sigma_\rho^2 \left(\boldsymbol{D}\boldsymbol{y}^{(m)}\right)^{\mathrm{T}} (\boldsymbol{I}-\boldsymbol{C})\mathbf{1} + \nu_\rho}{\sigma_\rho^2 \left(\boldsymbol{D}\boldsymbol{y}^{(m)}\right)^{\mathrm{T}} (\boldsymbol{I}-\boldsymbol{C})\boldsymbol{D}\boldsymbol{y}^{(m)} + 1}, \frac{\sigma_\rho^2}{\sigma_\rho^2 \left(\boldsymbol{D}\boldsymbol{y}^{(m)}\right)^{\mathrm{T}} (\boldsymbol{I}-\boldsymbol{C})\boldsymbol{D}\boldsymbol{y}^{(m)} + 1}\right).$$

The Gibbs update consists of taking the new value of ρ from this full conditional distribution.

8.4.3 m Update

Propose a new value m' from $q(m'|m)$, then the proposal ratio is $q(m|m')/q(m'|m)$, and if $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{z}) > 0.0$ the posterior ratio is

$$\frac{p(\boldsymbol{\zeta}|\boldsymbol{\phi}, m')}{p(\boldsymbol{\zeta}|\boldsymbol{\phi}, m)}.$$

The acceptance probability for m' is the minimum of 1.0 and the product of the posterior ratio and proposal ratio. Suitable proposal distributions are discussed in Section 7.3.1 and the Robin Hood method for sampling from a discrete distribution is outlined in Section 7.3.2.

8.4.4 (a, b) Update

Following Weir and Pettitt (1999) we could adopt a proposal that is uniform on a square of given side, centred on the current value, with sides parallel to the parameter region boundaries, and conditional on lying within the feasible parameter space. However, for many prior specifications the posterior density for (a, b) is concentrated in a small region of the parameter space close to the boundary $|a| + |b| = 0.5$. In this case proposals that take (a, b) away from the boundary are

generally rejected. We found mixing of the MCMC algorithm was improved by using two perpendicular one-dimensional updates.

Let $\phi_1 = a + b$ and $\phi_2 = a - b$, then $-0.5 < \phi_1, \phi_2 < 0.5$ and we will update ϕ_1 and ϕ_2 independently. Propose a new value ϕ_1' from $\mathcal{U}(\max(\phi_1 - f, -0.5), \min(\phi_1 + f, 0.5))$, then calculate $a' = (\phi_1' + \phi_2)/2$ and $b' = (\phi_1' - \phi_2)/2$. The length of a proposal centred at $\phi_1 = a + b$ is

$$L(a, b) = \min(a + b + f, 0.5) - \max(a + b - f, -0.5).$$

Then, assuming $p(\boldsymbol{\zeta}, \mu, \rho, c, d, m, \boldsymbol{z}) > 0.0$, the acceptance probability is the minimum of 1.0 and

$$\frac{p(\boldsymbol{\zeta}|\mu, \rho, a', b', c, d, m)p(a', b')}{p(\boldsymbol{\zeta}|\mu, \rho, a, b, c, d, m)p(a, b)} \frac{L(a, b)}{L(a', b')}.$$

8.4.5 (c, d) Update

We take as our proposal density, $q(c', d'|c, d)$, a Uniform distribution on a square centred on the current value, with sides parallel to the parameter axes, and constrained to lie within the feasible parameter space (see Figure 8.3). Let $A(c, d)$ be the area of the proposal region when the proposal is centred at (c, d) then, assuming that $p(\boldsymbol{\zeta}, \mu, \rho, a, b, m, \boldsymbol{z}) > 0.0$ and $p(c, d) > 0.0$, the acceptance ratio is the minimum of 1.0 and

$$\frac{p(\boldsymbol{\zeta}|\mu, \rho, \boldsymbol{D'}, \boldsymbol{y}, \boldsymbol{C})}{p(\boldsymbol{\zeta}|\mu, \rho, \boldsymbol{D}, \boldsymbol{y}, \boldsymbol{C})} \frac{A(c, d)}{A(c', d')}.$$

8.4.6 ζ_i Update

In Weir and Pettitt (1999) the same Metropolis-Hastings update is used for all ζ_i. However, for some ζ_i mixing may be poor for one proposal distribution whilst for others mixing may be poor for another. This problem does not occur if we use

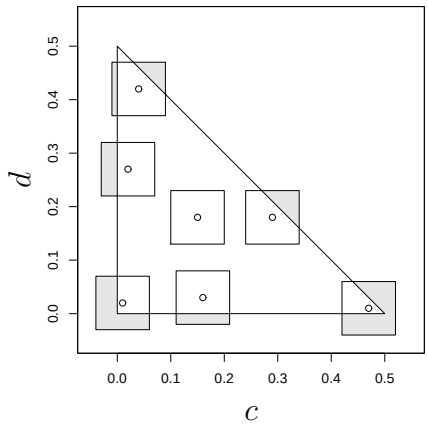


Figure 8.3: Possible proposal regions using a Uniform distribution on a square centred on the current value, with sides parallel to the parameter axes, and constrained to lie within the feasible parameter space.

Gibbs updates for each ζ_i. Assuming $p(\boldsymbol{\zeta}_{-i}, \boldsymbol{\phi}, m, \boldsymbol{z}) > 0.0$ we find

$$\begin{aligned} & p(\zeta_i|\boldsymbol{\zeta}_{-i}, \boldsymbol{\phi}, m, \boldsymbol{z}) \\ & \quad \propto p(\boldsymbol{z}|\zeta_i, \boldsymbol{\zeta}_{-i}, \boldsymbol{\phi}, m) p(\zeta_i|\boldsymbol{\zeta}_{-i}, \boldsymbol{\phi}, m) \\ & \quad = p(z_i|\zeta_i) p(\zeta_i|\boldsymbol{\zeta}_{-i}, \boldsymbol{\phi}, m) \\ & \quad = \mathbf{1}[z_i = \mathbf{1}_{\{-1,1\}}[\zeta_i > 0]] p(\zeta_i|\boldsymbol{\zeta}_{-i}, \boldsymbol{\phi}, m). \end{aligned}$$

So the full conditional for ζ_i is a truncated Normal distribution which can be sampled from using the following result.

Suppose $X \sim \mathcal{N}(\mu, \sigma^2)$ and Y is X truncated to be > 0, we write $Y \sim \mathbf{1}[Y > 0]\mathcal{N}(\mu, \sigma^2)$. Then

$$p_Y(y) = \begin{cases} p_X(y)/P(X > 0) & \text{if } y > 0 \\ 0 & \text{if } y \leq 0. \end{cases}$$

We want to find the inverse of the cumulative distribution function so a sample from Y can be obtained by taking a sample $u \sim \mathcal{U}[0, 1]$. The cdf for Y is

$$\begin{aligned} P(Y < y) &= \int_0^y p_Y(y)\, \mathrm{d}y \\ &= \int_0^y p_X(y)/P(X > 0)\, \mathrm{d}y \\ &= \frac{1}{P(X > 0)} \left(P(X < y) - P(X < 0)\right). \end{aligned}$$

Finding y such that $P(Y < y) = u$ is equivalent to solving

$$P(X < y) = uP(X > 0) + P(X < 0),$$

for y. Using $X = \sigma Z + \mu$ where $Z \sim \mathcal{N}(0, 1)$ we find

$$y = \sigma\Phi^{-1}\left(u(1 - \Phi\left(-\frac{\mu}{\sigma}\right)) + \Phi\left(-\frac{\mu}{\sigma}\right)\right) + \mu.$$

This is equivalent to taking $u \sim U[\Phi(-\mu/\sigma), 1]$ and applying the inverse cdf for X, $F_X^{-1}(u)$.

The convergence of the MCMC algorithm would probably be improved by observing that $\boldsymbol{\zeta}|\boldsymbol{z}, \mu, \rho, \boldsymbol{D}, m, \boldsymbol{C}$ is truncated multivariate Normal. However, sampling efficiently from a truncated multivariate Normal distribution is a difficult problem that, as yet, has no satisfactory solution (personal communication with Håvard Rue).

8.4.7 Initial Values

The initial values do not affect the stationary distribution of the Markov chain but may affect the time to convergence. Weir and Pettitt (1999) found the initial values of the mean parameters and $\boldsymbol{\zeta}$ did influence the time to convergence, although the spatial interaction parameters did not. Our algorithm differs from that of Weir and Pettitt (1999) in that we have opted for Gibbs updates where possible, which are less dependent on the initial value of the chain. However, we will adopt a similar method for defining sensible initial values.

We choose to take an arbitrary simulation, $m^{(0)} = 1$, and set the mean parameter $\mu^{(0)} = 0.0$ and the regression parameter $\rho^{(0)} = 1.0$. We assume spatial independence $a^{(0)} = b^{(0)} = 0.0$, and no blur $c^{(0)} = d^{(0)} = 0.0$.

In initialising $\boldsymbol{\zeta}$ we must respect $z_i = \mathbf{1}_{\{-1,1\}}[\zeta_i > 0]$ for $i = 1, \ldots, n$ to avoid division by zero in the posterior ratio. We take the mean of the full conditional distribution

$$\zeta_i|\boldsymbol{\zeta}_i, \boldsymbol{\phi}, m, \boldsymbol{z} \sim \mathbf{1}[z_i = \mathbf{1}_{\{-1,1\}}[\zeta_i > 0]]\,\mathcal{N}(y_i^{(m^{(0)})}, 1)$$

as the initial value $\zeta_i^{(0)}$ for $i = 1, \ldots, n$.

It just remains to calculate the mean of a truncated Normal. Suppose $X \sim \mathcal{N}(\mu, \sigma^2)$, and Y is X truncated ≤ 0, we write $Y \sim \mathbf{1}[Y \leq 0]\,\mathcal{N}(\mu, \sigma^2)$. Then

$$\begin{aligned}\mathrm{E}(Y) &= \int_{-\infty}^{\infty} y p_Y(y)\,\mathrm{d}y \\ &= \int_{-\infty}^{0} y p_X(y)/P(X \leq 0)\,\mathrm{d}y \\ &= -\frac{\sigma}{\sqrt{2\pi}\Phi(-\mu/\sigma)} \exp\left(-\frac{\mu^2}{2\sigma^2}\right) + \mu.\end{aligned}$$

Similarly, if Y is X truncated to > 0,

$$\mathrm{E}(Y) = \frac{\sigma}{\sqrt{2\pi}\Phi(\mu/\sigma)} \exp\left(-\frac{\mu^2}{2\sigma^2}\right) + \mu.$$

The parameters of the proposal distributions are determined by monitoring the convergence of the Markov chain.

8.4.8 Computational Efficiency

The iterative nature of MCMC, together with the large number of parameters, means the algorithm is very computer intensive. With this in mind we have tried to make use of the sparsity of the matrices involved to make the updates more efficient.

For example, in the Metropolis-Hastings updates the posterior ratio must be calculated; a quadratic of the form $\boldsymbol{v}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{C})\boldsymbol{v}$, where $\boldsymbol{v} = \boldsymbol{\zeta} - \mu\mathbf{1} - \rho\boldsymbol{D}\boldsymbol{y}^{(m)}$, appears in the numerator and denominator. However, in each row of $\boldsymbol{I} - \boldsymbol{C}$ there are only five nonzero elements so to calculate the quadratic we only need to loop over the rows and not the columns of the matrix. If the preceding update was also Metropolis-Hastings then one of the quadratics would already be known, but if it was Gibbs then both must be calculated.

8.5 Buscot Example

We return again to the Buscot dataset introduced in Section 2.4 so the results may be compared to those obtained with GLUE in Section 4.2.3, the BC model in Section 5.3 and the HBC model in Section 7.6.

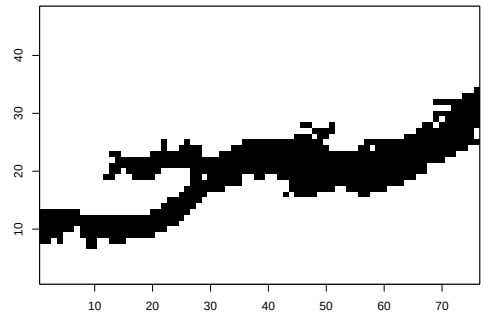
8.5.1 Realisations

In Figure 8.4 we present some realisations from the HCAR model using various values of the parameters μ, ρ, a, b, c and d and taking the simulation with index $m = 110$. For comparison $\boldsymbol{y}^{(110)}$ is shown in Figure 8.4(a). The effect of the precision matrix parameters, a and b, can be seen in Figure 8.4(c) where $a = 0.49$ and strong North-South dependence can be observed. Increasing ρ increases the probability that $z_i = \mathbf{1}_{\{-1,1\}}[\zeta_i > 0]$ equals y_i (compare Figures 8.4(b) and 8.4(d)). Increasing μ increases the probability that $z_i = 1$ regardless of the value of y_i (compare Figures 8.4(b) and 8.4(e)). Finally, the effect of the blur matrix parameters, c and d, is greatest at the boundary (see Figure 8.4(f)). Note that the priors we adopt in practice will typically force μ and ρ to take values much smaller than those investigated here, but for these values the realisations are both obvious and uninteresting.

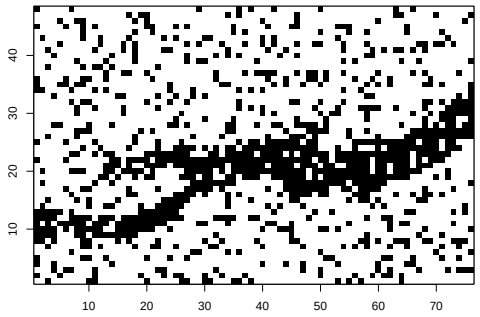
8.5.2 BC Model Examples

When we assume no blur, $\boldsymbol{D} = \boldsymbol{I}$, and spatial independence, $\boldsymbol{C} = \mathbf{0}$, the HCAR model is simply an alternative representation of the BC model (see Section 8.2.5). Figure 8.2 illustrates the density induced on (α, β) by the priors $\mu \sim \mathcal{N}(\nu_\mu, \sigma_\mu^2)$ and $\rho \sim \mathcal{N}(\nu_\rho, \sigma_\rho^2)$; together with the results using the BC model in Section 5.3, and with regard to the realisations in Figure 8.4, we can summarise the properties of the parameters μ and ρ as follows.

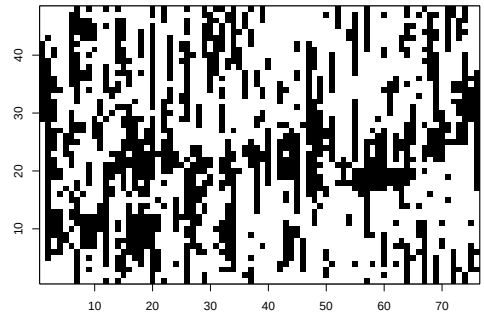
The mean parameter μ controls overall tendency toward 1 or -1. However, adopting a prior that allows $|\mu| \gg 0.0$ is equivalent to allowing $|\alpha - 0.5| \gg 0.0$ and $|\beta - 0.5| \gg 0.0$ which leads to the posterior for the simulation index, $p(m|\boldsymbol{z})$, being negligible for most values of m, as seen in Figure 5.5. The regression parameter ρ controls the dependence on $\boldsymbol{y}$, for example $\rho \gg 0.0$ implies we believe the simulations to be very accurate. As with μ, adopting a prior that allows $|\rho| \gg 0.0$ will result in $p(m|\boldsymbol{z})$ being negligible for most m.



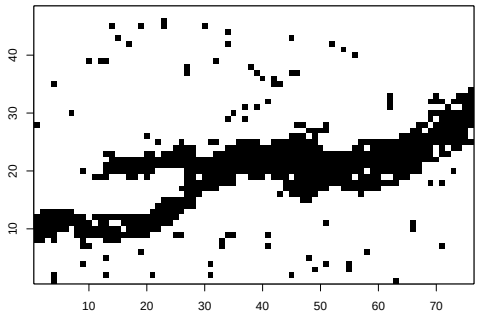
(a) Simulation $\boldsymbol{y}^{(110)}$.



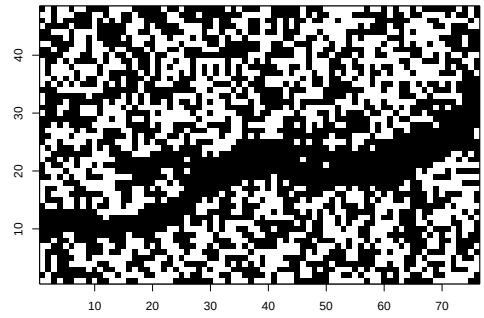
(b) $\mu = 0.0, \rho = 1.0, a = b = 0.0, c = d = 0.0$.



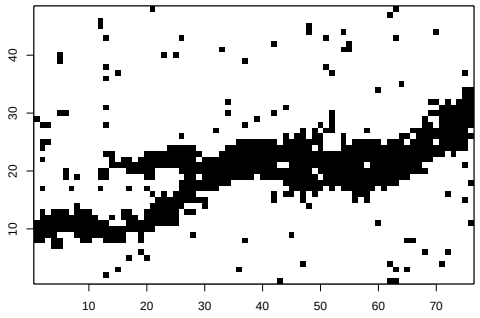
(c) $\mu = 0.0, \rho = 1.0, a = 0.49, b = 0.0, c = d = 0.0$.



(d) $\mu = 0.0, \rho = 2.0, a = b = 0.2, c = d = 0.0$.



(e) $\mu = 1.0, \rho = 1.0, a = b = 0.1, c = d = 0.0$.



(f) $\mu = 0.0, \rho = 2.0, a = b = 0.1, c = d = 0.2$.

Figure 8.4: Samples of $\boldsymbol{z}$ where $z_i = \mathbf{1}_{\{-1,1\}}[\zeta_i > 0]$ for $i = 1, \ldots, n$ and $\boldsymbol{\zeta} \sim \mathcal{MVN}(\mu\mathbf{1} + \rho\boldsymbol{D}\boldsymbol{y}^{(110)}, (\boldsymbol{I} - \boldsymbol{C})^{-1})$, where $\boldsymbol{y}^{(110)}$ is shown in Figure 8.4(a).

8.5.3 HCAR Model Examples

For comparison to the BC and HBC models, and for ease of interpretation, we plot the densities induced on $\alpha_i = p(z_i = 1|(\boldsymbol{Dy})_i = 1, \mu, \rho, a, b)$ and $\beta_i = p(z_i = -1|(\boldsymbol{Dy})_i = -1, \mu, \rho, a, b)$ by the prior $p(\mu, \rho, a, b)$ in the examples which follow. Similarly, we plot the densities induced on $\alpha_i = p(z'_i = 1|(\boldsymbol{Dy'})_i = 1, \mu, \rho, a, b)$ and $\beta_i = p(z'_i = -1|(\boldsymbol{Dy'})_i = -1, \mu, \rho, a, b)$ by the marginal posterior $p(\mu, \rho, a, b|\boldsymbol{z})$. The means of these distributions are $p(z_i = 1|(\boldsymbol{Dy})_i = 1)$, $p(z_i = -1|(\boldsymbol{Dy})_i = -1)$, $p(z'_i = 1|(\boldsymbol{Dy'})_i = 1, \boldsymbol{z})$ and $p(z'_i = -1|(\boldsymbol{Dy'})_i = -1, \boldsymbol{z})$ respectively.

In the first of our examples we look at the effect of spatial dependence. We set $\nu_\mu = \nu_\rho = 0.0$ and $\sigma_\mu = \sigma_\rho = 1/32$, and consider three cases for spatial dependence: spatially independent, $s = 100.0$ and $s = 1.0$, where $s = s_1 = t_1 = s_2 = t_2$. The results of calibration and calibrated prediction for this example are summarised in Figures 8.5 and 8.6.

The posterior for the simulation index, $p(m|\boldsymbol{z})$, becomes flatter as s decreases, leading to a flatter posterior for the calibration inputs, $p(\boldsymbol{\theta}|\boldsymbol{z})$. The density induced on $\alpha_i = p(z_i = 1|(\boldsymbol{Dy})_i = 1, \mu, \rho, a, b)$ and $\beta_i = p(z_i = -1|(\boldsymbol{Dy})_i = -1, \mu, \rho, a, b)$ by the prior of the likelihood parameters, $p(\mu, \rho, a, b)$, changes very little as s decreases. However, the density induced on $\alpha_i = p(z'_i = 1|(\boldsymbol{Dy'})_i = 1, \mu, \rho, a, b)$ and $\beta_i = p(z'_i = -1|(\boldsymbol{Dy'})_i = -1, \mu, \rho, a, b)$ by the posterior of the likelihood parameters, $p(\mu, \rho, a, b|\boldsymbol{z})$, becomes focused around 0.5 as s decreases. Correspondingly the calibrated predictions, $p(z'_i = 1|\boldsymbol{z})$, approach 0.5 as s decreases. In conclusion, allowing spatial dependence improves calibration by making $p(m|\boldsymbol{z})$ nonnegligible for more m, but increases the uncertainty in our calibrated predictions, which is undesirable.

The reason the calibrated prediction, $p(z_i = 1|\boldsymbol{z})$, approaches 0.5 as s decreases is because the posterior for $a+b$ becomes concentrated in a very small region close to the boundary $a + b = 0.5$. Now $1 - 2a - 2b$ is an eigenvalue of the precision matrix, $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{C}$, and the marginal variance of $\zeta_i|\mu, \rho, a, b, c, d, \boldsymbol{y}$ is the mean of the inverse eigenvalues (see Section 8.3). Therefore as s decreases this marginal variance becomes very large and $p(z'_i = 1|\boldsymbol{z})$ approaches 0.5.

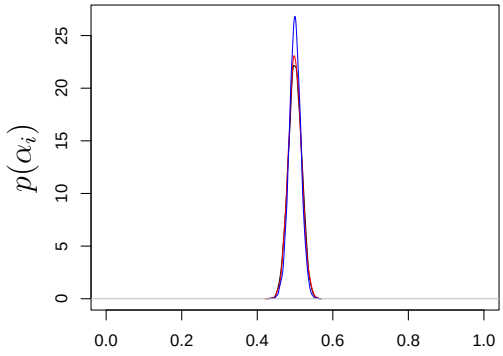
Note that because $1-2a-2b$ is an eigenvalue of the precision matrix, when $a+b = 0.5$ the precision matrix becomes singular and the HCAR model is undefined. Because the posterior for $a+b$ is concentrated close to the boundary $a+b = 0.5$, it is necessary to investigate what happens in the limit as $a+b \to 0.5$. In Section 8.7 we discover that there is a limit to this model but it is an improper distribution.

In the second example we look at the effect of the standard deviations σ_μ and σ_ρ. We set $\nu_\mu = \nu_\rho = 0.0$ and $s = 1.0$, and consider three values for $\sigma = \sigma_\mu = \sigma_\rho$: 1/4, 1/16 and 1/64. The results of calibration and calibrated prediction for this example are summarised in Figures 8.7 and 8.8.

The posterior for the simulation index, $p(m|\boldsymbol{z})$, becomes flatter as σ decreases, leading to a flatter posterior for the calibration inputs, $p(\boldsymbol{\theta}|\boldsymbol{z})$. This was to be expected from the relationship with the BC model (see Figure 8.2), and the properties of the BC model (see Section 5.3). The density induced on $\alpha_i = p(z_i = 1|(\boldsymbol{Dy})_i = 1, \mu, \rho, a, b)$ and $\beta_i = p(z_i = -1|(\boldsymbol{Dy})_i = -1, \mu, \rho, a, b)$ by the prior of the likelihood parameters, $p(\mu, \rho, a, b)$, becomes focused around 0.5 as σ decreases. Similarly, the density induced on $\alpha_i = p(z'_i = 1|(\boldsymbol{Dy'})_i = 1, \mu, \rho, a, b)$ and $\beta_i = p(z'_i = -1|(\boldsymbol{Dy'})_i = -1, \mu, \rho, a, b)$ by the posterior of the likelihood parameters, $p(\mu, \rho, a, b|\boldsymbol{z})$, becomes focused around 0.5 as σ decreases. Correspondingly the calibrated predictions, $p(z'_i = 1|\boldsymbol{z})$, approach 0.5 as σ decreases. In conclusion, decreasing σ improves calibration by making $p(m|\boldsymbol{z})$ nonnegligible for more m, but increases the uncertainty in our calibrated predictions, which is undesirable.

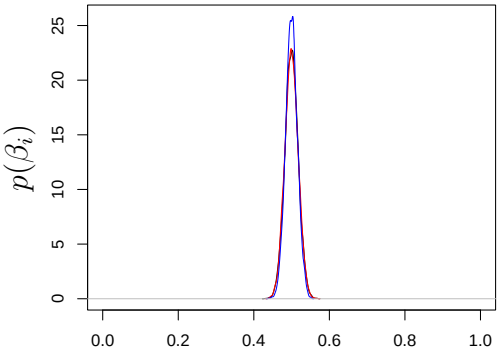
As for the BC model, local errors affect global fit, so we cannot obtain good results in calibration and calibrated prediction simultaneously. Therefore, following the logic that led us to consider the HBC model, it is natural to look at an extension of the HCAR model in which μ and ρ vary spatially. We present this extension in Section 8.8, and call it the heterogeneous hidden conditional autoregressive (HHCAR) model.

A final problem with the HCAR model is that of mixing of the MCMC algorithm. In the next section we investigate tools for diagnosing poor mixing and methods for improving mixing.



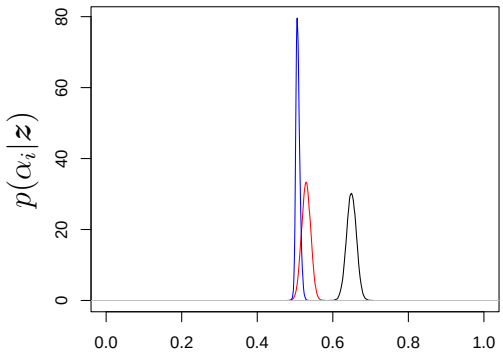
$$\alpha_i = p(z_i = 1|(\boldsymbol{Dy})_i = 1, \mu, \rho, a, b)$$

(a) Prior induced on α_i by $\mu \sim \mathcal{N}(\nu_\mu, \sigma_\mu^2)$, $\rho \sim \mathcal{N}(\nu_\rho, \sigma_\rho^2)$ and s.



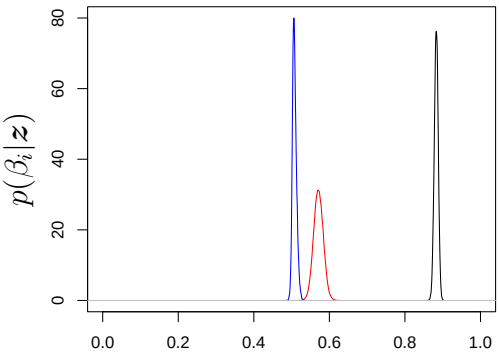
$$\beta_i = p(z_i = -1|(\boldsymbol{Dy})_i = -1, \mu, \rho, a, b)$$

(b) Prior induced on β_i by $\mu \sim \mathcal{N}(\nu_\mu, \sigma_\mu^2)$, $\rho \sim \mathcal{N}(\nu_\rho, \sigma_\rho^2)$ and s.



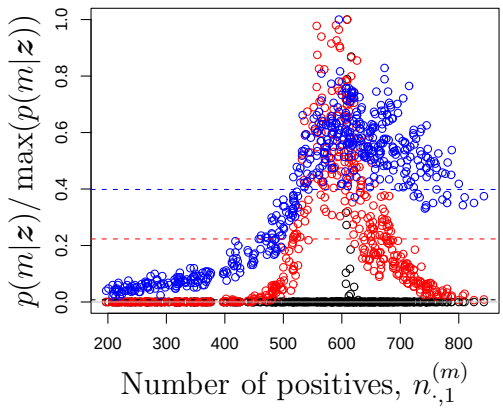
$$\alpha_i = p(z_i' = 1|(\boldsymbol{Dy}')_i = 1, \mu, \rho, a, b)$$

(c) Posterior induced on α_i by $p(\mu, \rho, a, b|\boldsymbol{z})$.

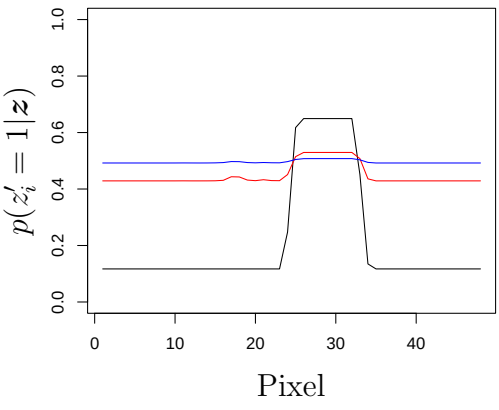


$$\beta_i = p(z_i' = -1|(\boldsymbol{Dy}')_i = -1, \mu, \rho, a, b)$$

(d) Posterior induced on β_i by $p(\mu, \rho, a, b|\boldsymbol{z})$.

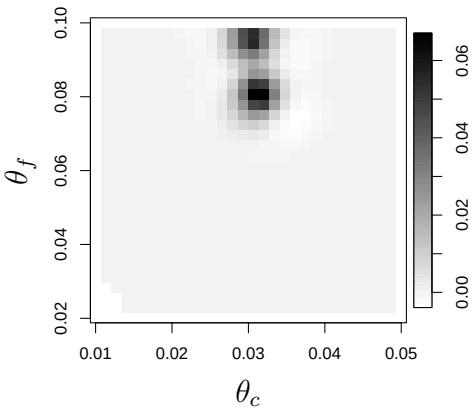


(e) Marginal posterior for m. The dashed lines show the means.

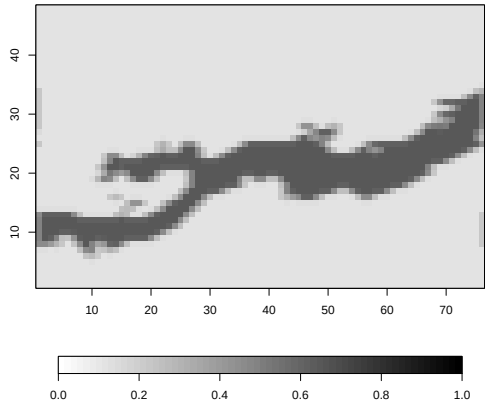


(f) Calibrated predictions for column 56.

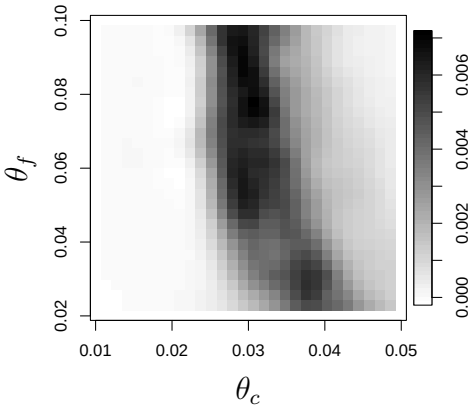
Figure 8.5: Three examples using the HCAR model and changing spatial dependence. The hyperparameters are $\nu_\mu = \nu_\rho = 0.0$ and $\sigma_\mu = \sigma_\rho = 1/32$ in all cases; and independent (black), $s = 100.0$ (red) and $s = 1.0$ (blue).



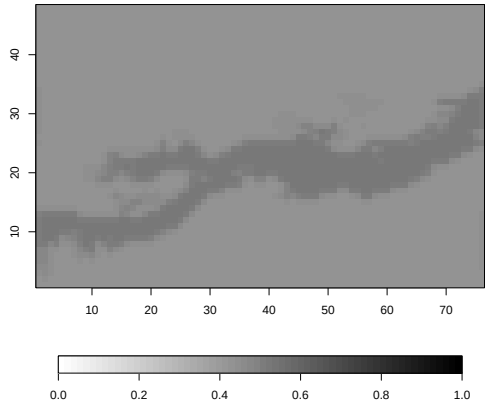
(a) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. Spatially independent.



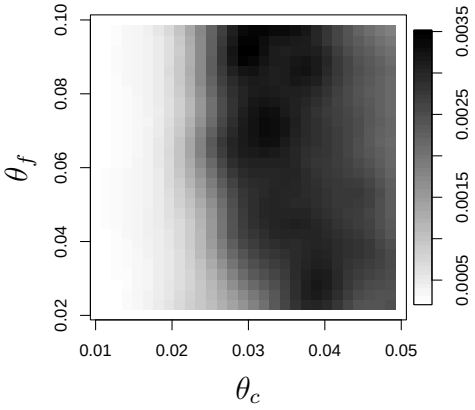
(b) $p(z_i' = 1|\boldsymbol{z})$, spatially independent.



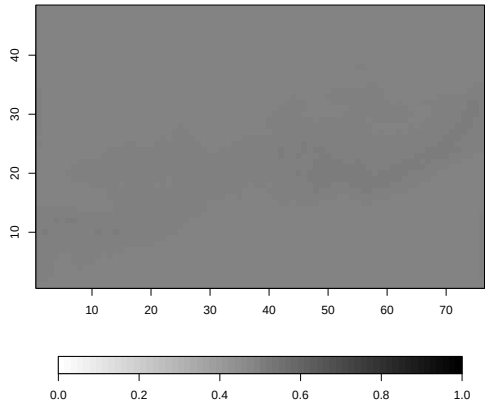
(c) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. s=100.0.



(d) $p(z_i' = 1|\boldsymbol{z})$, $s = 100.0$.

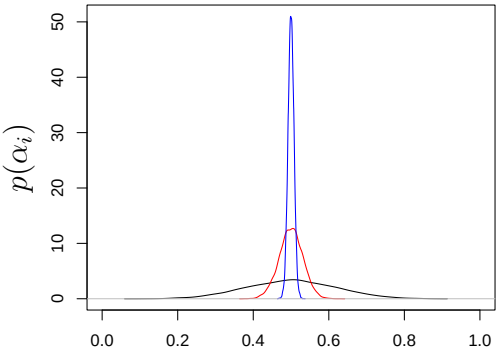


(e) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. s=1.0.



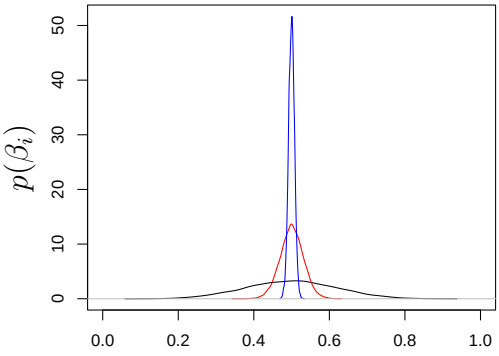
(f) $p(z_i' = 1|\boldsymbol{z})$, $s = 1.0$.

Figure 8.6: Three examples using the HCAR model and changing spatial dependence. The hyperparameters are $\nu_\mu = \nu_\rho = 0.0$ and $\sigma_\mu = \sigma_\rho = 1/32$ in all cases; and independent, $s = 100.0$ and $s = 1.0$.



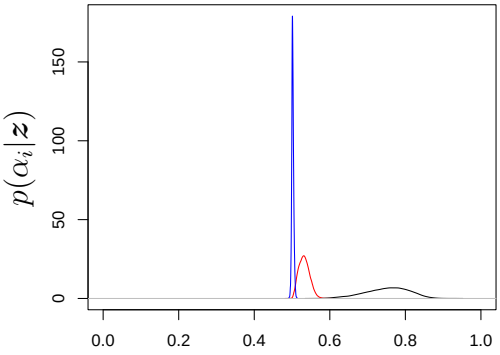
$$\alpha_i = p(z_i = 1|(\boldsymbol{Dy})_i = 1, \mu, \rho, a, b)$$

(a) Prior induced on α_i by $\mu \sim \mathcal{N}(\nu_\mu, \sigma_\mu^2)$, $\rho \sim \mathcal{N}(\nu_\rho, \sigma_\rho^2)$ and s.



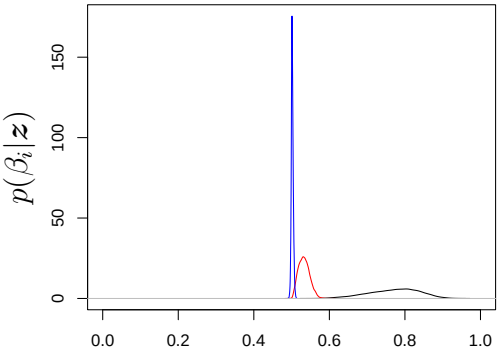
$$\beta_i = p(z_i = -1|(\boldsymbol{Dy})_i = -1, \mu, \rho, a, b)$$

(b) Prior induced on β_i by $\mu \sim \mathcal{N}(\nu_\mu, \sigma_\mu^2)$, $\rho \sim \mathcal{N}(\nu_\rho, \sigma_\rho^2)$ and s.



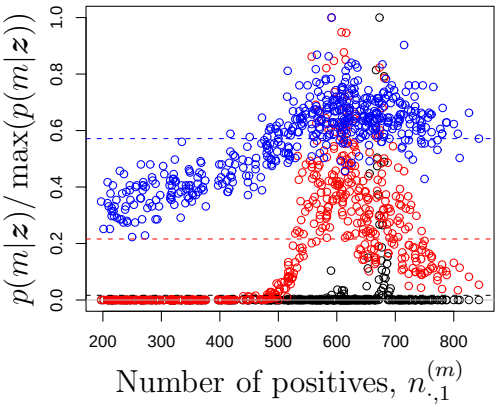
$$\alpha_i = p(z_i' = 1|(\boldsymbol{Dy}')_i = 1, \mu, \rho, a, b)$$

(c) Posterior induced on α_i by $p(\mu, \rho, a, b|\boldsymbol{z})$.

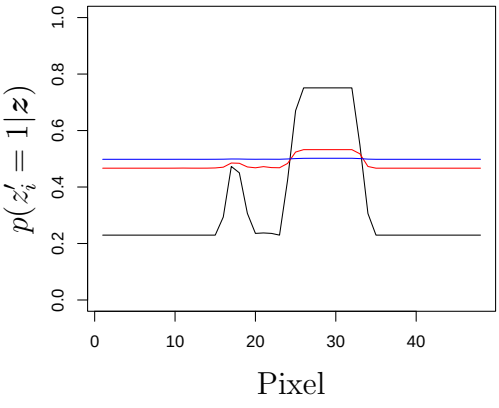


$$\beta_i = p(z_i' = -1|(\boldsymbol{Dy}')_i = -1, \mu, \rho, a, b)$$

(d) Posterior induced on β_i by $p(\mu, \rho, a, b|\boldsymbol{z})$.

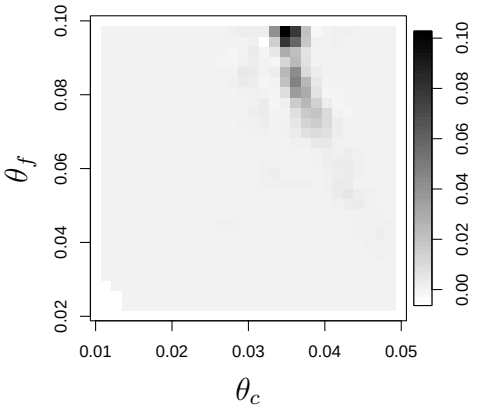


(e) Marginal posterior for m. The dashed lines show the means.

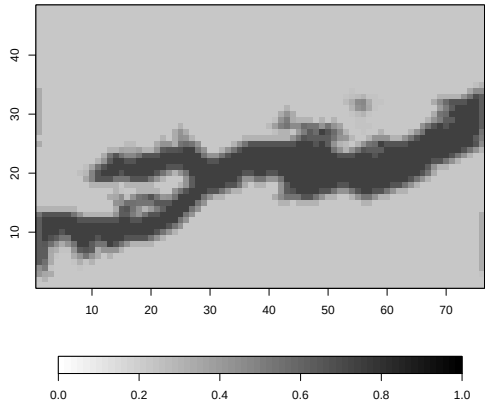


(f) Calibrated predictions for column 56.

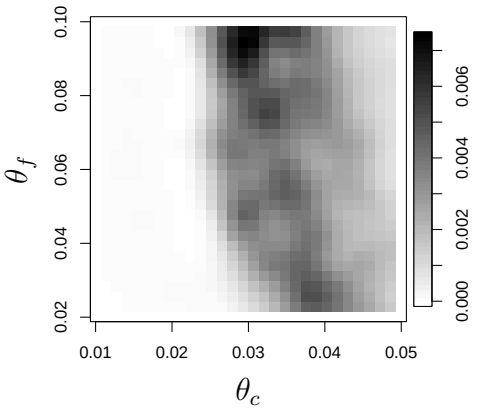
Figure 8.7: Three examples using the HCAR model and changing $\sigma = \sigma_\mu = \sigma_\rho$. The hyperparameters are $\nu_\mu = \nu_\rho = 0.0$ and $s = 1.0$ in all cases; and $\sigma = 1/4$ (black), $s = 1/16$ (red) and $s = 1/64$ (blue).



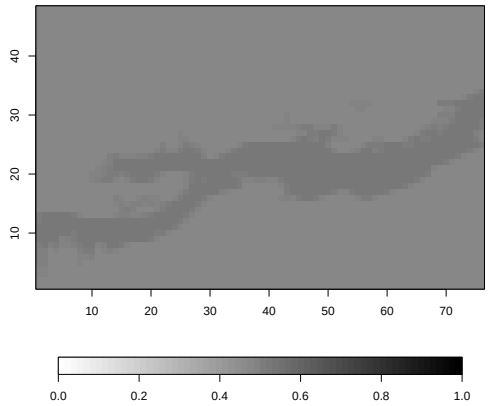
(a) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. $\sigma = 1/4$.



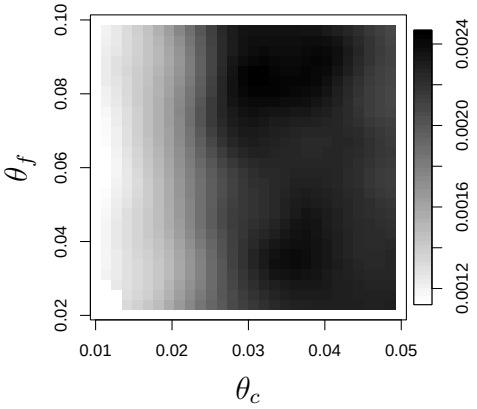
(b) $p(z'_i = 1|\boldsymbol{z})$, $\sigma = 1/4$.



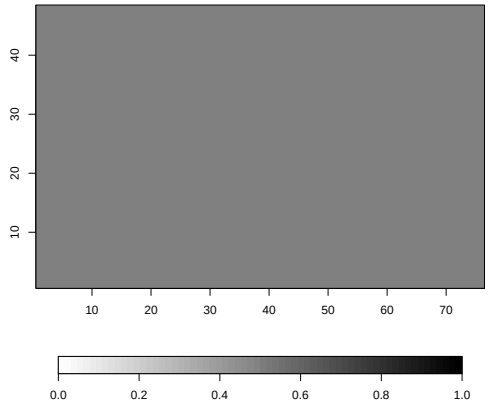
(c) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. $\sigma = 1/16$.



(d) $p(z'_i = 1|\boldsymbol{z})$, $\sigma = 1/16$.



(e) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. $\sigma = 1/64$.



(f) $p(z'_i = 1|\boldsymbol{z})$, $\sigma = 1/64$.

Figure 8.8: Three examples using the HCAR model and changing $\sigma = \sigma_\mu = \sigma_\rho$. The hyperparameters are $\nu_\mu = \nu_\rho = 0.0$ and $s = 1.0$ in all cases; and $\sigma = 1/4$, $s = 1/16$ and $s = 1/64$.

8.6 Improving Mixing

In theory, the Markov chain described in Section 8.4 is irreducible and the distribution of the Markov chain converges to the stationary distribution $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\boldsymbol{z})$. In practice, for certain values of the hyperparameters the Markov chain mixes very slowly, so a realisation $\{(\boldsymbol{\zeta}^{(1)}, \boldsymbol{\phi}^{(1)}, m^{(1)}), \ldots, (\boldsymbol{\zeta}^{(K)}, \boldsymbol{\phi}^{(K)}, m^{(K)})\}$ does not approximate a sample from the stationary distribution $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\boldsymbol{z})$ for practical values of K.

In this section we describe diagnostic tools for identifying convergence and investigating the reasons for slow mixing, then we present methods for improving mixing. We will not be illustrating these methods for improving mixing with examples because they were implemented using a code which we subsequently found to contain errors. These methods are nonessential to the discussion of the HCAR model, because the Markov chain mixes well for many values of the hyperparameters. Therefore, because of time constraints, we have chosen not to recode these methods.

8.6.1 Diagnostic Tools

The divergence of our Markov chain is characterised by a peculiar marginal posterior for the simulation index $p(m|\boldsymbol{z})$ and heterogeneous changes in plots of $m^{(k)}$ versus iteration k. An easy way to check convergence is to make a few realisations of the Markov chain, possibly using different initial values, and compare the estimates of the stationary distribution. This is called a test of robustness.

In the special case of spatial independence, $\boldsymbol{C} = \boldsymbol{0}$, and no blur, $\boldsymbol{D} = \boldsymbol{I}$, the estimated posteriors can be compared to exact results. The likelihood $p(\boldsymbol{z}|\boldsymbol{\phi}, \boldsymbol{y}) = \prod_{i=1}^{n} p(z_i|\boldsymbol{\phi}, y_i)$ where

$$\begin{aligned} p(z_i = 1|\boldsymbol{\phi}, y_i) &= \int_{-\infty}^{\infty} p(z_i = 1, \zeta_i|\boldsymbol{\phi}, y_i)\, \mathrm{d}\zeta_i \\ &= p(\zeta_i > 0|\boldsymbol{\phi}, y_i) \\ &= \Phi(\mu + \rho y_i), \end{aligned}$$

so the model is now a binomial, probit link GLM,

$$z_i \sim \text{Bin}(1, \Phi(\mu + \rho y_i)).$$

The maximum likelihood estimates are found by setting the partial differentials of the log likelihood with respect to μ and ρ to zero. Let $n_{r,s} = \sum_{i=1}^{n} \mathbf{1}[z_i = r]\mathbf{1}[y_i = s]$, then the maximum likelihood estimates $\hat{\mu}$ and $\hat{\rho}$ must satisfy

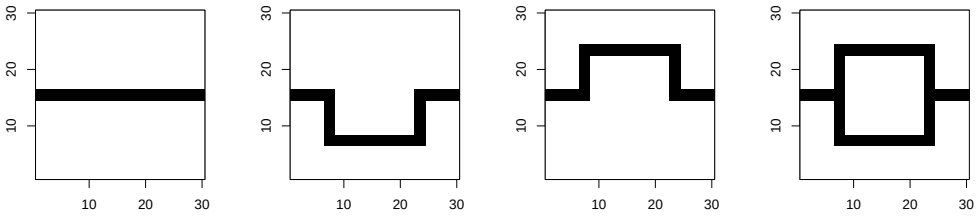
$$\Phi(\hat{\mu} + \hat{\rho}) = \frac{n_{1,1}}{n_{1,1} + n_{-1,1}} \quad \text{and}$$
$$\Phi(\hat{\mu} - \hat{\rho}) = \frac{n_{1,-1}}{n_{1,-1} + n_{-1,-1}}.$$

If the prior is relatively uninformative then the posterior modes should be close to these maximum likelihood estimates.

In Section 8.2.5 we showed how the HCAR model degenerates to a BC model when $\boldsymbol{C} = \mathbf{0}$ and $\boldsymbol{D} = \boldsymbol{I}$. The normal priors on μ and ρ induce a prior on the BC model parameters α and β (see Figure 8.2). If this prior can be approximated by beta distributions on α and β then we can find the posterior analytically (see Section 5.2), and compare this to the estimated posterior sample from the Markov chain.

In performing these analyses we identified that the rate of convergence of the Markov chain was most affected by the prior for ρ. If the prior, $\rho \sim \mathcal{N}(\nu_\rho, \sigma_\rho^2)$, allows the magnitude of ρ to be large then mixing is poor, for example with $\nu_\rho = 0.0$ mixing is poor for $\sigma_\rho > 1.0$. However, if the prior constrains ρ to be too close to 0.0 then $p(m|\boldsymbol{z})$ will be flat, which will rarely be appropriate. We want to be able to explore priors that lead to $p(m|\boldsymbol{z})$ being different for different m.

To identify a range of σ_ρ for which the Markov chain mixes well and the corresponding marginal posterior $p(m|\boldsymbol{z})$ is not flat, we devised the following experiment. First create a dataset consisting of an observation, $\boldsymbol{z}$, and three simulations $\boldsymbol{y}^{(1)}$, $\boldsymbol{y}^{(2)}$ and $\boldsymbol{y}^{(3)}$ such that $p(m = 1|\boldsymbol{z}) = p(m = 2|\boldsymbol{z})$ and $p(m = 3|\boldsymbol{z}) < p(m = 1|\boldsymbol{z})$ (for example see Figure 8.9). For a range of σ_ρ values generate realisations from the Markov chain to obtain estimates, $\tilde{p}(m|\boldsymbol{z})$, of $p(m|\boldsymbol{z})$. For large values of σ_ρ mixing is poor and realisations from the Markov chain are not good estimates of $p(m|\boldsymbol{z})$, this is characterised by $\tilde{p}(m = 1|\boldsymbol{z}) \neq$



(a) $\boldsymbol{z}$ (b) $\boldsymbol{y}^{(m^{(1)})}$ (c) $\boldsymbol{y}^{(m^{(2)})}$ (d) $\boldsymbol{y}^{(m^{(3)})}$

Figure 8.9: Example dataset for testing mixing of the Markov chain.

$\tilde{p}(m = 2|\boldsymbol{z})$. For small values of σ_ρ mixing is good and the estimate is good, but $\tilde{p}(m = 1|\boldsymbol{z}) = \tilde{p}(m = 2|\boldsymbol{z}) = \tilde{p}(m = 3|\boldsymbol{z})$, i.e. the model does not discriminate between simulations. We want to identify a region between these limiting cases where mixing is good and the marginal posterior for the simulation index is not flat, we do this by looking for results where $\tilde{p}(m = 1|\boldsymbol{z}) = \tilde{p}(m = 2|\boldsymbol{z})$ and $\tilde{p}(m = 3|\boldsymbol{z}) < \tilde{p}(m = 1|\boldsymbol{z})$. In Section 8.5 the examples for the Buscot dataset used $1/64 \leq \sigma_\rho \leq 1/4$ with $\nu_\rho = 0.0$.

Using the above methods we can identify *when* the Markov chain mixes poorly but not *why*. For a point $(\boldsymbol{\zeta}^{(k)}, \boldsymbol{\phi}^{(k)}, m^{(k)})$ in a realisation of the Markov chain and an arbitrary simulation indexed by m, we can calculate the log posterior ratio

$$\log\left(\frac{p(\boldsymbol{\zeta}^{(k)}, \boldsymbol{\phi}^{(k)}, m|\boldsymbol{z})}{p(\boldsymbol{\zeta}^{(k)}, \boldsymbol{\phi}^{(k)}, m^{(k)}|\boldsymbol{z})}\right).$$

For values of σ_ρ that result in poor mixing we find that the log posterior ratio is very small for different simulation indexes m, whether or not $\boldsymbol{y}^{(m)}$ is closer to $\boldsymbol{z}$. Whereas for Metropolis-Hastings updates of continuous parameters the size of the proposal can be reduced to improve mixing, for the simulation index m, this is not possible. The dependence on $\boldsymbol{y}^{(m)}$ is controlled by ρ, so this explains why the Markov chain convergence is so sensitive to $p(\rho)$.

The log posterior ratio

$$\log\left(\frac{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m = 2|\boldsymbol{z})}{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m = 1|\boldsymbol{z})}\right)$$

can be very small even when $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$ differ by only a few pixels. Create a dataset of observed data $\boldsymbol{z}$ and two simulations $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$ satisfying $p(m =$

$1|\boldsymbol{z}) = p(m = 2|\boldsymbol{z})$ (see for example Figure 8.9). Then fix $m = 1$ and make a realisation of the Markov chain to obtain a sample from $p(\boldsymbol{\zeta}, \boldsymbol{\phi}|m = 1, \boldsymbol{z})$, similarly for $p(\boldsymbol{\zeta}, \boldsymbol{\phi}|m = 2, \boldsymbol{z})$. (Note that mixing is not a problem because m is fixed.) Then because

$$\frac{p(\boldsymbol{\zeta}, m = 2, \boldsymbol{\phi}|\boldsymbol{z})}{p(\boldsymbol{\zeta}, m = 1, \boldsymbol{\phi}|\boldsymbol{z})} = \frac{p(\boldsymbol{\zeta}, \boldsymbol{\phi}|m = 2, \boldsymbol{z})}{p(\boldsymbol{\zeta}, \boldsymbol{\phi}|m = 1, \boldsymbol{z})},$$

appears in the Metropolis-Hastings update for m, comparing the conditional posteriors for the other parameters given the simulation index tells us something about mixing. If $m = 1$ it is likely that $\boldsymbol{\zeta}$ and $\boldsymbol{\phi}$ will take values for which $p(\boldsymbol{\zeta}, \boldsymbol{\phi}|m = 1, \boldsymbol{z})$ is large. We want to know if $p(\boldsymbol{\zeta}, \boldsymbol{\phi}|m = 2, \boldsymbol{z})$ is also large at this point, if not it is less likely a proposal of $m = 2$ will be accepted. We find that $p(\boldsymbol{\zeta}|m = 1, \boldsymbol{z})$ and $p(\boldsymbol{\zeta}|m = 2, \boldsymbol{z})$ are very different for priors which lead to poor mixing.

In summary, for two simulations indexed by $m = 1$ and $m = 2$ it may be that $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m = 1|\boldsymbol{z}) = p(\boldsymbol{\zeta}', \boldsymbol{\phi}', m = 2|\boldsymbol{z})$ for some parameters $\boldsymbol{\zeta}, \boldsymbol{\zeta}', \boldsymbol{\phi}$ and $\boldsymbol{\phi}'$, but for the Metropolis-Hastings update of m (see Section 8.4.3), only m changes and it may be that $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m = 1|\boldsymbol{z}) \gg p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m = 2|\boldsymbol{z})$. In the following sections we present a number of methods for improving mixing.

8.6.2 Linking Simulations with a Sequence of Images

When a new simulation index proposal is not accepted it may be because the new simulation differs from the current one by many pixels. If we reduce the number of pixels that are different between the simulations, we may improve the chance that the proposal is accepted.

Suppose there are two simulations $\boldsymbol{y}^{(a)}$ and $\boldsymbol{y}^{(b)}$, for which $p(m = a|\boldsymbol{z}) = p(m = b|\boldsymbol{z})$ but proposals between simulations are rarely accepted because $p(\boldsymbol{\zeta}, \boldsymbol{\phi}|m = a, \boldsymbol{z})$ and $p(\boldsymbol{\zeta}, \boldsymbol{\phi}|m = b, \boldsymbol{z})$ have little overlap. Then construct a series of images between the simulations that change by one pixel at a time, $\boldsymbol{y}^{(a)} = \boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(M)} = \boldsymbol{y}^{(b)}$. Let $Y_1 = (\boldsymbol{y}^{(a)}, \boldsymbol{y}^{(b)})$ and $Y_2 = (\boldsymbol{y}^{(a)} = \boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(M)} = \boldsymbol{y}^{(b)})$. Then generate a sample from the posterior $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\boldsymbol{z}, Y_2)$, $\{\boldsymbol{\zeta}^{(k)}, \boldsymbol{\phi}^{(k)}, m^{(k)}|k = 1, \ldots, K\}$, using an MCMC algorithm with proposal distribution $q(m' = i + 1|m = i) = q(m' = i - 1|m = i) = 0.5$ for

$1 < i < M$ and $q(m' = 2|m = 1) = q(m' = M - 1|m = M) = 1.0$. Because the proposed simulation, $\boldsymbol{y}^{(m')}$, only differs from the current simulation, $\boldsymbol{y}^{(m)}$, by one pixel the acceptance probability will hopefully be large.

Given the posterior $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\boldsymbol{z}, Y_2)$, we can calculate the posterior ratio given Y_1,

$$\frac{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m = a|\boldsymbol{z}, Y_1)}{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m = b|\boldsymbol{z}, Y_1)} = \frac{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m = a|\boldsymbol{z}, Y_2)}{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m = b|\boldsymbol{z}, Y_2)} \frac{p(m = b|Y_2)}{p(m = a|Y_2)} \frac{p(m = a|Y_1)}{p(m = b|Y_1)}.$$

In particular, if $p(m|Y_1) \propto 1.0$ and $p(m|Y_2) \propto 1.0$, then

$$\frac{p(m = a|\boldsymbol{z}, Y_1)}{p(m = b|\boldsymbol{z}, Y_1)} = \frac{\sum_{k=1}^{K} \mathbf{1}[m^{(k)} = a]}{\sum_{k=1}^{K} \mathbf{1}[m^{(k)} = b]}.$$

The problem with this method is that if the linking images are more probable as regressors of the observed data than the simulations, the Markov chain may rarely visit the simulations of interest.

8.6.3 Mixing Distributions

The Markov chain mixes well for some choices of the prior for ρ but poorly for others. In this section we describe three methods for conferring the mixing properties of one distribution, called the *mixing distribution*, onto the distribution of interest. An overview of the following methods can be found in Gilks and Roberts (1996).

Importance Sampling

Let Δ be a subset of the sample space Ω, then the posterior probability that $(\boldsymbol{\zeta}, \boldsymbol{\phi}, m) \in \Delta$ is

$$\begin{aligned} p((\boldsymbol{\zeta}, \boldsymbol{\phi}, m) \in \Delta|\sigma_\rho, \boldsymbol{z}) &= \iint \sum_{m=1}^{M} \mathbf{1}[(\boldsymbol{\zeta}, \boldsymbol{\phi}, m) \in \Delta] p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho, \boldsymbol{z}) \, \mathrm{d}\boldsymbol{\zeta} \, \mathrm{d}\boldsymbol{\phi} \\ &= \mathrm{E}\left(\mathbf{1}[(\boldsymbol{\zeta}, \boldsymbol{\phi}, m) \in \Delta]|\sigma_\rho, \boldsymbol{z}\right) \\ &\approx \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}[(\boldsymbol{\zeta}^{(k)}, \boldsymbol{\phi}^{(k)}, m^{(k)}) \in \Delta] \end{aligned}$$

where $\{(\boldsymbol{\zeta}^{(1)}, \boldsymbol{\phi}^{(1)}, m^{(1)}), \ldots, (\boldsymbol{\zeta}^{(K)}, \boldsymbol{\phi}^{(K)}, m^{(K)})\}$ is a sample from $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho, \boldsymbol{z})$, and we show the prior standard deviation for ρ explicitly. Now suppose it is

difficult to generate a sample from $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho, \boldsymbol{z})$ but easy to generate a sample from the mixing distribution $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma'_\rho, \boldsymbol{z})$, then

$$\begin{aligned}
&p((\boldsymbol{\zeta}, \boldsymbol{\phi}, m) \in \Delta|\sigma_\rho, \boldsymbol{z}) \\
&\quad = \iint \sum_{m=1}^{M} \mathbf{1}[(\boldsymbol{\zeta}, \boldsymbol{\phi}, m) \in \Delta] \frac{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho, \boldsymbol{z})}{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma'_\rho, \boldsymbol{z})} p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma'_\rho, \boldsymbol{z}) \, \mathrm{d}\boldsymbol{\zeta} \, \mathrm{d}\boldsymbol{\phi} \\
&\quad = \mathrm{E}\left(\mathbf{1}[(\boldsymbol{\zeta}, \boldsymbol{\phi}, m) \in \Delta] \frac{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho, \boldsymbol{z})}{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma'_\rho, \boldsymbol{z})} \middle| \sigma'_\rho, \boldsymbol{z} \right) \\
&\quad \approx \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}[\left(\boldsymbol{\zeta}'^{(k)}, \boldsymbol{\phi}'^{(k)}, m'^{(k)}\right) \in \Delta] \frac{p(\boldsymbol{\zeta}'^{(k)}, \boldsymbol{\phi}'^{(k)}, m'^{(k)}|\sigma_\rho, \boldsymbol{z})}{p(\boldsymbol{\zeta}'^{(k)}, \boldsymbol{\phi}'^{(k)}, m'^{(k)}|\sigma'_\rho, \boldsymbol{z})}
\end{aligned}$$

where $\{(\boldsymbol{\zeta}'^{(1)}, \boldsymbol{\phi}'^{(1)}, m'^{(1)}), \ldots, (\boldsymbol{\zeta}'^{(K)}, \boldsymbol{\phi}'^{(K)}, m'^{(K)})\}$ is a sample from $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma'_\rho, \boldsymbol{z})$.

The mixing distribution must be different from the distribution of interest to aid mixing. However, if it is too different then the weights, $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho, \boldsymbol{z})/p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma'_\rho, \boldsymbol{z})$, will be close to zero for values which occur regularly, and will be very large for values which occur rarely. Consequently, our estimates will be dominated by a very small subset of the Markov chain.

Assuming $p(\boldsymbol{z}|\boldsymbol{\zeta}') > 0.0$, the weights may be written

$$\frac{p(\boldsymbol{\zeta}'^{(k)}, \boldsymbol{\phi}'^{(k)}, m'^{(k)}|\sigma_\rho, \boldsymbol{z})}{p(\boldsymbol{\zeta}'^{(k)}, \boldsymbol{\phi}'^{(k)}, m'^{(k)}|\sigma'_\rho, \boldsymbol{z})} = \frac{p(\rho'^{(k)}|\sigma_\rho)}{p(\rho'^{(k)}|\sigma'_\rho)} \frac{p(\boldsymbol{z}|\sigma'_\rho)}{p(\boldsymbol{z}|\sigma_\rho)},$$

where the ratio $p(\boldsymbol{z}|\sigma'_\rho)/p(\boldsymbol{z}|\sigma_\rho)$ does not cancel. From the importance sampling identity for ratios of normalising constants (see Equation (6.13)), we find

$$\frac{p(\boldsymbol{z}|\sigma_\rho)}{p(\boldsymbol{z}|\sigma'_\rho)} = \mathrm{E}_{\sigma'_\rho}\left(\frac{p(\boldsymbol{z}, \boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho)}{p(\boldsymbol{z}, \boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma'_\rho)} \right)$$

where $\mathrm{E}_{\sigma'_\rho}(\cdot)$ is the expectation with respect to $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma'_\rho, \boldsymbol{z})$. Furthermore,

$$\frac{p(\boldsymbol{z}, \boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho)}{p(\boldsymbol{z}, \boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma'_\rho)} = \frac{p(\rho|\sigma_\rho)}{p(\rho|\sigma'_\rho)}$$

provided $p(\boldsymbol{z}|\boldsymbol{\zeta}) \neq 0$. So we can estimate the ratio $p(\boldsymbol{z}|\sigma'_\rho)/p(\boldsymbol{z}|\sigma_\rho)$ using the sample from the mixing distribution.

Simulated Tempering

Simulated tempering extends the idea of importance sampling to a long chain of variable length runs from different samplers. Suppose $\{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho^{(i)}, \boldsymbol{z})|i =$

$1, \ldots, s\}$ is a sequence of distributions that differ only by the prior standard deviation of ρ, and let $\sigma_\rho^{(1)} = \sigma_\rho$ and $\sigma_\rho^{(i)} < \sigma_\rho^{(i-1)}$ so as the index increases we move further from distribution of interest but improve mixing. At the end of each MCMC iteration a new standard deviation index j is proposed with probability $q_{i,j}$ where $q_{i,i+1} = q_{i,i-1} = 0.5$ for $1 < i < s$ and $q_{1,2} = q_{s,s-1} = 1.0$. The proposal is accepted with probability

$$\min\left\{1, \frac{c_j p(\boldsymbol{z}, \boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho^{(j)}) q_{j,i}}{c_i p(\boldsymbol{z}, \boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho^{(i)}) q_{i,j}}\right\}$$

where the constants $\{c_i | i = 1, \ldots, s\}$ are chosen so the chain divides its time equally between all samplers. Reject all samples for which $i \neq 1$ to obtain a sample from the distribution of interest, $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\boldsymbol{z}, \sigma_\rho)$. The problem with this method is the specification of the constants. The acceptance probability will be optimal if $c_i \propto p(\boldsymbol{z}|\sigma_\rho^{(i)})$ (see Section 3.2). In this case we could estimate the ratios, c_j/c_i, offline using importance sampling as described above.

MCMCMC

A variation on the above method which avoids the need for the normalising constant ratio is the Metropolis-Coupled MCMC (MCMCMC) method. Chains are run in parallel with stationary distributions $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\sigma_\rho^{(i)}, \boldsymbol{z})$ for $i = 1, 2, \ldots, s$. After each iteration a swap is proposed between chains i and j and accepted with probability

$$\min\left\{1, \frac{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m^{(j)}|\sigma_\rho^{(i)}, \boldsymbol{z})\, p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m^{(i)}|\sigma_\rho^{(j)}, \boldsymbol{z})}{p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m^{(i)}|\sigma_\rho^{(i)}, \boldsymbol{z})\, p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m^{(j)}|\sigma_\rho^{(j)}, \boldsymbol{z})}\right\}.$$

The normalising constants cancel in this ratio unlike the simulated tempering method. Output from the mixing chains is discarded and the original chain, $i = 1$, provides a sample from the distribution of interest.

8.6.4 Multidimensional Proposals

We may be able to improve mixing by using multidimensional proposals, i.e. by updating two or more parameters together. For example, we can propose m' from $q(m'|m)$, then given this value propose new values of other parameters, and accept

or reject the whole set. The proposal and posterior ratios may become quite complex. We develop two methods: one for parameters for which we can sample from the full conditionals, μ, ρ and ζ_i, $i = 1, 2, \ldots, n$, and one for parameters for which we cannot, a, b, c and d.

We present our algorithm for updating m together with a parameter for which it is possible to sample from the full conditional, using μ. The algorithm is as follows:

1. Propose m' from $q(m'|m)$.

2. Propose μ' from $p(\mu|\boldsymbol{\zeta}, \rho, a, b, c, d, m', \boldsymbol{z})$.

3. Accept (m', μ') with probability

$$\alpha = \min\left(1, \frac{q(m|m')p(\mu|\boldsymbol{\zeta}, \rho, a, b, c, d, m, \boldsymbol{z})}{q(m'|m)p(\mu'|\boldsymbol{\zeta}, \rho, a, b, c, d, m', \boldsymbol{z})} \frac{p(\boldsymbol{\zeta}, \mu', \rho, a, b, c, d, m'|\boldsymbol{z})}{p(\boldsymbol{\zeta}, \mu, \rho, a, b, c, d, m|\boldsymbol{z})}\right).$$

Similar algorithms can be constructed for (m', ρ') and (m', ζ_i') for $i = 1, 2, \ldots, n$. Furthermore, the algorithm can be extended to more than two parameters, for example propose m', then μ', then ρ', and accept or reject (m', μ', ρ').

Sampling from the full conditional should increase the posterior probability of the whole set, e.g. $p(\boldsymbol{\zeta}, \mu', \rho, a, b, c, d, m'|\boldsymbol{z})$ will probably be larger than $p(\boldsymbol{\zeta}, \mu, \rho, a, b, c, d, m'|\boldsymbol{z})$. The simulation index together with parameters for which the full conditional is not available, could be updated using this method, by replacing the full conditional with an arbitrary proposal distribution. However, in this case there is no reason to suppose the probability of the whole set will increase.

We present our algorithm for updating m together with parameters for which it is not possible to sample from the full conditional, using (c, d). The algorithm is as follows:

1. Propose m' from $q(m'|m)$.

2. Propose (c', d') from $q(c', d'|c, d)$.

3. Accept (c', d') *as part of the proposal* with probability

$$f(c', d'|c, d, m') = \min\left(1, \frac{q(c, d|c', d')}{q(c', d'|c, d)} \frac{p(\boldsymbol{\zeta}, \mu, \rho, a, b, c', d', m'|\boldsymbol{z})}{p(\boldsymbol{\zeta}, \mu, \rho, a, b, c, d, m'|\boldsymbol{z})}\right).$$

If (c', d') is rejected return to step 2.

4. Accept (m', c', d') with probability

$$\alpha = \min\left(1, \frac{q(m|m')q(c,d|c',d')f(c,d|c',d',m))}{q(m'|m)q(c',d'|c,d)f(c',d'|c,d,m')} \frac{p(\boldsymbol{\zeta}, \mu, \rho, a, b, c', d', m'|\boldsymbol{z})}{p(\boldsymbol{\zeta}, \mu, \rho, a, b, c, d, m|\boldsymbol{z})}\right).$$

This choice of acceptance probability preserves detailed balance. A similar algorithm can be constructed for $(m', a,' b')$. Furthermore, we can combine our two algorithms for parameters for which the full conditional is and is not available, e.g. $(m', a', b', \mu', \rho')$.

Ideally we would update m together with $\boldsymbol{\zeta}$, because $p(\boldsymbol{\zeta}|m, \boldsymbol{z})$ is very different for different m. However, the full conditional for $\boldsymbol{\zeta}$, $p(\boldsymbol{\zeta}|\boldsymbol{\phi}, m, \boldsymbol{z})$, is a truncated multivariate Normal distribution, for which no efficient sampling algorithms currently exist.

8.6.5 Integrating Out $\boldsymbol{\zeta}$

The main reason it is difficult to update the simulation index m is because $p(\boldsymbol{\zeta}|m, \boldsymbol{z})$ varies greatly with changes in m. Therefore we considered integrating $\boldsymbol{\zeta}$ out of the likelihood,

$$\begin{aligned} p(\boldsymbol{z}|\boldsymbol{\phi}, \boldsymbol{y}) &= \int p(\boldsymbol{z}, \boldsymbol{\zeta}|\boldsymbol{\phi}, \boldsymbol{y}) \, \mathrm{d}\boldsymbol{\zeta} \\ &= \int p(\boldsymbol{z}|\boldsymbol{\zeta}) p(\boldsymbol{\zeta}|\boldsymbol{\phi}, \boldsymbol{y}) \, \mathrm{d}\boldsymbol{\zeta} \\ &= \int \prod_{i=1}^{n} p(z_i|\zeta_i) p(\boldsymbol{\zeta}|\boldsymbol{\phi}, \boldsymbol{y}) \, \mathrm{d}\boldsymbol{\zeta} \\ &= \int \prod_{i=1}^{n} \mathbf{1}[z_i = \mathbf{1}_{\{-1,1\}}[\zeta_i > 0]] p(\boldsymbol{\zeta}|\boldsymbol{\phi}, \boldsymbol{y}) \, \mathrm{d}\boldsymbol{\zeta} \end{aligned}$$

where $\boldsymbol{\zeta}|\boldsymbol{\phi}, \boldsymbol{y} \sim \mathcal{MVN}(\mu\mathbf{1} + \rho\boldsymbol{D}\boldsymbol{y}, (\boldsymbol{I} - \boldsymbol{C})^{-1})$. The right-hand side of this equation is the multivariate Normal integral, which cannot be evaluated sufficiently quickly. Therefore, in general, it is not possible to integrate out $\boldsymbol{\zeta}$. However, in the independent case,

$$p(\boldsymbol{z}|\boldsymbol{\phi}, \boldsymbol{y}) = \prod_{i=1}^{n} \Phi\left((-1)^{\frac{1-z_i}{2}}(\mu + \rho(\boldsymbol{D}\boldsymbol{y})_i)\right)$$

which can be evaluated sufficiently quickly. We found this led to improved mixing of the MCMC algorithm.

8.7 The Hidden Intrinsic Autoregressive Model

In the Buscot examples of the HCAR model (see Section 8.5), we found the posterior density for the spatial dependence parameters, a and b, was often focused close to $a+b=0.5$. However, when $a+b=0.5$ the precision matrix $\boldsymbol{Q}$ is singular because the eigenvalue $1-2a-2b=0.0$. Quoting from Besag and Kooperberg (1995):

> [A] common disadvantage of conditional autoregressions is that appreciable correlations between the [variables] at neighbouring sites require parameter values extremely close to a particular boundary of the parameter space.

They turn this to their advantage by considering intrinsic limits of conditional autoregressions. In this section we first look at the density of a CAR process on 2 variables as the intrinsic limit is approached, then we introduce the intrinsic autoregressive (IAR) model by considering a CAR with a linear constraint. We consider the implications of using a hidden IAR model instead of the HCAR model in our framework, and look at some examples.

Motivation

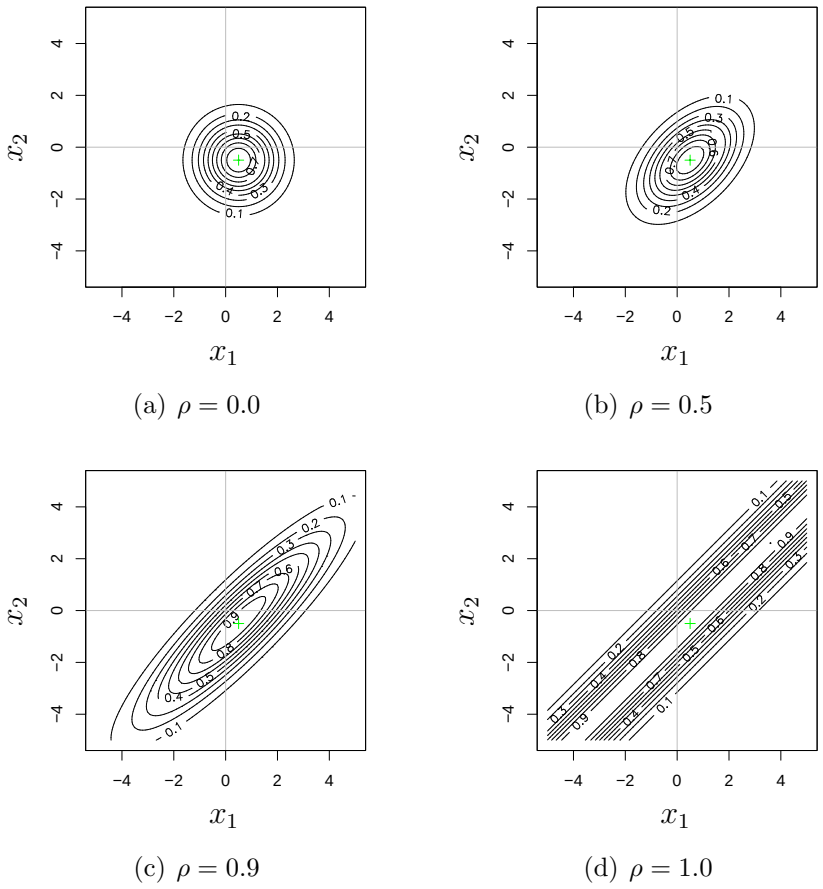
Consider a CAR on 2 variables x_1 and x_2 with

$$\begin{aligned} \mathrm{E}\left(x_i|x_{-i}\right) &= \mu_i + \rho(x_{-i}-\mu_{-i}) \quad \text{and} \\ \mathrm{Var}\left(x_i|x_{-i}\right) &= 1.0 \end{aligned}$$

for $i=1,2$. The joint density is bivariate Normal

$$p(x_1,x_2) \propto \exp\left(-\frac{1}{2}\left\{(x_1-\mu_1)^2 - 2\rho(x_1-\mu_1)(x_2-\mu_2) + (x_2-\mu_2)^2\right\}\right).$$

Figure 8.10 shows the effect of increasing ρ to the critical value of 1.0. Increasing ρ from 0.0 turns the circular contours into ellipses with main axes on a gradient



(a) $\rho = 0.0$ (b) $\rho = 0.5$

(c) $\rho = 0.9$ (d) $\rho = 1.0$

Figure 8.10: The density of the CAR model on two variables as the precision matrix becomes singular. The means are $\mu_1 = 0.5$ and $\mu_2 = -0.5$. Figure 8.10(d) shows the (improper) density for the limiting IAR model.

of 1.0. When $\rho = 1.0$ the contours become straight lines. We find that taking a slice perpendicular to these lines gives a Normal distribution. So we can think of the density as an infinite ridge of Normal cross-section, which clearly does not integrate to 1.0: it is an improper density.

The improper density can be written

$$p(x_1, x_2) \propto \exp\left(-\frac{1}{2}\left((x_1 - \mu_1) - (x_2 - \mu_2)\right)^2\right).$$

Let $d = (x_1 - x_2)$ which is proportional to the perpendicular distance from the main axis of the density, then $d \sim \mathcal{N}(\mu_1 - \mu_2, 1)$, i.e. the density is proper on this lower dimension. The joint density can be loosely regarded as the product of the proper density on $x_1 - x_2$ and an improper (diffuse) density on $x_1 + x_2$.

The Intrinsic Autoregressive Model

As we have seen in Figure 8.10(d), the IAR density is invariant to the addition of a constant to x_1 and x_2. Therefore to define the multivariate IAR we consider the density of a CAR with the constraint that the mean is fixed. Later we will define this density to hold for all values regardless of the value of the mean. We will see the resulting distribution is invariant to additions to the mean.

Suppose $\boldsymbol{\zeta}$ is a zero mean CAR with precision matrix $\boldsymbol{Q}$, so $\boldsymbol{\zeta} \sim \mathcal{MVN}(\mathbf{0}, \boldsymbol{Q}^{-1})$,. Let the eigenvalue and eigenvector matrices be

$$\begin{aligned}\boldsymbol{\Lambda} &= \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \quad \text{and} \\ \boldsymbol{V} &= (\boldsymbol{v}_1 | \boldsymbol{v}_2 | \ldots | \boldsymbol{v}_n)\end{aligned}$$

where $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V} = \boldsymbol{I}$. Assuming $\boldsymbol{Q}$ is block-circulant, one eigenvector will be $(1, 1, \ldots, 1)/\sqrt{n}$, without loss of generality let this be $\boldsymbol{v}_1$.

We want to calculate the density $p(\boldsymbol{\zeta} | \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{\zeta}/\sqrt{n} = w)$ for some constant w. Let $\boldsymbol{t} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{\zeta}$ then $\boldsymbol{t} \sim \mathcal{MVN}(\mathbf{0}, \boldsymbol{\Lambda}^{-1})$, i.e. $t_i \overset{iid}{\sim} \mathcal{N}(0, \lambda_i^{-1})$ for $i = 1, \ldots, n$. We have $\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{\zeta} = t_1 = \sqrt{n}w$. Then

$$\begin{aligned} p(\boldsymbol{t} | t_1 = \sqrt{n}w) &= \mathbf{1}[t_1 = \sqrt{n}w] \prod_{i=2}^{n} p(t_i), \\ \text{E}\left(\boldsymbol{t} | t_1 = \sqrt{n}w\right) &= (\sqrt{n}w, \mathbf{0}^{\mathrm{T}})^{\mathrm{T}} \quad \text{and} \\ \text{Prec}\left(\boldsymbol{t} | t_1 = \sqrt{n}w\right) &= \tilde{\boldsymbol{\Lambda}} \end{aligned}$$

where $\tilde{\boldsymbol{\Lambda}} = \text{diag}(0, \lambda_2, \ldots, \lambda_n)$. Converting back to $\boldsymbol{\zeta} = \boldsymbol{V}\boldsymbol{t}$,

$$\text{E}\left(\boldsymbol{\zeta} | \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{\zeta}/\sqrt{n} = w\right) = \boldsymbol{V}\begin{pmatrix}\sqrt{n}w \\ \mathbf{0}\end{pmatrix} = \begin{pmatrix} w \\ w \\ \vdots \\ w \end{pmatrix} \quad \text{and}$$

$$\text{Prec}\left(\boldsymbol{\zeta} | \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{\zeta}/\sqrt{n} = w\right) = \boldsymbol{V}\tilde{\boldsymbol{\Lambda}}\boldsymbol{V}^{\mathrm{T}},$$

so the density is

$$p(\boldsymbol{\zeta} | \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{\zeta}/\sqrt{n} = w) = (2\pi)^{-\frac{n-1}{2}} (\prod_{i=2}^{n} \lambda_i)^{\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{\zeta}^{\mathrm{T}}(\boldsymbol{V}\tilde{\boldsymbol{\Lambda}}\boldsymbol{V}^{\mathrm{T}})\boldsymbol{\zeta}\right).$$

Note that w does not appear in the density, it is implicit in the fact that the density is only nonzero where $\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{\zeta}/\sqrt{n} = w$. The only changes we needed to make to apply the constraint was to zero the appropriate eigenvalues and then renormalize.

The *intrinsic autoregressive (IAR) model* for $\boldsymbol{\zeta}$ is defined to have improper density

$$q(\boldsymbol{\zeta}) = (2\pi)^{-\frac{n-1}{2}}(|\boldsymbol{Q}|^{\star})^{\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{\zeta}^{\mathrm{T}}(\boldsymbol{V}\tilde{\boldsymbol{\Lambda}}\boldsymbol{V}^{\mathrm{T}})\boldsymbol{\zeta}\right)$$

for all $\boldsymbol{\zeta}$, where the generalised determinant, $|\boldsymbol{Q}|^{\star}$, is the product of the nonzero eigenvalues. This is the same density as for the CAR with the constraint that the mean is fixed, but now it is defined for all $\boldsymbol{\zeta}$. Compare this to the earlier example for two variables for which the joint density can loosely be defined as the product of a Normal density on $x_1 - x_2$ and an improper density on $x_1 + x_2$. The density is invariant to the addition of a constant to all variables. The density is proper on a lower dimension, namely that defined by a constant mean constraint.

Likelihood

The HIAR limit of our HCAR model has improper density

$$q(\boldsymbol{\zeta}|\rho, a, b, c, d, \boldsymbol{y}) = (2\pi)^{-\frac{n-1}{2}}(|\boldsymbol{Q}|^{\star})^{\frac{1}{2}} \times \exp\left(-\frac{1}{2}(\boldsymbol{\zeta} - \rho\boldsymbol{D}\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{C})(\boldsymbol{\zeta} - \rho\boldsymbol{D}\boldsymbol{y})\right)$$

where μ drops out of the model because of the invariance to the addition of a constant to all variables. The *mean* is $\rho\boldsymbol{D}\boldsymbol{y}^{(m)}$ and *precision matrix* is $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{C}$, although they are not strictly means and precisions because the density is improper we will continue to refer to them as such for convenience.

Posterior, Calibration and Calibrated Prediction

We now consider how Bayesian calibration and calibrated prediction is affected by replacing the HCAR model with the HIAR model. In Section 8.3 we calculated the marginal variance, $\mathrm{Var}\,(\zeta_i|\boldsymbol{\phi}, m)$, to be

$$\frac{1}{rc}\sum_{i'=0}^{r-1}\sum_{j'=0}^{c-1}(1 - 2a\cos(2\pi i'/r) - 2b\cos(2\pi j'/c))^{-1}.$$

As $a + b \to 0.5$ this variance will tend to ∞. Consequently we cannot produce calibrated predictions using the HIAR model. Furthermore, when using the HCAR model we should use the prior $p(a, b)$ to ensure the that the parameter values do not approach the boundary $a + b = 0.5$.

Calibration is possible using the HIAR model, because although the density is improper the posterior for $p(m|\boldsymbol{z})$ is proper provided that not all z_i take the same value. We will prove this for the two parameter case described earlier.

First we will show that it is sufficient to prove that $q(\boldsymbol{\zeta}|\boldsymbol{\phi}, m, \boldsymbol{z})$ is proper, where $\boldsymbol{\phi} = (\rho, a, b, c, d)$. The prior $p(\boldsymbol{\phi})p(m) = p(\rho)p(a, b)p(c, d)p(m)$ is proper but the prior and likelihood do not combine to define a proper joint probability model, $p(\boldsymbol{\zeta}, \boldsymbol{\phi}, m, \boldsymbol{z})$. However, using Bayesian algebra we can write the unnormalised posterior density function as

$$q(\boldsymbol{\zeta}, \boldsymbol{\phi}, m|\boldsymbol{z}) \propto p(\boldsymbol{z}|\boldsymbol{\zeta})q(\boldsymbol{\zeta}|\boldsymbol{\phi}, m)p(\boldsymbol{\phi})p(m)$$

The integral of the right-hand side of this equation is equal to the marginal for $\boldsymbol{z}$, $m(\boldsymbol{z})$. So proving that the posterior is proper is equivalent to proving that the marginal for $\boldsymbol{z}$ is finite,

$$\begin{aligned} m(z) &= \sum_{m=1}^{M} \iint p(\boldsymbol{z}|\boldsymbol{\zeta})q(\boldsymbol{\zeta}|\boldsymbol{\phi}, m)p(\boldsymbol{\phi})p(m)\, \mathrm{d}\boldsymbol{\zeta}\, \mathrm{d}\boldsymbol{\phi} \\ &= \sum_{m=1}^{M} \int \left\{ \int_B q(\boldsymbol{\zeta}|\boldsymbol{\phi}, m)\, \mathrm{d}\boldsymbol{\zeta} \right\} p(\boldsymbol{\phi})p(m)\, \mathrm{d}\boldsymbol{\phi}, \end{aligned}$$

where $B = \left\{ \boldsymbol{\zeta} \in \mathbb{R}^n | \mathbf{1}_{\{-1,1\}}[\zeta_i > 0] = z_i \text{ for } i = 1, \ldots, n \right\}$. If we can prove

$$\int_B q(\boldsymbol{\zeta}|\boldsymbol{\phi}, m)\, \mathrm{d}\boldsymbol{\zeta} \leq U, \tag{8.13}$$

for some finite constant U then $m(\boldsymbol{z}) \leq \sum_{m=1}^{M} \int U p(\boldsymbol{\phi})p(m)\, \mathrm{d}\boldsymbol{\phi} = U$. So we only need to prove Equation (8.13). It is logical that the propriety of the posterior should be connected to the values $\boldsymbol{z}$ takes, because we expect that if all z_i take the same value the posterior is not proper.

With only two pixels the precision matrix, $\boldsymbol{Q}$, is

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

and

$$q(\boldsymbol{\zeta}|\boldsymbol{\phi}, m) \propto \exp\left(-\frac{1}{2}\left((\zeta_1 - \rho(\boldsymbol{D}\boldsymbol{y}^{(m)})_1) - (\zeta_2 - \rho(\boldsymbol{D}\boldsymbol{y}^{(m)})_2\right)^2\right).$$

If z_1 and z_2 have the same value then the integral is not finite. Consider Figure 8.10(d) and suppose $z_1 = z_2 = 1$, then the integral corresponds to finding the volume under the ridge with the Normal cross-section in the top right quadrant, but this is clearly infinite. Now consider $z_1 = -1$ and $z_2 = 1$, we now need the volume of the ridge in the top-left quadrant. Although the region of integration is still infinite, the function decreases exponentially so the integral should be finite,

$$\int_B q(\boldsymbol{\zeta}|\boldsymbol{\phi}, m)\,\mathrm{d}\boldsymbol{\zeta} \propto \int_{\zeta_1 \leq 0}\int_{\zeta_2 > 0} \exp\left(-\frac{1}{2}((\zeta_1 - \rho(\boldsymbol{D}\boldsymbol{y}^{(m)})_1) - (\zeta_2 - \rho(\boldsymbol{D}\boldsymbol{y}^{(m)})_2))^2\right)\,\mathrm{d}\zeta_1\,\mathrm{d}\zeta_2.$$

Let $a_1 = -\rho(\boldsymbol{D}\boldsymbol{y}^{(m)})_1$, $a_2 = -\rho(\boldsymbol{D}\boldsymbol{y}^{(m)})_2$, $d_1 = (\zeta_1 + a_1) - (\zeta_2 + a_2)$ and $d_2 = (\zeta_1 + a_1) + (\zeta_2 + a_2)$, then the boundaries $\zeta_1 = 0.0$ and $\zeta_2 = 0.0$ correspond to $d_1 + d_2 = 2a_1$ and $d_2 - d_1 = 2a_2$ respectively,

$$\begin{aligned}
&\int_{-\infty}^{a_1-a_2}\int_{2a_2+d_1}^{2a_1-d_1} \exp\left(-\frac{1}{2}d_1^2\right)\,\mathrm{d}d_2\,\mathrm{d}d_1 \\
&\quad= \int_{-\infty}^{a_1-a_2} 2(a_1 - a_2 - d_1)\exp\left(-\frac{1}{2}d_1^2\right)\,\mathrm{d}d_1 \\
&\quad= -2\int_{-\infty}^{a_1-a_2} d_1 \exp\left(-\frac{1}{2}d_1^2\right)\,\mathrm{d}d_1 + 2(a_1 - a_2)\int_{-\infty}^{a_1-a_2}\exp\left(-\frac{1}{2}d_1^2\right)\,\mathrm{d}d_1 \\
&\quad= 2\exp\left(-\frac{1}{2}(a_1 - a_2)^2\right) + 2\sqrt{2\pi}(a_1 - a_2)\Phi(a_1 - a_2),
\end{aligned}$$

which is finite for all a_1 and a_2.

To summarise, if all the z_is take the same value then the posterior mean for $\boldsymbol{\zeta}$ is only constrained to be positive or negative so the posterior is improper. However, if just one z_i is different, the strong correlation between pixels requires that the posterior have a well defined mean.

Buscot Example

We now present an example of calibration for the Buscot dataset using the HIAR model. For comparison to the second HCAR model example, see Figures 8.7 and

8.8, we look at the effect of the standard deviation σ_ρ. We set $\nu_\rho = 0.0$ and $s = 1.0$, and consider three values for σ_ρ: 1/4, 1/16 and 1/64. The results of calibration for this example are shown in Figure 8.11.

The results of calibration for the HCAR and HIAR models become closer as σ_ρ decreases, but the posterior for the calibration inputs, $\boldsymbol{\theta}$, is always flatter for the HIAR model. It is interesting to note that the type of predictions possible using GLUE, i.e. the expected future prediction $\mathrm{E}(\boldsymbol{y}'|\boldsymbol{z})$, can be produced using the HIAR model, although this likelihood model is improper (see Section 4.2.3). However, the HIAR model is not a practical likelihood model because it does not allow calibrated predictions to be calculated, $p(z_i' = 1|\boldsymbol{z})$.

8.8 Heterogeneous Hidden Conditional Autoregressive Model

In much the same way that the fit of the BC model was improved by allowing the parameters to be heterogeneous, we believe that the fit of the HCAR model will improve by allowing some of the parameters to be heterogeneous. The difference between the HCAR and HHCAR models is that the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\rho}$ now vary spatially.

Likelihood

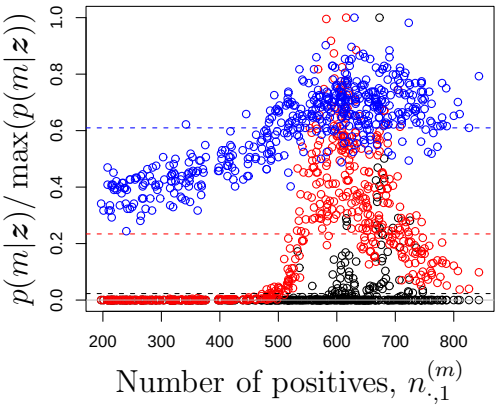
Following the approach used in Section 8.2, we define the *heterogeneous hidden conditional autoregressive (HHCAR) model* through the full conditionals. The conditional mean and variance are

$$\mathrm{E}(\zeta_i|\boldsymbol{\zeta}_{-i}, \boldsymbol{\mu}, \boldsymbol{\rho}, a, b, \boldsymbol{y}) = \mu_i + \rho_i y_i + \sum_{j=1}^{n} C_{ij}(\zeta_j - \mu_j - \rho_j y_j) \quad \text{and}$$

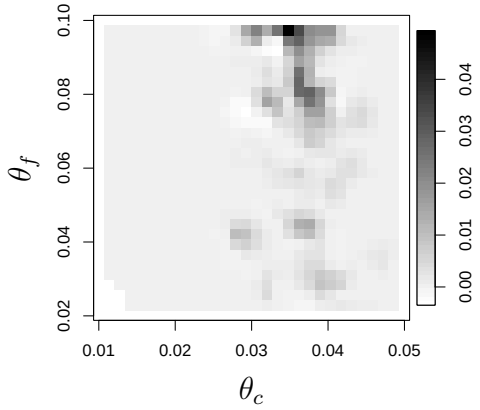
$$\mathrm{Var}(\zeta_i|\boldsymbol{\zeta}_{-i}, \boldsymbol{\mu}, \boldsymbol{\rho}, a, b, \boldsymbol{y}) = 1.0,$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\rho}$ are now vectors. We define the spatial interactions matrix as

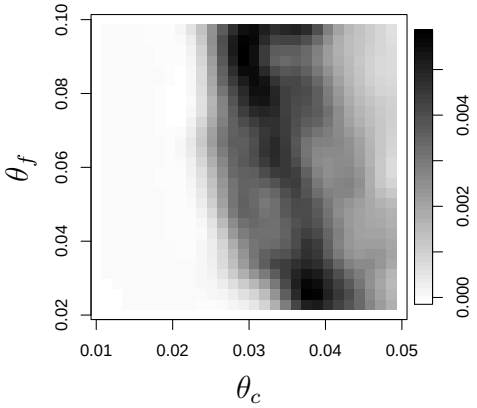
$$C_{ij} = \begin{cases} a & \text{if } i \overset{ew}{\sim} j \\ b & \text{if } i \overset{ns}{\sim} j \\ 0.0 & \text{otherwise,} \end{cases}$$



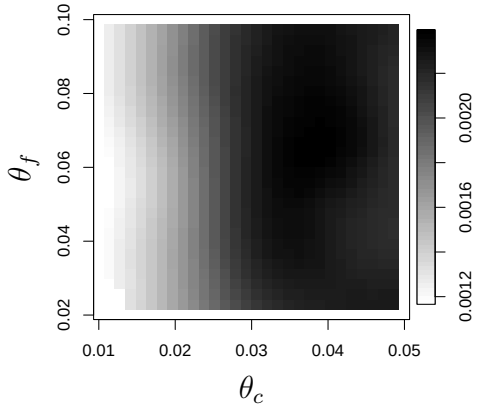
(a) Marginal posterior for m. The dashed lines show the means.



(b) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. $\sigma = 1/4$.



(c) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. $\sigma = 1/16$.



(d) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. $\sigma = 1/64$.

Figure 8.11: Three examples using the HIAR model and changing σ_ρ. The hyper-parameters are $\nu_\rho = 0.0$ and $s = 1.0$ in all cases; and $\sigma_\rho = 1/4$ (black), $s = 1/16$ (red) and $s = 1/64$ (blue).

where $|a| + |b| < 0.5$, and we assume no blur, $\boldsymbol{D} = \boldsymbol{I}$. The joint distribution is

$$\boldsymbol{\zeta}|\boldsymbol{\mu}, \boldsymbol{\rho}, a, b, m \sim \mathcal{MVN}\left(\boldsymbol{\mu} + \text{diag}(\boldsymbol{\rho})\boldsymbol{y}, (\boldsymbol{I} - \boldsymbol{C})^{-1}\right).$$

Priors

The priors for the spatial interaction parameters, a and b, and the simulation index, m, and the conditional distributions for $\boldsymbol{y}$ and $\boldsymbol{y}'$ given m, are as for the HCAR model, see Section 8.2.3. We define the priors for $\boldsymbol{\mu}$ and $\boldsymbol{\rho}$ through the full conditionals,

$$\mu_i|\boldsymbol{\mu}_{-i} \sim \mathcal{N}\left((1-\lambda_\mu)\nu_\mu + \frac{\lambda_\mu}{4}\sum_{j\in\delta i}\mu_j, \sigma_\mu^2\right) \quad \text{and}$$
$$\rho_i|\boldsymbol{\rho}_{-i} \sim \mathcal{N}\left((1-\lambda_\rho)\nu_\rho + \frac{\lambda_\rho}{4}\sum_{j\in\delta i}\rho_j, \sigma_\rho^2\right)$$

where $\nu_\mu, \nu_\rho \in \mathbb{R}$, $\sigma_\mu, \sigma_\rho \in \mathbb{R}_{\geq 0}$, $0.0 \leq \lambda_\mu, \lambda_\rho < 1.0$, and δi is the set of first-order neighbours of pixel i. Let $\boldsymbol{G}$ be a matrix with elements

$$G_{ij} = \begin{cases} \frac{1}{4} & \text{if } i \overset{ns}{\sim} j \text{ or } i \overset{ew}{\sim} j \\ 0 & \text{otherwise,} \end{cases}$$

then

$$\boldsymbol{\mu} \sim \mathcal{MVN}((1-\lambda_\mu)\nu_\mu\mathbf{1}, \sigma_\mu^2(\boldsymbol{I} - \lambda_\mu\boldsymbol{G})^{-1}) \quad \text{and}$$
$$\boldsymbol{\rho} \sim \mathcal{MVN}((1-\lambda_\rho)\nu_\rho\mathbf{1}, \sigma_\rho^2(\boldsymbol{I} - \lambda_\rho\boldsymbol{G})^{-1}).$$

Posterior, Calibration and Calibrated Prediction

The posterior distribution is

$$p(\boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\rho}, a, b, m|\boldsymbol{z}) \propto p(\boldsymbol{z}|\boldsymbol{\zeta})p(\boldsymbol{\zeta}|\boldsymbol{\mu}, \boldsymbol{\rho}, a, b, m)p(\boldsymbol{\mu})p(\boldsymbol{\rho})p(a, b)p(m).$$

We cannot evaluate this density directly because we do not know the normalising constant, but if we can generate a sample, $\{\boldsymbol{\zeta}^{(k)}, \boldsymbol{\mu}^{(k)}, \boldsymbol{\rho}^{(k)}, a^{(k)}, b^{(k)}, m^{(k)}|k = 1, \ldots, K\}$, from the posterior then we can perform calibration and make calibrated predictions.

MCMC Algorithm

Our MCMC algorithm for sampling from the posterior is similar to that used for the HCAR model, see Section 8.4. For the Metropolis-Hastings updates of the parameters a, b and m, we only need to modify the posterior ratios to take account of the new likelihood. For the Gibbs update of ζ_i the full conditional is now

$$\zeta_i|\boldsymbol{\zeta}_{-i}, \boldsymbol{\phi}, m, \boldsymbol{z} \sim \mathbf{1}[z_i = \mathbf{1}_{\{-1,1\}}[\zeta_i > 0]]\mathcal{N}(\mu_i + \rho_i y_i + \sum_{j=1}^{n} C_{ij}(\zeta_j - \mu_j - \rho_j y_j), 1.0).$$

The parameters $\boldsymbol{\mu}$ and $\boldsymbol{\rho}$ can be updated term by term using Gibbs updates. The full conditional for ρ_i, assuming $p(\boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\rho}_{-i}, a, b, m, \boldsymbol{z}) > 0.0$, is

$$\begin{aligned} p(\rho_i|\boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\rho}_{-i}, a, b, m, \boldsymbol{z}) &\propto p(\boldsymbol{\zeta}|\boldsymbol{\mu}, \rho_i, \boldsymbol{\rho}_{-i}, a, b, m)p(\rho_i|\boldsymbol{\rho}_{-i}) \\ &\propto \exp\left(-\frac{1}{2}(\boldsymbol{\zeta} - \boldsymbol{\mu} - \text{diag}(\boldsymbol{\rho})\boldsymbol{y})^{\text{T}}(\boldsymbol{I} - \boldsymbol{C})(\boldsymbol{\zeta} - \boldsymbol{\mu} - \text{diag}(\boldsymbol{\rho})\boldsymbol{y})\right) \\ &\quad \times \exp\left(-\frac{1}{2\sigma_\rho^2}(\rho_i - (1-\lambda_\rho)\nu_\rho - \lambda_\rho \bar{\boldsymbol{\rho}}_{\delta i})^2\right). \end{aligned}$$

Completing the square for ρ_i, we find

$$\begin{aligned} &\rho_i|\boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\rho}_{-i}, a, b, m, \boldsymbol{z} \sim \\ &\quad \mathcal{N}\left(\frac{1}{1+\sigma_\rho^2 y_i^2}\left(\sigma_\rho^2 y_i((\boldsymbol{I} - \boldsymbol{C})\boldsymbol{\zeta})_i - \sigma_\rho^2 y_i((\boldsymbol{I} - \boldsymbol{C})\boldsymbol{\mu})_i + \sigma_\rho^2 a y_i \sum_{j \in ew(i)} \rho_j y_j \right.\right. \\ &\qquad \left.\left. + \sigma_\rho^2 b y_i \sum_{k \in ns(i)} \rho_k y_k + (1-\lambda_\rho)\nu_\rho + \lambda_\rho \bar{\boldsymbol{\rho}}_{\delta i}\right), \frac{\sigma_\rho^2}{1+\sigma_\rho^2 y_i^2}\right). \end{aligned}$$

Similarly, provided $p(\boldsymbol{\zeta}, \boldsymbol{\mu}_{-i}, \boldsymbol{\rho}, a, b, m, \boldsymbol{z}) > 0.0$, the full conditional for μ_i is

$$\begin{aligned} &\mu_i|\boldsymbol{\zeta}, \boldsymbol{\mu}_{-i}, \boldsymbol{\rho}, a, b, m, \boldsymbol{z} \\ &\quad \sim \mathcal{N}\left(\frac{1}{1+\sigma_\mu^2}\left(\sigma_\mu^2((\boldsymbol{I} - \boldsymbol{C})\boldsymbol{\zeta})_i - \sigma_\mu^2((\boldsymbol{I} - \boldsymbol{C})\text{diag}(\boldsymbol{\rho})\boldsymbol{y})_i + \sigma_\mu^2 a \sum_{j \in ew(i)} \mu_j \right.\right. \\ &\qquad \left.\left. + \sigma_\mu^2 b \sum_{k \in ns(i)} \mu_k + (1-\lambda_\mu)\nu_\mu + \lambda_\mu \bar{\boldsymbol{\mu}}_{\delta i}\right), \frac{\sigma_\mu^2}{1+\sigma_\mu^2}\right). \end{aligned}$$

Buscot Example

Ideally, we would now present examples of calibration and calibrated prediction for the Buscot dataset using the HHCAR model, for comparison to the HCAR

model (see Section 8.5). However, for the MCMC algorithm described above mixing is poor in all cases of interest. The poor mixing can be attributed to the heterogeneous parameters $\boldsymbol{\mu}$, $\boldsymbol{\rho}$ and $\boldsymbol{\zeta}$. Mixing was better for the HBC model, see Section 7.6, because each parameter in the HHCAR model appears in rather a lot of the different factors in the probability model compared to the HBC model, and intuitively this means that the parameters will be *a posteriori* more correlated.

Although it is not possible to carry out calibration and calibrated prediction using our MCMC algorithm, by fixing the simulation index m we can demonstrate the impact of different prior assumptions about $\boldsymbol{\mu}$, $\boldsymbol{\rho}$, a and b on posterior inference.

For the following example we select the simulation with the fewest falses, $m = 110$. Then the calibrated prediction given $m = 110$ is

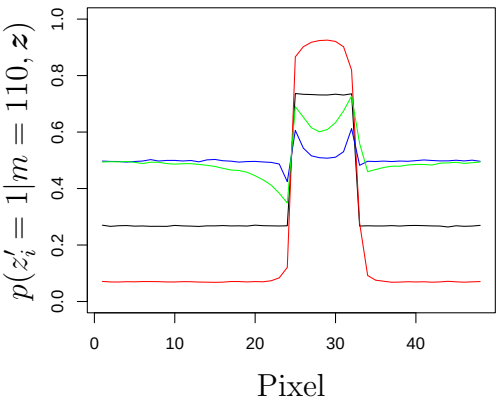
$$p(z'_i = 1|m = 110, \boldsymbol{z}) = \iiiint p(\zeta'_i > 0|\boldsymbol{\mu}, \boldsymbol{\rho}, a, b, m = 110)p(\boldsymbol{\mu}, \boldsymbol{\rho}, a, b|m = 110, \boldsymbol{z})\, \mathrm{d}\boldsymbol{\mu}\, \mathrm{d}\boldsymbol{\rho}\, \mathrm{d}a\, \mathrm{d}b$$

where $\zeta'_i|\boldsymbol{\mu}, \boldsymbol{\rho}, a, b, m = 110$ is Normal. We run our MCMC algorithm with $m = 110$ fixed to obtain a sample, $\{\boldsymbol{\mu}^{(k)}, \boldsymbol{\rho}^{(k)}, a^{(k)}, b^{(k)}|k = 1, \ldots, K\}$, from the posterior $p(\boldsymbol{\mu}, \boldsymbol{\rho}, a, b|m = 110, \boldsymbol{z})$, then

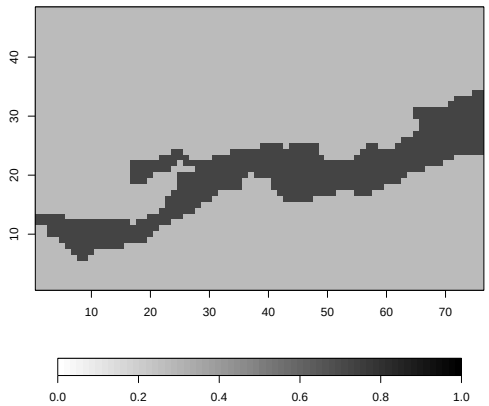
$$p(z'_i = 1|m = 110, \boldsymbol{z}) \approx \frac{1}{K}\sum_{k=1}^{K} p(\zeta'_i > 0|\boldsymbol{\mu}^{(k)}, \boldsymbol{\rho}^{(k)}, a^{(k)}, b^{(k)}, m = 110).$$

The results of calibrated prediction with $m = 110$ fixed using the HHCAR model are shown in Figure 8.12. The main feature of the results is that allowing $a = b \neq 0.0$ leads to calibrated predictions with more certainty at the boundary than away from it. We expect that these seemingly counterintuitive results arise because the large regions of true-positives and true-negatives, within the channel and on the floodplain away from the flood extent boundary, are accounted for by the values of a and b and not by the values of $\boldsymbol{\mu}$ and $\boldsymbol{\rho}$. Whereas, the behaviour close to the flood extent boundary is still accounted for by the values of $\boldsymbol{\mu}$ and $\boldsymbol{\rho}$. These results suggest that we should set $a = b = 0.0$ in the HHCAR model.

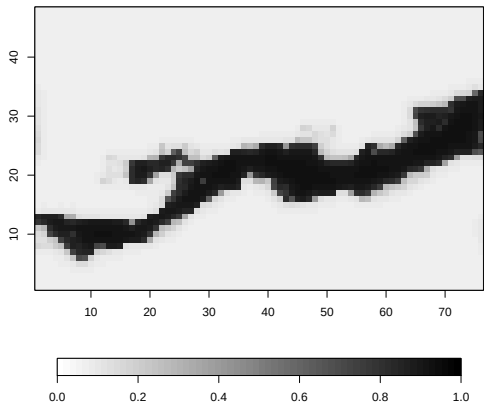
If we assume $a = b = 0.0$, the HHCAR model is essentially an alternative parameterisation of the HBC model. Furthermore, we can integrate out $\boldsymbol{\zeta}$ because



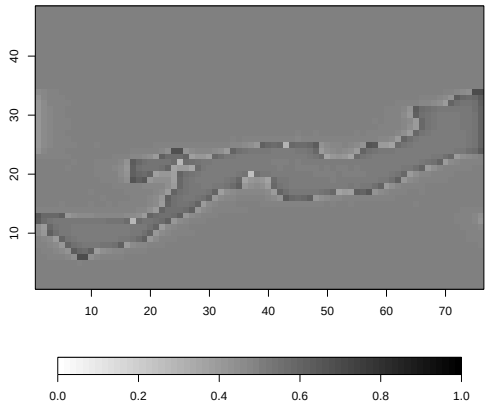
(a) Calibrated predictions with $m = 110$ for column 56.



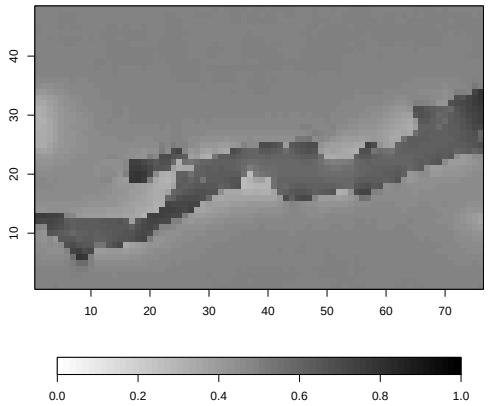
(b) $p(z_i' = 1|m = 110, \boldsymbol{z})$, $a = b = 0.0$ and $\lambda_\mu = \lambda_\rho = 0.0$.



(c) $p(z_i' = 1|m = 110, \boldsymbol{z})$, $a = b = 0.0$ and $\lambda_\mu = \lambda_\rho = 0.9$.



(d) $p(z_i' = 1|m = 110, \boldsymbol{z})$, $s = 10.0$ and $\lambda_\mu = \lambda_\rho = 0.0$.



(e) $p(z_i' = 1|m = 110, \boldsymbol{z})$, $s = 10.0$ and $\lambda_\mu = \lambda_\rho = 0.9$.

Figure 8.12: Four examples of calibrated prediction with $m = 110$ fixed using the HHCAR model. The hyperparameters are $\nu_\mu = \nu_\rho = 0.0$ and $\sigma_\mu = \sigma_\rho = 1.0$ in all cases; and $a = b = 0.0$ and $\lambda_\mu = \lambda_\rho = 0.0$ (black), $a = b = 0.0$ and $\lambda_\mu = \lambda_\rho = 0.9$ (red), $s = 10.0$ and $\lambda_\mu = \lambda_\rho = 0.0$ (blue), and $s = 10.0$ and $\lambda_\mu = \lambda_\rho = 0.9$ (green).

the multivariate Normal integral is now the product of Normal integrals, see Section 8.6.5. Finally, to improve mixing of the MCMC algorithm, instead of updating the vectors $\boldsymbol{\mu}$ and $\boldsymbol{\rho}$ term by term, we can update the whole vectors using Gibbs updates where the full conditionals are multivariate Normals with block-circulant precision matrices. We could also investigate treating m as a model index and using one of the various within-model sampling methods, one of which was applied to the HBC model in Section 7.7. We leave these ideas to be explored in future work, and in the next section consider a likelihood that uses continuous simulation values.

8.9 Continuous Hidden Conditional Autoregressive Model

In all the likelihood models considered so far the output of the flood inundation simulator, $\boldsymbol{y}$, is modelled as a binary image. In reality, the flood inundation simulator outputs the water depth in each pixel and we threshold these values to get the binary image.

Let $\boldsymbol{d}$ be an array of simulated water depths, where $d_i = 0.0$ if pixel i is dry and $d_i > 0.0$ if pixel i is wet. The magnitude of d_i is an indicator of *how wet* pixel i is – near the flood extent boundary we expect d_i to be close to 0.0, whereas in the channel we expect d_i to be larger. However, d_i does not indicate *how dry* pixel i is.

By combining the simulated water depths, $\boldsymbol{d}$, with the topography, $\boldsymbol{t}$, we can produce an indicator of how dry a pixel is. Let

$$y_i = -t_i + t_{i_w} + d_{i_w}$$

where i_w is the closest wet pixel to pixel i. If pixel i is wet, then $i_w = i$ and y_i is just the water depth d_i. If pixel i is dry, then $-y_i$ measures the height of the topography above the closest water surface.

Likelihood

We can define the *continuous hidden conditional autoregressive (CHCAR) model* through the full conditionals as we did the HCAR and HHCAR models. We omit a detailed derivation for the sake of brevity. The likelihood is

$$\boldsymbol{\zeta}|\mu, \rho, a, b, \boldsymbol{y} \sim \mathcal{MVN}(\mu\mathbf{1} + \rho\boldsymbol{y}, (\boldsymbol{I} - \boldsymbol{C})^{-1}),$$

where we have assumed no blur, $\boldsymbol{D} = \boldsymbol{I}$.

Priors

The priors for μ and ρ, the spatial interaction parameters, a and b, and the simulation index, m, and the conditional distributions for $\boldsymbol{y}$ and $\boldsymbol{y}'$ given m, are as for the HCAR model, see Section 8.2.3.

Posterior, Calibration and Calibrated Prediction

The posterior distribution is

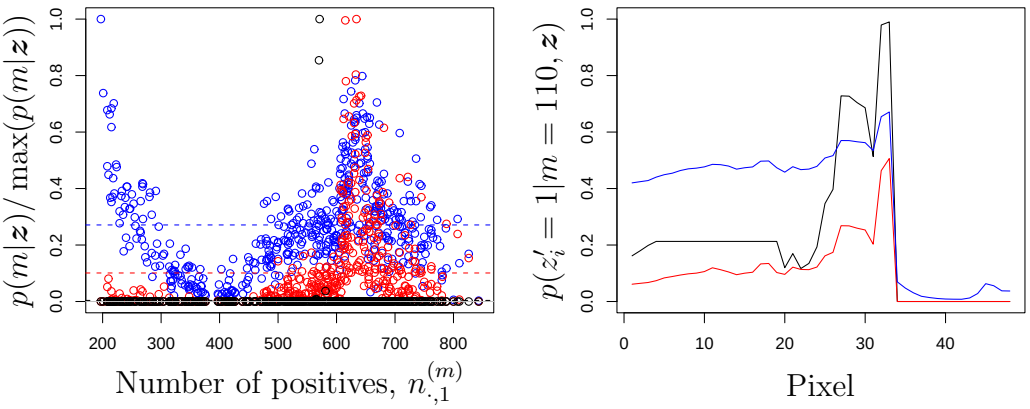
$$p(\boldsymbol{\zeta}, \mu, \rho, a, b, m|\boldsymbol{z}) \propto p(\boldsymbol{z}|\boldsymbol{\zeta})p(\boldsymbol{\zeta}|\mu, \rho, a, b, m)p(\mu)p(\rho)p(a, b)p(m).$$

We cannot evaluate this density directly because we do not know the normalising constant, but if we can generate a sample, $\{\boldsymbol{\zeta}^{(k)}, \mu^{(k)}, \rho^{(k)}, a^{(k)}, b^{(k)}, m^{(k)}|k = 1, \ldots, K\}$, from the posterior then we can perform calibration and make calibrated predictions.

Buscot Example

We now present an example of calibration and calibrated prediction using the CHCAR model. We look at the effects of spatial dependence in $\boldsymbol{\zeta}$, and $\sigma_\mu = \sigma_\rho$. We set $\nu_\mu = \nu_\rho = 0.0$, and consider three cases: spatially independent and $\sigma_\mu = \sigma_\rho = 1.0$, $s = 10.0$ and $\sigma_\mu = \sigma_\rho = 1.0$, and $s = 10.0$ and $\sigma_\mu = \sigma_\rho = 0.1$. The results of calibration and calibrated prediction are shown in Figures 8.13 and 8.14.

Introducing spatial dependence in $\boldsymbol{\zeta}$ leads to the marginal posteriors for the simulation index, m, and the calibration inputs, $\boldsymbol{\theta}$, being flatter (see Figures 8.13(a), 8.14(a) and 8.14(c)). Furthermore, the calibrated prediction $p(z_i' = 1|\boldsymbol{z})$ decreases



(a) Marginal posterior for m. The dashed lines show the means.

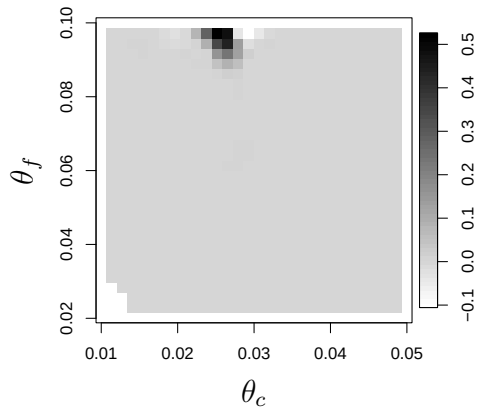
(b) Calibrated predictions for column 56.

Figure 8.13: Three examples using the CHCAR model. The hyperparameters are $\nu_\mu = \nu_\rho = 0.0$ in all cases; and spatially independent and $\sigma_\mu = \sigma_\rho = 1.0$ (black), $s = 10.0$ and $\sigma_\mu = \sigma_\rho = 1.0$ (red), and $s = 10.0$ and $\sigma_\mu = \sigma_\rho = 0.1$ (blue).

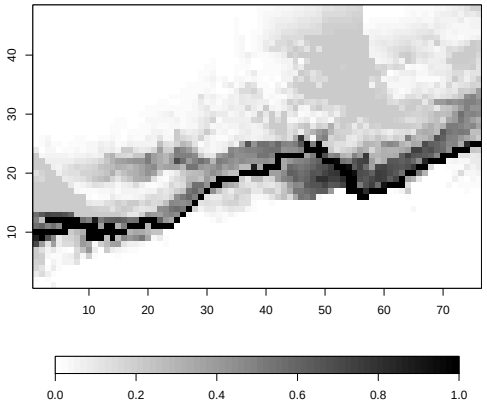
for all pixels, so the uncertainty in the calibrated predictions becomes larger in regions that we had predicted to be wet and smaller in regions we had predicted to be dry (see Figure 8.13(b)).

As $\sigma_\mu = \sigma_\rho$ decreases, the calibrated prediction $p(z_i' = 1|\boldsymbol{z})$ increases for all pixels. This increase is substantial for the low-lying floodplain. However, the posterior for the simulation index, m, is now bimodal. In addition to the expected peak corresponding to the simulation with the fewest falses, there is a peak corresponding to the driest simulations (see Figure 8.13(a)). This leads to an unusual posterior for the calibration inputs, $\boldsymbol{\theta}$ (see Figure 8.14(c)). We propose that this occurs for the following reason. The number of dry pixels in any given simulation is much greater than the number of wet pixels, and $|y_i|$ is much larger for dry pixels than for wet pixels. If the number of negatives in $\boldsymbol{y}^\star$ is greater than the number of negatives in $\boldsymbol{y}$, then $y_i^\star < y_i$ for almost all i because the water surface changes. Under certain hyperparameter settings, the increased probability of the observed negatives will outweigh the decreased probability of the observed positives.

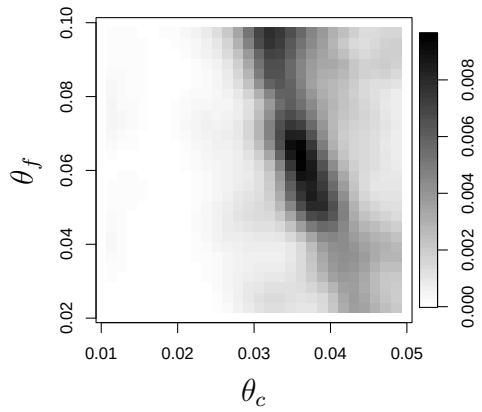
Future work should develop and test alternative measures of how dry a pixel is. It should also investigate further the effect of prior specifications on posterior inference, particularly the preference for dry simulations under certain hyperparameter



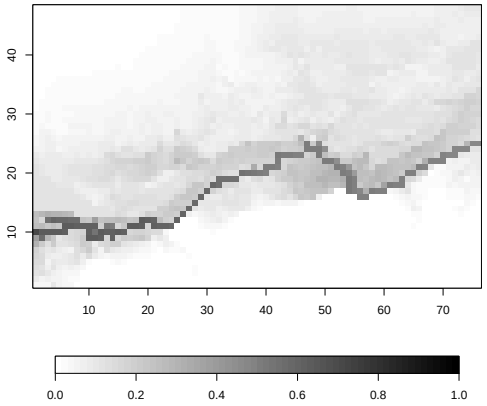
(a) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. Spatially independent and $\sigma_\mu = \sigma_\rho = 1.0$.



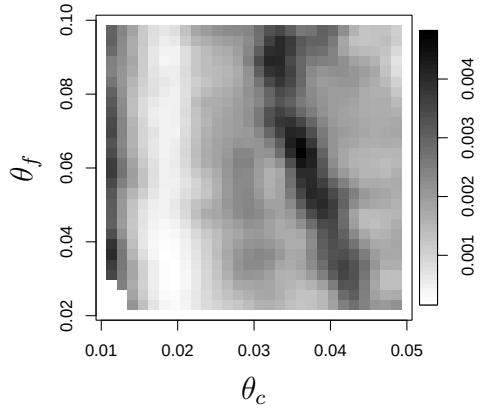
(b) $p(z'_i = 1|\boldsymbol{z})$, spatially independent and $\sigma_\mu = \sigma_\rho = 1.0$.



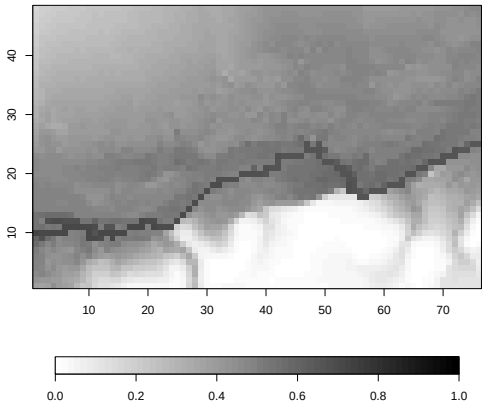
(c) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. $s = 10.0$ and $\sigma_\mu = \sigma_\rho = 1.0$.



(d) $p(z'_i = 1|\boldsymbol{z})$, $s = 10.0$ and $\sigma_\mu = \sigma_\rho = 1.0$.



(e) $p(\boldsymbol{\theta}|\boldsymbol{z})$ approximated from $p(\boldsymbol{\theta}^{(m)}|\boldsymbol{z})$ for $m = 1, \ldots, M$ using a thin-plate spline. $s = 10.0$ and $\sigma_\mu = \sigma_\rho = 0.1$.



(f) $p(z'_i = 1|\boldsymbol{z})$, $s = 10.0$ and $\sigma_\mu = \sigma_\rho = 0.1$.

Figure 8.14: Three examples using the CHCAR model. The hyperparameters are $\nu_\mu = \nu_\rho = 0.0$ in all cases; and spatially independent and $\sigma_\mu = \sigma_\rho = 1.0$, $s = 10.0$ and $\sigma_\mu = \sigma_\rho = 1.0$, and $s = 10.0$ and $\sigma_\mu = \sigma_\rho = 0.1$.

settings.

In this chapter we developed the HCAR model and a number of variants of this model: the HIAR, HHCAR and CHCAR models. For each likelihood model we demonstrated the effect of prior assumptions on posterior inference. Using the HCAR model it is not possible to obtain good results for calibration and calibrated prediction simultaneously. Using the HIAR model calibrated predictions are not possible at all. The HHCAR and CHCAR models showed promising results for calibration and calibrated prediction. The development of the HHCAR and CHCAR models would be worth considering in future work. Mixing of the MCMC algorithm was a problem for all the likelihood models and we described a number of methods for improving mixing. In the next chapter we present our conclusions and future work.

Chapter 9

Conclusions and Future Work

In this chapter we present conclusions for each of the research chapters in the thesis and then look more generally at future work.

In Chapter 5 we introduced a Bayesian framework for calibrating flood inundation simulators on an observation of flood extent, and then making calibrated predictions of a future event. By illustrating the framework using a directed acyclic graph (DAG) it is clear how the problem can be broken down into a number of smaller tasks. We identified likelihood specification as the most important task for calibration, and therefore assumed there was no variable input uncertainty and no observation error, so we could focus solely on this task. Using our Bayesian framework we were able to produce maps of the probability of flooding for a particular level flood event (e.g. the 1 in 100 year flood). A sensible extension of this approach would be to develop a framework which can produce maps of the probability of flooding in any given year from any level flood event.

We showed how to calibrate flood inundation simulators and make calibrated predictions of a future event using our Bayesian framework. Then we gave an example using the binary channel (BC) model for the likelihood and the Buscot dataset (introduced in Section 2.4). There is no validation data for the Buscot dataset, so we can only assess the performance of a particular likelihood model on what we believe to constitute sensible results for calibration and calibrated prediction. We recognise that this is not ideal, and suggest that validation on observations of future events should be a topic for future research. For this example we provided plots of $\mathrm{E}(\boldsymbol{y}'|\boldsymbol{z})$ for comparison to the maps of flood probability in

GLUE.

For calibration, if two simulations differ by only one pixel we expect the posterior for the simulations to be similar. For calibrated prediction, simulations are invariably correct within the channel and on the floodplain away from the flood boundary. Therefore we expect calibrated predictions to be relatively certain in these regions. The BC model does not represent spatial dependence, heterogeneity or blur. Consequently, using the BC model for the likelihood it was not possible to meet both these criteria simultaneously (i.e. using the same prior specification). This motivated the search for a more appropriate likelihood model.

In Chapter 6 we extended the Ising model (see Besag, 1974) to regression on a binary image. We reviewed methods for dealing with the intractable normalising constant and proposed novel applications of path sampling to paths between images and parameterisations. We also extended path sampling to sampling over areas. When these methods still proved too inefficient for practical use we proposed a number of approximations to path sampling and devised an experiment to test their adequacy. Unfortunately we did not identify a method which was both efficient and accurate enough for use in our Bayesian framework. Future work might consider more variants of the path sampling methodology, or investigate further the possibility of avoiding the calculation of the normalising constant using the auxiliary variable method from Møller *et al.* (2004). The latter method would require fast simulation methods to be developed for the Ising model with regression on a binary image.

In Chapter 7 the heterogeneous binary channel (HBC) model was developed, which extended the BC model to account for heterogeneity and spatial dependence. Using the HBC model for the likelihood it was possible to meet our criteria for calibration and calibrated prediction. However, the HBC model allows negative regression on the simulator output, so the probability that the observed value is different from the simulated value can be greater than 0.5. To investigate whether this was important we developed the positive heterogeneous binary channel (PHBC) model, which forced the regression to be always positive. For

the Buscot dataset there was no obvious advantage using the PHBC model, and we found that provided there was spatial dependence in the HBC model negative regression was rare. To test the necessity of forcing positive regression further we require observations of flood extent at various magnitudes.

By constructing a one-dimensional toy dataset we were able to show how the distribution of t false-positives in the simulator output affects the posterior. We found that a simulation with a block of false-positives away from the flood boundary had greater posterior density than a simulation with a block of false-positives on the flood boundary, because of spatial dependence. This is intuitively an undesirable property because blocks of false-positives near the flood boundary are to be expected, whereas blocks of false-positives away from the boundary are not. Another undesirable property of the HBC and PHBC models is that there are no explicit links between true-positives and false-negatives, or between true-negatives and false-positives.

For some prior specifications we found that the mixing of the MCMC algorithm was poor, so a realisation of the Markov chain is a poor estimate of a sample from the distribution of interest. We considered a within-model sampling (WMS) strategy for sampling from the posterior when mixing is poor, but this suffered high variance as very few sample points contributed to the estimate.

In Chapter 8 we extended the hidden conditional autoregressive (HCAR) model (see Weir and Pettitt, 1999) to regression on a binary image. By adopting toroidal boundary conditions we showed that the determinant calculation necessary for calibration, and the matrix inversion necessary for calibrated prediction, are computationally feasible through the use of block-circulant matrix results. We found that allowing more spatial dependence by a suitable prior choice leads to a flatter posterior for the simulation index in calibration, but calibrated predictions become more uncertain. As for the BC model, using the HCAR model as the likelihood it is not possible to meet our criteria for calibration and calibrated prediction simultaneously.

We found that, unless prevented from doing so by suitable prior choice, the posterior for the spatial interaction parameters will be focused close to a boundary of the parameter space at which the HCAR model does not hold. We showed that the limit of the HCAR model as these parameters approach this boundary is improper, and we call this the hidden intrinsic autoregressive (HIAR) model. Calibrated predictions are not possible using the HIAR model because it is improper, but we showed that calibration is possible provided the observed value is not the same for all pixels.

As for the HBC model, mixing of the MCMC algorithm was poor for some prior specifications. We described diagnostic tools for identifying poor mixing and investigating reasons for poor mixing. Then we presented methods for improving mixing: by linking simulations by a sequence of images; by conferring properties of a mixing distribution onto the distribution of interest; by updating two or more parameters together; and by integrating out the continuous process.

We explored two extensions of the HCAR model. In the first extension, the heterogeneous hidden conditional autoregressive (HHCAR) model, the mean and regression parameters were allowed to vary spatially. Unfortunately, for all priors of interest mixing of the MCMC algorithm was poor. The poor mixing is confined entirely to the update of the simulation index. Therefore we fixed the simulation index, and demonstrated the impact of different prior assumptions about the likelihood parameters on posterior inference. Modelling spatial dependence in the hidden continuous process and in the heterogeneous likelihood parameters, led to calibrated predictions with more certainty at the flood boundary than away from it. We concluded that for the HHCAR model we should assume spatial independence in the hidden continuous process, and noted that in this case the HHCAR model is simply an alternative parameterisation of the HBC model.

In the second extension, the continuous hidden conditional autoregressive (CHCAR) model, we use continuous valued simulator output. Flood inundation simulators output water depths, so we have a measure of *how wet* but not *how dry* a pixel is. It is for this reason that we had previously focused on binary models.

However, we show how we can form a measure of *how dry* a pixel is, by combining the simulator output with the topography. For some prior choices the posterior for the simulation index was bimodal. We proposed that this is because the number of dry pixels is much greater than the number of wet pixels, and *how dry* measurements are typically much larger in magnitude than *how wet* measurements.

We could extend the HCAR model by extending the underlying CAR, for example by allowing different variances for each pixel or by allowing the spatial interaction parameters to vary spatially.

We will now discuss ideas for future work and more general conclusions.

Friction parameters are not stationary between events of different magnitude, but without data available for a number of different magnitude events, there is no way to predict how the parameters change. The assumption of parameter stationarity remains a concern, and should be a topic for future research as data become more readily available. Another way of addressing this issue is to develop flood simulators with parameters that are more stationary between events of different magnitude. A single observation of flood extent would be of greater value to this type of model.

Methods for calibration using multiple sources of observed data should be developed. Ideally, we would have spatio-temporal data, e.g. a sequence of observations of flood extent over time. Failing this a selection of spatial and temporal data should be used. Observations of flood extent are very useful because this is the very quantity we want to predict and we do not need to work out how to translate the simulator inadequacy to the appropriate space, as we would if we had used a hydrograph. Future research may look at improving the satellite segmentation algorithm to account for the topography.

A serious practical limitation of all the likelihood models we have developed is that the associated calibrated predictions either have the deficiency of not tending to zero probability of flooding on high ground or they show no uncertainty around the flood outline. We may improve on these calibrated predictions by developing heterogeneous likelihood models which make use of topographic data.

It would be interesting to investigate alternatives to the pixel based models presented here. Shape deformation models which will focus solely on the flooded area may be of interest, but also any model that uses the water depth from the simulator output together with the topography.

BACCO is a comprehensive Bayesian method for handling uncertainty in computer codes, and future research might look at extending this method to flood inundation modelling, using some of the ideas developed within this thesis. The BACCO method is constructed around Gaussian processes so rather than use a thresholded Gaussian Markov random field or CAR, we might use a thresholded Gaussian process. It should be recognised that BACCO is not fully Bayesian because hyperparameters are fixed for posterior inference. The value of non-probabilistic methods should be assessed both for situations in which probabilistic methods are not possible and for those where they are possible but computationally intensive. All of these methods are concerned with making decisions, if the right decision is made using a method that violates the Bayesian paradigm but is far more efficient, then it has a value. The most important problem that must be addressed for non-probabilistic approaches is the way in which the results are represented. To represent arbitrary measures of skill as probabilities is misleading, but if the skill can be represented in a non-misleading way then the method becomes useful.

In conclusion, the main features of this thesis are the development of a Bayesian framework for calibrating flood inundation simulators, and then making calibrated predictions of a future event; together with a thorough investigation of a number of candidate likelihoods. We have shown that the non-probabilistic results obtained using GLUE can be obtained in a rigorous statistical way, and that with our method we can make probabilistic predictions of flooding in a future event, which is not possible with GLUE.

Bibliography

Living with the risk: The floods in Boscastle and North Cornwall 16th August 2004. Environment Agency, Manley House, Kestrel Way, Exeter, Devon, EX2 7LQ.

State of the environment 2005: A better place? Environment Agency, Waterside Drive, AztecWest, Almondsbury, Bristol BS32 4UD.

Anderson, M., Walling, D., and Bates, P. (1996). The General Context for Floodplain Process Research. In M. Anderson, D. Walling, and P. Bates (Eds.), *Floodplain Processes*, pp. 1–13. John Wiley & Sons, Chichester.

Aronica, G., Bates, P., and Horritt, M. (2002). Assessing the uncertainty in distributed model predictions using observed binary pattern information within GLUE. *Hydrological Processes*, **16**, 2001–2016.

Aronica, G., Hankin, B., and Beven, K. (1998). Uncertainty and equifinality in calibrating distributed roughness coefficients in a flood propagation model with limited data. *Advances in Water Resources*, **22**(4), 349–365.

Bates, P. (2004). Remote sensing and flood inundation modelling. *Hydrological Processes*, **18**(2593–2597).

Bates, P. (2005). Flood routing and inundation prediction. In M. Anderson (Ed.), *Encyclopedia of Hydrological Sciences*, pp. 1897–1922. John Wiley & Sons, Chichester.

Bates, P. and De Roo, A. (2000). A simple raster-based model for floodplain inundation. *Journal of Hydrology*, **236**.

Bates, P., Horritt, M., Aronica, G., and Beven, K. (2004). Bayesian updating of flood inundation likelihoods conditioned on flood extent data. *Hydrological Processes*, **18**(17), 3347–3370.

Bates, P., Horritt, M., and Hervouet, J.-M. (1998). Investigating two dimensional finite element predictions of floodplain inundation using fractal generated topography. *Hydrological Processes*, **12**, 1257–1277.

Bates, P., Horritt, M., Hunter, N., Mason, D., and Cobby, D. (2005). Numerical modelling of floodplain flow. In P. Bates, S. Lane, and R. Ferguson (Eds.), *Computational Fluid Dynamics: Applications in Environmental Hydraulics*, pp. 271–304. John Wiley & Sons, Chichester.

Bates, S., Cullen, A., and Raftery, A. (2003). Bayesian uncertainty assessment in multicompartment deterministic simulation models for environmental risk assessment. *Environmetrics*, **44**, 355–371.

Bayarri, M. and Berger, J. (2004). The interplay of Bayesian and frequentist analysis. *Statistical Science*, **19**, 58–80.

Ben-Haim, Y. (2001). *Information-Gap Decision Theory: Decisions Under Severe Uncertainty*. Academic Press, San Diego.

Berger, J. and Wolpert, R. (1988). *The Likelihood Principle* (Second ed.). Institute of Mathematical Statistics: Hayward, California.

Besag, J. (1972). Nearest-neighbour systems and the auto-logistic model for binary data. *Journal of the Royal Statistical Society B*, **34**, 75–83.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society B*, **36**, 192–236.

Besag, J., Green, P., Higdon, D., and Mengersen, K. (1995). Bayesian computation and stochastic systems. *Statistical Science*, **10**, 3–41.

Besag, J. and Kooperberg, C. (1995). On conditional and intrinsic autoregression. *Biometrika*, **82**(4), 733–746.

Best, N. and Green, P. (2005). Structure and uncertainty: Graphical models for understanding complex data. *Significance*, **2**(4), 177–181.

Beven, K. (2006). A manifesto for the equifinality thesis. *Journal of Hydrology*, **320**, 18–36.

Beven, K. and Binley, A. (1992). The future of distributed models: model calibration and uncertainty prediction. *Hydrological Processes*, **6**, 279–298.

Binley, A. and Beven, K. (1991). Physically-based modelling of catchment hydrology: a likelihood approach to reducing predictive uncertainty. In D. Farmer and M. Rycroft (Eds.), *Computer Modelling in the Environmental Sciences*, pp. 75–88. Clarendon Press, Oxford.

Blazkova, S., Beven, K., and Kulasova, A. (2002). On constraining TOPMODEL hydrograph simulations using partial saturated area information. *Hydrological Processes*, **16**(2), 441–458.

Box, G. (1976). Science and Statistics. *J. Am. Statist. Assoc.*, 791–799.

Campbell, K. (2002). Exploring bayesian model calibration: A guide to intuition. Technical Report LA-UR-02-7175, Los Alamos National Laboratory.

Craig, P., Goldstein, M., Rougier, J., and Seheult, A. (2001). Bayesian forecasting for complex systems using computer simulators. *Journal of the American Statistical Association*, **96**(454), 717–729.

Craig, P., Goldstein, M., Seheult, A., and Smith, J. (1996). Bayes Linear strategies for matching hydrocarbon reservoir history (with discussion). In J. Bernado, J. Berger, A. Dawid, and A. Smith (Eds.), *Bayesian Statistics 5*, pp. 69–95. Oxford University Press, Oxford.

Cressie, N. (1993). *Statistics for Spatial Data* (Revised ed.). Wiley, New York.

Cunge, J., Holly, F., and Verwey, A. (1980). *Practical Aspects of Computational River Hydraulics*. Pitman, London.

de Finetti, B. (1974). *Theory of Probability*. Wiley, New York.

Department for Environment, Food and Rural Affairs (2005). Making space for water: Taking forward a new Government strategy for flood and coastal erosion risk management in england. Technical report, Executive Summary.

Ebel, B. and Loague, K. (2006). Physics-based hydrologic-response simulation: Seeing through the fog of equifinality. *Hydrological Processes*, **20**, 2887–2900.

Fleming, G. (2002). Learning to live with rivers – the ICE's report to government. *Civil Engineering*, **150**, 15–21.

Foresight (2004). Future Flooding Executive Summary. Technical report, Office of Science and Technology.

Frenjel, D. (1986). Free-energy computation and first-order phase transition. In G. Ciccotti and W. Hoover (Eds.), *Molecular-Dynamics Simulation of Statistical-Mechanical Systems*, pp. 151–188. Amsterdam: North-Holland.

Friel, N. and Pettitt, A. (2004). Likelihood estimation and inference for the autologistic model. *Journal of Computational and Graphical Statistics*, **13**(1), 232–246.

Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2004). *Bayesian Data Analysis*. CRC Press.

Gelman, A. and Meng, X. (1998). Simulating normalising constants: from importance sampling to bridge sampling to path sampling. *Statistical Science*, **13**(2), 163–185.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721–741.

Gilks, W., Richardson, S., and Spiedelhalter, D. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London.

Gilks, W. and Roberts, G. (1996). Strategies for improving MCMC. In W. Gilks, S. Richardson, and D. Spiedelhalter (Eds.), *Markov Chain Monte Carlo in Practice*, pp. 89–114. Chapman & Hall, London.

Goldstein, M. (1995). Bayes linear methods I – Adjusting beliefs: concepts and properties. Technical Report 1995/1, Department of Mathematical Sciences, University of Durham.

Goldstein, M. and Rougier, J. (2004). Probabilistic formulations for transferring inferences from mathematical models to physical systems. *SIAM Journal on Scientific Computing*, **26**(2), 467–487.

Green, P. (2001). A primer on Markov chain Monte Carlo. In O. Barndorff-Nielsen, D. Cox, and C. Kluppelberg (Eds.), *Complex Stochastic Systems*, pp. 1–62. Chapman and Hall, London.

Green, P. (2003). Trans-dimensional Markov chain Monte Carlo. In P. Green, N. Hjort, and S. Richardson (Eds.), *Highly Structured Stochastic Systems*, pp. 179–198. Oxford University Press.

Grimmett, G. and Stirzaker, D. (2002). *Probability and Random Processes* (Third ed.). Oxford University Press.

Gupta, H., Sorooshian, S., and Yapo, P. (1998). Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research*, **34**(4), 751–763.

Halcrow Group Ltd (2001). National Appraisal of Assets at Risk from Flooding and Coastal Erosion, including the potential impact of climate change. Technical report, Department for Environment, Food and Rural Affairs.

Hall, J. (2003). Handling uncertainty in the hydroinformatic process. *Journal of Hydroinformatics*, **5**(4), 215–232.

Hall, J. and Anderson, M. (2002). Handling uncertainty in extreme or unrepeatable hydrological processes – the need for an alternative paradigm. *Hydrological Processes*, **16**(9), 1867–1870.

Hankin, B., Hardy, R., Kettle, H., and Beven, K. (2001). Using CFD in a GLUE framework to model the flow and dispersion characteristics of a natural fluvial dead zone. *Earth Surface Processes and Landforms*, **26**(6), 667–687.

Hankin, R. (2005). Introducing BACCO, an R bundle for Bayesian analysis of computer code output. *Journal of Statistical Software*, **14**(16).

Harman, J., Bramley, M., and Funnell, M. (2002). Sustainable flood defence in England and Wales. *Civil Engineering*, **150**, 3–9.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

Hervouet, J.-M. and Van Haren, L. (1996). Recent advances in numerical methods for fluid flows. In M. Anderson, D. Walling, and P. Bates (Eds.), *Floodplain Processes*, pp. 183–214. John Wiley & Sons, Chichester.

Horritt, M. (1999). A statistical active contour model for SAR image segmentation. *Image and Vision Computing*, **17**(3), 213–224.

Horritt, M. and Bates, P. (2001). Predicting floodplain inundation: Raster-based modelling versus the finite element approach. *Hydrological Processes*, **15**, 825–842.

Horritt, M. and Bates, P. (2002). Evaluation of 1-D and 2-D numerical models for predicting river flood inundation. *Journal of Hydrology*, **268**, 87–99.

Horritt, M., Mason, D., and Luckman, A. (2001). Flood boundary delineation from Synthetic Aperture Radar imagery using a statistical active contour model. *International Journal of Remote Sensing*, **22**(13), 2489–2507.

Hunter, N., Bates, P., Horritt, M., De Roo, A., and Werner, M. (2005). Utility of different data types for calibrating flood inundation models within a GLUE framework. *Hydrology and Earth System Sciences*, **9**(4), 412–430.

Hunter, N., Horritt, M., Bates, P., Wilson, M., and Werner, M. (2005). An adaptive time step solution for raster-based storage cell modelling of floodplain inundation. *Advances in Water Resources*, **28**(9), 975–991.

Hurn, M., Husby, O., and Rue, H. (2003). A tutorial on image analysis. In J. Møller (Ed.), *Spatial Statistics and Computational Methods*, Number 173 in Lecture Notes in Statistics, pp. 87–141. Springer, New York.

Institution of Civil Engineers (2001). *Learning to live with rivers: Final report of the ICE's presidential commission to review the technical aspects of flood risk management in England and Wales*. ICE, London.

International Federation of Red Cross and Red Crescent Societies (2001). *World Disasters Report 2001: Focus on Recovery*. Oxford University Press.

Ising, E. (1925). Beitrag zur Theorie des Ferromagnetismus. *Zeitschr. f. Physik*, **31**, 253–258. [German].

Jolliffe, I. and Stephenson, D. (2003). *Forecast Verification: A Practitioner's Guide in Atmospheric Science*. John Wiley & Sons, Chichester.

Jordan, M. (Ed.) (1999). *Learning in Graphical Models*. MIT Press, Cambridge, MA, USA.

Julien, P. (2002). *River Mechanics*. Cambridge University Press.

Kac, M. and Ulam, S. (1968). *Mathematics and Logic*. Dover Publications.

Kavetski, D., Franks, S., and Kuczera, G. (2002). Confronting input uncertainty in environmental modelling. In Q. Duan, H. Gupta, S. Sorooshian, A. Rousseau, and R. Turcotte (Eds.), *Calibration of Watershed Models*, Volume 6 of *AGU Water Science and Applications Series*, pp. 49–68. AGU.

Kennedy, M. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B*, **63**(3), 425–464.

Kennedy, M., O'Hagan, A., and Higgins, N. (2002). Bayesian analysis of computer code outputs. In C. Anderson, V. Barnett, P. Chatwin, and A. El-Shaarawi (Eds.), *Quantitative Methods for Current Environmental Issues*, pp. 227–243. Springer-Verlag.

Knight, D. and Shiono, K. (1996). River channel and floodplain hydraulics. In M. Anderson, D. Walling, and P. Bates (Eds.), *Floodplain Processes*, pp. 139–182. John Wiley & Sons, Chichester.

Krzysztofowicz, R. (2002). Bayesian system for probabilistic river stage forecasting. *Journal of Hydrology*, **268**, 16–40.

Lane, S. (1998). Hydraulic modelling in hydrology and geomorphology: A review of high resolution approaches. *Hydrological Processes*, **12**, 1131–1150.

Lane, S., Bradbrook, K., Richards, K., Biron, P., and Roy, A. (1999). The application of computational fluid dynamics to natural river channels: Three-dimensional versus two-dimensional approaches. *Geomorphology*, **29**, 1–20.

Lauritzen, S. (1996). *Graphical Models*. Clarendon Press, Oxford.

Lindley, D. (1971). *Bayesian Statistics: A Review*. CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia.

Mantovan, P. and Todini, E. Hydrological forecasting: Incoherence of the GLUE methodology. In press.

Marsaglia, G., Tsang, W., and Wang, J. (2004). Fast generation of discrete random variables. *Journal of Statistical Software*, **11**.

Mason, D., Cobby, D., Horritt, M., and Bates, P. (2003). Floodplain friction parameterization in two-dimensional river flood models using vegetation heights derived from airborne scanning laser altimetry. *Hydrological Processes*, **17**, 1711–1732.

Meng, X. and Wong, W. (1996). Simulating ratios of normalising constants via a simple identity: a theoretical exploration. *Statist. Sinica*, **6**, 831–860.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equations for state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1091.

Ministry of Agriculture, Fisheries and Food (2000). Flood and Coastal Defense Project Appraisal Guidance. Technical report, Flood and Coastal Defence with Emergencies Division.

Møller, J. (2003). *Spatial Statistics and Computational Methods*. Springer, New York.

Møller, J., Pettitt, A., Berthelsen, K., and Reeves, R. (2004). An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. Research Report R-2004-02, Aalborg University, Department of Mathematical Sciences, Aalborg University.

Moran, P. (1973). A Gaussian Markovian process on a square lattice. *Journal of Applied Probability*, **10**, 54–62.

Murphy, K. (2001). An introduction to graphical models. Web based tutorial. http://www.cs.ubc.ca/~murphyk/Papers/intro_gm.pdf.

Oakley, J. and O'Hagan, A. (2002). Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, **89**, 769–784.

Oakley, J. and O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society, Series B*, **66**, 751–769.

O'Hagan, A. (1994). *Bayesian Inference* (First ed.), Volume 2B of *Kendall's Advanced Theory of Statistics*. Edward Arnold, London.

O'Hagan, A. (2004a). Bayesian analysis of computer code outputs: A tutorial. Technical report, University of Sheffield, UK.

O'Hagan, A. (2004b). Dicing with the unknown. *Significance*, **1**(3), 132–133.

Pappenberger, F., Beven, K., Horritt, M., and Blazkova, S. (2005). Uncertainty in the calibration of effective roughness parameters in HEC-RAS using inundation and downstream level observations. *Journal of Hydrology*, **302**(1–4), 46–69.

Pappenberger, F., Matgen, P., and Beven, K. The influence of rating curve and structural uncertainty on flood inundation predictions. Accepted for publication in *Advances in Water Resources*.

Paterson, A. (1997). *A first course in fluid dynamics*. Cambridge University Press.

Pender, G. (2006). Briefing: Introducing flood risk management research consortium. *Proceedings of the Institution of Civil Engineers, Water Management*, **159**, 3–8.

Peskun, P. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, **60**, 607–612.

Pettitt, A., Friel, N., and Reeves, R. (2003). Efficient calculation of the normalising constant of the autologistic and related models on the cylinder and lattice. *J. R. Statist. Soc. B*, **65**(1), 235–246.

Pettitt, A., Weir, I., and Hart, G. (2002). A conditional autoregressive Gaussian process for irregularly spaced multivariate data with application to modelling large sets of binary data. *Statistics and Computing*, **12**, 353–367.

Propp, J. and Wilson, D. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, **9**(1–2), 223–252.

Rice, J. (1995). *Mathematical statistics and data analysis* (second ed.). Duxbury Press, Belmont.

Robert, C. and Casella, G. (2004). *Monte Carlo Statistical Methods* (second ed.). Springer-Verlag.

Rodi, W. (1980). *Turbulence models and their application in hydraulics: A state of the art review*. International Association for Hydraulic Research, Delft.

Romanowicz, R. and Beven, K. (2003). Estimation of flood inundation probabilities as conditioned on event inundation maps. *Water Resources Research*, **39**(3), SWC 4–1. doi:10.1029/2001WR001056.

Romanowicz, R., Beven, K., and Tawn, J. (1996). Bayesian calibration of flood inundation models. In M. Anderson, D. Walling, and P. Bates (Eds.), *Floodplain Processes*, pp. 333–360. John Wiley & Sons, Chichester.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*, Volume 104 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London.

Sacks, J., Welch, W., Mitchell, T., and Wynn, H. (1989). Design and analysis of computer experiments. *Statist. Sci.*, **4**, 409–435.

Saltelli, A., Chan, K., and Scott, E. (2000). *Sensitivity Analysis*. John Wiley & Sons, Chichester.

Sellin, R. and Willets, B. (1996). Three-dimensional structures, memory and energy dissipation in meandering compound channel flow. In M. Anderson, D. Walling, and P. Bates (Eds.), *Floodplain Processes*, pp. 255–298. John Wiley & Sons, Chichester.

Shachter, R. (1998). Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In G. Cooper and S. Moral (Eds.), *UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, July 24–26, 1998, University of Wisconsin Business School, Madison, Wisconsin, USA*, pp. 480–487. Morgan Kaufmann.

Spiegelhalter, D., Best, N., Carlin, B., and van der Linde, A. (2002). Bayesian measures of model complexity and fit. *J. R. Statist. Soc. B*, **64**, 583–639.

Stoesser, T., Wilson, C., Bates, P., and Dittrich, A. (2003). Application of a 3D model to a river with vegetated floodplains. *Journal of Hydroinformatics*, **5**, 99–112.

Swartz, T., Haitovsky, Y., Vexler, A., and Yang, T. (2004). Bayesian identifiability and misclassification in multinomial data. *The Canadian Journal of Statistics*, **32**, 1–18.

The Oxford English Dictionary (2nd ed.) *"flood, n."* (1989). OED Online. Oxford University Press. 1st June 2006. http://dictionary.oed.com/cgi/entry/50086562.

Thomas, T. and Williams, J. (1995). Large-Eddy Simulation of turbulent flow in an asymmetric compound open channel. *Journal of Hydraulic Research*, **33**(1), 27–41.

Wagener, T. and Gupta, H. (2005). Model identification for hydrological forecasting under uncertainty. *Stoch. Environ. Res. Risk Assess.*, **19**, 378–387.

Walker, A. (1977). An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, **3**(3), 253–256.

Weir, I. and Pettitt, A. (1999). Spatial modelling for binary data using a hidden conditional autoregressive Gaussian process: a multivariate extension of the probit model. *Statistics and Computing*, **9**, 77–86.

Werner, M. (2001). Impact of grid size in GIS based flood extent mapping using a 1D flow model. *Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere*, **26**(7), 517–522.

Werner, M. and Lambert, M. Evaluation of modelling approaches for river reach scale inundation modelling. Accepted for publication in the *Journal of Hydraulic Research*.

Wilson, M. and Atkinson, P. (2005). Prediction uncertainty in elevation and its effect on flood inundation modelling. In P. Atkinson, G. Foody, S. Darby, and F. Wu (Eds.), *GeoDynamics*, pp. 185–202. CRC Press.

Yapo, P., Gupta, H., and Sorooshian, S. (1998). Multi-objective global optimization for hydrologic models. *Journal of Hydrology*, **204**, 83–97.