RANDOMIZED POLYA TREES: BAYESIAN NONPARAMETRICS FOR MULTIVARIATE DATA ANALYSIS

by

Susan Mary Paddock

Institute of Statistics and Decision Sciences
Duke University

Date: _____

Approved:

_____
Mike West, Supervisor

_____
Michael Lavine

_____
James O. Berger

_____
Fabrizio Ruggeri

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Institute of Statistics and Decision Sciences
in the Graduate School of
Duke University

1999

ABSTRACT

(Statistics)

RANDOMIZED POLYA TREES: BAYESIAN NONPARAMETRICS FOR MULTIVARIATE DATA ANALYSIS

by

Susan Mary Paddock

Institute of Statistics and Decision Sciences
Duke University

Date: _____

Approved:

_____
Mike West, Supervisor

_____
Michael Lavine

_____
James O. Berger

_____
Fabrizio Ruggeri

An abstract of a dissertation submitted in partial
fulfillment of the requirements for the degree
of Doctor of Philosophy in the
Institute of Statistics and Decision Sciences in the Graduate School of
Duke University

1999

Abstract

The nonparametric approach to statistical modeling is appealing because it readily accommodates non-standard relationships in data. This dissertation is a first step to understanding the usefulness of Polya tree priors for modeling in multidimensional Euclidean spaces. In particular, the Polya tree prior is applied to a multidimensional Euclidean space. Using binary perpendicular recursive partitioning of a hypercube in $\Re^K$, it is shown that marginal distributions of Polya tree priors are Polya trees, and a conditional predictive distribution simulation scheme for exploring conditonal relationships among K variables in a K-dimensional space is developed. Its usefulness for missing data imputation is also discussed. To address partition dependence – a critical limitation of Polya trees – the Randomized Polya tree is defined and developed. This new framework inherits the structure of Polya trees but induces smoothing of discontinuities in predictive distributions. Theoretical aspects of the new framework are developed, followed by discussion of methodological and computational issues arising in implementation. Analyses of two data sets highlight aspects of inference with randomized trees. Future directions for research are discussed.

Acknowledgements

I would like to thank my advisor, Mike West, for the guidance he has given to me over the past four years. I would also like to thank Fabrizio Ruggeri and Jim Berger for helpful discussions of my dissertation research. Thanks to Michael Lavine, for his advice and insights on this research and for providing computer code which served as a helpful guide at the start of my dissertation research. I also had helpful conversations with Yiannis Vlassopoulos of the Duke Math Department. Special thanks goes to the ISDS graduate students, who have struggled with me through the courses, research, and career decision-making that are part of the graduate student experience.

Above all, I want to thank those who were with me through the good and bad times: my parents; my sister, Sandra; and friends Amado Sleiman, Kristina Przybilla, Courtney Johnson, Dalene Stangl, Cheryl McGhee, and Maria de Iorio.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Motivation

Bayesian nonparametric and semiparametric approaches to flexible modeling in $\Re^K$ have long been active areas of research. Discussion of all Bayesian methods for exploration of dependence structure, density estimation, and prediction in multidimensional spaces would go far beyond the scope of this dissertation. A small sampling of the vast collection of such methods includes mixture models (see West (1992) for an overview) and mixtures of Dirichlet processes (Ferguson (1973), Antoniak (1974), Lo (1984), Kuo (1986), Escobar and West (1995), Müller *et al.* (1997), and many others).

In a nonparametric framework for modeling random samples, observations X_1, X_2, $\ldots$, X_n belong to a set Ω, and for $j \in 1, \ldots, n$, $X_j \sim P$, where the distribution P is unknown and comes from a family of distribution functions $\mathcal{P}$; the Bayesian approach to this problem requires P to be regarded as an unknown parameter and a prior distribution placed upon it. The nonparametric approach is appealing because it, in principle, provides opportunity for accommodating non-standard relationships in data relative to more constrained parametric models. Polya tree priors (Lavine

(1992), Mauldin *et al.* (1992)) can be used to model an unknown distribution P by recursively partitioning the sample space. While some authors anticipate the usefulness of Polya tree priors in $\Re^K$ for applied statistical modeling — *e.g.*, Fienberg *et al.* (1996) for nonparametric function estimation in a data disclosure context and Walker and Mallick (1997) for a multivariate hierarchical random effects models — the framework for modeling with Polya trees in $\Re^K$ has yet to be applied to multivariate problems for $K > 1$. This dissertation is a first step to fully understanding the Polya tree priors for applications in $\Re^K$.

Partition dependence is a major drawback of the Polya tree prior (Ferguson (1974), Lavine (1992)). Basically, partition dependence is when the partition specified for the model strongly influences resulting inferences. The problem of partition dependence is not restricted to Polya trees; any statistical method relying upon a partition of a Euclidean space can meet this criticism. Hartigan (1996) mentions concerns about lack of 'amalgamation' of observations in neighboring bins of a Bayesian histogram build from a fixed set of candidate partition points as an issue, as do Wolpert and Lavine (1995) in their work with Markov random fields for univariate density estimation. Our initial work with Polya trees confirmed that partition dependence is a problem which must be addressed when modeling with Polya trees. In this dissertation, the 'randomized Polya tree' method is developed for smoothing discontinuities of the posterior predictive density that are induced by partition dependence.

The remainder of this chapter is devoted to a review of theoretical work pertinent to the development of Polya tree prior, from Ferguson's 1974 unification of several classes of random distribution functions under the tailfree process framework to a review of the Polya tree prior (Lavine (1992), Mauldin *et al.* (1992), Lavine (1994)).

1.2 Previous Work

1.2.1 Tailfree Processes

In an important early review of Bayesian nonparametrics, Ferguson (1974) describes tailfree processes, which were first defined by Freedman (1963) and Fabius (1964). He unifies several classes of random distribution functions previously described in the literature, including Dirichlet processes (DP) (Ferguson, 1973), the continuous singular distributions of Dubins and Freedman (1967), and the absolutely continuous (with probability 1) distributions of Kraft (1964) and Métivier (1971), by showing them all to be tailfree processes. The tailfree process described by Ferguson is essentially the Polya tree prior which is later defined and investigated by Mauldin *et al.* (1992), Lavine (1992), and Mauldin and Williams (1990).

Before discussing these papers in detail, tailfree processes are explained. Let $\Pi = \{\pi_m; m = 0, 1, \ldots\}$ be such that $\pi_0, \pi_1, \ldots$ is a sequence of measurable partitions on a space $(\Omega, \mathcal{B})$; π_{m+1} is a refinement of π_m for each m; and $\bigcup_{m=0}^{\infty} \pi_m$ generates $\mathcal{B}$, a $\sigma-$algebra of subsets of Ω. Let P be a random probability measure on $(\Omega, \mathcal{B})$.

Definition 1.1 (Tailfree Process (Ferguson, 1974)) *The distribution of a random probability P on $(\Omega, \mathcal{B})$ is* tailfree *with respect to $\{\pi_m\}$ if there exists a family of nonnegative random variables $\{Y_{m,B}; m = 1, 2, \ldots, B \in \pi_m\}$ such that*

- *the families $\{Y_{1,B}; B \in \pi_1\}, \{Y_{2,B}; B \in \pi_2\}, \ldots$ are independent*
- *for every $m = 1, 2, \ldots$, if $B_j \in \pi_j$ $(j = 1, \ldots, m)$ is such that $B_1 \supset B_2 \supset \cdots \supset B_m$, then $P(B_m) = \prod_{j=1}^{m} Y_{j,B_j}$*

Tailfree processes are conjugate; if the distribution of P is tailfree with respect to $\{\pi_m\}$ and if $X_1, \ldots, X_n$ is a sample from P, then the posterior distribution of P given $X_1, \ldots, X_n$ is tailfree with respect to $\{\pi_m\}$.

Some key results related to tailfree processes and the development of Polya tree priors are now presented.

1.2.2 Dirichlet Processes

The Dirichlet process (DP), one of the most widely-used and researched Bayesian nonparametric methods, is very closely related to the Polya tree. The DP will now be defined and notation will be introduced to motivate upcoming comparisons with Polya tree priors.

A random probability measure P on $(\Omega, \mathcal{B})$ is a *Dirichlet process* on $(\Omega, \mathcal{B})$ with parameter αP_0 (denoted $P \sim DP(\alpha P_0)$) if for every $n = 1, 2, \ldots$ and measurable partition $B_1, B_2, \ldots, B_n$ of Ω, the joint distribution of random probabilities $(P(B_1), \ldots, P(B_n))$ is Dirichlet with parameters $(\alpha P_0(B_1), \ldots, \alpha P_0(B_n))$. The prespecified probability measure, P_0, the *base measure*, is the prior guess and expectation of P: $E(P(B_i)) = P_0(B_i)$. The *precision parameter*, α, describes the degree of faith in the choice of P_0; α could be regarded as a "prior sample size" (Antoniak, 1974). The Dirichlet process selects discrete distributions P with probability 1 (Ferguson (1973), Blackwell (1973) and Blackwell and MacQueen (1973)).

Antoniak (1974) develops mixtures of Dirichlet processes (MDP) to handle the case in which the unknown random distribution is a mixing distribution for a parameter which selects a distribution — e.g., $X \sim N(\mu, 1)$, $\mu \sim P$, $P \sim DP(\alpha P_0)$. The MDP approach is appealing for applications where there is a natural mixture component intrinsic to the problem, or for Bayesian kernel density estimation, in which mixtures of parametric kernels are mixed with respect to a DP (Ferguson (1973), Ferguson (1974), Antoniak (1974), Lo (1984), Kuo (1986), West (1992), Escobar and West (1995), Müller *et al.* (1997)).

The DP selects a discrete distribution with probability 1, which is a drawback

for many applications. Using MDP could be a solution when the mixing parameter indexes continuous distributions; despite this, the discreteness of the DP remains a common criticism of the DP.

1.2.3 Polya Urn Scheme and Tailfree Processes

Tailfree processes were first described in terms of the Polya urn scheme by Blackwell and MacQueen (1973), who show that the limit of the sampling distribution of random variables $X_1, \ldots, X_n$ drawn from a Polya urn scheme as $n \to \infty$ converges to a distribution P that arises from a DP.

Mauldin and Williams (1990) also employ the Polya urn scheme to generate random distribution functions — moreover, their construction involves generating a random distribution from a tree of Polya urns. Their strategy is as follows: Let $E = \{0, 1\}$ represent a set of outcomes – *e.g.,* labeled balls in an urn. Let E^m be the set of all sequences in the $m-$fold product $E \times E \times \ldots \times E$, and let $E^* = \cup_{k=0}^{\infty} E^m$ be the set of all finite sequences of zeros and ones, including the empty sequence $\emptyset$. Associate with each $\varepsilon \in E^*$ an urn B_ε containing one ball labeled 0 and another ball labeled 1. Generate a number $X_1 \in [0, 1]$ as follows:

- Draw a ball from urn $B_\emptyset$; call its label ϵ_1
- Replace the drawn ball with label ϵ_1 with two identically labeled balls
- Draw a ball from urn B_{ϵ_1}; call its label ϵ_2
- Replace ϵ_2 with two identical balls
- Go to urn $B_{\epsilon_1 \epsilon_2}$
- Continue the process

The random variable X_1 is equivalent to the sequence $(\epsilon_1, \epsilon_2, \dots)$ and can be written in terms of its dyadic expansion.

Draw a second number, $X_2 \in [0,1]$, in the same manner as X_1 was generated, only now using the same set of urns B_ε – both updated and non-updated urns – following the generation of X_1. The resulting empirical distribution function converges to a random distribution function which was first constructed by Dubins and Freedman (1967).

1.2.4 Polya Tree Prior

Lavine (1992) and Mauldin *et al.* (1992) formally define and develop the Polya tree prior. The Polya tree prior is a tailfree process that allows probability 1 to be given to sets of continuous, absolutely continuous, and discrete distribution functions – the Polya tree prior is a generalization of the Dirichlet process. The development of the Polya tree prior by Lavine (1992) focuses on the binary tree construction used by Mauldin and Williams (1990) and Ferguson (1974) for constructing random distributions on the real line.

First, we describe the Polya tree prior as described by Lavine (1992). As before, let $E = \{0,1\}$, $E^0 = \emptyset$, E^m be the m-fold product $E \times E \times \ldots \times E$, $E^* = \bigcup_{m=0}^{\infty} E^m$, and $\varepsilon \in E^*$. Let Ω be a separable measurable space, and $\Pi = \{\pi_m : m = 0, 1, \dots\}$, where $\pi_0, \pi_1, \dots$ is a sequence of partitions such that $\bigcup_{m=0}^{\infty} \pi_m$ generates the measurable sets. Every $B \in \pi_{m+1}$ is obtained by splitting some $B' \in \pi_m$ into two pieces. Let the support Ω be denoted by $B_\emptyset$ or π_0. For every $\varepsilon \in E^*$, $B_{\varepsilon 0}$ and $B_{\varepsilon 1}$ result from the splitting of B_ε in two: $B_\varepsilon = B_{\varepsilon 0} \bigcup B_{\varepsilon 1}$.

Definition 1.2 (Polya Tree Prior (Lavine, 1992)) *A random probability measure $\mathcal{P}$ on Ω has a Polya tree prior with parameter $(\Pi, \mathcal{A})$, and is written $\mathcal{P} \sim$*

$PT(\Pi, \mathcal{A})$, *if there exist nonnegative numbers* $\mathcal{A} = \{\alpha_\varepsilon : \varepsilon \in E^*\}$ *and random variables* $\mathcal{Y} = \{Y_\varepsilon : \varepsilon \in E^*\}$ *such that:*

(i) The random variables in $\mathcal{Y}$ *are independent*

(ii) For each $\varepsilon \in E^*$*, the distribution of* Y_ε *is Beta with parameters* $(\alpha_{\varepsilon 0}, \alpha_{\varepsilon 1})$

(iii) For every $n = 1, 2, \ldots$ *and every* $\varepsilon \in E^m$,

$$\mathcal{P}(B_{\epsilon_1 \ldots \epsilon_n}) = \prod_{i=1}^{n} Y_{\epsilon_1 \ldots \epsilon_i}$$

The Y_ε represent the probabilities of partition elements $B_{\varepsilon 0}$ given B_ε. Let $Y_\emptyset$ and $1 - Y_\emptyset$ be the probabilities that $X_i \in B_0$ and $X_i \in B_1$ and, more generally, let Y_ε and $1 - Y_\varepsilon$ be the conditional probabilities of $X_i \in B_{\varepsilon 0}$ and $X_i \in B_{\varepsilon 1}$, respectively. Polya trees are conjugate; for the prior given above, $Y_\varepsilon | X \sim Beta(\alpha_{\varepsilon 0} + \sum \delta_{B_{\varepsilon 0}}(X_i), \alpha_{\varepsilon 1} + \sum \delta_{B_{\varepsilon 1}}(X_i))$, where $\delta_{B_{\varepsilon *}}(X)$ is the indicator function of $X \in B_{\varepsilon *}$.

Lavine (1992) provides a canonical construction of the Polya tree prior on $\Re$. One issue to consider is how to construct the partition Π. Center the PT prior about a distribution Q as follows. Let $X \sim Q$, where Q is known and continuous, and the distribution function of Q is G. Then, select Π to be the set such that all $B_\varepsilon \in \{(G^{-1}(k/2^m), G^{-1}((k+1)/2^m)]\}$ for $k = 0, \ldots, 2^m - 1$ at each level m of the tree. For the resulting partition Π, $Q(B_0) = Q(B_1) = 0.5$ and $Q(B_{\varepsilon 0} | Q_\varepsilon) = Q(B_{\varepsilon 1} | Q_\varepsilon) = 0.5$. To center the prior around $Q = U(0, 1)$, select all $B_\varepsilon \in \{(k/2^m, (k+1)/2^m]\}$ for all ε.

Another issue to consider is the selection of the parameters in $\mathcal{A}$. Parameters α_ε in $\mathcal{A}$ determine both how strong the prior distribution will be with respect to data and the general structure of F. As previously mentioned, Ferguson (1974) highlights conditions on $\mathcal{A}$ which yield discrete, continuous singular, and absolutely continuous distributions with probability one. While further details and results will be given

in the next chapter, note that for level $m = 1, 2, \ldots$, $\alpha = 2^{-m}$ implies a Dirichlet process, $\alpha = 1$ implies a Dubins-Freedman distribution, and $\alpha = m^2$ implies an absolutely continuous distribution with probability 1 (Ferguson, 1974). Therefore, through selection of $\mathcal{A}$ and Q, one can center the Polya tree prior about a distribution Q arbitrarily closely, as determined by $\mathcal{A}$, in a manner directly analogous to the specification of a baseline measure and precision parameter in the Dirichlet process – $\mathcal{A}$ can again be thought of as a precision parameter and Q as a base measure.

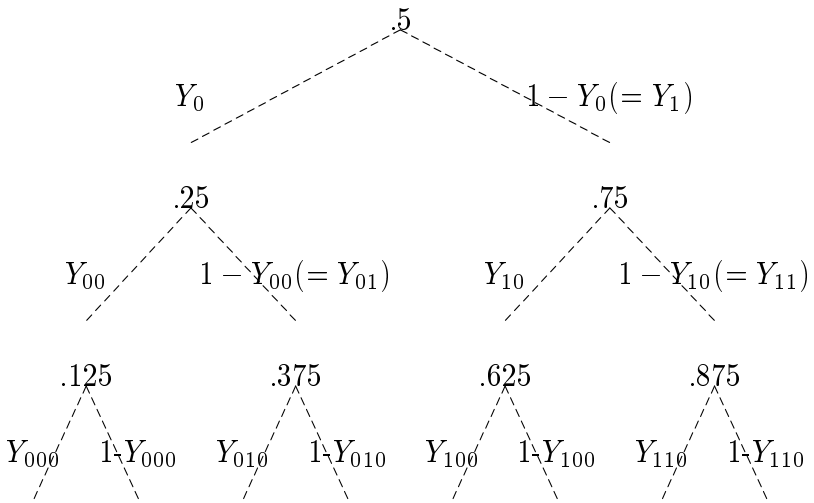
Figure 1.1 shows an example of the construction of a Polya tree prior on $(0, 1]$ (Ferguson, 1974). At the top level of the tree, the top partition element, $B_\emptyset = \Omega = [0, 1]$ is split in half at the dyadic rational, 0.5. At the second level, the resulting halves, $B_0 = (0, 0.5]$ and $B_1 = (0.5, 1]$, are *children*, or *child urns*, or *subcells* of the *partition element* $B_\emptyset : B_\emptyset = B_0 \bigcup B_1$.

The notation of Mauldin *et al.* (1992) differs from the above notation in that $E = \{0, 1, \ldots, k\}$ for some integer k, where $k \geq 1$. Polya tree priors are defined directly on $E^N = E \times E \times \ldots$ and Polya tree distributions are induced on Ω via measurable functions $d : E^N \to \Omega$. For example, if $\Omega = [0, 1]$ as in Ferguson (1974), $k = 1$, $g : E^N \to [0, 1]$:

$$d(\epsilon_1, \epsilon_2, \ldots) = \sum_{i=1}^{\infty} \epsilon_i 2^{-i}.$$

1.2.5 Finite Polya Tree

Computation with Polya trees might be hindered by the need to update the infinite number of parameters which describe the tree. The finite Polya tree, which is also called a 'partially specified Polya tree' (Lavine (1994); Mauldin *et al.* (1992)) addresses this concern. The finite Polya tree is constructed to be identical to the Polya tree up to a finite, pre-specified level m. However, the Polya tree parameters in the



$$F(0.25) = Pr(0 < x < 0.25) = Y_1 Y_{00}$$
$$Pr(0.125 < x < 0.25) = Y_0 Y_{00}(1 - Y_{000})$$

Figure 1.1: Construction of a Polya tree prior on $(0, 1]$ (Ferguson, 1974)

set $\mathcal{A}$ are updated only to level m in the finite Polya tree. Lavine (1994) discusses two scenarios for which it might be reasonable to update only to a pre-specified level M. The first case is when the parameters in $\mathcal{A}$ are constructed to increase rapidly enough that as the level of the tree increases, the posterior updating of the distributions of Y_ϵ beyond level M based on a data set of size n does not affect the prior strongly. The second scenario in which finite Polya trees are appealing arises from concerns of prior elicitation; it might be possible to elicit prior information about parameters near the top of the Polya tree and information about aspects of the distribution such as shape and modality, but it could be unreasonable to expect to elicit meaningful prior distributions for each and every parameter of the Polya tree prior. The former of these two scenarios is most of interest here; most, if not all, of the prior distributions to be used in this thesis entail a specification of the α_ϵ's to be increasing rapidly as the level of the tree increases.

The finite Polya tree will now be defined. The notation to be used here builds upon that set forth in the Polya tree prior definition of the previous Section. Let S be a finite subset of E^* such that, for every $\epsilon = \epsilon_1 \cdots \epsilon_m \in S$, $\epsilon_1 \cdots \epsilon_j \in S$ $(j < m)$ as well, and suppose we have specified values for the parameters describing the partition elements and the subset of $\mathcal{A}$ of interest, $\{B_{\epsilon 0}, B_{\epsilon 1}, \alpha_{\epsilon 0}, \alpha_{\epsilon 1} : \epsilon \in S\}$. Let $T_1 = \{\mathcal{P}(B_{\epsilon 0}), \mathcal{P}(B_{\epsilon 1}) : \epsilon \in S\}$ be the random probabilities assigned by the partially specified Polya tree. Let T_2 be the mass distribution of $\mathcal{P}$ given T_1; then, $\mathcal{P} = (T_1, T_2)$ and the distribution of $\mathcal{P}$ is equal to the distribution of T_1 times the distribution of T_2 given T_1.

Definition 1.3 (Finite Polya Tree (Lavine, 1994)) *The random variable T_1 has a finite Polya tree distribution with parameter $(\mathcal{B}^S, \mathcal{A}^S)$ if there exist sets $\mathcal{B}^S = \{B_{\epsilon 0}, B_{\epsilon 1} : \epsilon \in S\}$, numbers $\mathcal{A}^S = \{\alpha_{\epsilon 0}, \alpha_{\epsilon 1} : \epsilon \in S\}$ and random variables $\mathcal{Y}^S = \{Y_\epsilon : \epsilon \in S\}$ such that:*

1. *all the random variables in* $\mathcal{Y}^S$ *are independent;*

2. *for every* $\epsilon \in S$, Y_ϵ *has a Beta distribution with parameters* $\alpha_{\epsilon 0}, \alpha_{\epsilon 1}$;

3. *for every* $\epsilon = \epsilon_1 \cdots \epsilon_m \in S$,

$$\mathcal{P}(B_{\epsilon_1 \cdots \epsilon_m 0}) = \left(\prod_{j=1;\epsilon_j=0}^{m+1} Y_{\epsilon_1 \cdots \epsilon_{j-1}} \right) \left(\prod_{j=1;\epsilon_j=1}^{m+1} (1 - Y_{\epsilon_1 \cdots \epsilon_{j-1}}) \right).$$

T_2 should be selected so that (T_1, T_2) is consistent with prior beliefs about $\mathcal{P}$. Let ν be a measure on the support of $\mathcal{P}$. Let $\mathbf{P}$ be the set of all possible distributions of the form (T_1, T_2) which are to receive mass 1 under any reasonable prior. Let E be the event that $(T_1, T_2) \in \mathbf{P}$. So long as (T_1, T_2) given E is such that T_1 has a finite Polya tree distribution and $Pr[\mathcal{P} << \nu] = Pr[d\mathcal{P}/d\nu > 0 \ a.e.] = 1$, T_2 can be selected in any fashion. Lavine (1994) searches over the class of prior distributions which are consistent with the available prior information to obtain bounds on prior (and posterior) quantities of interest.

1.2.6 Some Other Methods Relying on a Partition

Andreev and Arjas (1996) argue that in nonparametric Bayesian density estimation, 'the space of all density functions is too large for one to set up a prior supported by the whole space,' and because Bayesians are concerned with finding posterior distributions over the set of all possible density estimates, it might be desirable to approximate a 'true' density by examining a restricted set of possible functions which are arbitrarily close to the 'true' density in some prespecified fashion. The finite Polya tree approach addresses this consideration by considering how to estimate a 'true' distribution function reasonably well by assessing the error of estimates of quantities of interest, such as predictive distributions, means, etc. The practical considerations

of computation are accounted for as well by the finite Polya tree, in that computation can only be done up to a finite number of levels of a tree.

In addition to the finite Polya tree, there have been many other Bayesian methods in which unknown distribution function is modeled by somehow partitioning the sample space of interest. Perhaps the simplest method entails placing a Dirichlet conjugate prior on the (multinomial) probabilities of the partition elements which would be updated by a count of the observations falling into the various partition elements. Histograms are often used to summarize information about arbitrary densities. Hartigan (1996) develops an envelope histogram to reflect the uncertainty associated with selecting a histogram. Usually, the histogram is not of primary interest but is rather a tool for estimating a 'true' underlying density. Lavine (1994) discusses searching for upper and lower bounds on posterior and posterior predictive quantities of interest when using finite Polya trees. The class of histograms can be used to approximate a 'true' (absolutely continuous) distribution (density) arbitrarily well (Andreev and Arjas (1996); Lavine (1996)).

Wolpert and Lavine (1995) place a Markov random field on a partition of the real line to estimate a density function. They illustrate that beliefs about the density, such as monotonicity and continuity, can be incorporated into the model via parameter choice; this is also the case with the Polya tree prior (Lavine 1992, 1994). Sometimes, in Markov random field modeling, the partition is of intrinsic interest; one could easily imagine the points of a lattice of a Markov random field (MRF) applied to a spatial process as representing geographical observation posts of particular interest – *e.g.*, lakes, factories, etc. For other scenarios, the partition is not of primary interest and is employed solely for convenience or necessity.

The main conceptual difference among Polya trees and these other methods is the underlying assumptions about the dependence structure. The dependence structure

induced by the tree entails specification of probabilities at level j, conditionally independent given the parent partition element at level $j-1$. This obviously differs from a histogram, in which the partition and probabilities of partition elements can be chosen in any manner, so long as the sum of the piecewise constant densities is 1. For the Markov random field prior, the probabilities of the partition elements are independent given the nearest neighbors of the partition elements. The definition of 'neighbor' can be made to suit one's modeling purposes; some common specifications entail either taking the partition elements that are directly connected to each cell by a single line segment on a regular lattice, or defining neighbors to be all adjacent partition elements to the partition element of interest. For some applications, the dependence structure will be of primary importance, while for others it is merely chosen for convenience.

For all of these methods, parameter specification is critical, as is true for the Polya tree. Each method has its own set of advantages and disadvantages. Computation is a major factor to consider. While arbitrarily fine partitions lead to better approximations of densities via histograms (Andreev and Arjas, 1996), numerical problems could result with histograms if there is such a huge number of partition elements that computing probabilities of the cells results in small enough probabilities for numerical problems to enter in. Computational time involved with implementing a Markov random field prior is of concern to Wolpert and Lavine (1995) and Lavine and Lozier (1998), who simulate posterior distributions of interest with the Markov chain Monte Carlo methods of Metropolis-Hastings and Gibbs sampling, respectively; Wolpert and Lavine (1995) elect to implement Metropolis-Hastings over a 'glacially slow' Gibbs. Lavine (1998) shows how to compute exact posterior distributions and to simulate from the exact posterior distribution of interest in a conditionally Gaussian Markov random field, which is an improvement over implementing a Gibbs sampler or other

MCMC simulation algorithm. The method is promising in terms of computational efficiency, but thus far is limited application to conditionally Gaussian Markov random fields, and the full impact of computational costs of operations such as matrix inversion and random variate simulation, which are related to factors such as sample size and dimension of the sample space, on posterior computation and simulation remains to be seen.

Chapter 2

MULTIVARIATE POLYA TREE PRIOR

This Chapter begins with notation to be used in describing the Polya tree priors in $\Re^K$, building upon that introduced in Chapter 1. Issues such as specifying absolute continuity of distributions and marginal distributions of Polya trees in $\Re^K$ are considered, and a simulation scheme to obtain conditional predictive distributions and its applicability to missing data imputation is described.

2.1 Notation

The notation and some terminology to be introduced in this Section will be used throughout the dissertation. Much of this directly extends from the development in Chapter 1.

Define a *perpendicular split* of an axis of a hypercube in $\Re^K$ to be a split of the axis by a line which is perpendicular to that axis. The "perpendicular splits" of a hypercube will be all perpendicular splits of each axis of the hypercube made in this fashion. Let K be the dimension of the random vector $(X_1, \ldots, X_K)$ that lies in a $K-$dimensional hypercube. There are 2^K children for each urn B, resulting from binary perpendicular splits along each axis of a $K-$dimensional hypercube. Let $E = \{0, 1, \ldots, 2^K - 1\}$ and ϵ be a element of E. Define a function $b_v : E \to \{0, 1\}$

to be of value equal to the v^{th} bit in the binary representation of ϵ; *e.g.*, if $\epsilon = 6$ and $K = 3$, $b_1(\epsilon) = b_2(\epsilon) = 1$ and $b_3(\epsilon) = 0$. Let $\Theta = (\theta_1, \ldots, \theta_K)$ be the $K-$length vector that is the binary representation of ϵ; *i.e.*, $\theta_v = b_v(\epsilon)$, the v^{th} component of Θ. For $K = 1$, $\epsilon = \Theta$.

Figure 2.1 highlights the binary perpendicular partitioning applied to the axes of the unit square, and shows the equivalence relationship of ϵ and Θ in $\Re^2$. As shown, ϵ is an integer-valued index of the children of the unit square, while the equivalent binary vector $\Theta = (\theta_1, \theta_2)$ represents the left/right (0/1) orientation of each child partition element with respect to the cut points along the axes (the cut points are 0.5 in this case, as denoted along the axes of Figure 2.1); θ_v corresponds to X_v, the v^{th} component of the random vector $X_1, \ldots, X_K$. Just as for the Polya tree in Figure 1.1, this notation is employed in this fashion at subsequent levels of the tree. The relationship between ϵ and Θ illustrated in Figure 2.1 extends to other dimensions of the hypercube.

Unless otherwise stated, E will equal $\{0, 1, \ldots, 2^K - 1\}$ in the following discussion. Let $\epsilon \in E$. Let $E^m = E \times E \times \ldots \times E$ be the $m-$fold product of E. Let $E^* = \bigcup_{m=0}^{\infty} E^m$ and ε be an element of E^*. Let $E^N = E \times E \times \cdots$. Let Ω be a separable measurable space, $\pi_0 = \Omega$ and $\Pi = \{\pi_m; m = 0, 1, \ldots\}$ be a sequence of partitions such that $\bigcup_{m=0}^{\infty} \pi_m$ generates all the measurable sets and such that every $B \in \pi_{m+1}$ is obtained by splitting some $B' \in \pi_m$ into 2^K pieces.

2.2 Polya Tree Priors on $\Re^K$

Polya tree priors on $\Re^k$ are now constructed, using the framework and notation presented by Mauldin *et al.* (1992). Lavine's 1992 presentation is equivalent, though the notation of Mauldin *et al.* (1992) is more convenient for the purposes of this

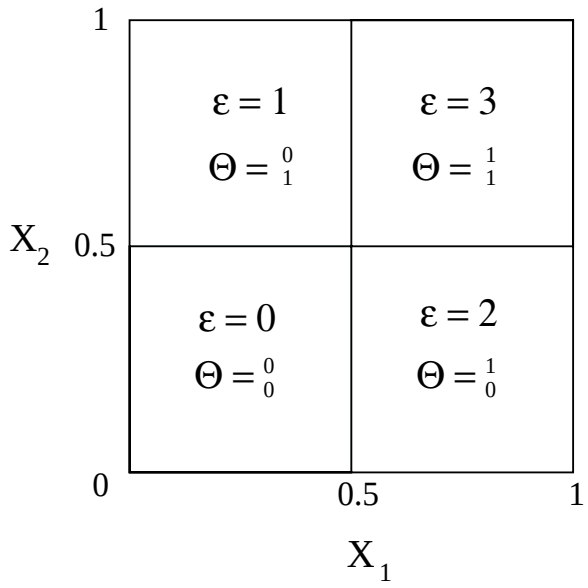


Figure 2.1: Binary splitting along axes of a cube in $\Re^2$. For axis v, corresponding to X_v $(v = 1, 2)$, $\theta_v = 0$ if the urn is below the cut point 0.5.

Chapter. Mauldin *et al.* (1992) point out that if there is a measurable mapping ψ of E^N into another space S, then random distributions on S can be obtained via Polya tree priors. One strategy for measurably mapping the Polya tree structure from E^N to $\Re^K$ will be given in two steps: a mapping from E^N to I^K (where $I = [0, 1]$) will be given, and then a second mapping from I^K to $\Re^K$.

One can measurably map $\psi : E^N \to I^K$ via

$$\psi(\epsilon_1, \epsilon_2, \dots) = \sum_{n=1}^{\infty} \frac{1}{2^n}(b_1(\epsilon_n), \dots, b_K(\epsilon_n))'$$

where $b_i(\epsilon)$ is as previously described. The measurability of the components ψ_i of ψ:

$$\psi_i(\epsilon_1, \epsilon_2, \dots) = \sum_{n=1}^{\infty} \frac{b_i(\epsilon_n)}{2^n}$$

implies the measurability of ψ (Billingsley (1995), Section 13, p. 183). Ferguson (1974), Lavine (1992) and Mauldin *et al.* (1992) discuss $\psi : E \to I$, where $K = 1$ and $b_i(\epsilon) = \epsilon$: $\psi(\varepsilon) = \psi(\epsilon_1, \epsilon_2, \dots) = \sum_{n=1}^{\infty} \frac{\epsilon_n}{2^n}$. Define a second measurable function,

$\gamma : I^K \to \Re^K$. Map $E^N \to \Re^K$ via

$$\phi = \gamma \circ \psi$$

The composition ϕ is measurable, as the composition of two measurable functions is measurable (Billingsley (1995), Theorem 13.1(ii)).

In $\Re^1$, a suitable Borel-measurable function, γ, would be the inverse CDF, G, of a pre-specified distribution function (Lavine, 1992); for example, if $X \sim Q$ *a priori* and Q has CDF G, where G can be of parametric form (*e.g.,* U(0,1)) or any other form. Similarly, $\{\gamma_i\}$ for $i = 1, \ldots, K$ could be chosen to be an inverse CDF.

In this case, let Π be the partition of $\Re^K$ (or a subset thereof) induced by binary perpendicular splits of the axes of the hypercube. Let

$$\mathcal{A} = \{\alpha_{\varepsilon\bullet} = (\alpha_{\varepsilon 0}, \alpha_{\varepsilon 1}, \ldots, \alpha_{\varepsilon(2^K-1)}) : \varepsilon \in E^*\}.$$

A random probability measure P on $\Re^K$ has a Polya tree prior with parameter $(\mathcal{A}, \Pi)$. One small difference from Definition 1.2 is that the random variables $\mathcal{Y} = \{Y_{\varepsilon\bullet} = (Y_{\varepsilon 0}, Y_{\varepsilon 1}, \ldots, Y_{\varepsilon(2^K-1)}) : \varepsilon \in E^*\}$ now have independent Dirichlet distributions, rather than the Beta distributions.

2.3 Continuity of Predictive Distributions in $\Re^K$

Kraft (1964) and Métivier (1971) discuss conditions for which Polya trees on $(0, 1]$ will yield absolutely continuous distributions with probability one. Schervish (1995) generalizes these theorems to arbitrary tailfree processes.

First, let P be a random probability measure which follows a Polya tree prior. By Theorems 1.121 and Lemma 1.124 (pp. 66-68) (Schervish, 1995):

Theorem 2.1 (Schervish (1995)) *Suppose each element of the partition* Π *has a positive measure with respect to a measure* ν *and for every* j, $B_{\epsilon_1 \ldots, \epsilon_k} \in \pi_k$ *(the set*

of partition elements at level k in a tree). Let $Y_{\epsilon_1 \ldots \epsilon_k}$ *be such that, for all k and all* $B_{\epsilon_1 \ldots \epsilon_k}$, $E(Y_{\epsilon_1 \ldots \epsilon_k}) = \nu(B_{\epsilon_1 \ldots \epsilon_k})/\nu(B_{\epsilon_1 \ldots, \epsilon_{k-1}})$. *If*

$$\sum_{m=1}^{\infty} \underset{B \in \pi_m}{sup} \frac{Var(Y_{\epsilon_1 \ldots \epsilon_m})}{(E(Y_{\epsilon_1 \ldots \epsilon_m}))^2} < \infty$$

then with probability 1, P *is absolutely continuous with respect to Lebesgue measure.*

The above theorem provides a criterion by which to select a probability density function that is absolutely continuous with probability 1, provided the conditions of the Theorem hold, of course. For example, if the PT prior is such that $\alpha_{\varepsilon \bullet} = (cm^2, \ldots, cm^2)$ for all $\varepsilon \in E^{m-1}$ and constant $c > 0$, then

$$\begin{aligned}
\sum_{m=1}^{\infty} \underset{B \in \pi_m}{sup} \frac{Var(Y_{\epsilon_1 \ldots \epsilon_m})}{(E(Y_{\epsilon_1 \ldots \epsilon_m}))^2} &= \sum_{m=1}^{\infty} \frac{2^K - 1}{2^{4K}(2^K cm^2 + 1)} \\
&\leq \sum_{m=1}^{\infty} \frac{2^K}{2^{4K}(2^K cm^2)} \\
&= (2^{-4K}/c) \sum_{m=1}^{\infty} m^{-2} \\
&< \infty.
\end{aligned}$$

Of course, if $E(Y_{\epsilon_1 \ldots \epsilon_k}) \neq \nu(B_{\epsilon_1 \ldots \epsilon_k})/\nu(B_{\epsilon_1 \ldots, \epsilon_{k-1}})$, then the Theorem cannot be applied. However, this result can be applied when the partitioning results from dyadic rational perpendicular splits.

Other results with less stringent criteria ensuring P is continuous with probability 1 are given by Mauldin *et al.* (1992).

2.4 Conditional Predictive Simulation

To address the issue of nonparametric learning about relationships among variables throughout $\Re^K$ a method for simulating the conditional predictive distribution of ran-

dom variables $X_1, \ldots, X_j$ given $X_{j+1}, \ldots, X_K$ $(j < K)$ based on the Polya tree prior is now developed. Conditional predictive distributions are of interest because one can learn about relationships among random variables $X_1, \ldots, X_K$ by exploring the distributions of a subset of the random variables, *e.g,* $X_1, \ldots, X_j$, given another subset of variables, *e.g,* $X_{j+1}, \ldots, X_K$. The relationship of $X_1, \ldots, X_j$ given $X_{j+1}, \ldots, X_K$ may vary according to the particular values of $X_{j+1}, \ldots, X_K$.

A computational scheme to simulate the conditional predictive distribution of interest is presented here. The quantities which are necessary for the simulation of the conditional predictive distributions can only be computed up to a finite level m in the tree. Thus, a conditional predictive distribution simulation scheme is developed here which is based on the finite Polya tree framework in $\Re^K$, where the space is partitioned via recursive binary perpendicular splits of the axes of $\Re^K$ up to level m. The points $X_1, \ldots, X_K$ all fall along the partitioned space. Let $X_1, \ldots, X_K$ be a random vector in $\Re^K$.

Conditional Predictive Distribution Simulation

Let $X = (X_1, \ldots, X_j, X_{j+1}, \ldots, X_K)$ be a random vector with distribution F, where F follows a Polya tree prior on $\Re^K$, where the partition is induced by recursive binary perpendicular splits of $\Re^K$. For each component X_i of $X_1, \ldots, X_K$, let $\Theta^{(i)} = (\theta_1^{(i)}, \theta_2^{(i)}, \ldots, \theta_m^{(i)})$ be the binary representation of X_i up to level m. Let

$$\{\theta_{(a:b)}^{(l)}\}_{l=1}^{j} = \{(\theta_a^{(1)}, \theta_{a+1}^{(1)}, \ldots, \theta_b^{(1)}), \ldots, (\theta_a^{(j)}, \theta_{a+1}^{(j)}, \ldots, \theta_b^{(j)})\}$$

be the collection of vectors corresponding to $X_1, \ldots X_j$ from levels a to b. Let $\nu(\cdot)$ be Lebesgue measure; of particular interest here, $\nu(B_{\{\theta_{1:m}^{(i)}\}_{i=1}^{K}})$ is the Lebesgue measure of the partition element $B_{\{\theta_{1:m}^{(i)}\}_{i=1}^{K}}$, which is computed as the product of the lengths of the K axes of the $K-$dimensional hypercube, $B_{\{\theta_{1:m}^{(i)}\}_{i=1}^{K}}$.

Notice that:

$$
\begin{aligned}
p(X_1, \ldots, X_j \quad | \quad & X_{j+1}, \ldots, X_K) \qquad (2.1)\\
\propto \quad & p(X_1, \ldots, X_j, X_{j+1}, \ldots, X_K)\\
\propto \quad & p(\{\theta^{(i)}_{1:m}\}_{i=1}^K)\frac{1}{\nu(B_{\{\theta^{(i)}_{1:m}\}_{i=1}^K})}\\
= \quad & p(\{\theta^{(i)}_{1:c}\}_{i=1}^K)\frac{1}{\nu(B_{\{\theta^{(i)}_{1:c}\}_{i=1}^K})}\\
& \times p(\{\theta^{(i)}_{c+1:m}\}_{i=1}^K|\{\theta^{(i)}_{1:c}\}_{i=1}^K) \times \frac{\nu(B_{\{\theta^{(i)}_{1:c}\}_{i=1}^K})}{\nu(B_{\{\theta^{(i)}_{1:m}\}_{i=1}^K})}\\
= \quad & p(\{\theta^{(i)}_{1:c}\}_{i=1}^K)\frac{1}{\nu(B_{\{\theta^{(i)}_{1:c}\}_{i=1}^K})}\\
& \times p(\{\theta^{(i)}_{c+1:m}\}_{i=1}^j|\{\theta^{(i)}_{1:c}\}_{i=1}^j, \{\theta^{(i)}_{1:m}\}_{i=j+1}^K)\frac{\nu(B_{\{\theta^{(i)}_{1:c}\}_{i=1}^j})}{\nu(B_{\{\theta^{(i)}_{1:m}\}_{i=1}^j})}\\
& \times p(\{\theta^{(i)}_{c+1:m}\}_{i=j+1}^K|\{\theta^{(i)}_{1:c}\}_{i=1}^K)\frac{\nu(B_{\{\theta^{(i)}_{1:c}\}_{i=j+1}^K})}{\nu(B_{\{\theta^{(i)}_{1:m}\}_{i=j+1}^K})}\\
= \quad & p(\{\theta^{(i)}_{c}\}_{i=1}^j|\{\theta^{(i)}_{1:c-1}\}_{i=1}^j, \{\theta^{(i)}_{1:c}\}_{i=j+1}^K)\frac{\nu(B_{\{\theta^{(i)}_{1:c-1}\}_{i=1}^j})}{\nu(B_{\{\theta^{(i)}_{1:c}\}_{i=1}^j})}\\
& \times p(\{\theta^{(i)}_{1:c-1}\}_{i=1}^j, \{\theta^{(i)}_{1:c}\}_{i=j+1}^K)\frac{1}{\nu(B_{\{\theta^{(i)}_{1:c-1}\}_{i=1}^j}) \times \nu(B_{\{\theta^{(i)}_{1:c}\}_{i=j+1}^K})}\\
& \times p(\{\theta^{(i)}_{c+1:m}\}_{i=1}^j|\{\theta^{(i)}_{1:c}\}_{i=1}^j, \{\theta^{(i)}_{1:m}\}_{i=j+1}^K)\frac{\nu(B_{\{\theta^{(i)}_{1:c}\}_{i=1}^j})}{\nu(B_{\{\theta^{(i)}_{1:m}\}_{i=1}^j})}\\
& \times p(\{\theta^{(i)}_{c+1:m}\}_{i=j+1}^K|\{\theta^{(i)}_{1:c}\}_{i=1}^K)\frac{\nu(B_{\{\theta^{(i)}_{1:c}\}_{i=j+1}^K})}{\nu(B_{\{\theta^{(i)}_{1:m}\}_{i=j+1}^K})}
\end{aligned}
$$

The first two lines of the equation result from proportionality. On the third line, the arguments to the function, the $X_1, \ldots, X_K$, are rewritten in terms of their binary vectors up to level m; the $\nu(\cdot)$ terms are included because the binary vectors

representing $X_1, \ldots, X_K$ included here only represents levels 1 to m. The following equalities result from rewriting the second line in terms of conditionally independent components.

Let

$$a = \{\theta^{(i)}_{c+1:m}\}^j_{i=1} \quad b = \{\theta^{(i)}_c\}^j_{i=1} \quad c = \{\theta^{(i)}_{1:c-1}\}^j_{i=1} \quad d = \{\theta^{(i)}_{1:c}\}^K_{i=j+1} \quad e = \{\theta^{(i)}_{c+1:m}\}^K_{i=j+1}.$$

In terms of these identities for (a-e) and $\{\theta^{(i)}_c\}^j_{i=1}$, the last line of Equation 2.1 is proportional to $p(b|cde)$, and:

$$\begin{aligned} p(b|cde) &\propto p(a|bcde) \times p(e|bcd) \times p(b|cd) \\ &= p(ae|bcd) \times p(b|cd) \end{aligned}$$

Since a is unknown at level c, $p(ae|bcd)$ must be computed (up to a normalizing constant) by summing over all possible values of a; this results in $p(e|bcd)$:

$$\begin{aligned} p(e|bcd) &= \sum_{all\{a\}} p(ae|bcd) \\ &\propto \sum_{all\{\{\theta^{(i)}_{(c+1:m)}\}^j_{i=1}\}} \left\{ \prod_{l=c+1}^{m} \frac{\alpha_{\{\theta^{(i)}_{1:l)}\}^K_{i=1}\}}}{\sum_{all\{\{\theta^{(i)}_{1:l}\}^K_{i=1}\}} \alpha_{\{\theta^{(i)}_{(1:l)}\}^K_{i=1}}} \right\} \times \frac{\nu(B_{\{\theta^{(i)}_{1:c-1}\}^j_{i=1}})}{\nu(B_{\{\theta^{(i)}_{1:c}\}^j_{i=1}})} \end{aligned}$$

Also,

$$p(b|cd) \propto \left\{ \frac{\alpha_{\{\theta^{(i)}_{1:c}\}^K_{i=1}}}{\sum_{all\{\{\theta^{(i)}_{1:c}\}^K_{i=1}\}} \alpha_{\{\theta^{(i)}_{1:c}\}^K_{i=1}}} \right\} \times \frac{\nu(B_{\{\theta^{(i)}_{1:c-1}\}^j_{i=1}})}{\nu(B_{\{\theta^{(i)}_{1:m}\}^j_{i=1}})}$$

If the partition elements at each level are all of equal size, then the $\nu(\cdot)$ terms cancel out in the computation of the multinomial probabilities of the 2^j possible values for $\{\theta^{(l)}_c\}^j_{l=1}$. Otherwise, they must be evaluated.

To summarize, simulate $b = \{\theta^{(l)}_c\}^j_{l=1}$ by first computing

$$p(b|cde) \propto p(b|cd) \times p(e|bcd)$$

for all 2^j possible values of b (*i.e.,* $\{\theta_c^{(i)}\}_{i=1}^j$), where:

$$p(b|cd) \propto \left\{ \frac{\alpha_{\{\theta_{1:c}^{(i)}\}_{i=1}^K}}{\sum_{all\{\{\theta_{1:c}^{(i)}\}_{i=1}^K\}} \alpha_{\{\theta_{1:c}^{(i)}\}_{i=1}^K}} \right\} \times \frac{\nu(B_{\{\theta_{1:c-1}^{(i)}\}_{i=1}^j})}{\nu(B_{\{\theta_{1:m}^{(i)}\}_{i=1}^j})}$$

and

$$p(e|bcd) \propto \sum_{all\{\{\theta_{(c+1:m)}^{(i)}\}_{i=1}^j\}} \left\{ \prod_{l=c+1}^{m} \frac{\alpha_{\{\theta_{1:l)}^{(i)}\}_{i=1}^K\}}}{\sum_{all\{\{\theta_{1:l}^{(i)}\}_{i=1}^K\}} \alpha_{\{\theta_{(1:l)}^{(i)}\}_{i=1}^K}} \right\} \times \frac{\nu(B_{\{\theta_{1:c-1}^{(i)}\}_{i=1}^j})}{\nu(B_{\{\theta_{1:c}^{(i)}\}_{i=1}^j})},$$

and secondly by drawing $\{\theta_c^{(i)}\}_{i=1}^j$ based on the resulting multinomial probability vector.

The two key components of the computation at each level c are:

a) The "prior" probability of $\{\theta_c^{(l)}\}_{l=1}^j$ given $X_1, \ldots, X_j$ up to level $c-1$ and $X_{j+1}, \ldots, X_K$ up to level c.

b) The "likelihood" component, *i.e.,* the probability of $X_{j+1}, \ldots, X_K$ beyond level c given the possible values for $X_1, \ldots, X_j$ at level c (i.e., the probability of the "tail" of $X_{j+1}, \ldots, X_K$).

Once the Polya tree prior is updated to the posterior, simulation of $X_1, \ldots, X_j$ given $X_{j+1}, \ldots, X_K$ can be done, bit by bit, at each level of the tree by simulating $\{\theta_c^{(l)}\}_{l=1}^j$ at each level $c = 1, \ldots, m$.

To illustrate the conditional predictive simulation with a small Polya tree with two levels, an example of a Polya tree in $\Re^2$ is given. Let (X_1, X_2) be a random vector in $[0, 1]$. For this two-level tree, denote the binary representations of X_1 and X_2 by $(\theta_1^{(1)}, \theta_2^{(1)})$ and $(\theta_1^{(2)}, \theta_2^{(2)})$, each vector corresponding to bits at levels 1 and 2. Suppose one is interested in simulating the distribution of X_1 given $x_2 = 0.05$. An illustration of the Polya tree is presented in Figure 2.2; the conditional probabilities

of the partition elements at level 1 are given on the Figure as $(1/3, 1/4, 1/6, 1/4)$. The value that is being conditioned on, $x_2 = 0.05$, is denoted by a dashed line in Figure 2.2. Since $x_2 = 0.05$ implies $(\theta_1^{(2)}, \theta_2^{(2)})' = (0,0)'$, the partition elements that have probabilities $(Y_{00'}, Y_{10'}) = (1/3, 1/6)$ are the only ones of interest; the other two partition elements at level 1 can be ignored. In this case, the partition elements at a given level are all of the same size, and so the $\nu(\cdot)$ terms are constants in the multinomial probability vector and thus can be ignored.

To simulate $\theta_1^{(1)}$, the probabilities of $\theta_1^{(1)}$ being equal to 0 or 1 must be computed:

$$\begin{aligned}
p(\theta_1^{(1)} = 0 \quad | \quad & (\theta_1^{(2)}, \theta_2^{(2)}) = (0,0)) \\
\propto \; & p(\theta_1^{(1)} = 0 | \theta_1^{(2)} = 0) \times p(\theta_2^{(2)} = 0 | \theta_1^{(2)} = 0, \theta_1^{(1)} = 0) \\
= \; & p(\theta_1^{(1)} = 0 | \theta_1^{(2)} = 0) \times \{p(\theta_2^{(2)} = 0, \theta_2^{(1)} = 0 | \theta_1^{(2)} = 0, \theta_1^{(1)} = 0) \\
& + p(\theta_2^{(2)} = 0, \theta_2^{(1)} = 1 | \theta_1^{(2)} = 0, \theta_1^{(1)} = 0)\} \\
= \; & \frac{1/3}{1/3 + 1/6}(1/20 + 1/10) \\
= \; & 1/10
\end{aligned}$$

and, similarly,

$$p(\theta_1^{(1)} = 1 | (\theta_1^{(2)}, \theta_2^{(2)}) = (0,0)) \propto \frac{1/6}{1/3 + 1/6}(1/4 + 1/4) = 1/6.$$

Normalizing 1/10 and 1/6 results in the probability of $\theta_1^{(1)} = 1$ to be 0.625. Next, simulate $\theta_1^{(1)}$ by drawing from a Bernoulli with parameter 0.625. Suppose $\theta_1^{(1)}$ is drawn to be 1. At level two, go to the lower right hand partition element corresponding to $Y_{10'} = 1/6$. Compute the probabilities of $\theta_1^{(2)}$ being 0 or 1 given $\theta_2^{(2)} = 0$ as $\frac{1/4}{1/4+1/4} = 1/2$. Then, simulate $\theta_1^{(2)} \sim Bernoulli(1/2)$.

As can be gleaned from this example, one could equivalently draw at point $p(X_1, \ldots, X_j | X_{j+1}, \ldots, X_K)$ from a multinomial distribution with probabilities based

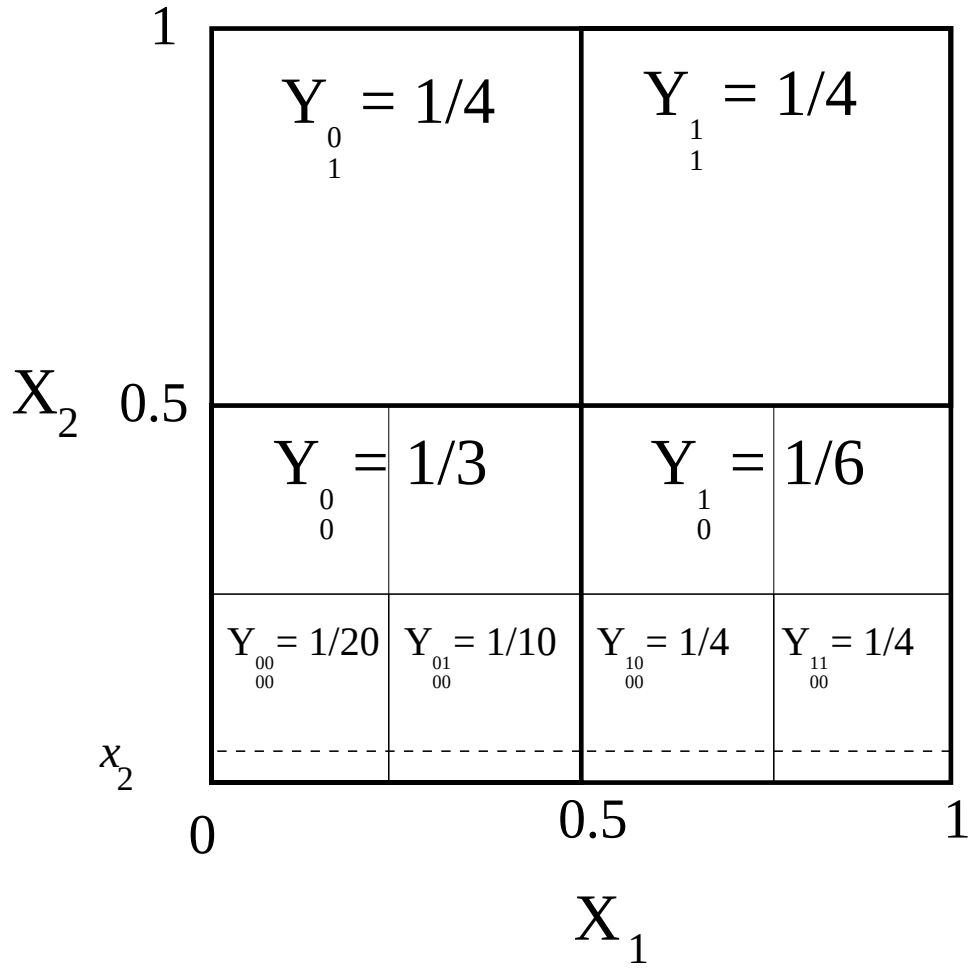


Figure 2.2: Example: Conditional Predictive Simulation. The unit square is split into quadrants (bold solid lines) at $(X_1, X_2) = (0.5, 0.5)$ at level 1, resulting in four partition elements at level 1; the probabilities of these partition elements are indicated by the following Y's: (1/3, 1/4, 1/6, 1/4). At level 2, the quadrants corresponding to $x_2 = 0.05$ (indicated by the dashed line) are further split into quadrants (thin solid lines). The conditional probabilities of the resulting partition elements at level 2 given $X_1 \leq 0.5$ and $x_2 = 0.05$ are denoted by the Y's at level 2 $(1/20, 1/10)$, and the conditional probabilities at level 2 given $X_1 > 0.5$ and $x_2 = 0.05$ are $(1/4, 1/4)$.

on the set of 2^{jm} candidate paths that the point $(X_1 \ldots, X_K)$ can possibly follow down the tree. Such a direct multinomial draw based on 2^{jm} cells is computationally infeasible for even moderate-sized problems: for a tree updated to $m = 10$ levels and for $j = 3$, the resulting multinomial probability vector would be of length 2^{30}, which would require a vector of size 2^{10} megabytes for 8-byte double-precision numbers! The simulation scheme described in this section circumvents this problem by exploiting the conditional independencies in the tree. As mentioned in Chapter 1, the precision with which a finite Polya tree models the 'true' underlying distribution can be regulated by selecting a partition of desired fineness. In the same vein, a density can be estimated by a histogram arbitrarily well by taking arbitrarily fine partition elements (Andreev and Arjas, 1996), as mentioned in Chapter 1. With the conditional predictive simulation scheme, one can explore in greater detail conditional predictive distributions on higher dimensional Euclidean spaces than one would be able to with a brute force multinomial-Dirichlet histogram-like method.

2.5 Missing Data

For examining and processing arbitrary data sets, missing data will inevitably be encountered. Missing data can appear in a problem via censoring or by complete omission. Lavine (1992) discusses the special case of censored data; he illustrates how to build a partition Π on the real line when all that is known is that an observable x is less than some quantity x_0 (p. 1226). Another aspect of missing data – especially in $K-$dimensions – is if for a vector $X = (X_1, X_2, \ldots, X_K)$, components $(X_{i_1}, \ldots, X_{i_j})$ are missing. Imputation of the missing quantities may be desirable in such a case. Missing data can be treated as an unknown variable in the problem, and via Bayesian imputation one can numerically integrate out the missing data in the posterior. In this case, missing data are assumed to be missing at random (MAR) and the missing

data mechanism is ignorable (Little and Rubin, 1987); that is, the probability of a data point missing is independent of X.

As will be shown in this Section, the conditional predictive simulation scheme can also be readily employed for handling missing data that arises in the form of missing components of an observed random vector $X_1, \ldots, X_K$, which follows a distribution F. This F comes from a finite Polya tree distribution with parameter $(\Pi^S, \mathcal{A})$, where Π^S is a partition induced by binary recursive perpendicular splitting, $\mathcal{A}^S$ is the usual finite Polya tree parameter set, and $\mathcal{Y}^S$ is the usual set of Dirichlet-distributed parameters in the finite Polya tree.

Because the conditional predictive simulation scheme is employed here to impute data, the imputation of missing data can only be done up to a finite level m. Let n_i be the number of missing components of the vector $X_{i_1}, \ldots, X_{i_K}$ ($n_i \in \{0, 1, \ldots, K-1\}$), so that at least one component is observed. Let $j_1, \ldots, j_{n_i}, j_{n_i+1}, \ldots, j_K$ be a rearrangement of the indices $\{1, \ldots, K\}$ of the vector $X_{i_1}, \ldots, X_{i_K}$ such that components $X_{i_{j_1}}, \ldots, X_{i_{j_{n_i}}}$ of X_i are missing, and components $X_{i_{j_{n_i+1}}}, \ldots, X_{i_{j_K}}$ are observed. Let

$$D = \{\{\theta^{(l)}_{i_{1:m}}\}_{l=j_{n_i+1}}^{K}\}_{i=1}^{n}$$

represent the observed data (*i.e.,* observed components of the data vectors), and

$$M = \{\{\theta^{(l)}_{i_{1:m}}\}_{l=1}^{j_{n_i}}\}_{i=1}^{n}$$

represent the missing data (*i.e.,* missing components of the data vectors).

The observed-data likelihood is:

$$p(D|\mathcal{Y}^S) \propto \prod_{i=1}^{n}\prod_{c=1}^{m}\left[Y_{\{\theta^{(l)}_{i_{1:c}}\}_{l=1}^{K}}\right]^{I\{n_i=0\}} \times \left[\sum_{all\{\{\theta_{i_{c+1:m}}\}_{l=1}^{n_i}\}}\prod_{t=c+1}^{m} Y_{\{\theta^{(l)}_{i_{1:t}}\}_{l=1}^{K}}\right]^{I\{n_i>0\}} \tag{2.2}$$

where the component denoted by $I\{n_i = 0\}$ results from the usual likelihood under the Polya tree if X_i is fully observed, and the component indicated by $I\{n_i > 0\}$

results otherwise. The resulting posterior distribution of $\mathcal{Y}^S$ based on the observed data is, of course,

$$p(\mathcal{Y}^S|D) \propto p(D|\mathcal{Y}^S)p(\mathcal{Y}^S).$$

Since the posterior quantities of interest — $\mathcal{Y}^S$ and M — are available via the following analytical forms,

$$p(\mathcal{Y}^S|D) = \int p(\mathcal{Y}^S|M, D)p(M|D)dM$$

and

$$p(M|D) = \int p(M|\mathcal{Y}^S, D)p(\mathcal{Y}^S|D)d\mathcal{Y}^S,$$

a chained data augmentation algorithm (Tanner, 1996) can be implemented to obtain samples from $p(\mathcal{Y}^S|D)$ and $p(M|D)$ by iterating over the following steps. Simulate

$$\mathcal{Y}^S \sim p(\mathcal{Y}^S|M, D)$$

$$M \sim p(M|D, \mathcal{Y}^S)$$

where $p(\mathcal{Y}^S|M, D)$ is the usual Polya tree posterior given the "completed" (*i.e.,* the imputed and observed) data and $p(M|D, \mathcal{Y}^S) \propto p(M, D|\mathcal{Y}^S)$.

$p(M|D, \mathcal{Y}^S)$ is simulated as follows. Impute the missing components of X_i, $\{\theta^{(l)}_{i_{1:m}}\}_{l=1}^{j_{n_i}}$ via a multinomial probability vector of length $2^{j_{n_i}m}$, which is the number of all possible paths observation X_i can follow down the tree. Each component of the multinomial vector is of the form:

$$p(\{\theta_{i_{1:m}}\}_{l=1}^{j_{n_i}}|\{\theta_{i_{1:m}}\}_{l=j_{n_i}+1}^{K}, \mathcal{Y}^S) = \frac{Y_{\{\theta^{(l)}_{1:c}\}_{l=1}^{K}}}{\sum_{all\ \{\theta^{(l)}_{i_{1:m}}\}_{l=1}^{j_{n_i}}} \prod_{c=1}^{m} Y_{\{\theta^{(l)}_{1:c}\}_{l=1}^{K}}}$$

Notice, however, that:

$$p(\{\theta_{i_{1:m}}\}_{l=1}^{j_{n_i}}|\{\theta_{i_{1:m}}\}_{l=j_{n_i}+1}^{K}, \mathcal{Y}^S) = \prod_{c=1}^{m} p(\{\theta^{(l)}_{i_c}\}_{l=1}^{j_{n_i}}|\{\theta^{(l)}_{i_{i:c-1}}\}_{l=1}^{j_{n_i}}, \{\theta^{(l)}_{i_{1:m}}\}_{l=j_{n_i}+1}^{K}, \mathcal{Y}^S)$$

Each term in this product is written in exactly the same form as the equation used in the previous Section, upon which the conditional predictive simulation scheme is based. Thus, one can implement the conditional predictive simulation scheme to draw the missing components of X_i by simulating $\{\theta_{i_{1:m}}\}_{l=1}^{j_{n_i}}$ in the manner described in the previous Section.

2.6 Marginal Distributions of Polya Trees

In this section, marginal distributions of multivariate Polya tree priors will be derived. The proof relies on the fact that a random vector of length 2^k that follows a Dirichlet distribution has the same distribution as a vector of 2^K independent Gamma-distributed random variables (with common scale parameter and shape parameters coming from the Dirichlet parameter vector) divided by the sum of the 2^K independent Gamma random variables.

Theorem 2.2 (Marginal Distributions of Polya Trees) *Suppose F is a Polya tree prior on $\Re^k$, with parameters $(\Pi, \mathcal{A})$, where Π is a recursive binary perpendicular partition. Then, the $j-$dimensional marginal distribution of F is a Polya tree.*

Proof: Let $\{Y_{\{\theta_{1:m}^{(l)}\}_{l=1}^k}\}$ be a Dirichlet-distributed random vector of length 2^k, with a 2^k length parameter vector $\{\alpha_{\{\theta_{1:m}^{(l)}\}_{l=1}^k}\}$:

$$\{Y_{\{\theta_{1:m}^{(l)}\}_{l=1}^k}\} \sim Dirichlet(\{\alpha_{\{\theta_{1:m}^{(l)}\}_{l=1}^k}\})$$

$\{Y_{\{\theta_{1:m}^{(l)}\}_{l=1}^k}\}$ has the following equivalent parameterization: the distribution of a Dirichlet random vector is that of a vector of written in terms of a collection of independent Gamma random variables with common scale parameter (Johnson and Kotz, 1976). For each random vector $\{Y_{\{\theta_{1:m}^{(l)}\}_{l=1}^k}\}$ of length 2^k, introduce 2^k random variables,

$Z_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}$, each of which has a Gamma distribution with parameter equal to one of the 2^k parameters in the Dirichlet parameter vector, $\{\alpha_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}\}$, for one of the 2^k possible values of $\{\theta_{1:m}^{(l)}\}_{l=1}^{k}$. Each $Z_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}$ follows a Gamma distribution with shape parameter $\alpha_{\{\theta_{1:m}^{(l)}\}_{l=1}^{K}}$; the $Z_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}$ have a common scale parameter. Then, $Y_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}$ has the same distribution as

$$\frac{Z_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}}{\sum\limits_{all\{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}\}} Z_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}}.$$

To obtain the j-dimensional marginal distribution of F, compute 2^j random variables by summing over $Z_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}$ with respect to all possible values of $\{\theta_{1:m}^{(l)}\}_{l=j+1}^{k}$, for all $Y_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}$:

$$M_{\{\theta_{1:m}^{(l)}\}_{l=1}^{j}} = \sum_{all\{\{\theta_{1:m}^{(l)}\}_{l=j+1}^{k}\}} Z_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}$$

Since the sum of independent Gamma random variables with the same scale parameter is Gamma (DeGroot (1986), p. 289): $M_{\{\theta_{1:m}^{(l)}\}_{l=1}^{j}}$ follows a Gamma distribution with shape parameter $\sum_{all\{\{\theta_{1:m}^{(l)}\}_{l=j+1}^{k}\}} \alpha_{\{\theta_{1:m}^{(l)}\}_{l=1}^{k}}$ and with scale parameter that is common to the 2^K Gamma distributed random variables introduced above.

Then, for each $\{\theta_{1:m}^{(l)}\}_{l=1}^{j}$, there is a random variable, $Y^*_{\{\theta_{1:m}^{(l)}\}_{l=1}^{j}}$, which has the same distribution as

$$\frac{M_{\{\theta_{1:m}^{(l)}\}_{l=1}^{j}}}{\sum\limits_{all\{\{\theta_{1:m}^{(l)}\}_{l=1}^{j}\}} M_{\{\theta_{1:m}^{(l)}\}_{l=1}^{j}}}.$$

The collection of 2^j components of the Dirichlet random vector $\{Y^*_{\{\theta^{(l)}_{1:m}\}^j_{l=1}}\}$ describe the marginalized components of the Dirichlet vector:

$$\{Y^*_{\{\theta^{(l)}_{1:m}\}^j_{l=1}}\} \sim Dirichlet\left(\{\alpha^*\} = \left\{ \sum_{all\{\{\theta^{(l)}_{1:m}\}^k_{l=j+1}\}} \alpha_{\{\theta^{(l)}_{1:m}\}^k_{l=1}} \right\}\right)$$

Thus, repeat the above calculations for every vector Y at level m, and for all levels m. The set of all $\{Y^*_{\{\theta^{(l)}_{1:m}\}^j_{l=1}}\}$ and all $\{\alpha^*_{\{\theta^{(l)}_{1:m}\}^j_{l=1}}\}$ describe the marginal Polya tree on the partition induced by the recursive binary perpendicular splitting on $\Re^j$.

To illustrate, suppose F is a Polya tree in $\Re^2$. The parameters at level 1 follow a Dirichlet distribution as usual:

$$(Y_{00'}, Y_{01'}, Y_{10'}, Y_{11'}) \sim Dirichlet(\alpha_{00'}, \alpha_{01'}, \alpha_{10'}, \alpha_{11'}),$$

just as in Figure 2.1. If one is interested in finding the one-dimensional marginal distribution of F with respect to the first axis of the square, one must sum over the cells created by splitting the support of second axis via a binary perpendicular split. So, in Figure 2.1, this marginalization occurs at level 1 at which $\alpha_{00'}$ and $\alpha_{01'}$ are added together and $\alpha_{10'}$ and $\alpha_{11'}$ are added as well. The marginal distribution of Y would then be $(Y_{00'}+Y_{01'}, Y_{10'}+Y_{11'}) \sim Dirichlet(\alpha_{00'}+\alpha_{01'}, \alpha_{10'}+\alpha_{11'})$. This type of summation can be repeated at all levels of the tree. Notice that the marginalization leads to increased Dirichlet parameters; for example, for a tree in $\Re^2$ for which the prior parameterization is $Y \sim Dirichlet(l, l, l, l)$, where l is the level of the tree, the marginalization results in the Dirichlet parameters of the marginal Polya tree at level 1 being equal to $(2l, 2l)$.

2.7 Ordinal Data

Modeling with Polya trees in $\Re^K$ when a component of X is ordinal can be implemented via a recursive partition of C categories, by initially splitting the C categories

into two partitions. The partitions can be of any size, but if the partitions are of as equal of size as possible, given the previous level, then the ordinal category can be described by a minimum number of levels of the tree. Consider the latter partitioning strategy. This partitioning scheme will yield a tree with its deepest branch describing the ordinal data to be of $\lfloor log_2(C) \rfloor$ levels. For example, if $X = (X_1, X_2)$ and X_1 is continuous while $X_2 \in \{0, 1\}$, the support $B_\emptyset$ is split into four child partition elements as usual, for some cut point $c_\emptyset$:

$$\{X_1 \leq c_\emptyset; X_2 = 0\}; \{X_1 \leq c_\emptyset; X_2 = 1\}; \{X_1 > c_\emptyset; X_2 = 0\}; \{X_1 > c_\emptyset; X_2 = 1\}$$

At level 2, the partitioning can be done with respect to only X_1, so eight child partition elements result; each element above for which $X_1 \leq c_\emptyset$ will become two elements, based on whether $X_1 \leq c_0$ and similarly, for $X_1 > c_\emptyset$, two child partition elements result based on whether $X_1 \leq c_1$ for some cut points $c_0 < c_\emptyset < c_1$.

If C is a power of 2, the partitioning can be done up to level $log_2(C)$. Otherwise, the partitioning can be done on subgroups of size based on powers of 2; for example, for $C = 12$ the split can be based on whether a data point falls in a subset of 4 elements or in the other 8 elements. The subtree based on the subset of size 4 can be updated $\lfloor log_2(4) \rfloor$ levels, while the other subtree extends $\lfloor log_2(8) \rfloor$ levels. This structure can be expanded to accommodate any number of categories.

Chapter 3

RANDOMIZED POLYA TREES

3.1 Motivation: Partition Dependence

One critical drawback of Polya trees is that the partition Π has a strong influence on posterior and posterior predictive distributions. For example, suppose X is a random variable on the sample space $(0, 1]$ and $X|F \sim F$, where the Polya tree prior for F is constructed via the following construction:

$$\begin{aligned} F \sim & PT(\Pi, \mathcal{A}) \\ \Pi = & \{(k/2^m, (k+1)/2^m]\} \qquad (k = 0, \ldots, 2^m - 1) \\ \mathcal{A} = & \{\alpha_{0_m} = \alpha_{1_m} = m^2\} \qquad \text{at level } m \end{aligned} \tag{3.1}$$

Suppose one data point is observed, $x_1 = 0.51$. Figure 3.1 shows the posterior predictive density with respect to Lebesgue measure on $(0, 1]$; computations are done to 15 levels of the tree. The strong role of the partition, Π, on the posterior predictive density is clearly depicted in Figure 3.1; there is a sizable jump at the dyadic rational, 0.5, which is the split point of $(0, 1]$ at level 1. Similarly, jumps due to the partitioning at levels 2, 3, etc. of the tree are visible at the dyadic rationals 0.75, 0.625, etc.

The partition can be chosen for convenience. The only non-trivial tailfree prior for which partition dependence is not a problem is the Dirichlet process (Ferguson,

1974). Lavine (1992) suggests using a dyadic rational recursive partition when there is no pressing reason to select the partition in any special manner; even for other modeling approaches that require a choice of partition, such as Bayesian histograms, the ease of using a dyadic rational partition might outweigh concerns about partition dependence (Hartigan, 1996). The inferential focus of many Polya tree applications is not on how a particular partition affects posterior inference, which makes the strong influence of the partition on the posterior particularly disconcerting. It is therefore desirable to develop a method which reduces the role of the partition on posterior distributions.

One strategy for reducing the effect of partition dependence is to employ a randomized Polya tree approach, which is developed in this Chapter. The general idea is to randomly jitter a fixed partition to induce smoothing of the discontinuities that result from using a fixed partition. The randomized tree is constructed here by building upon the recursive dyadic partition as follows. At each level of the tree, a bit of randomness is added to the selection of partition element cut points. To partition $(0, 1]$, a partition cut point is selected to be "near" the dyadic rational, 0.5, at level 1; at the second level, the cut points are selected to be "near" 0.25 and 0.75; and cut points at subsequent levels of the tree are selected in this fashion. Concrete details of this particular construction follow in the upcoming Sections. For now, just note that the partitioning in the randomized tree that is developed here is based on the dyadic rational partition specification, but it is conceivable that many other variants on this construction could be developed as well.

Using this idea, each observation x_i in a data set of n observations is given its own partition, with each partition being a small, random perturbation of the underlying dyadic rational partition. The resulting n partitions can be intuitively regarded to be centered about the dyadic partition. An obvious question might be why one would

choose to use n partitions rather than just one random partition applied to all n observations. This model would imply that there is a single underlying partition which would be interesting to learn about. By introducing n random partitions, the partitioning is hierarchical in nature. This is reasonable in light of the motivation for building the randomized tree model, in which one is trying to reduce the effect of the partition on inference rather than to learn about an "underlying" partition of interest. In addition, for many applications, numerical problems are likely to result for moderate to large n if just one random partition is used in the analysis; the resulting posterior distributions would consist of factors which are powers of n; consideration of this small detail should be clear once the model specification is given in the upcoming Sections. Another variant of the approach to be developed here is to jitter partition elements just at level 1, or just at a few levels near the top of the tree; this is appealing because the discontinuities that are induced by a fixed partitioning at the top levels of the tree are the most influential of the set of all discontinuities that result from partition choice. It is straightforward to accommodate this special case by modifying the methods presented in this Chapter.

In this Chapter, the randomized tree approach will be developed. Throughout this chapter, the sample space will be assumed to be $(0, 1]$; the methods presented here can be extended to any space on the real line by transformation, and then to multivariate problems directly, as is done in Chapter 4.

3.2 Construction of Partition for the Randomized Polya Tree

Just as the partition for the Polya tree is constructed recursively, so is the partition for the randomized tree. To motivate the illustration of the recursive partitioning in the randomized tree, first recall from Equation 3.1 that for the Polya tree, Π

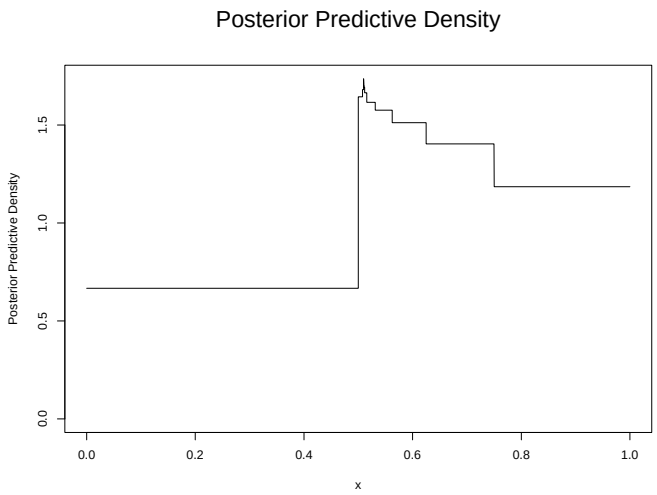


Figure 3.1: Posterior predictive density for a Polya tree prior on $(0, 1]$, computed to 15 levels, with $\Pi = \{(k/2^j, (k+1)/2^j]\} \quad k = 0 \ldots, 2^j - 1, j = 1, 2, \ldots$, with prior $Y_{\theta_1 \ldots \theta_j} \sim Beta(j^2, j^2)$, based on one observation, $x_1 = 0.51$.

is constructed on $(0, 1]$, via recursive dyadic rational partitioning, by splitting the interval into halves at 0.5, resulting in $B_0 = (0, 0.5]$ and $B_1 = (0.5, 1]$ so that $B_\emptyset = B_0 \cup B_1$. The resulting B_0 and B_1 are further split into halves to yield $B_0 = (0, 0.25] \cup (0.25, 0.5]$ and $B_1 = (0.5, 0.75] \cup (0.75, 1]$.

For the randomized tree, partitioning is observation specific. For a single observation x, $(0, 1]$ will be split not into halves but into pieces of sizes β_1 and $(1 - \beta_1)$, so that $B_0 = (0, \beta_1]$ and $B_1 = (\beta_1, 1]$. Then, B_0 and B_1 will each be further split into two pieces of size determined by proportion β_2 (or $1 - \beta_2$) of their length: $B_0 = (0, \beta_1]$ becomes $B_0 = (0, \beta_1\beta_2] \cup (\beta_1\beta_2, \beta_1]$, and similarly $B_1 = (\beta_1, \beta_1 + (1 - \beta_1)\beta_2] \cup (\beta_1 + (1 - \beta_1)\beta_2, 1]$. At level 3, a new parameter β_3 is introduced, and the partition elements of level 2 are split according to proportion β_3. Figure 3.2 shows this recursive partitioning of $(0, 1]$ to level 3 of the tree. The partition element cut points are denoted on Figure 3.2 as functions of $\beta_1, \beta_2, \beta_3$.

This method of recursive partitioning occurs at all subsequent levels of the tree, as a parameter β_j is introduced at each level j of the tree. Further, the $\{\beta_j\}$ are

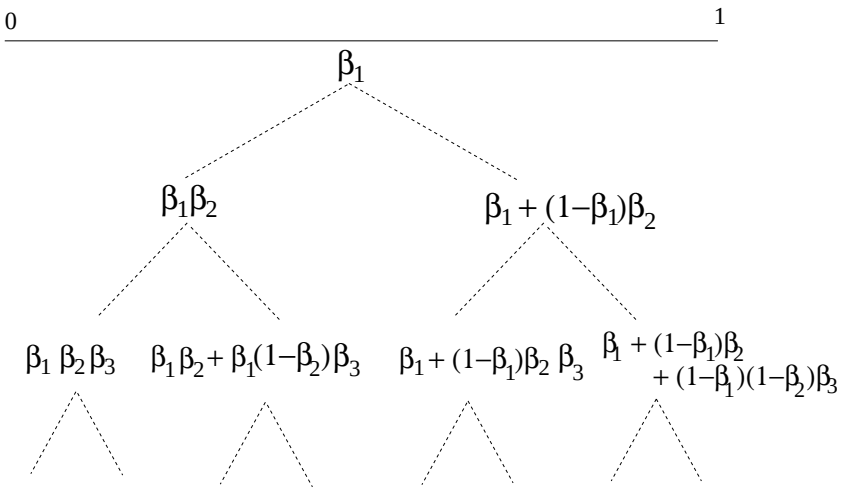


Figure 3.2: Recursive partitioning of $(0, 1]$ via randomized tree to three levels of tree. Cut points appear on the graph as functions of $\beta_1, \beta_2, \beta_3$

taken to be independent. Define the vector $(\theta_1, \ldots, \theta_{m-1})$ as the path taken down the tree by this specific x. The vector, $\beta = (\beta_1, \ldots, \beta_m)$, determines the cut points $\lambda_{\theta_1 \ldots \theta_{m-1}}$ of the partition elements can be computed at any level m as:

$$\lambda_{\theta_1 \ldots \theta_{m-1}} = \sum_{j=1}^{m} \beta_j \theta_j^{I\{j \neq m\}} \prod_{l=1}^{j-1} \beta_l^{1-\theta_l} (1 - \beta_l)^{\theta_l} \tag{3.2}$$

($\lambda_\emptyset$ is defined to be the cut point at level $m = 1$, as shown in Figure 3.3.) Each vector β is specifically tied to an observation x; a second observation, x^*, has its own vector, β^* and hence its own partition points λ^*.

From comparison of Figure 3.2 and Equation 3.2 with Figure 1.1 of Chapter 1, it is clear that $\beta_j = 0.5$ would yield the dyadic partition Π of the Polya tree (Equation 3.1). Because the objective is to add just enough variation to the choice of $\{\beta_j\}$ to move the partition away from Π, it will be desirable to select $\{\beta_j\}$ to be close to 0.5.

Each x_i has its own partition, which is determined by its own set of parameters $\{\beta_{i_j}\}$, with partition cut points as determined in Equation 3.2. For observation x_i $(i = 1, \ldots, n)$, the collection $\{\lambda_{\theta_{i_1} \ldots \theta_{i_{m-1}}} : m = 1, 2, \ldots\}$ induces a sequence of

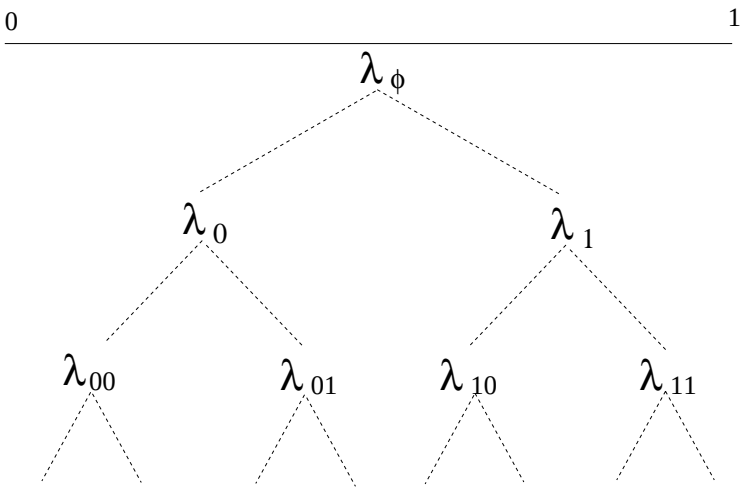


Figure 3.3: Recursive partitioning of $(0, 1]$ via randomized tree to three levels of tree. Cut points appear as subscripted $\{\lambda\}$ on the graph.

partitions:

$$\Lambda_i = \{\ (0,1];\ (0,\lambda_\emptyset) \cup [\lambda_\emptyset, 1];\ (0,\lambda_0) \cup [\lambda_0, \lambda_\emptyset) \cup [\lambda_\emptyset, \lambda_1) \cup [\lambda_1, 1];\ etc.\ \} \tag{3.3}$$

where the cut points are now subscripted by i to emphasize that this partition corresponds to observation x_i.

Figure 3.3 shows the partitioning in terms of the cut points $\lambda_{\theta_1 \ldots \theta_{m-1}}$ rather than in terms of $\{\beta_{i_j}\}$ (Figure 3.2). In comparison to Figure 1.1 of Chapter 1, the structure of Figure 3.3 implies that given Λ_i, x_i conditionally follows some distribution F_i that is constructed by the partitioning. To study F_i, observe the following by comparing Figures 3.3 and 1.1:

Relating F ***and*** F_i ***given*** Λ_i***:***

$$F(0.5) = F_i(\lambda_{\emptyset_i}) \tag{3.4}$$

$$F(0.25) = F_i(\lambda_{0_i}),\ F(0.75) = F_i(\lambda_{1_i})$$

$$F(0.125) = F_i(\lambda_{00_i}),\ F(0.375) = F_i(\lambda_{01_i}),\ F(0.625) = F_i(\lambda_{10_i}),\ F(0.875) = F_i(\lambda_{11_i})$$

$$etc.$$

Conditional on the function Λ_i and the underlying distribution F, this infinite collection of equalities identities F_i. This structure is studied in detail in the following

two Sections.

3.3 Definition of the Random Map $\Lambda_i(\cdot)$

The construction of Λ_i from a collection of independent terms, $\{\beta_{i_j}\}$, implies a function $\Lambda_i(\cdot)$, which determines the partition of $(0,1]$ for the observation, x_i. In this section, this function, $\Lambda_i(\cdot)$, will be studied. The key result of this Section concerns invertibility of $\Lambda_i(\cdot)$.

Definition 3.1 (Functional Form of $\Lambda_i(\cdot)$) *Let z_i be some point in $(0,1]$, written in dyadic expansion as $z_i = \sum_{j=1}^{\infty} \theta_{i_j}/2^j$. Consider unique dyadic expansions of z_i by selecting θ_{i_j} according to the rule that $\theta_{i_j} = 0$ if $z_i < \sum_{k=1}^{j} \theta_{i_k} 2^{-k}$, and $\theta_{i_j} = 1$ otherwise. The collection of independent terms $\{\beta_{i_j}\}$ determines the map $\Lambda_i(\cdot)$ from $(0,1]$ to $(0,1]$ such that:*

$$x_i = \Lambda_i(z_i) = \Lambda_i\left(\sum_{j=1}^{\infty} \theta_{i_j}/2^j\right) = \sum_{j=1}^{\infty} \theta_{i_j}\beta_{i_j}H_{i_j}$$

where $H_{i_j} = \prod_{l=1}^{j-1} \beta_l^{1-\theta_{i_l}}(1-\beta_{i_l})^{\theta_{i_l}}$, and $H_{i_1} = 1.0$

As mentioned in Section 3.1, the cut points will be selected to be "near" the dyadic rationals. To do this, one can select β_{i_j} to be near 0.5, say, within a small distance τ from 0.5; thus constrain β_{i_j} to fall in an interval $(0.5 \pm \tau)$ for "small" τ (*e.g*, 0.01, 0.05, 0.10); much more will be said about the choice of τ later. This constraint also bounds β_{i_j} away from 0 and 1.

Theorem 3.1 (Invertibility of $\Lambda_i(\cdot)$) *The function $\Lambda_i(\cdot)$ is invertible; i.e., for every $x_i \in (0,1]$ there is a unique $z_i \in (0,1]$ such that $\Lambda_i(z_i) = x_i$.*

Proof: The structure of the tree is exploited to show that $\Lambda_i(\cdot)$ does indeed have an inverse. Select any $x_i \in (0,1]$ according to the rule that yields a unique dyadic expansion for x_i, as put forth in the Definition 3.1. The function Λ_i defines a partition of

$(0,1]$, Π_{Λ_i}, which guides us to finding $z_i = \Lambda_i^{-1}(x_i)$ via finding the vector $(\theta_{i_1}, \theta_{i_2}, \dots)$, which represents the unique binary expansion of z_i. To find this $(\theta_{i_1}, \theta_{i_2}, \dots)$, follow x_i down the tree: at each level m, set $\theta_{i_m} = 1$ if $\sum_{j=1}^{m} \beta_{i_j} \theta_{i_j}^{I\{j \neq m\}} \prod_{l=1}^{j-1} \beta_{i_j}^{1-\theta_{i_l}} (1-\beta_{i_l})^{\theta_{i_l}} < x_i$; otherwise $\theta_{i_m} = 0$. After finding $(\theta_{i_1}, \theta_{i_2}, \dots)$, set $z_i = \sum_{j=1}^{\infty} \theta_{i_j} 2^{-j}$. To see that this z_i is unique and satisfies $\Lambda_i(z_i) = x_i$, note that by construction it is not possible to select the θ_{i_j} in such a way that there are two binary representations of the same number. To see that $\Lambda_i(z_i) = x_i$ is indeed true, notice that for all m,

$$\sum_{j=1}^{m} \beta_{i_j} \theta_{i_j} \prod_{l=1}^{j-1} \beta_{i_l}^{1-\theta_{i_l}} (1-\beta_{i_l})^{\theta_{i_l}} \leq x_i < \sum_{j=1}^{m} \beta_{i_j} \theta_{i_j} \prod_{l=1}^{j-1} \beta_{i_l}^{1-\theta_{i_l}} (1-\beta_{i_l})^{\theta_{i_l}} + \prod_{l=1}^{m} \beta_{i_l}^{1-\theta_{i_l}} (1-\beta_{i_l})^{\theta_{i_l}}.$$

(Recall that β_{i_j} is "near" $1/2$, and thus the case of $\beta_{i_j} = 1$ is automatically ruled out.) The difference between the lower and upper bounds on the interval containing x_i converges to 0 as $m \to \infty$. The convergence of the partial sums defining $\Lambda_i(\cdot)$ implies $\Lambda_i(z_i) = x_i$ almost surely. $\square$

3.4 Relating F and F_i via $\Lambda_i(\cdot)$

Given the invertibility of $\Lambda_i(\cdot)$, the connection between F_i and F conditional on Λ_i can be explored. It is known that $x_i | \Lambda_i, F \sim F_i$; that is, conditional on Λ_i and F, observation x_i has distribution F_i. Further, the x_i are conditionally independent given the Λ_i and F. By construction (Equation 3.4), the following equality holds:

$$F(z_i) = F_i(\Lambda_i(z_i)) \quad \text{for } z_i \in (0,1) \tag{3.5}$$

In other words, if $z_i \sim F(\cdot)$, then F_i is the distribution implied by $x_i = \Lambda_i(z_i)$.

The invertibility of $\Lambda_i(\cdot)$ allows for $\Lambda_i^{-1}(x_i) = z_i$ and a variation on Equation 3.5 is:

$$F(\Lambda_i^{-1}(x_i)) = F_i(x_i) \tag{3.6}$$

This is an explicit form of F_i in terms of Λ_i and F. Note that Λ_i now can be seen to be a random perturbation of F but on the quantile scale, as will now be discussed.

3.4.1 $\Lambda_i(\cdot)$ and Random Quantile Functions

In the Polya tree literature, the results of Dubins and Freedman (1967) and Mauldin and Williams (1990) pertain to random quantile functions; these authors develop continuous and strictly increasing, and hence invertible, random distribution functions on $(0, 1]$. While their development is much more general than that of the development of the function $\Lambda_i(\cdot)$, where each Λ_i is constructed for a single observation x_i, it is interesting to notice that $\Lambda_i(\cdot)$ is similar, but not identical, to one of the functions studied by these authors. Dubins and Freedman (1967) generate a probability measure, P_μ, where μ is a uniform distribution on the line $x = 1/2;\ 0 \leq y \leq 1$; Mauldin and Williams (1990) demonstrate how to generate this P_μ via constructing a distribution function, h, via the dyadic rationals as follows. Set $h(0) = 0$ and $h(1) = 1$; draw $h(1/2) \sim U(0, 1)$ at step 1; independently draw $h(1/4) \sim U(0, h(1/2))$ and $h(3/4) \sim U(h(1/2), 1)$ at step 2; and so on. $\Lambda_i(\cdot)$ differs from $h(\cdot)$ in that the collection of points drawn by Mauldin and Williams (1990) at step m: $\{h(c) : c \in \{k/2^m, (k+1)2^m\}; k = 0, 1, \ldots, 2^m - 1\}$ are all drawn independently, conditional on steps $1, \ldots, m-1$, whereas the values of $\Lambda_i(\cdot)$ at step m are all functions of $\beta_{i_1}, \ldots, \beta_{i_m}$ and thus are not independent. It is beyond the scope of this subsection and this dissertation to explore further connections between examining a single $\Lambda_i(\cdot)$ and the random functions of Dubins and Freedman (1967) and Mauldin and Williams (1990), though future research on this topic could be interesting. With the randomized tree, a random distribution is constructed, in part, by constructing a collection of n Λ_i functions, which is a different approach than that taken by these authors.

Let u be be a random variable in (0,1) and let x_i be such that $F_i(x_i) = u$. By

Equation (3.6),

$$u = F_i(x_i) = F(\Lambda_i^{-1}(x_i)).$$

By invertibility of F and F_i, and the fact that $F_i^{-1}(u) = x_i$,

$$\begin{aligned} F^{-1}(u) &= \Lambda_i^{-1}(x_i) \\ &= \Lambda_i^{-1}(F_i^{-1}(u)) \end{aligned}$$

which implies

$$\Lambda_i(F^{-1}(u)) = F_i^{-1}(u), \text{ or equivalently, } \Lambda_i(Q(u)) = Q_i(u)$$

where Q is the quantile function of F and Q_i is the quantile function of F_i. Thus, $\Lambda_i(Q(\cdot))$ is the quantile function corresponding to F_i. The nature of the randomized tree is clear here: the Λ_i randomly "distorts" Q, the quantile function corresponding to F.

3.4.2 Marginal Distribution $P(x_i|F)$

Recall from construction and Equation 3.5, to model an observation x_i as coming from a distribution F_i, first model $z_i \sim F$ and then set $x_i = \Lambda_i(z_i)$. This is equivalent to saying that x_i follows F_i given F and Λ_i. Given this construction, the marginal distribution of x_i, $p(x_i|F)$, which results from integrating $P(x_i|F, \Lambda_i)$ with respect to $dP(\Lambda_i)$ (and noting that $\Lambda_i(\cdot)$ and F are independent by definition), is discussed. As the Λ_i are independent, and independent of F, it follows that given F the x_i are still independent. The prior for Λ_i, derived from that for the $\{\beta_{i_j}\}$, will be assumed to be the same for all i. Then the x_i have a common distribution $P(x_i|F)$. An explicit form for this cannot be derived, but note the following. Conditional on F,

$$P(x_i|F) = \int F_i(x_i) dP(\Lambda_i) = \int F(\Lambda_i^{-1}(x_i)) dP(\Lambda_i).$$

In the special case that $Z_i \sim U(0,1)$, the above expressions imply $P(x_i|F) = E\{\Lambda_i^{-1}(x_i)\}$.

The exact form of $P(x_i|F)$ above is difficult to interpret analytically. However, it may be explored via simulation. The effect of various choices of prior distributions for β on $\Lambda_i(\cdot)$ is now examined to assess how "different" the simulated distribution for x is with respect to a fixed Polya tree distribution F; that is, how much variability is added by $\Lambda_i(\cdot)$? It is important to understand how much variability $\Lambda_i(\cdot)$ adds via the randomized tree. For a fixed F, what is the prior predictive distribution of $P(x_i|F)$? This prior will be influenced by the choice of the prior distribution on $\beta_{i_1}, \beta_{i_2}, \ldots$ which in turn determines the function $\Lambda_i(\cdot)$.

First, select the prior distribution of the $\{\beta_{i_j}\}$ as follows. The $\{\beta_{i_j}\}$ will be independent *a priori*, and will come from a uniform distribution which is concentrated about 0.5:

$$p(\beta_{i_j}) = U(0.5 - \tau, 0.5 + \tau) \qquad j = 1, 2, \ldots \tag{3.7}$$

The prior distributions for $\{\beta_{i_j}\}$ will determine how variable the function $\Lambda_i(\cdot)$ will be. The hyperparameter, τ, will typically be selected to be small. The objective of the randomized tree is to allow for some jitter of the dyadic partition to induce smoothing. It is not necessary to select Λ_i to be very noisy. The objective pursued by specifying a prior for β_{i_j} is to induce a reasonable amount of randomness to the partition by effectively centering $\Lambda_i(z_i)$ about the line $\Lambda_i(z_i) = z_i$ (which corresponds to the dyadic partition) by centering and concentrating the prior for β_{i_j} about 0.5. The influence of various choices of τ will be explored in this Section.

F is fixed to be a $U(0,1)$ distribution ($Y_{\varepsilon_m} = 0.5$ for all m, and Π = dyadic rational partition), and computation of the tree is carried to 15 levels. Figures 3.4–3.5 display 100000 MCMC draws from the simulated distribution for x_i. Each row of subfigures in the Figures corresponds to one choice of τ. In each row, a histogram of the simulated

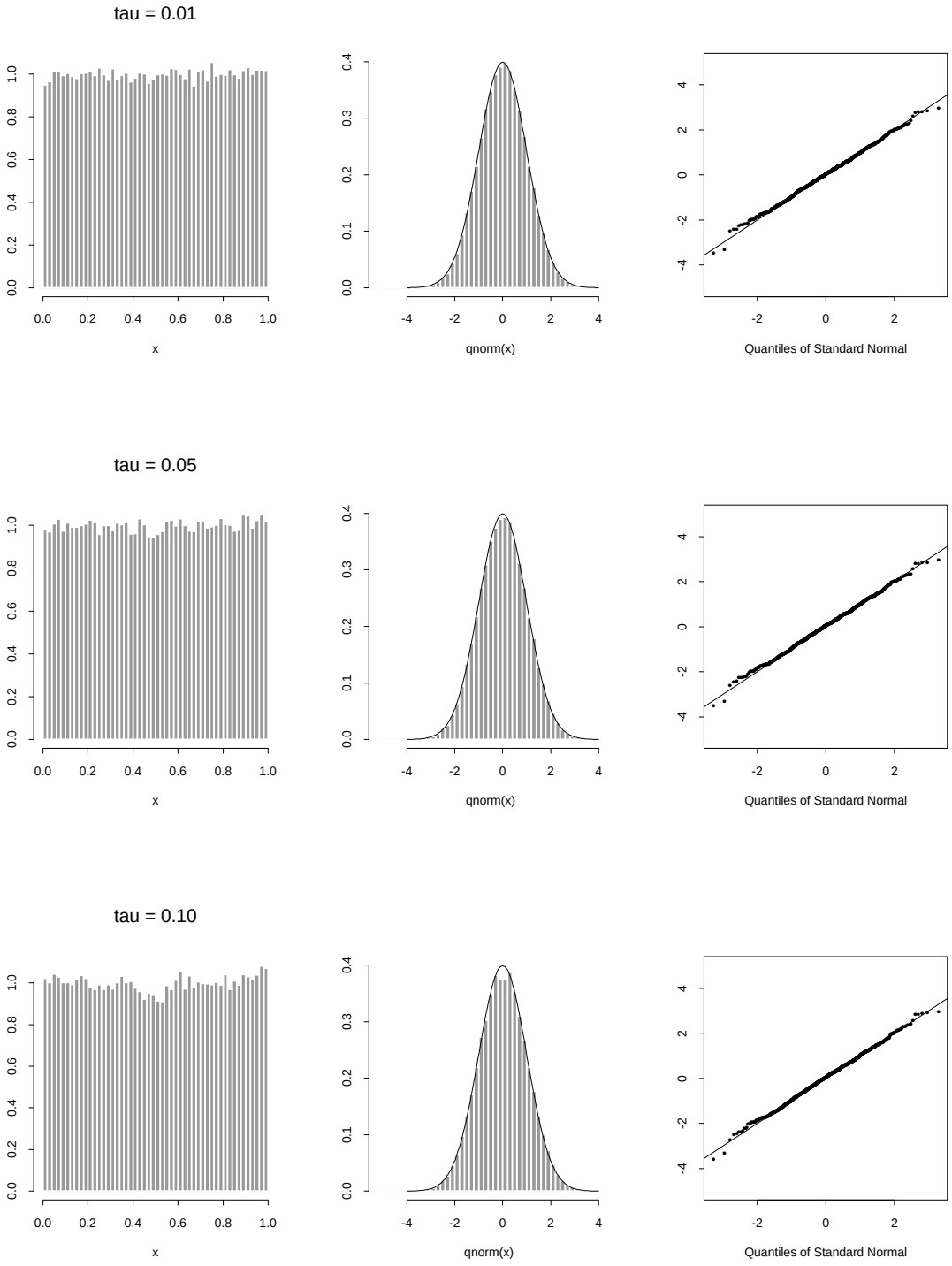


Figure 3.4: Simulated distribution under prior choices for $\tau \in \{0.01, 0.05, 0.10\}$ for F=U(0,1).

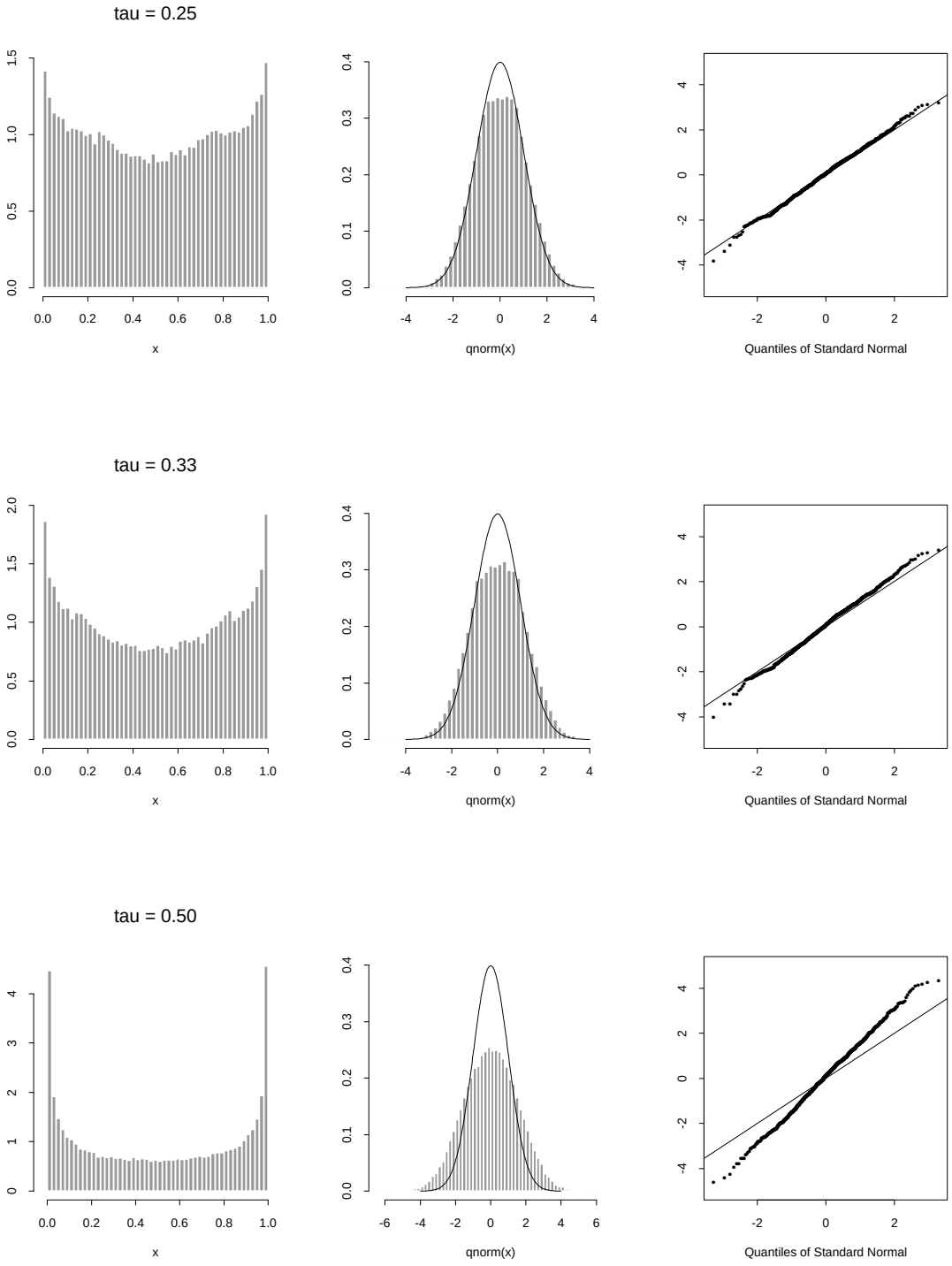


Figure 3.5: Simulated distribution under prior choices for $\tau \in \{0.25, 0.33, 0.50\}$ for F=U(0,1).

distribution for x_1 appears on the left; in the center column appears a histogram of $\Phi^{-1}(\cdot)$, the inverse of the standard normal cumulative distribution function (CDF) of x_1, with a superimposed line indicating the standard normal density function; and in the right hand column appears a normal quantile plot of a random subsample of 1000 of the 100000 simulations of $\Phi^{-1}(x_1)$, plotted against quantiles of the standard normal distribution. The disparity or agreement between the samples and the $N(0,1)$ density, as reflected by the subfigures in the center and right hand side columns, will reveal how close or far the simulated distribution is from uniformity. The simulated distributions for $\tau = \{0.01, 0.05, 0.10\}$ are presented in Figure 3.4, and those for $\tau = \{0.25, 0.33, 0.5\}$ are displayed in Figure 3.5. The lower the value of τ, the closer to uniformity of the simulated distribution. As τ increases, the median of the simulated distribution is sampled from a wider range, which pushes the mass of the distribution toward the edges of the unit interval. Another feature of the histograms of the samples in Figure 3.5 is that there are bumps visible at approximately 0.25 and 0.75, which are clear for $\tau = 0.25$ and $\tau = 0.33$. A hint of the cup-shaped pattern in the distribution which is so clear in Figure 3.5 for larger values of τ appears in the form of small dips at about 0.5 for $\tau = 0.05$ and $\tau = 0.10$ (Figure 3.4). These bumps could be due to the effect of the variable median being replicated at all levels of the tree. Another explanation for the bumps is that the Lebesgue measure of the partition element sizes plays a role in the predictive distribution; small values of the Lebesgue measure of a partition element will increase the predictive density, while comparatively larger values will decrease it. The presence of these small bumps at about 0.25 and 0.75, as well at other locations throughout the distribution, could be due to the effect of the variable partition element size. Note that for all values of τ examined in Figures 3.4-3.5, this effect does not seem to dominate the overall pattern of the simulated predictive distributions. The normal quantile plots reveal

generally linear patterns to the quantiles of the simulated values for x_1, suggesting the distributions of $\Phi^{-1}(\cdot)$ (center column) may approximate normal distributions with variances greater than 1.

3.5 Model Specification for the Randomized Polya Tree

For the randomized tree, the partition is regarded as random. This entails specification of the prior distributions on the β_{i_j}'s. These details are now provided in this section. The likelihood and prior distributions for the β_{i_j}'s as well as for $\mathcal{Y}$, the collection of parameters following Dirichlet distributions which describe the conditional probabilities of belonging to various partition elements in the tree, will be presented. A computational scheme to be implemented to obtain simulations of conditional posterior distributions will be presented in Section 3.6.

Notation

First, some notation is set forth. Let $\epsilon_{1:k} = \epsilon_1 \cdots \epsilon_k$ be a vector of zeros and ones which denotes a general path down the tree to level k. Let $\theta_{i_{1:k}} = (\theta_{i_1}, \ldots, \theta_{i_k})$ be a 0/1 valued random vector denoting the path that x_i takes down the tree to level k. The distinction between $\epsilon_{1:k}$, a *general* path down the tree, and $\theta_{i_{1:k}}$, which describes the path followed down the tree by *observation* x_i, must be made. There is an $\epsilon_{1:k}$ to describe every possible path down the tree to level k, but there are only n $\theta_{i_{1:k}}$'s — one for each observation. Let $\beta_i = (\beta_{i_1}, \ldots, \beta_{i_k})$ be vectors of random variables corresponding to observation x_i, up to level k of the tree. The set of all $\beta_{i_1}, \ldots, \beta_{i_m}$ parameters corresponding to observations will be written as the collection $\{\beta_i\}_{i=1}^n$, or $\{\beta_i\}$ for short. Finally, denote the observed data by $(x_1, \ldots, x_n)$, or $\{x_i\}$ for short.

Finite (to level m) Model

The development of the randomized tree and the function $\Lambda_i(\cdot)$ in the previous sections entails specification of the model for an arbitrary number of levels in the tree. In practice, it is not possible to compute the tree for an arbitrary number of levels when β_{i_j} and Y_{ϵ_j} are unknown beyond a pre-specified finite level, m, as will be clear from the model to be specified. However, one can specify a probability model based on a randomized tree which parallels the finite Polya tree prior (Lavine, 1994) which is described in Chapter 1. For the parameterizations to be used in this thesis – the $\alpha_{\epsilon_1 \cdots \epsilon_j}$ increase rapidly (*e.g.*, $\alpha_{\epsilon_1 \cdots \epsilon_j} = cj^2$ for level j) as the level j increases – it makes sense to stop updating the tree parameters beyond some chosen level m if the sample size n is small enough relative to parameter values of $\alpha_{\epsilon_1 \cdots \epsilon_{m+j}}$ that the prior on $\{Y_{\epsilon_1 \cdots \epsilon_m \cdots \epsilon_{m+j}}\}$ will not be strongly affected by n observations.

From here on, the finite tree model presented here will have a piecewise constant density on each interval $B_{\epsilon_1 \cdots \epsilon_m}$ at level m of the tree; *i.e.,* the distribution on each $B_{\epsilon_1 \cdots \epsilon_m}$ will be uniform. This finite Polya tree specification is analogous to setting $\beta_{i_{m \cdots m+j}} = 0.5$, and to setting $Y_{\epsilon_1 \cdots \epsilon_m \cdots \epsilon_{m+j}}$ beyond level m ($j = 1, 2, \ldots$) equal to $1/2$. Let $\mathcal{Y}^S$ and $\mathcal{A}^S$ be parameters of the finite Polya tree updated to level m (see Definition 1.3).

To re-emphasize, the process studied here is defined on $[0, 1]$, and suitable transformation to the real line can be done to handle any subset of $\Re$. Extension to $\Re^K$ is straightforward.

3.5.1 Prior Distributions

The prior distribution for $\{\beta_{i_j}\}$ was given in the previous Section (Equation 3.7). The prior for $Y_{\epsilon_{1:k-1}0}$ and $k = 0, \ldots, m-1$ is:

$$Y_{\epsilon_{1:k-1}0} \sim Beta(\alpha_{\epsilon_{1:k-1}0}, \alpha_{\epsilon_{1:k-1}1}),$$

which is the usual (finite) Polya tree prior as in Definition 1.3 of Chapter 1.

3.5.2 Likelihood

By construction, the resulting likelihood function is:

$$p(x_1, \ldots, x_n | \mathcal{Y}^S, \{\beta_{i_1}, \ldots, \beta_{i_m}\}_{i=1}^n) \propto \prod_{i=1}^{n} \frac{1}{\nu(B_{\theta_{i_{1:m}}})} \prod_{k=1}^{m} Y_{\theta_{i_{1:k}}} \tag{3.8}$$

Here the $\nu(\cdot)$ terms are the Lebesgue measures of $B_{\theta_{i_{1:m}}}$, and is a function of the parameters which describe the partition for F_i, $(\beta_{i_1}, \ldots, \beta_{i_m})$, and of the path which x_i follows to level m, $\theta_{i_{1:m}}$:

$$\nu(B_{\theta_{i_{1:m}}}) = \prod_{k=1}^{m} \beta_{i_k}^{1-\theta_{i_k}} (1 - \beta_{i_k})^{\theta_{i_k}} \tag{3.9}$$

The product, $\prod_{k=1}^{m} Y_{\theta_{i_{1:k}}}$, results from the multiplication of the conditionally independent probabilities of x_i belonging to $B_{\theta_{i_{1:k}}}$ given x_i is in parent partition element $B_{\theta_{i_{1:k-1}}}$ at level $k-1$.

3.5.3 Posterior Distributions of Model Parameters

Combining the likelihood and prior distributions yields the joint posterior distribution:

$$p(\{\beta_{i_1}, \ldots, \beta_{i_m}\}_{i=1}^n, \mathcal{Y}^S | x_1, \ldots, x_n)$$

$$\propto \prod_{i=1}^{n} \frac{1}{\nu(B_{\theta_{i_{1:m}}})} \prod_{k=1}^{m} Y_{\theta_{i_{1:k}}} \times p(\beta_{i_k}) \prod_{all\{\epsilon_{1:k-1}\}} Y_{\epsilon_{1:k-1}0}^{\alpha_{\epsilon_{1:k-1}0}-1} Y_{\epsilon_{1:k-1}1}^{\alpha_{\epsilon_{1:k-1}1}-1}$$

Exact computation of the marginal posterior distributions of model parameters given data is not possible. Simulation of these marginal posterior distributions by implementing Markov chain Monte Carlo (MCMC) (Gelfand and Smith (1990); Tierney

(1994); Gilks *et al.* (1996)) to obtain draws from the full conditional posterior distributions of the parameters of interest will be performed. Computational details follow the model specification.

Conditional Posterior Distribution for $\mathcal{Y}^S$

The conditional posterior distribution for $Y_{\epsilon_{1:j-1}0} \in \mathcal{Y}^S$ given $\{\beta_i\}$ and $\{x_i\}$ is a Beta distribution with parameters a_1 and a_2, as in the usual Polya tree model:

$$Y_{\epsilon_{1:j-1}0}|\{\beta_i\},\{x_i\} \sim Beta(a_1, a_2) \tag{3.10}$$

where

$$a_1 = \alpha_{\epsilon_{1:j-1}0} + \sum_{i=1}^{n} I\left[\epsilon_{1:j-1}0 = \theta_{i_{1:j}}\right] \qquad a_2 = \alpha_{\epsilon_{1:j-1}1} + \sum_{i=1}^{n} I\left[\epsilon_{1:j-1}1 = \theta_{i_{1:j}}\right] \tag{3.10a}$$

The posterior updating of the Beta parameters $(\alpha_{\epsilon_{1:j-1}0}, \alpha_{\epsilon_{1:j-1}1})$ to the values (a_1, a_2) occurs as follows: if x_i follows the path $\epsilon_{1:j}$ down the tree to level j (*i.e.*, $\theta_{i_{1:j}} = \epsilon_{1:j}$), the Beta parameters for $Y_{\epsilon_{1:j}}$ will be incremented by 1. Each of the parameters a_1 and a_2 is equal to its prior value plus a count of the number of observations falling into partition elements $B_{\epsilon_{1:j-1}0}$ and $B_{\epsilon_{1:j-1}1}$, respectively.

Conditional Posterior Distribution for β_{i_j}

By inspection of the joint posterior distribution, it is clear that the posterior factorizes; *i.e.,* the β_i are conditionally independent over $i = 1, \ldots, n$. The conditional posterior distribution of each β_i is:

$$p(\beta_{i_1}, \ldots, \beta_{i_m}|\mathcal{Y}^S, x_i) \propto \nu(B_{\theta_{i_{1:m}}})^{-1} \prod_{k=1}^{m} Y_{\theta_{i_{1:k}}} p(\beta_{i_k}) \tag{3.11}$$

A complication in simulating from the conditional posterior distribution for β_i above is that one does not know which $B_{\epsilon_1 \cdots \epsilon_m}$ contains x_i. Prior to sampling β_i, one

of the possible 2^m partition elements at level m, $B_{\epsilon_1 \ldots \epsilon_m}$ that is to contain x_i must be selected. Computational approaches for this scenario will be discussed in Section 3.6.

3.5.4 Posterior Predictive Distribution

Now that conditional posterior distributions for the parameters have been derived, the posterior predictive distribution for x_{n+1} can be computed. As usual, the predictive distribution is derived by integrating with respect to the posterior distribution of parameters and conditioning on observed data. Integration with respect to $\{\beta_i\}_{i=1}^n$ averages over the effect of the partition:

$$\begin{aligned} p(x_{n+1}|x_1, \ldots, x_n) &= \int p(x_{n+1}|\{\beta_i\}_{i=1}^n, \mathcal{Y}^S) \times p(\{\beta_i\}_{i=1}^n, \mathcal{Y}^S|x_1, \ldots, x_n) d\beta_1 \ldots d\beta_n d\mathcal{Y}^S \\ &= \int \left\{ \int p(x_{n+1}|\beta_{n+1}, \mathcal{Y}^S) p(\beta_{n+1}) d\beta_{n+1} \right\} \\ &\quad \times p(\{\beta_i\}_{i=1}^n, \mathcal{Y}^S|x_1, \ldots, x_n) d\beta_1 \ldots d\beta_n d\mathcal{Y}^S \end{aligned} \tag{3.12}$$

The second line in the above equation is due to the independence of β_{n+1} and $\beta_1, \ldots, \beta_n$. Given the data $x_1, \ldots, x_n$, simulate the $p(x_{n+1}|x_1, \ldots, x_n)$ via simulation of the posterior distributions of $\{\beta_i\}_{i=1}^n$ and $\mathcal{Y}^S$. Then draw β_{n+1} from its prior distribution, along with an indicator for the path x_{n+1} would take down the tree, $\theta_{n+1_1}, \ldots, \theta_{n+1_m}$, which would be drawn from multinomial distributions based on the updated distributions for $\mathcal{Y}^S$.

3.6 Implementation of the Randomized Tree

A hybrid MCMC sampler is constructed here by combining a Gibbs step to sample $\mathcal{Y}^S$ from its exact full conditional posterior distribution, $p(\mathcal{Y}^S|\{\beta_i\}, x_1, \ldots, x_n)$ (Equation 3.10), and by sampling β via an independence Metropolis step. First, updating

the parameters $\mathcal{A}^S$ for the distribution of $\mathcal{Y}^S$ and how to simulate the distribution of $\mathcal{Y}^S|x_1, \ldots, x_n, \{\beta_i\}$ is discussed, and then how to construct an independence Metropolis sampler for β_i. This is followed by a discussion of the convergence of the resulting transition distribution for this hybrid MCMC sampler.

3.6.1 Updating the Conditional Posterior Distribution for $\mathcal{Y}$

The conditional posterior distribution for $Y_{\epsilon_{1:j-1}0}$ follows a Beta distribution (Equation 3.10). As described earlier, the parameters for the Beta distribution of $Y_{\epsilon_{1:j-1}0}$, a_1 and a_2, reflect the number of observations falling in $B_{\epsilon_{1:j}}$ plus prior parameter values (Equation 3.10a). It is only necessary to keep track of parameters for $Y_{\epsilon_{1:j}}$ corresponding to partition elements $B_{\epsilon_{1:j}}$ where data fall. Conditional on $\{\{\beta_{i_j}\}_{j=1}^m\}_{i=1}^n$, if data do not fall in a partition element $B_{\epsilon_{1:j}}$, the prior values of the distribution of $Y_{\epsilon_{1:j}}$ are taken for a_1 and a_2. Details on how to keep track of parameters a_1 and a_2 while computing with the tree structure are provided in Appendix C.

3.6.2 Simulation of the Conditional Posterior Distribution for β_i

An independence Metropolis-Hastings algorithm is implemented to simulate a vector β_i from a proposal distribution, and then accept the draw with a certain probability. The stationary distribution of β_i given x_i and $\mathcal{Y}^S$ is given by Equation 3.11. Define a proposal density, $q(\beta_i^{prop})$, to be the prior:

$$\text{Proposal distribution: } q(\beta_i^{prop}) = \prod_{j=1}^{m} U(0.5-\tau, 0.5+\tau)$$

Because of the tight priors on the β_i, one would expect that the priors and posteriors for the β_i should be similar. Thus, the independence chain constructed in this way should yield reasonably high acceptance rates. It is also very easy to implement, and

is very fast in comparison to Gibbs sampling. β_i is drawn from a proposal distribution and then the draw is accepted with acceptance probability

$$\alpha(\beta_i^t, \beta_i^{prop}) = min\left\{1, \frac{w(\beta_i^{prop})}{w(\beta_i^t)}\right\},$$

where $\beta_i^{(t)}$ comes from the last simulation cycle, and $w(\beta_i^{(t)})$, the weight function, is equal to $p(\beta_i|\mathcal{Y}^S, x_i)/q(\beta_i)$, which is the target density divided by the proposal density. The distribution $q(= p$, in this case) is bounded away from 0 given the choices of τ, n and m.

3.6.3 Convergence of the MCMC Algorithm

Completion of the above simulations $\{\beta_i^t, \mathcal{Y}^{S\,t}\}$ at times $t = 1, 2, \ldots$, means one cycles though the following steps:

$$\begin{aligned} \mathcal{Y}^{S\,(t)} &\sim p(\mathcal{Y}^S|\{\beta_i^{(t-1)}\}, x_1, \ldots, x_n) \\ \beta_i^{(t)} &\sim p(\beta_i|\mathcal{Y}^{S\,(t)}, x_1, \ldots, x_n) \end{aligned}$$

In order to ensure that the Markov chain constructed here indeed converges to the target distribution of interest — the posterior distribution $p(\{\beta_i\}, \mathcal{Y}^S|x_1, \ldots, x_n)$ – it must be shown that the transition kernel of our hybrid, cyclic MCMC scheme is ergodic – *i.e.,* the transition kernel must be irreducible, aperiodic, and positive Harris recurrent. Two results from Tierney (1994) (Propositions 3 and 4) support uniform ergodicity of our Markov chain, which in turn implies ergodicity. First, an independence kernel with a bounded weight function $w(\cdot)$ is uniformly ergodic. Second, any transition kernel constructed as a cycle which contains a uniformly ergodic kernel is uniformly ergodic. Our weight function, $w(\beta_i)$, is bounded:

$$0 < w(\beta_i) = (2\tau)^m \prod_{j=1}^{m} \frac{Y_{\theta_{i_{1:j}}}}{\beta_i^{1-\theta_{i_j}}(1-\beta_{i_j})^{\theta_{i_j}}} \leq \left\{\frac{2\tau}{(0.5-\tau)}\right\}^m$$

The quantity, $\{\prod_{j=1}^{m} \beta_{i_j}^{1-\theta_{i_j}}(1-\beta_{i_j})^{\theta_{i_{1:j}}}\}^{-1}$, is bounded above by $(0.5-\tau)^{-m}$, the product, $\prod_{j=1}^{m} Y_{\theta_{i_{1:j}}}$ is bounded above by one, and τ and m are fixed.

3.7 Randomized Tree versus Polya Tree

An experiment was done to highlight how the randomized Polya tree prior differs from the Polya tree prior in addressing the partition dependence problem exemplified in Figure 3.1. The analysis shown in that figure is repeated, only this time employing simulation of the randomized tree and of the Polya tree. The trees are updated to 15 levels.

Figure 3.7 shows simulations of 100000 MCMC draws from the posterior distribution of a Polya tree prior updated after observing one data point equal to 0.51, along with three analyses from randomized trees for $\tau = 0.025$, 0.05, and 0.10. The randomized tree appears to be doing well at smoothing discontinuities. Figure 3.6 shows the histograms of the resulting samples for $\beta_1, \ldots, \beta_{15}$, for $\tau = 0.05$. The histograms indicate that the marginal posterior distributions of the $\{\beta_i\}$ are roughly uniform, with small deviations for particular β_i, such as β_2, β_5, and β_7, for example.

Figure 3.8 displays another analysis in which a sample size of 10 was used, in which the data points were equidistant from each other and fall along the sequence $(0.5, 0.55)$. Similar benefits arise from employing the RT framework as are apparent from Figure 3.7.

Of course, it is unlikely that a tree would be applied to an analysis of a data set with one observation. These analyses are simply to illustrate the effect of the partition on inference via Polya trees and randomized trees. The analysis in which the value of x is 0.51 could be criticized for "exaggerating" the effect of the randomized tree on reducing the influence of the partition on inference. In fact, analyses of a data set using Polya trees and randomized trees are compared in Chapter 4. Suppose the

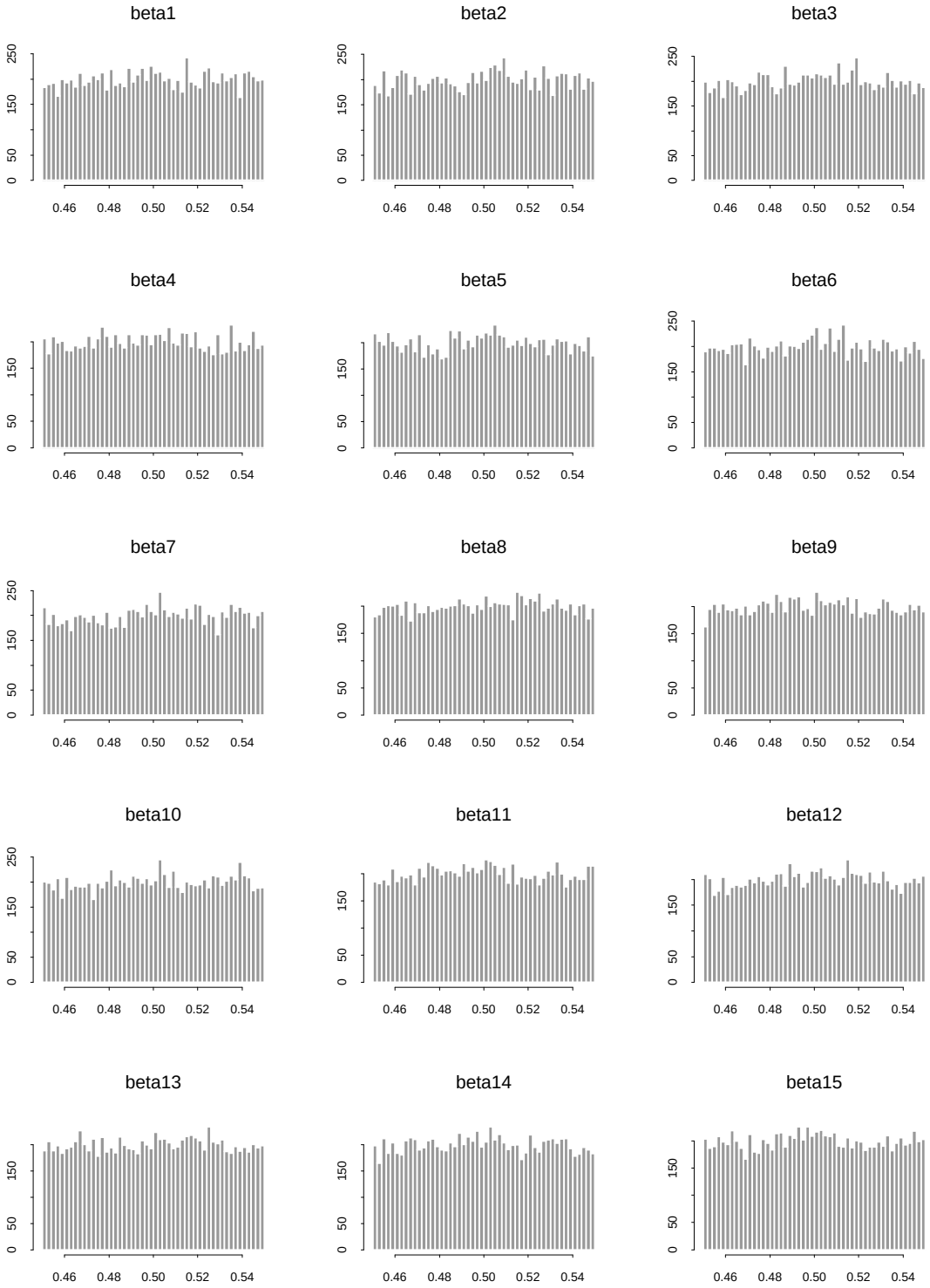


Figure 3.6: Histograms of $\beta_1, \ldots, \beta_{15}$ (10000 subsampled from 100000 iterations) for the analysis with $n = 1$, $\tau = 0.05$.

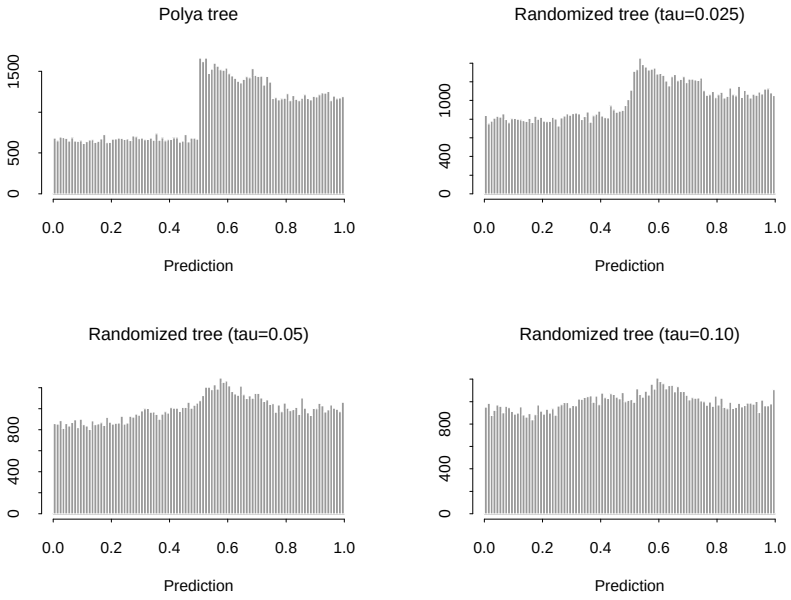


Figure 3.7: n=1 ($x_1 = 0.51$): Simulations of posterior predictive distributions for Polya tree prior (upper left) and randomized Polya trees computed to level 15. $\tau \in \{0.025, 0.05, 0.10\}$.

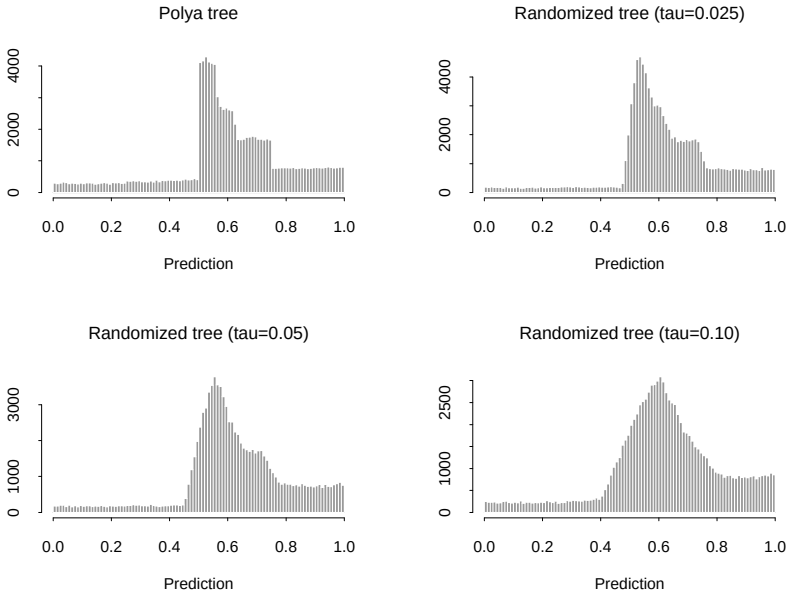


Figure 3.8: n=10: Simulations of posterior predictive distributions for Polya tree prior (upper left) and randomized Polya trees computed to level 15. $\tau \in \{0.025, 0.05, 0.10\}$.

same analysis is repeated on a data set consisting of $x = 0.05$; Figure 3.9 shows the results of this analysis. In the left column of the Figure, the prior parameters of the trees are set to be equal to the square of the level of the tree, just as was the case in Figures 3.7–3.8. The randomized tree is still doing a nice job of reducing the effect of the partition. Of course, with respect to the nesting of the partition elements on $(0, 1]$, 0.05 is an extreme enough value that one would not expect the randomized tree to reduce the effect of the partition as dramatically as is demonstrated in Figure 3.7. However, it is encouraging to see from the Figure that the effect of the partition at the first two levels of the tree is reduced in this case.

In the right column, the prior parameters are set to 0.1 times the square of the level. The point of this sub-analysis is that the randomized tree is still doing a nice job of reducing the effect of the partition under a different parameterization, and, more importantly, the re-emphasize the fact that there are many parameters and combinations of parameter choices that will influence the inference. The partition (and randomizing of the partition) is just one of many factors that must be accounted for in data analysis.

3.7.1 Monitoring Markov Chain Monte Carlo Trajectories

The results of Section 3.6.3 ensure that the transition kernel resulting from the Markov chain Monte Carlo algorithm for simulating the randomized tree parameters converges to the target distribution of interest. To address the question of whether convergence has been reached by the Markov chain implemented here, several standard diagnostic measures are employed in this Section to examine the MCMC output from one of the analyses of Section 3.7, the analysis of the data set consisting of the single point, $x_1 = 0.51$, and parameter $\tau = 0.05$. The acceptance rate for the β vector which is sampled by an independence Metropolis step is 75%.

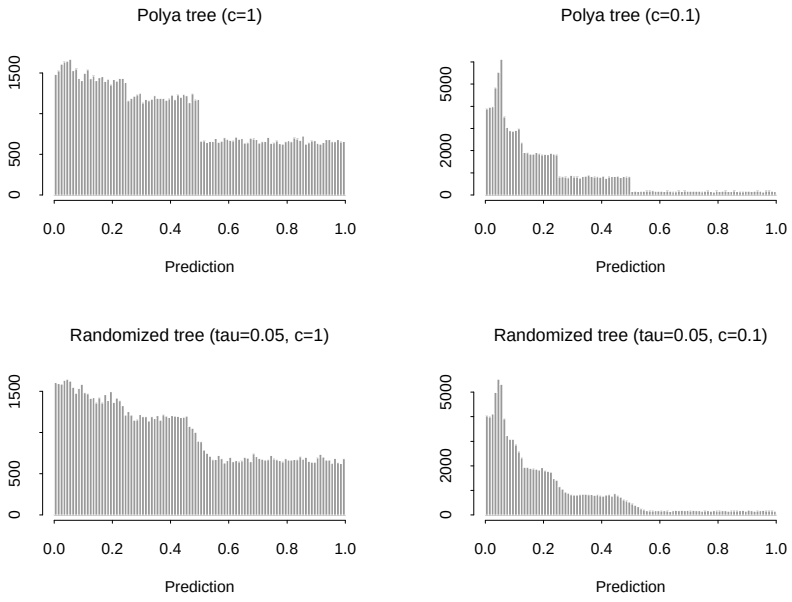


Figure 3.9: Simulations of posterior predictive distributions for Polya tree prior computed to level 15 with $\alpha = level^2$ (upper left); a Polya tree with $\alpha = 0.1\alpha^2$ (upper right); randomized Polya tree (lower left) with $\tau = 0.05$ and $\alpha = level^2$; randomized Polya tree (lower left) with $\tau = 0.05$ and $\alpha = 0.1level^2$.

The trajectories of the MCMC samples are examined visually, and diagnostics which are part of the Bayesian Output Analysis program (BOA) (Smith, 1999) are implemented. BOA is available from

`http://www.public-health.uiowa.edu/boa/`

In particular, autocorrelations and cross-correlations of sampled output are examined, and the Brooks-Gelman-Rubin (Brooks and Gelman, 1998), Heidelberger-Welch, and Raftery-Lewis diagnostics are implemented. Descriptions of these diagnostics are available in Brooks and Gelman (1998) and in Best *et al.* (1995). (Many of these diagnostics are also available in CODA (Best *et al.*, 1995), but BOA is better able to handle numerous MCMC chains than CODA, which will be important for upcoming analyses in Chapter 4.)

The MCMC trajectories were monitored for the example in which $x_1 = 0.51$.

	Lag 1	Lag 5	Lag 10	Lag 50
β_1	-0.0074	0.0057	-0.0100	0.0076
β_2	0.0095	0.0013	0.0021	0.0061
β_3	-0.0061	-0.0081	0.0221	-0.0084
β_4	0.0173	0.0023	0.0007	0.0076
β_5	-0.0211	0.0024	0.0055	-0.0111
β_6	0.0064	-0.0047	-0.0060	0.0044
β_7	0.0035	-0.0035	0.0015	0.0049
β_8	0.0105	-0.0093	-0.0038	-0.0095
β_9	0.0162	-0.0021	-0.0026	-0.0136
β_{10}	-0.0049	-0.0145	-0.0031	-0.0171
β_{11}	0.0020	-0.0004	-0.0067	-0.0021
β_{12}	-0.0025	-0.0115	-0.0092	-0.0040
β_{13}	-0.0144	0.0045	-0.0093	-0.0015
β_{14}	0.0203	0.0011	-0.0129	0.0172
β_{15}	0.0126	0.0087	0.0001	-0.0188

Table 3.1: Autocorrelations of the MCMC samples for β

Figures 3.10–3.12 show MCMC trajectories for an evenly-spaced subsample of 2000 of the 100000 simulations of $\beta_1, \ldots, \beta_{15}$. The trajectories do not indicate any serious signs of lack of convergence; the trajectories do not appear to become "stuck" in any parts of the space, nor do the trajectories appear to wander about the space in any sort of erratic fashion, which would indicate a lack of convergence; in fact, the trajectories appear to follow rather regular patterns of oscillating about the space, which indicates that mixing of the Markov chain is good.

High autocorrelations suggest a slow-mixing chain, which is important to know for assessing how long to run a Markov chain in order to adequately sample all interesting regions of the space. Dependence among the sampled Markov chains, which would be indicated by high cross-correlations and would suggest slow mixing as the high dependence might restrict where in a highly-dimensional space the chain travels, appears to be absent from the MCMC output; Tables 3.1–3.2 show the autocorrelations and cross-correlations of the Markov chains of the $\{\beta_i\}_{i=1}^{15}$. The lack of notable

	β_1	β_2	β_3	β_4	β_5	β_6	β_7
β_2	-0.0254	1					
β_3	-0.0157	-0.0086	1				
β_4	-0.0008	0.0026	0.0242	1			
β_5	0.0035	0.0130	-0.0103	-0.0137	1		
β_6	-0.0023	-0.0045	-0.0069	0.0174	-0.0102	1	
β_7	0.0118	0.0075	0.0044	-0.0006	-0.0050	-0.0058	1
β_8	-0.0096	0.0039	0.0068	-0.0031	0.0165	0.0113	0.0026
β_9	-0.0045	0.0146	0.0142	-0.0105	0.0112	-0.0052	0.0012
β_{10}	0.0022	-0.0134	-0.0048	-0.0107	-0.0028	-0.0007	-0.0104
β_{11}	0.0109	0.0024	0.0047	-0.0015	-0.0095	-0.0007	0.0094
β_{12}	-0.0156	0.0188	-0.0008	0.0014	-0.0154	-0.0131	-2e-06
β_{13}	0.0071	-0.0068	-0.0022	0.0046	0.0101	0.0037	-0.0077
β_{14}	0.0136	-0.0118	0.0002	0.0029	0.0009	-0.0033	-0.0003
β_{15}	-0.0290	0.0037	0.0016	-0.0010	-0.0023	-0.0204	-0.0086

	β_8	β_9	β_{10}	β_{11}	β_{12}	β_{13}	β_{14}
β_9	-0.0228	1					
β_{10}	0.01020	0.008	1				
β_{11}	-0.0073	-0.0132	0.0112	1			
β_{12}	-0.0160	0.0004	0.0083	0.0119	1		
β_{13}	-0.0064	-0.0029	-0.0054	0.0177	0.0055	1	
β_{14}	-0.0008	-0.0095	-0.0128	-0.0170	-0.0036	-0.0113	1
β_{15}	0.0182	-0.0143	-0.0071	-0.0097	0.003613	-0.002654	0.005127

Table 3.2: Cross-correlations of the MCMC samples for β

	Estimate	75%		Estimate	75%		Estimate	75%
β_1	0.999916	0.999940	β_6	0.999904	0.999906	β_{11}	0.999908	0.999937
β_2	1.000028	1.000463	β_7	1.000345	1.001923	β_{12}	1.000011	1.000413
β_3	0.999992	1.000347	β_8	1.000270	1.000697	β_{13}	0.999954	1.000111
β_4	1.000364	1.002201	β_9	1.000306	1.001164	β_{14}	0.999911	0.999935
β_5	0.999913	0.999941	β_{10}	1.000116	1.000853	β_{15}	1.000286	1.001764

Table 3.3: Estimated correction reduction scale factors and 97.5% quantiles of the estimated correction reduction scale factors. This is an estimate of the upper bound on how much the confidence interval of the posterior distributions for β will shrink if the iterative simulation is continued indefinitely, based on the Brooks-Gelman-Rubin convergence diagnostic applied to two Markov chains with over-dispersed starting values. Values in both columns are nearly 1, indicating that the two Markov chains are essentially overlapping.

correlations among the trajectories is a positive sign that the sampler has reached its stationary distribution, as strong correlations would suggest that the chain is slow-mixing and thus quite likely may have yet to reach the stationary distribution.

The Heidelberger and Welch diagnostic showed that the chains for β all passed the stationarity test; for one chain, the diagnostic indicated that all samples came from the stationary distribution, while the diagnostic applied to the second set of chains suggested that initial iterations need to be omitted from four of the fifteen chains (1000-4000 of 10000 iterations). The Raftery-Lewis diagnostic reported a "burn-in" of two steps; the measures of dependence between samples of the chain, dependence factors, were computed to be in the range of 0.95 to 1.06, which are close enough to 1 to indicate that the dependence in samples is not high enough to cause alarm that convergence has yet to be reached. The main idea of the Brooks and Gelman (Brooks and Gelman, 1998) diagnostic compares the simulated distributions obtained from independent multiple simulations, started at over-dispersed starting values, and compare them to the resulting distribution based on combining multiple chains to form one chain. BOA output provides potential scale reduction factors for each chain, β_{1_j} for $j = 1, \ldots, 30$, as well as a multivariate summary, the multivariate potential

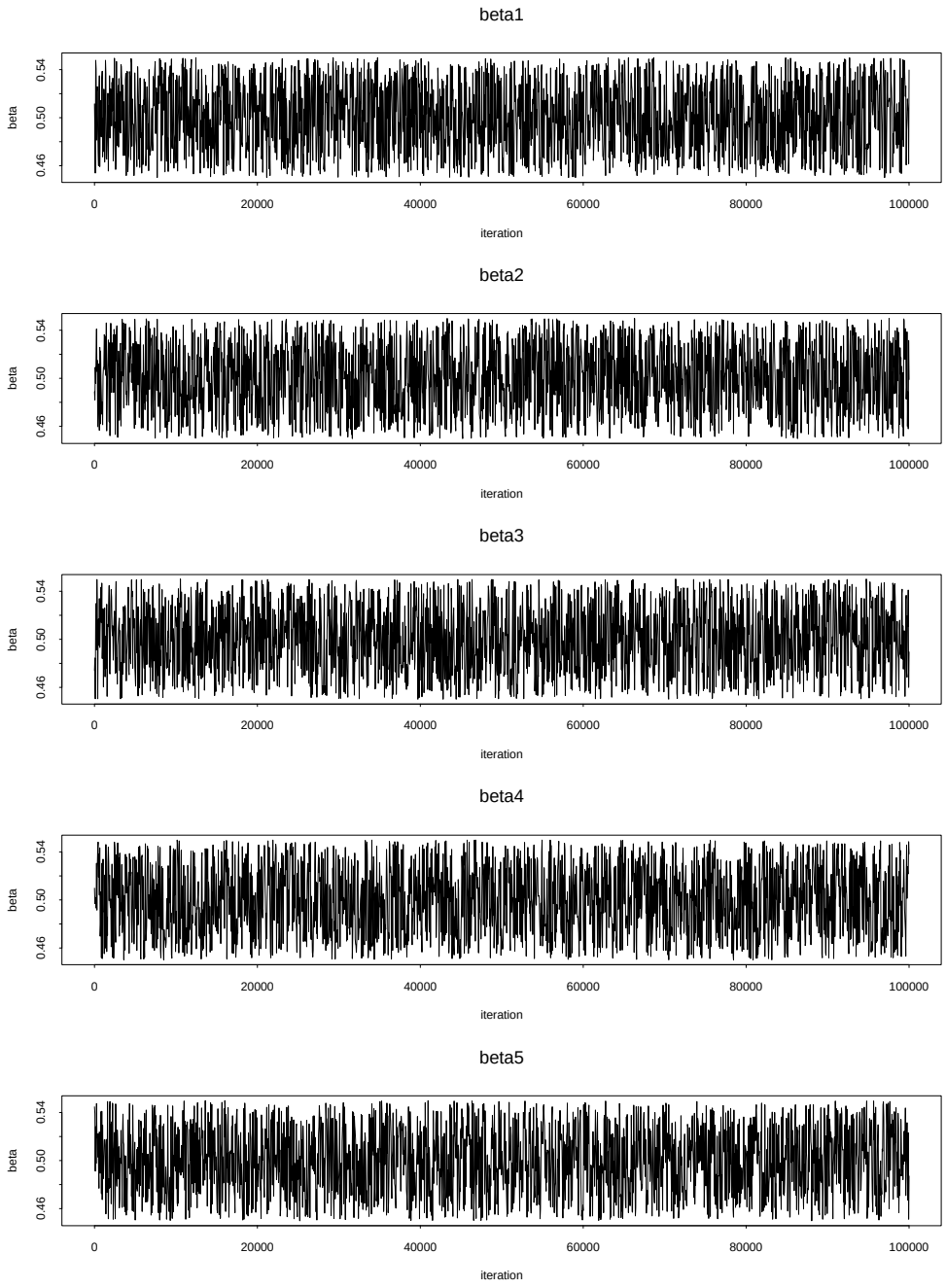


Figure 3.10: MCMC trajectories (a subsample of 2000 of 100000 simulations) for $\beta_1, \ldots, \beta_5$ for the analysis with $n = 1$, $\tau = 0.05$

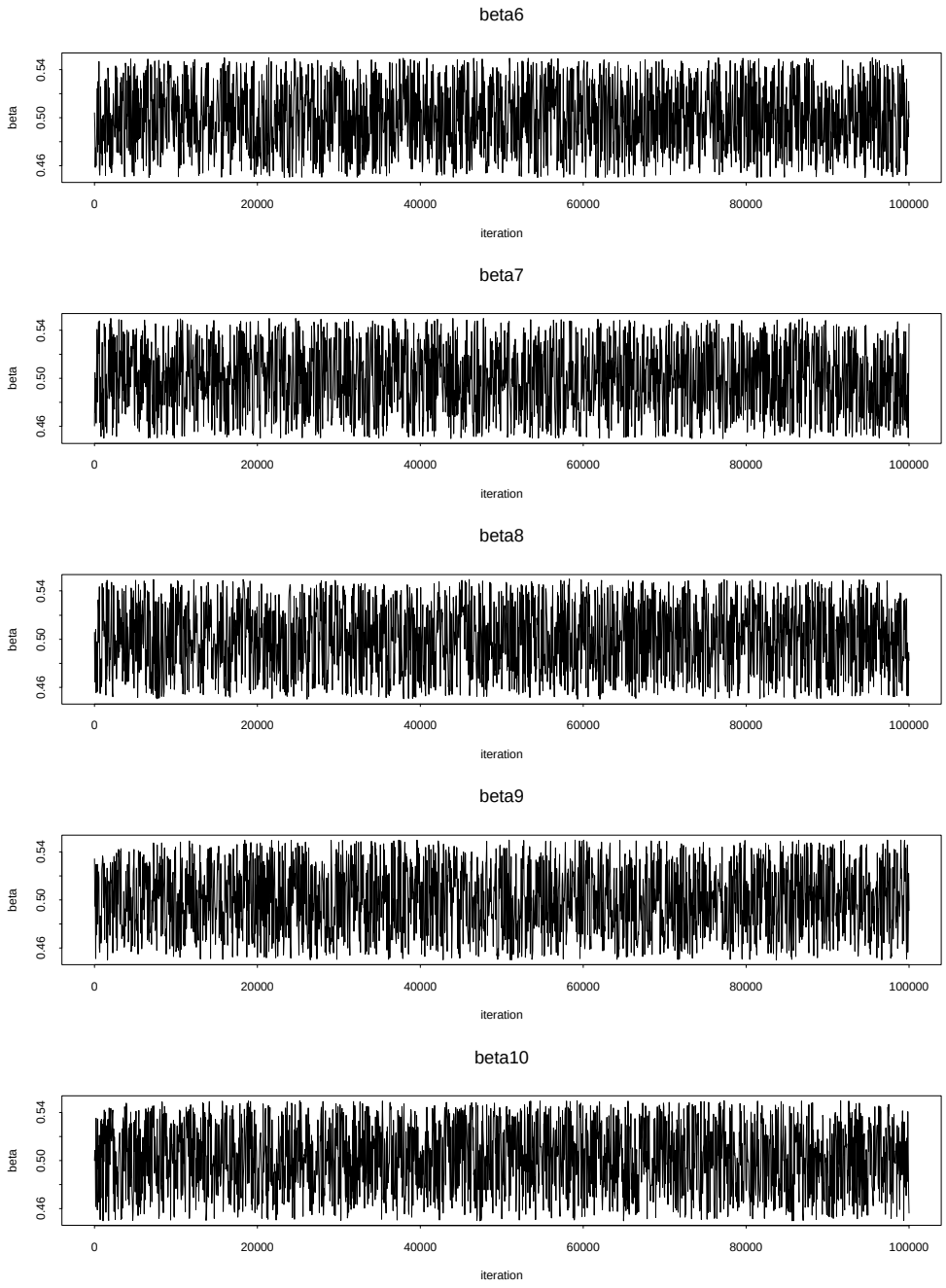


Figure 3.11: MCMC trajectories (a subsample of 2000 of 100000 simulations) for $\beta_6, \ldots, \beta_{10}$ for the analysis with $n = 1$, $\tau = 0.05$.

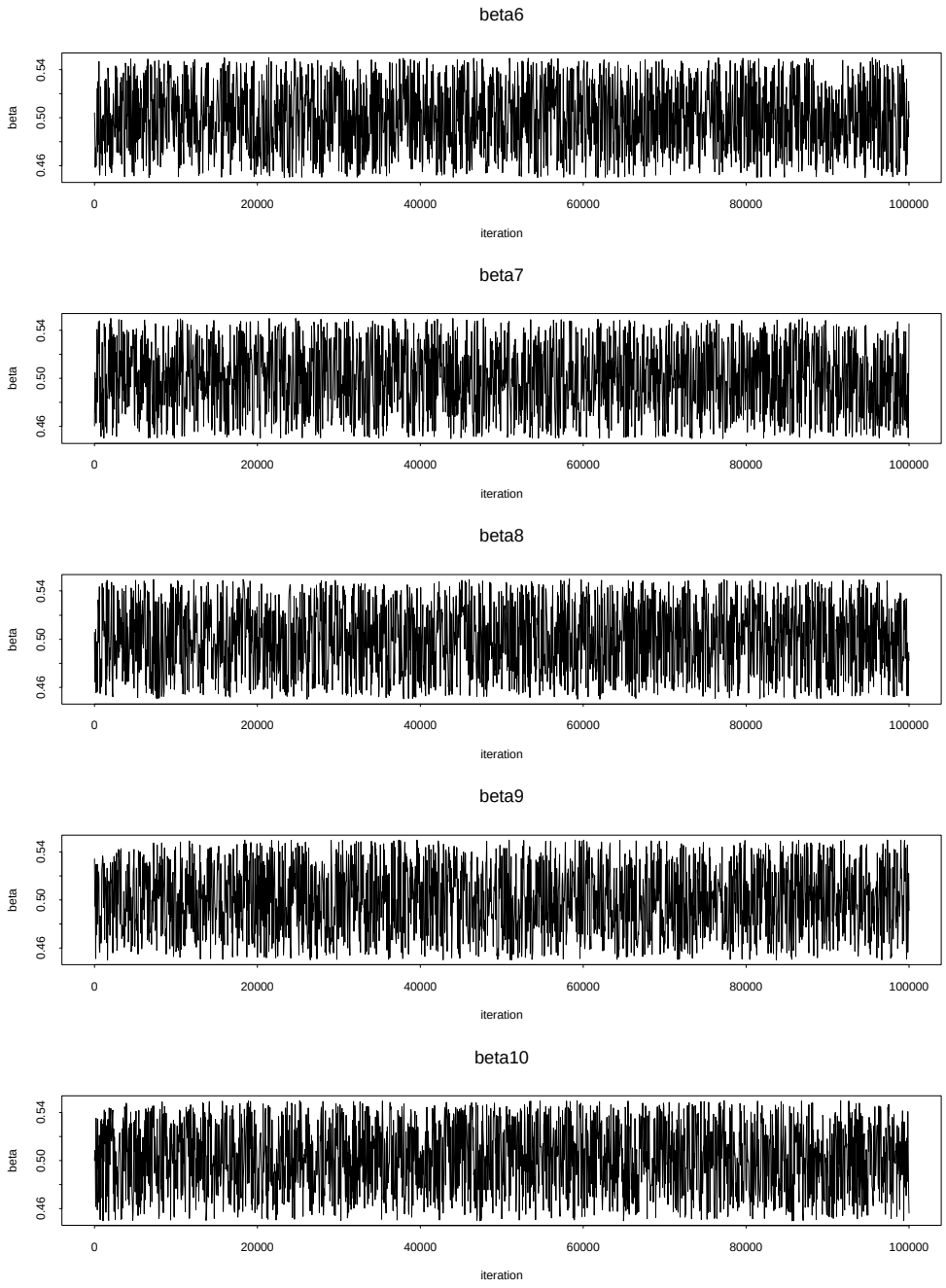


Figure 3.12: MCMC trajectories (a subsample of 2000 of 100000 simulations) for $\beta_{11}, \ldots, \beta_{15}$ for the analysis with $n = 1$, $\tau = 0.05$.

scale reduction factor, which treats the vector β_1 as one entity. In our analyses, two Markov chains are implemented, each with over-dispersed starting values and different random seeds. Table 3.3 shows the estimate and 97.5% quantiles of an estimated upper bound on how much the confidence interval of the posterior distributions would change if the chains were run to infinity. The "shrink factors" are basically equal to 1, which supports the assertion that convergence has been reached. The multivariate potential scale reduction factor is 1.001973.

3.8 Randomized Trees in $[0,1]^K$

In this Chapter, randomized trees have been developed on the unit interval (and to $\Re^1$ by transformation). Application of randomized trees to the hypercube $[0,1]^K$ (and thus to $\Re^K$, by transformation of each axis) is straightforward. Each axis of the hypercube has its own set of $\{\beta_i\}$ which are independent *a priori*. Let $\beta_i^{(j)}$ be the vector of β parameters for axis j $(j = 1, \ldots, K)$ for observation x_i. Then, the prior becomes:

$$p(\{\{\beta_{i_1}^{(j)}, \ldots, \beta_{i_m}^{(j)}\}_{j=1}^K\}_{i=1}^n) \propto \prod_{i=1}^n \prod_{j=1}^K \prod_{l=1}^m p(\beta_{i_l}^{(j)})$$

The resulting likelihood is:

$$p(x_1, \ldots, x_n | \mathcal{Y}^S, \{\{\beta_{i_1}^{(j)}, \ldots, \beta_{i_m}^{(j)}\}_{j=1}^K\}_{i=1}^n) = \prod_{i=1}^n \frac{1}{\nu(B_{\{\theta_{i_{1:m}}^{(j)}\}_{j=1}^K})} \prod_{l=1}^m Y_{\theta_{i_{1:l}}}$$

where $\nu(B_{\{\theta_{i_{1:m}}^{(j)}\}_{j=1}^K}) = \prod_{j=1}^K \prod_{l=1}^m \beta_{i_l}^{(j)^{1-\theta_{i_l}^{(j)}}} (1 - \beta_{i_l}^{(j)})^{\theta_{i_l}^{(j)}}$. The resulting conditional posterior distributions of the parameters follow immediately. In the next Chapter, the randomized tree will be applied to a multidimensional Euclidean space.

Chapter 4

APPLICATIONS

4.1 Analysis of Earthquake Data

The purpose of this analysis is to demonstrate how randomized trees can be used to explore data sets consisting of continuous variables, and how conditional predictive distributions based on randomized trees can be used to explore interesting features of the data. A comparison is made to an analysis using a Polya tree prior.

Simonoff (1996) analyzes a data set of earthquakes which were analyzed and described by Frohlich and Davis (1990). $n = 2178$ earthquake events of magnitude 5.8 to 6.9 occurring between January 1964 and February 1986 were reported by the International Seismological Center. Frohlich and Davis (1990) show how single-link clustering (SLC) could be used to evaluate spatial clustering of earthquakes and how SLC allows for quantitative assessment of whether certain earthquake activity is clustered or isolated. Simonoff (1996) approaches the problem from a frequentist density estimation standpoint, examining, for example, how location (latitude and longitude) of earthquakes varies with depth of earthquake occurrence, using conditional density estimation for exploring multivariate densities.

Four variables are provided in the data set – latitude, longitude, magnitude, and

depth of earthquake. The depth variable describes how many kilometers below the Earth's surface the earthquake occurred. The distinct pattern of the depth data (Figures 4.5–4.6) is explained by Simonoff (1996) as follows. When earthquakes are initially reported, depth is reported initially in rough terms – shallow, or zero depth, or to within 100km depths. Some further processing of the data attempts to confirm whether certain quakes are indeed shallow. Quakes which are confirmed to be shallow but cannot be measured accurately are given a depth of 33km, the assumed depth of the Earth's crust. Magnitude on the Richter scale is always greater or equal to 5.8 and less than or equal to 6.9 in this dataset.

Figures 4.1–4.6 display the data. As is clear from Figure 4.1, the earthquakes occur in distinct clusters, for the most part; most of the activity recorded occurs off of the coast of southeast Asia in the south Pacific Ocean and around Indonesia. Another notable cluster is detectable along the west coast of the Americas. Clusters of activity are also apparent in the middle east, Mediterranean and south Atlantic Ocean. Isolated activity, which is also of interest, is notable at various points throughout the world. Figure 4.2 displays the data for earthquake location by magnitude. The frequency of earthquake occurrence decreases dramatically as magnitude increases. Figure 4.3 shows earthquake occurrence for three groupings of magnitude values – low, medium and high. Note that the earthquakes of highest magnitude, with magnitudes in the range of 6.5-6.8 (high) and colored as red in Figure 4.3, occur in areas of highest earthquake density, with a couple of exceptions – one earthquake on what appears to be a fault line just below South Africa, and a cluster of earthquakes north of Russia at approximately 60 degrees longitude at Novaya Zemlya – this cluster turns out to represent 22 underground nuclear explosions. Occurrence of earthquakes by ranges of depth (km) is presented in Figure 4.4. The marginal plots of data (Figure 4.5) and pairwise plot (Figure 4.6) are shown here as well. As is clear from Figure

4.5, magnitude is measured in discrete units which approximate a true, underlying continuous distribution. In Section 4.1.2, the continuity of magnitude is assumed and the discreteness of the actual magnitude measure is ignored.

One question of interest is, can location of earthquake be predicted, given magnitude? This question is focused on in the upcoming analyses. Before proceeding with the data analysis, a couple of caveats regarding the analysis of this data are mentioned. The first concerns 'edge effects;' the data actually fall on a sphere – *i.e.,* Earth – with location on that sphere that is determined by latitude and longitude; the methods used here assume the data fall along a hypercube, and that each variable is represented by an axis along the hypercube. In reality, this is not correct if we are to examine the entire globe/Earth at once. In order to rectify the matter, one must account for edge effects on the longitude scale. It does not appear that this feature of the data was explicitly incorporated into Simonoff's 1996 density estimation examples. However, it will be shown that this is not a serious obstacle. It does, however, pose an interesting wrinkle that might be worth pursuing if the randomized tree method is to be applied to spatial data in the future. A second caveat is that prior information about specific fault lines is not incorporated into this analysis. For the purposes of illustrating the randomized tree methods this is not a problem. However, for future analysis one might include this information into the model. It is hoped that the nonparametric nature of our methods will detect and explain in further detail what these fault lines are like and where they might be located; one also might expect to learn about isolated vs. non-isolated earthquakes. All graphs from the conditional predictive distribution simulations are presented in Appendix A.

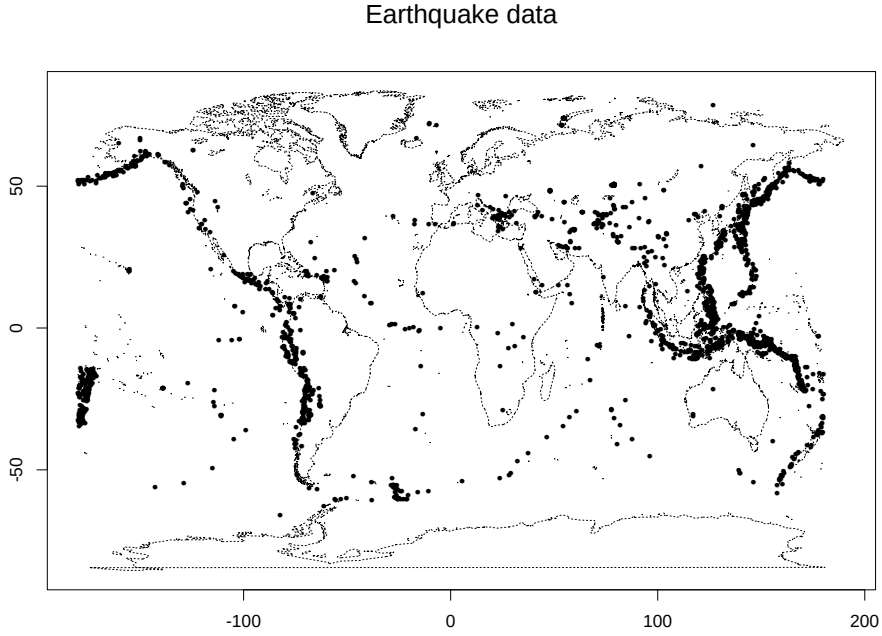


Figure 4.1: Data of earthquake occurrence.

4.1.1 Conditional Predictive Distribution via the Polya Tree

The conditional predictive distributions of location given magnitude are simulated. Before conditional predictive distributions are simulated via the randomized tree approach, these analyses using the finite Polya tree prior are included so that comparisons to the randomized tree can later be made. For the analyses, the Polya tree prior is updated to 10 levels. The α parameters are set to $0.1(level)^2$ at each level $1, \ldots, 10$ of the tree, and G is selected to be the uniform CDF. Figures A.1–A.3 display the results. Judging from the data on location of earthquakes given various values of magnitude (Figures 4.2–4.3), these conditional predictive simulations appear to be quite reasonable. The effect of the partition is clear in all of the simulations via the box like patterns, and it becomes more pronounced as magnitude increases and the number of observations of higher magnitude decreases. Compared to lower magnitudes, there is not a lot of information about higher magnitudes for

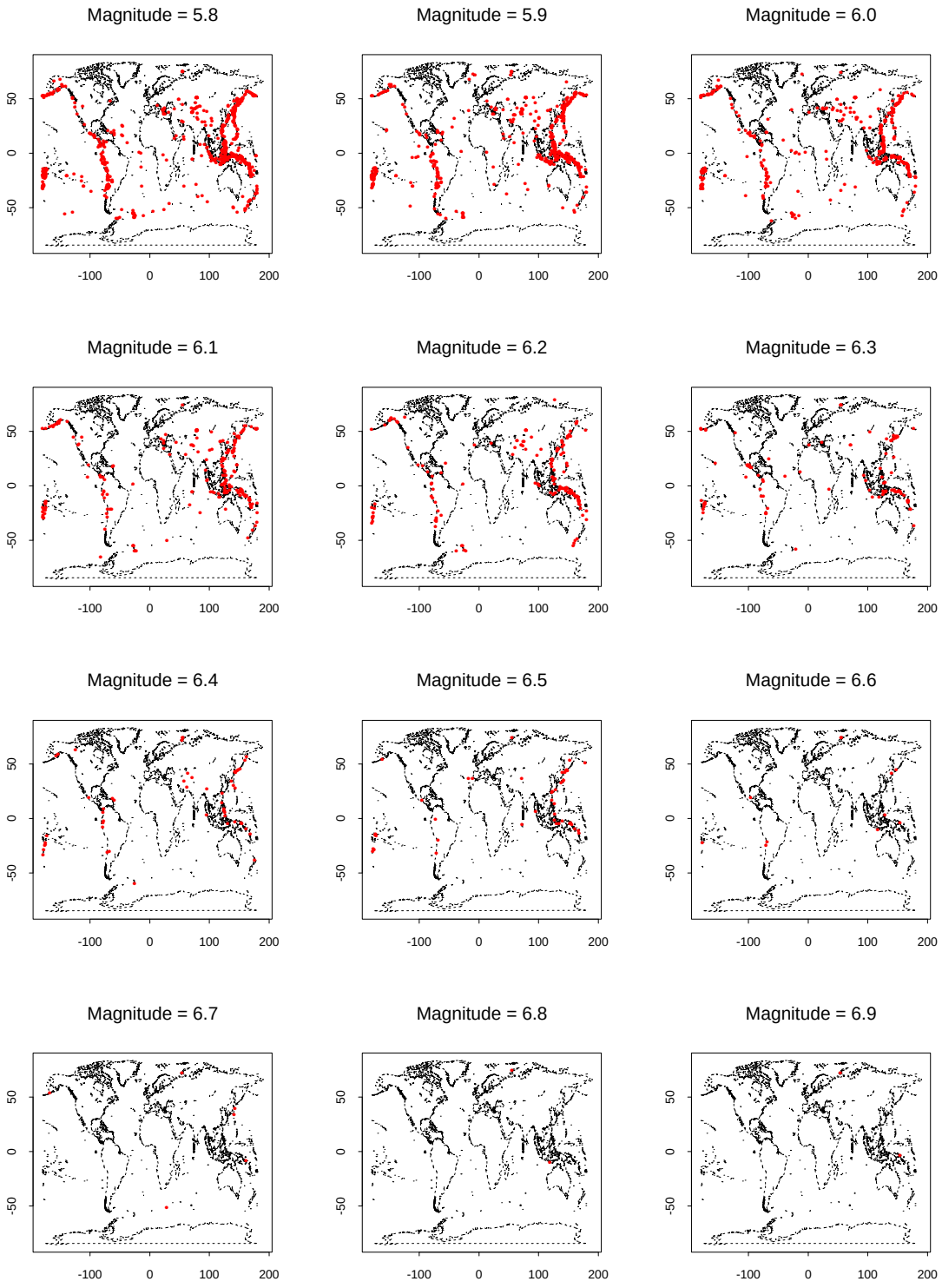


Figure 4.2: Data of earthquake occurrence (red), by magnitude.

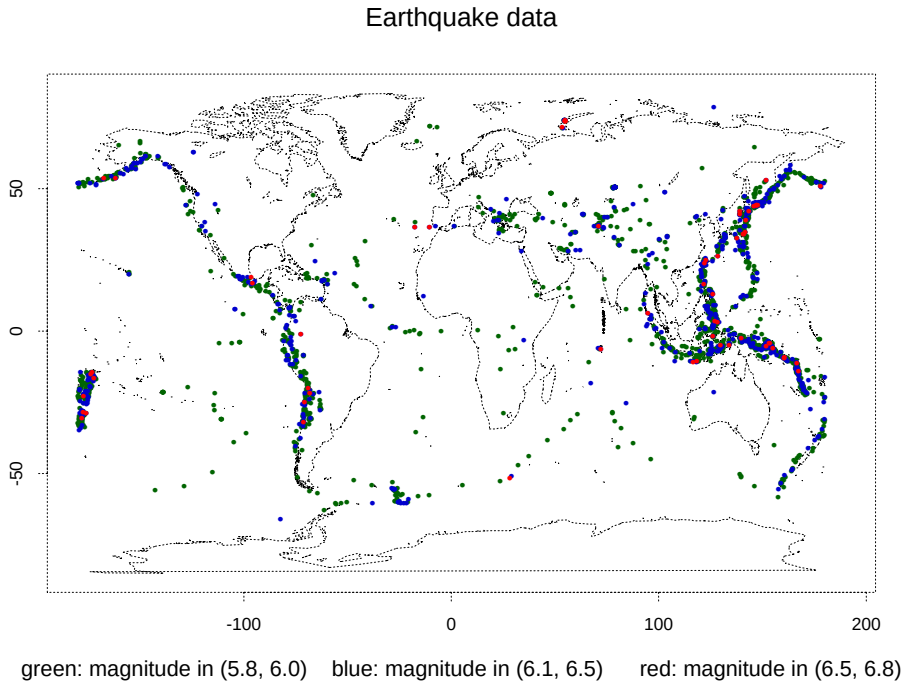


Figure 4.3: Data of earthquake occurrence, by range of magnitude.

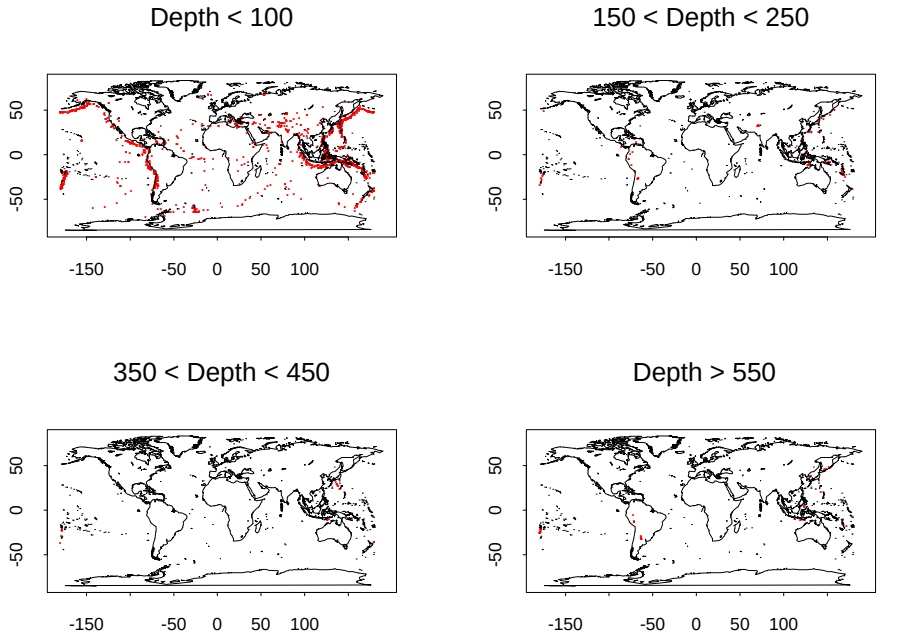


Figure 4.4: Data of earthquake occurrence, by range of depth.

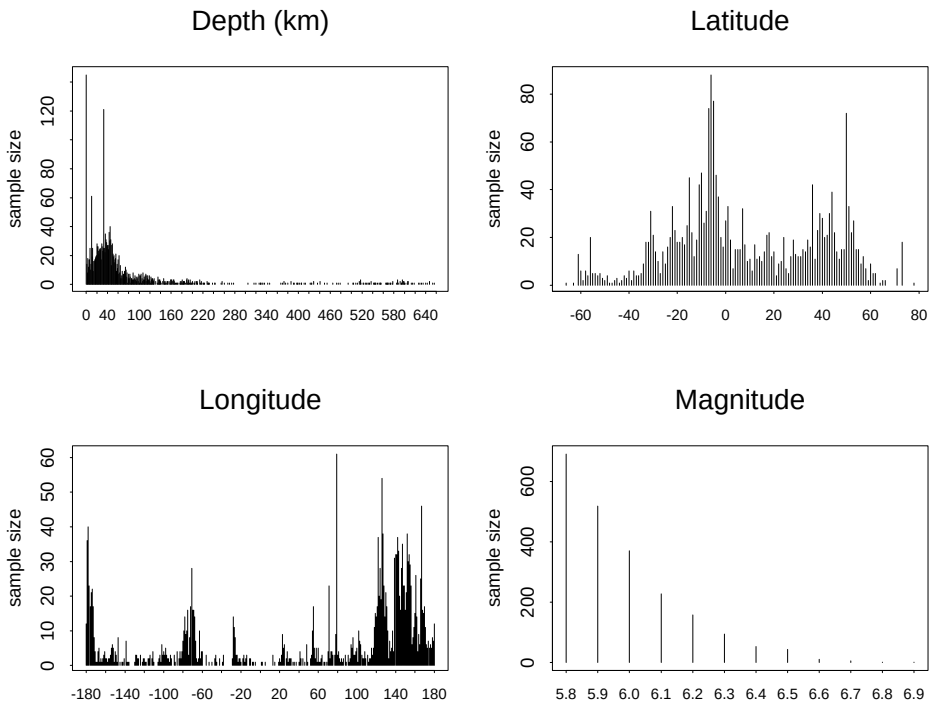


Figure 4.5: Marginal barplots of earthquake occurrence data.

updating the relatively flat prior. Thus, it seems that in these analyses, the problem of partition dependence is more pronounced in cases when there are few observations.

4.1.2 Conditional Predictive Distribution via the Randomized Tree

Several analyses using the randomized tree are presented. To assess the effect of various modeling assumptions on the randomized tree, parameter values of τ and G (the CDF by which data are transformed via G^{-1} from $\Re$ to $[0, 1]$) will be explored. In some analyses, the longitude variable is in the range $(-180, 180)$, and for other analyses, longitude is in $(-120, 240)$; the range of this variable is altered to see how sensitive our results are to 'edge effects,' *i.e.,* to the selection of the endpoints of the sample space. While edge effects are not a central focus of the research nor the motivation for this analysis, it is important, nevertheless, to recognize that this is an

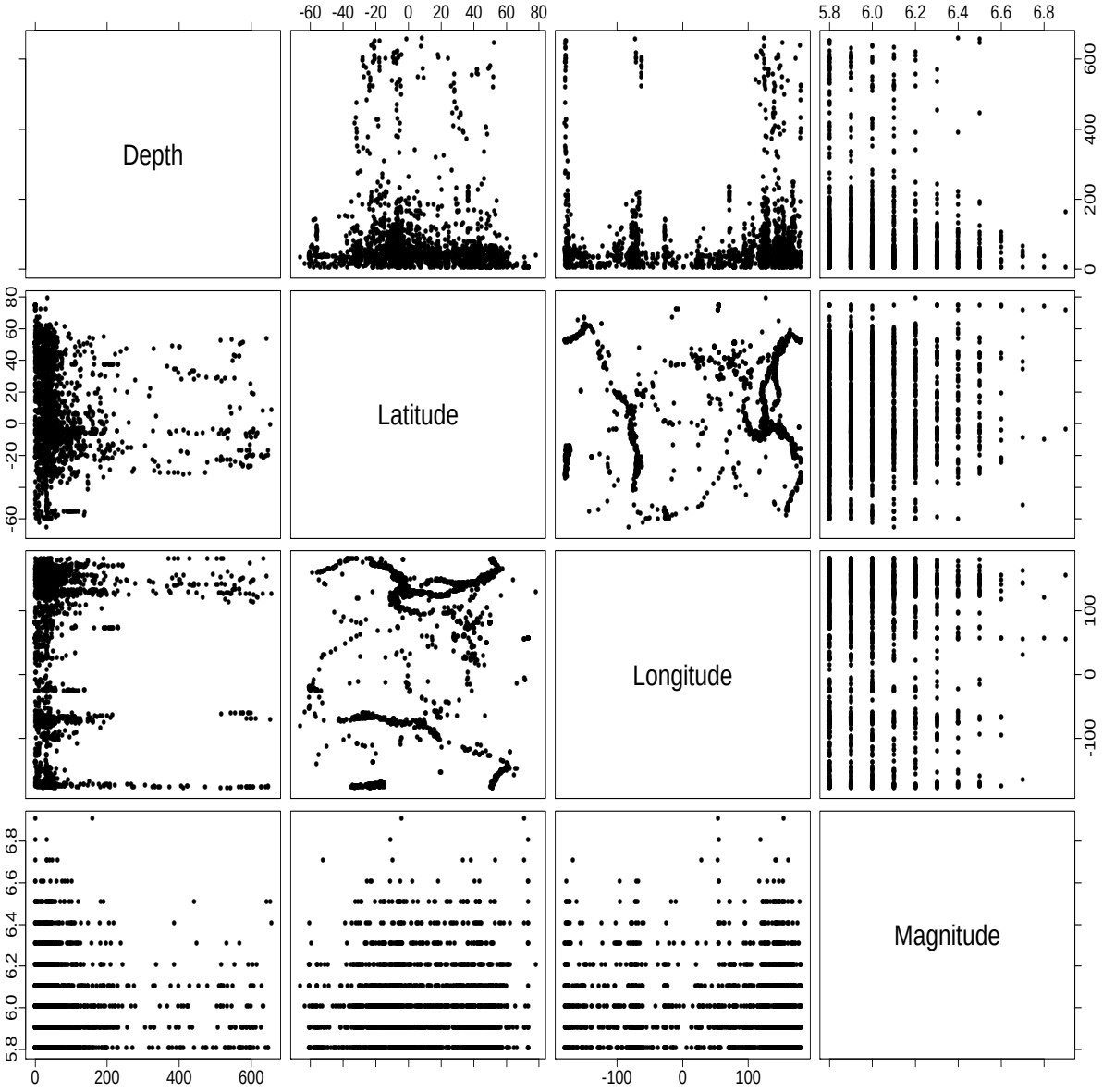


Figure 4.6: Pairwise plot of earthquake occurrence data.

issue. For the first few sets of analyses, the conditional predictive distribution of location given magnitude will be simulated, and then that of location given depth will be simulated. The joint predictive distributions of latitude, longitude, and magnitude, as well as the joint predictive distributions of latitude, longitude, and depth, are simulated using the conditional predictive simulation scheme developed in Chapter 2 and the randomized tree approach of Chapter 3.

The prior values of the Dirichlet parameters $\alpha \in \mathcal{A}^S$ at levels m of the tree are denoted in Table 4.1. In particular, $\alpha = cm^2$ for some $c > 0$. The scalar factor, c, controls how large the parameters are at the top of the tree; the particular choice of $c < 1$ allows for the data to more strongly influence the posterior distribution than would occur for larger scalar values such as $c \geq 1$, which can have a significant impact at low levels of the tree. Also of importance is the effect of the choice of parameters τ and G, where τ is as described in the randomized Polya tree formulation (Chapter 3). The choices of G to be examined here include G as the CDF of a Uniform and G as an empirical distribution function, based on a training sample of $n = 1178$ of the data, leaving $n = 1000$ observations for analysis; the empirical CDF is computed via routines `dgcdf` and `dgcin` from the IMSL Library (1984). A final concern, 'edge effects,' is assessed – does the choice of modeling longitude as going from -180 to 180 degrees or, for instance, from -120 to 240 degrees have an impact on the results of the analysis? As will be shown below, the choice of longitude range does not affect the results in any meaningful way.

All figures presented in the following discussion (Figures A.4 – A.21) are based upon 50000 draws from the posterior predictive randomized tree distributions, via the Markov chain Monte Carlo sampling scheme described in Chapter 3, Section 3.6.

Table 4.1 summarizes the analyses presented in this Section. The choices for τ, G, and the range of longitude are varied (Table 4.1). Predictions of location (latitude

Analysis	α	τ	G	range
1	$0.1m^2$	0.025	Uniform	(-180, 180)
2	$0.1m^2$	0.05	Uniform	(-180, 180)
3	$0.1m^2$	0.025	Empirical CDF	(-180, 180)
4	$0.1m^2$	0.05	Empirical CDF	(-180, 180)
5	$0.1m^2$	0.025	Uniform	(-120, 240)
6	$0.1m^2$	0.05	Uniform	(-120, 240)

Table 4.1: Summary of parameters for analyses in Section 4.1.2.

and longitude) given a fixed scalar value for magnitude are presented.

Comparison of $\tau = 0.025$ (Analysis 1) and $\tau = 0.05$ (Analysis 2)

Figures A.4–A.6 display the results of Analysis 1; prior parameter values for α are as specified above, while the hyperparameter τ is set to 0.025. Figures A.7–A.9 display the results of Analysis 2, for which $\tau = 0.05$. Analyses 1 and 2 can be compared to assess the effect of τ. The results are largely the same – as to be expected – though slight differences exist. Figures A.4 –A.6 and Figures A.7–A.9 show the conditional predictive distributions for location given magnitude. The same general patterns are detectable in all of the Figures, though the influence of the choice of τ is readily apparent by comparing, for example, Figures A.4 and A.7; the conditional predictive distribution diffuses mass about a larger area for $\tau = 0.05$ than for $\tau = 0.025$. Note the image plots and scatterplots of simulated subsamples for magnitude = 6.5 (Figures A.6 and A.9); the greater smoothing effect of $\tau = 0.05$ is clear in the latter figure, as the for regions of highest predictive probability (particular just above the equator on the far right–hand side of the graph), box-like patterns are more distinct. Another interesting feature is the relatively high predictive value in central Asia apparent in Analysis 1, but not as apparent in Analysis 2; while the data show earthquake activity in that region for the given magnitude (Figure 4.2), the mass is

more diffused about the region in Analysis 1 than in Analysis 2.

Analyses 3 and 4

Analyses 3 and 4 shed light on the effect of τ when G is the empirical CDF. As previously mentioned, $n = 1178$ data points are used as a "training set" to compute the empirical CDF, while 1000 points remain for analysis. Comparison of the conditional predictive distributions for Analysis 3 (Figures A.10 –A.12) and Analysis 4 (Figures A.13–A.12) shows that both analyses yield the same results, largely, though there is a bit more smoothing in the predictive distributions for $\tau = 0.05$ in Analysis 4 than for $\tau = 0.025$ in Analysis 3.

Comparison of Analyses 3 and 4 with Analyses 1 and 2

First, Analyses 1 and 3 ($\tau = 0.025$), and then Analyses 2 and 4 ($\tau = 0.05$), will be compared to understand the effect of the choice of G in the randomized tree framework for this analysis.

Comparison of Figures A.4–A.6 of the conditional predictive distributions for Analysis 1 with those of Analysis 3 (Figures A.10–A.12) reveals that the same general geographical areas are given high predictive probability in both analyses, though it seems that the choice of G as the Uniform CDF does better at magnitude = 5.8, for which there are more observations (Figure 4.2), and the choice of G as the empirical CDF does better for magnitude = 6.5, for which there are fewer observations (Figure 4.2). Most notably, in Figure A.4, the cluster of earthquakes between 0 and -50 latitude and at about -180 longitude on the far left hand side of the map is detected quite well, when compared to the actual data concentration (Figure 4.2). Also, in Figure A.12, the nuclear explosions at Novaya Zemlya are detected, which is the case but to a much lesser extent in Analysis 1.

Comparison of Analyses 2 and 4, in which $\tau = 0.05$, provide an identical story.

Comparison of Longitude $\in (-180, 180)$ and Longitude $\in (-120, 240)$

In order to explore how much a difference is made by the choice of the range of longitude on the results of the analysis, Analysis 1 is repeated, with the only difference being that longitude runs from -120 to 240 degrees here, rather than -180 to 180.

Comparison of Analyses 1 and 2 with 5 and 6

Figures A.4 – A.9, which correspond to Analyses 1 and 2, and Figures A.10–A.12, which correspond to Analyses 5 and 6 – demonstrate that Analyses 1 and 2, when compared to Analyses 5 and 6, yield basically identical results, with the minor differences in the predictive density of the region just to the northeast of New Zealand, on the far left of the graph, near longitude of -180. This could be due to the fact that the cells of that region are now neighbor cells with those on the far right of the graph, near Indonesia and Australia. Other differences include that earthquake activity in central Asia, near Afghanistan and southern Russia, is more strongly detected under the choice of G as the Uniform CDF – this is clear from comparing, for example, Figures A.4–A.5 and A.10–A.11.

Summary

From the comparison of the Analyses of Sections 4.1.1 and 4.1.2, it is evident that the randomized trees do indeed smooth the effect of the partition on the conditional predictive simulations as compared to the Polya tree prior. The analyses in Section 4.1.2 show that the choice of τ makes a noticeable difference in how the predictive distributions are smoothed, while in this case, the choices of G do not have much effect. The 'edge effects' do not appear to be very pronounced in these analyses.

4.1.3 Conditional Predictive Distribution of Location Given Depth

For the next set of analyses, the conditional predictive distributions of earthquake location (latitude and longitude) given depth are simulated. Four analyses are completed; for each analysis $\tau = 0.025$ (Figures A.22–A.25) or $\tau = 0.05$ (Figures A.26–A.29), $\alpha = 0.1(level)^2$, and G is the CDF of the uniform distribution. Location is predicted for depths of $50, 200, 400$ and 600 km. Simonoff (1996) uses a Gaussian-based kernel density estimator to compute the conditional density of earthquake location given depth values of $50, 200, 400$, and 600 km; he uses the approach to illustrate how to explore a three-dimensional density by examining slices (based on conditioning) of it, which is similar to what is done with the conditional predictive simulations. Figure 4.4 shows scatterplots of the earthquakes given various ranges of depth. It is clear from this Figure, as well as from Figure 4.5, that most observations have a depth of less than 100 km, and that observations of depths greater than 100km decrease in frequency of occurrence.

Figures A.22–A.29 display the conditional predictive simulations. The overall pattern of location given depth resembles that of location given magnitude of the previous analyses. The simulated predictive distributions all appear to be quite reasonable given the data (Figure 4.4). As depth increases, the amount of data/information available decreases; the resulting increase in uncertainty is clear in comparing Figures A.22–A.29 and assessing the spread of the distribution; the conditional predictive distributions simulated for Figures A.22–A.23 are less diffuse than those for depths of 400 and 600km (Figures A.24–A.25).

In comparison to the conditional density estimates of Simonoff (1996), the overall patterns of the conditional predictive simulations are in agreement. The most interesting of the differences occurs for depths of 400 and 600 km, where data are

sparse. The randomized tree conditional predictive simulation and the Gaussian-based kernel density estimates are picking up slightly different regions of importance; at 400km, the randomized tree conditional predictive simulation basically detects little to practically no earthquake activity in South America, while the density estimator of Simonoff (1996) places a kernel over South America, indicating some activity of interest is occurring there. At 600km, Simonoff (1996)'s method seems to suggest there is greater activity around Japan than our method does; while there are simulated points in the region of Japan (Figure A.25) would our method does not indicate as much activity.

Similar results hold for Figures A.26–A.29 as for the above analyses, only this time $\tau = 0.05$, which induces more smoothing into the partition.

4.1.4 Convergence of Markov Chain Monte Carlo Simulation

Assessment of the convergence of the Markov chain Monte Carlo output are detailed in Section 3.7.1 of Chapter 3. A description of the diagnostics used in this Section is detailed there as well. Examination of MCMC output for a small example was explored to gain some understanding about aspects of convergence. For the much-larger data analyses presented in this Chapter, it is unfeasible to analyze all of the trajectories for the collection of $\{\beta_i\}_{i=1}^n$, as n is fairly large for these purposes. However, one β vector for one observation is examined, corresponding to the first data point in the data set (which was not sorted in any special order), and an analysis of convergence for one of the analyses (Analysis 1 of Section 4.1.2, in which $\tau = 0.025$ and G =Uniform CDF) is presented; its convergence behavior should be representative of all of the β_i. A thinned chain of 5000 of 50000 iterations is analyzed below. The acceptance rates for the β_i which are sampled by an independence Metropolis step are about 82% for most (over 50%) of the β_i samples. The minimum acceptance rate

from all analyses is 17% and the maximum acceptance rate is 83%; for the analyses in which G is the empirical CDF, the minimum acceptance rate is higher, 39%.

Just as in Section 3.7.1, autocorrelations, cross-correlations, visual inspection of convergence plots, and some diagnostics from the BOA package (Smith, 1999) are examined.

Table 4.1.4 displays the autocorrelations of the Markov chains for the β parameters. The columns in the Table correspond to β's for latitude, longitude and magnitude, respectively. As is clear, the autocorrelations are basically zero. Tables 4.1.4–4.1.4 displays the cross-correlations amongst all 30 β parameters (10 for each of the three axes of the three-dimensional Euclidean space). Just as for the autocorrelations, these correlations are effectively zero as well. The lack of auto- and cross-correlations is an encouraging sign that the Markov chain is searching all interesting areas of the sample space without getting "stuck" in modes due to high dependence of the MCMC samples.

Figures 4.10–4.12 show the trajectory plots of the β's; 2000 of the 5000 saved iterations are plotted here. None of the trajectory plots exhibit irregular patterns such as the chain getting "stuck" in any part of the space, and by all appearances nothing indicates that convergence has yet to be achieved.

Table 4.1.4 shows the results of the Brooks-Gelman-Rubin diagnostic. All estimates of the corrected scale reduction factors are basically equal to 1, which is evidence in support of convergence. The multivariate potential scale reduction factor is 1.010956.

The Heidelberger-Welch diagnostic is applied to two Markov chains of the same analysis, each chain was run with different starting values and different seeds. All chains passed the stationarity test, with the test suggesting that only two chains have the first 500 or 1000 samples removed. The dependence factors of the Raftery-Lewis

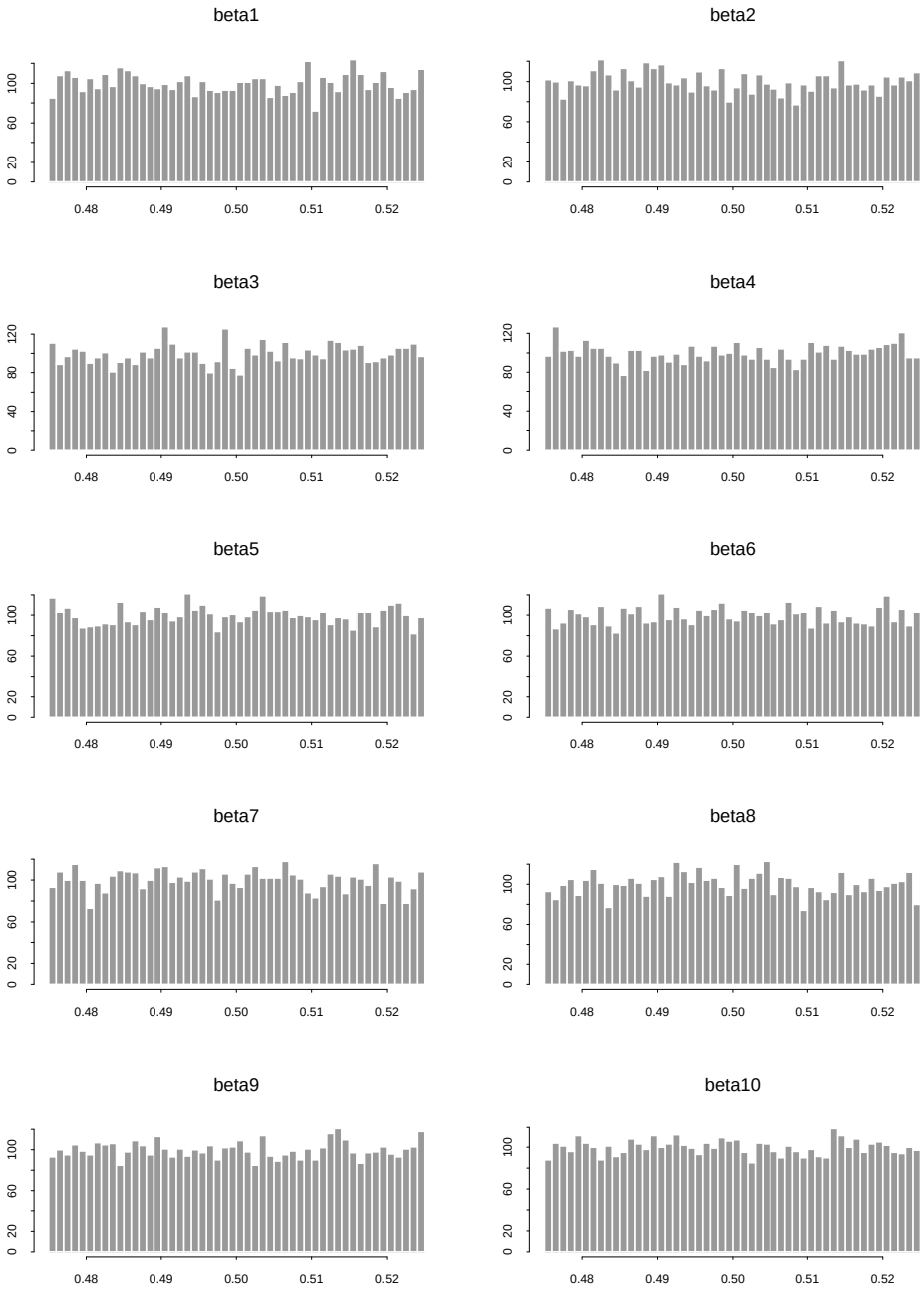


Figure 4.7: Histograms of MCMC samples for $\beta_1, \ldots, \beta_{10}$ (for the latitude axis) for the earthquake data analysis. These results are for Analysis 1, Section 4.1.2.

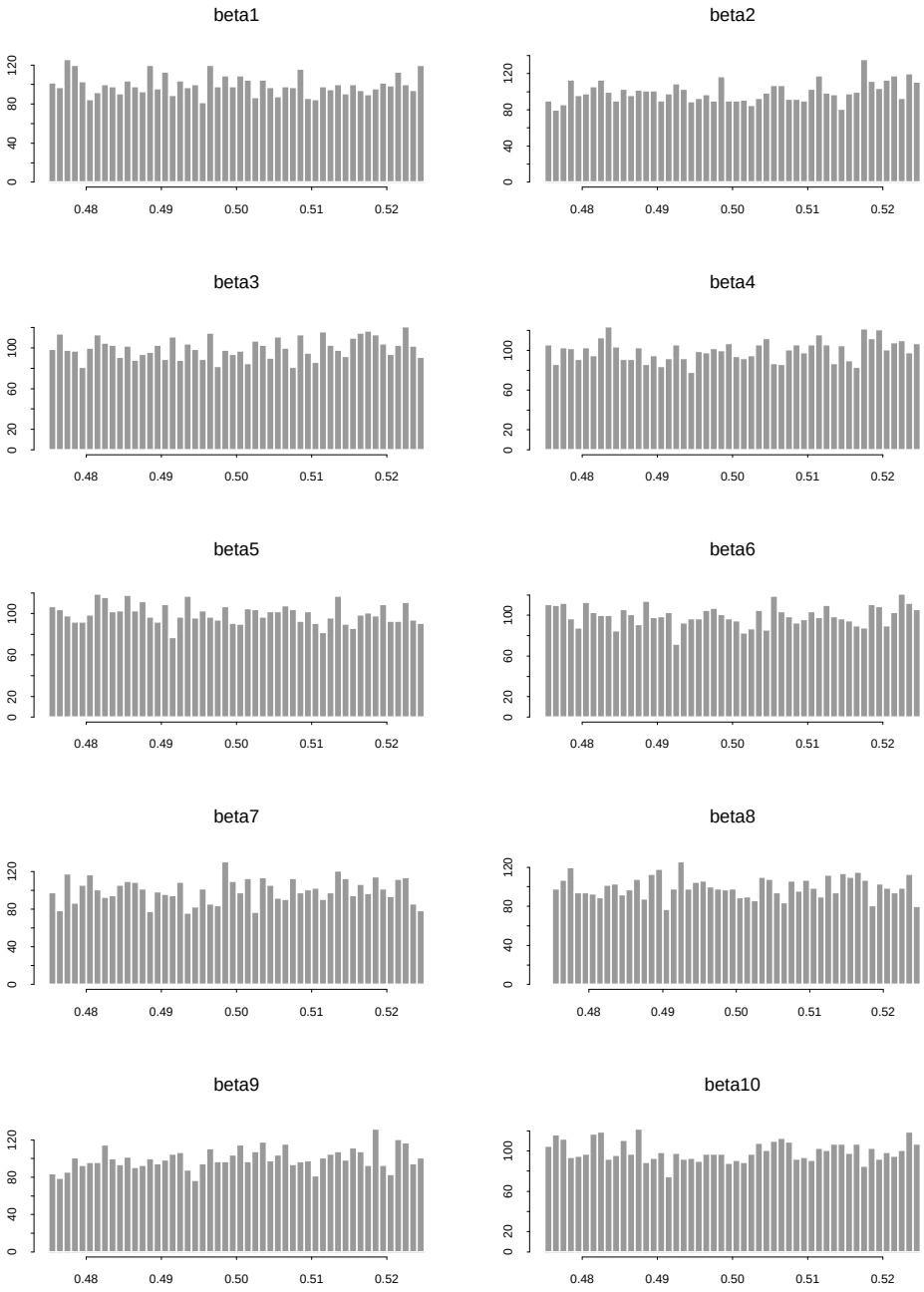


Figure 4.8: Histograms of MCMC samples for $\beta_1, \ldots, \beta_{10}$ (for the longitude axis) for the earthquake data analysis. These results are for Analysis 1, Section 4.1.2.

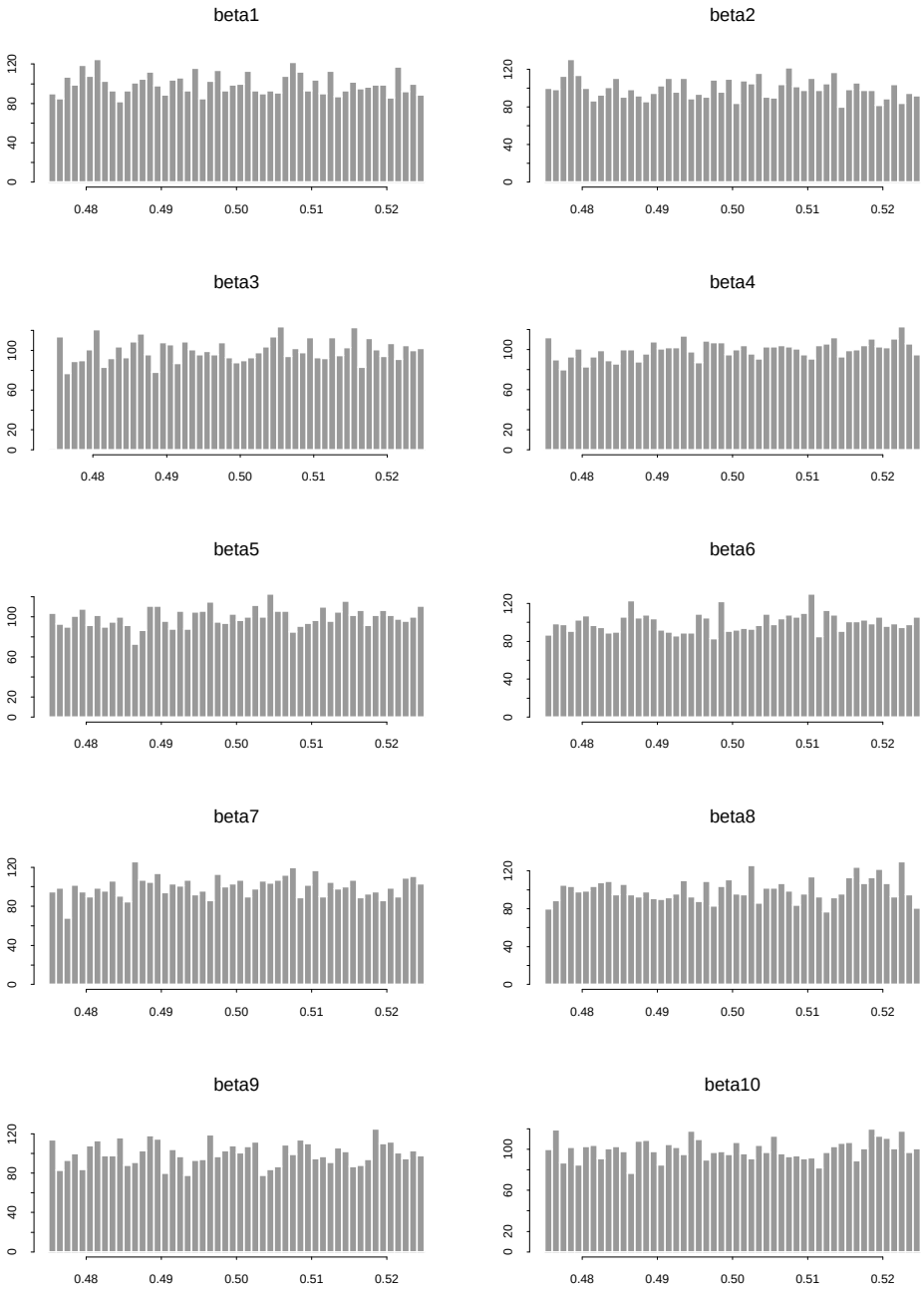


Figure 4.9: Histograms of MCMC samples for $\beta_1, \ldots, \beta_{10}$ (for the magnitude axis) for the earthquake data analysis. These results are for Analysis 1, Section 4.1.2.

	Lag 1	Lag 5	Lag 10	Lag 50
β_1	-1.13142e-02	-0.03863469	-0.00975542	0.020467807
β_2	1.41549e-02	0.02030589	0.00840857	0.012242291
β_3	-3.71350e-03	0.00462421	0.00266140	0.035743244
β_4	-1.79424e-02	-0.00871108	0.01854724	0.001592195
β_5	-1.48199e-02	0.01708816	-0.01751694	-0.021665938
β_6	-4.05211e-03	-0.01777273	-0.00689437	-0.008048644
β_7	2.52074e-03	0.00760041	0.00617219	-0.004995068
β_8	-7.99948e-03	-0.01800073	0.00617481	-0.010316793
β_9	8.30622e-05	-0.00600558	0.00915495	0.010491826
β_{10}	3.59739e-03	-0.00158121	0.00606918	-0.006096745
β_1	6.92923e-03	0.00695738	-0.01634004	-0.006289048
β_2	-4.92650e-03	0.02203083	0.03408827	0.025952516
β_3	6.97330e-03	0.01930547	-0.01922509	0.000826308
β_4	-6.00804e-03	-0.00876716	-0.02560697	0.010038770
β_5	1.34376e-02	-0.02015076	0.01639961	-0.000584564
β_6	-1.14011e-02	-0.00173774	0.00114775	-0.026821110
β_7	5.71071e-03	0.01907040	0.01919348	0.002502904
β_8	-1.62931e-02	-0.01476088	-0.00236415	-0.007824883
β_9	-5.32809e-03	-0.00798255	0.00332071	-0.006349508
β_{10}	2.36312e-03	0.00116208	0.00360539	0.028556559
β_1	1.74953e-02	-0.00961176	0.01499458	-0.015670234
β_2	-7.43450e-03	0.00912050	0.01500857	0.011408128
β_3	-1.49268e-02	-0.02485460	-0.01111205	0.006888865
β_4	-2.55243e-03	-0.00736635	-0.03866056	0.005524388
β_5	9.52371e-03	0.00273105	0.01325478	0.004738789
β_6	2.45128e-02	0.01206142	0.00976642	-0.012927135
β_7	-3.84964e-03	0.00310988	-0.02323175	-0.005247168
β_8	2.90632e-02	-0.01256916	0.00201675	0.005577800
β_9	-1.27914e-02	-0.00382356	0.01807854	0.011212721
β_{10}	6.47347e-03	-0.02739488	-0.03380559	-0.028972993

Table 4.2: Autocorrelations of the simulated trajectories at various lags. The first set of 10 β's corresponds to the latitude axis, the second set corresponds to the longitude axis, and the third set corresponds to the magnitude axis.

	β_1	β_2	β_3	β_4	β_5	β_6	β_7
β_1	1						
β_2	-0.0064	1					
β_3	-0.0154	0.0107	1				
β_4	-0.0160	-0.0087	-0.0078	1			
β_5	0.0073	-0.0043	-0.0275	0.0128	1		
β_6	0.0013	-0.0230	0.0042	-0.0117	0.0003	1	
β_7	0.0180	0.0239	-0.0013	0.0106	0.0012	-0.0173	1
β_8	-0.0158	0.0094	0.0199	0.0004	0.0191	0.0036	-0.0285
β_9	0.0115	0.0080	-0.0130	0.0026	0.0055	0.0095	-0.0154
β_{10}	0.0067	-0.0120	0.0188	0.0091	0.0081	-0.0116	-0.0025
β_{11}	-0.0235	0.0133	-0.0038	-0.0067	0.0039	0.0059	-0.0011
β_{12}	-0.0194	-0.0160	-0.0013	-0.0097	-0.0103	-0.0056	0.0158
β_{13}	0.0292	0.0171	-0.0241	-0.0096	0.0089	0.0011	-0.0060
β_{14}	-0.0188	0.0048	-0.0150	-0.0233	-0.0040	-0.0063	-0.0066
β_{15}	-0.0155	0.0056	-0.0009	0.0080	0.0134	0.0092	0.0024
β_{16}	-0.0111	0.0211	0.0222	0.0014	-0.0158	0.0038	-0.0054
β_{17}	0.0037	0.0195	-0.0029	0.0064	0.0038	-0.0207	-0.0218
β_{18}	0.0010	-0.0127	-0.0129	0.0047	-0.0223	-0.0029	0.0069
β_{19}	0.0230	-0.0084	-0.0042	-0.0186	0.0235	-0.0115	0.0281
β_{20}	0.0105	-0.0061	-0.0049	-0.0051	-0.0133	0.0010	0.0078
β_{21}	0.0002	0.0103	-1.9e-0	0.0179	-0.0049	0.0261	0.0257
β_{22}	0.0151	-0.0114	-0.0025	-0.0114	0.0118	0.0012	-0.0239
β_{23}	-0.0099	-0.0252	0.0135	-0.0059	0.0075	0.0043	-0.0183
β_{24}	0.0058	0.0136	-0.0069	0.0070	0.0127	-0.0057	-0.0218
β_{25}	0.0065	0.0030	-0.0088	0.0086	0.0066	0.0087	-0.0071
β_{26}	0.0121	-0.0036	-0.0026	-0.0126	0.0112	-0.0029	-0.0098
β_{27}	-0.0169	0.0049	0.0007	0.0034	0.0024	0.0090	-0.0155
β_{28}	-0.0152	0.0157	0.0010	0.0097	0.0040	0.0129	0.0062
β_{29}	-0.0129	0.0094	0.0102	-0.0169	-0.0180	-0.0172	0.0162
β_{30}	-0.0013	-0.0008	-0.0068	0.0108	-0.0101	-0.0133	0.0015

Table 4.3: Table 1 of 3: Cross-correlations of the MCMC samples for β. $\beta_1 - \beta_{10}$ correspond to the latitude axis, $\beta_{11} - \beta_{20}$ correspond to the longitude axis, and $\beta_{21} - \beta_{30}$ correspond to the magnitude axis.

	β_8	β_9	β_{10}	β_{11}	β_{12}	β_{13}	β_{14}
β_{08}	1						
β_{09}	-0.0261	1					
β_{10}	-0.0035	-0.0346	1				
β_{11}	0.0111	0.0164	0.0210	1			
β_{12}	0.0295	0.0078	0.0048	-0.0090	1		
β_{13}	-0.0075	0.0234	-0.0208	0.0095	0.0037	1	
β_{14}	0.0070	0.0075	-0.0131	-0.0080	0.0121	0.0039	1
β_{15}	0.0230	0.0003	-0.0371	-0.0067	0.0276	0.0141	0.0053
β_{16}	-0.0060	0.0097	-0.0004	0.0109	-0.0208	-0.0075	0.0122
β_{17}	0.0139	-0.0046	0.0097	0.0005	0.0197	0.0217	-0.0005
β_{18}	-0.0134	-0.0127	0.0159	0.0093	0.0124	0.0083	-0.0159
β_{19}	0.0162	-0.0014	-0.0281	0.0142	-0.0248	0.0057	-0.0176
β_{20}	-0.0080	0.0370	0.0046	0.0110	0.0115	-0.0097	0.0057
β_{21}	-0.0010	0.0194	-0.0113	0.0018	-0.0031	0.0217	0.0007
β_{22}	0.0302	-0.0047	0.0109	-0.0046	-0.0271	-0.0053	0.0143
β_{23}	0.0212	-0.0013	0.0160	-0.0068	0.0046	-0.0066	-0.0009
β_{24}	0.0050	-0.0150	-0.0186	-0.0157	-0.0032	-0.0060	-0.0173
β_{25}	-0.0029	0.0098	-0.0120	-0.0009	-0.0043	-0.0409	-0.0173
β_{26}	-0.0176	-0.0173	0.0059	0.0060	-0.0158	-0.0068	0.0107
β_{27}	0.0131	0.0218	0.0169	0.0230	-0.0028	0.0064	0.0270
β_{28}	0.0240	-0.0201	-0.0054	-0.0076	0.0154	0.0202	-0.0186
β_{29}	-0.0008	0.0042	0.0065	-0.0029	-0.0179	0.0031	0.0141
β_{30}	0.0025	0.0018	0.0013	0.0152	-0.0067	-0.0091	0.0387

Table 4.4: Table 2 of 3: Cross-correlations of the MCMC samples for β. Cross-correlations of the MCMC samples for β. $\beta_1 - \beta_{10}$ correspond to the latitude axis, $\beta_{11} - \beta_{20}$ correspond to the longitude axis, and $\beta_{21} - \beta_{30}$ correspond to the magnitude axis.

	β_{15}	β_{16}	β_{17}	β_{18}	β_{19}	β_{20}	β_{21}
β_{15}	1						
β_{16}	-0.0014	1					
β_{17}	0.0033	-0.0025	1				
β_{18}	-0.0076	-0.0430	0.0033	1			
β_{19}	-0.0087	-0.0153	0.0067	0.0558	1		
β_{20}	0.0377	0.0039	0.0029	0.0169	0.1579	1	
β_{21}	-0.0094	0.0076	0.0105	-0.0211	0.0235	0.0092	1
β_{22}	0.0290	0.0101	-0.0005	0.0219	-0.0435	0.0105	0.0023
β_{23}	0.0015	-0.0205	-0.0073	-0.0118	-0.0255	-0.0087	-0.0113
β_{24}	-0.0026	0.0189	0.0011	-0.0009	0.0109	0.0079	0.0128
β_{25}	-0.0108	0.0111	0.0077	-0.0112	0.0013	-0.0033	-0.0017
β_{26}	0.0034	-0.0141	-0.0127	0.0078	-0.0167	-0.0044	-0.0061
β_{27}	-0.0051	-0.0206	0.0054	-0.0031	-0.0008	-0.0276	0.0169
β_{28}	0.0145	-0.0132	0.0018	0.0022	0.0154	0.0222	0.0044
β_{29}	-0.0205	-0.0001	0.0017	0.0146	0.0156	0.0091	-0.0078
β_{30}	-0.0152	-0.0157	-0.0066	0.0123	0.0185	-0.0189	0.0096

	β_{22}	β_{23}	β_{24}	β_{25}	β_{26}	β_{27}	β_{28}
β_{22}	1						
β_{23}	-0.0126	1					
β_{24}	-0.0372	-0.0185	1				
β_{25}	-0.0110	-0.0032	-0.0353	1			
β_{26}	-0.0065	-0.0041	0.0226	0.0024	1		
β_{27}	-0.0089	0.0163	-0.0109	-0.0134	-0.0026	1	
β_{28}	0.0130	0.0101	0.0041	-0.0099	0.0020	-0.0218	1
β_{29}	0.0232	-0.0112	0.0173	-0.0082	0.0093	-0.0051	-0.0227
β_{30}	-0.0016	-0.0204	-0.0137	-0.0262	-0.0075	-0.0097	0.0010

	β_{29}	β_{30}
β_{29}	1	
β_{30}	-0.0011	1

Table 4.5: Table 3 of 3: Cross-correlations of the MCMC samples for β. Cross-correlations of the MCMC samples for β. $\beta_1 - \beta_{10}$ correspond to the latitude axis, $\beta_{11} - \beta_{20}$ correspond to the longitude axis, and $\beta_{21} - \beta_{30}$ correspond to the magnitude axis.

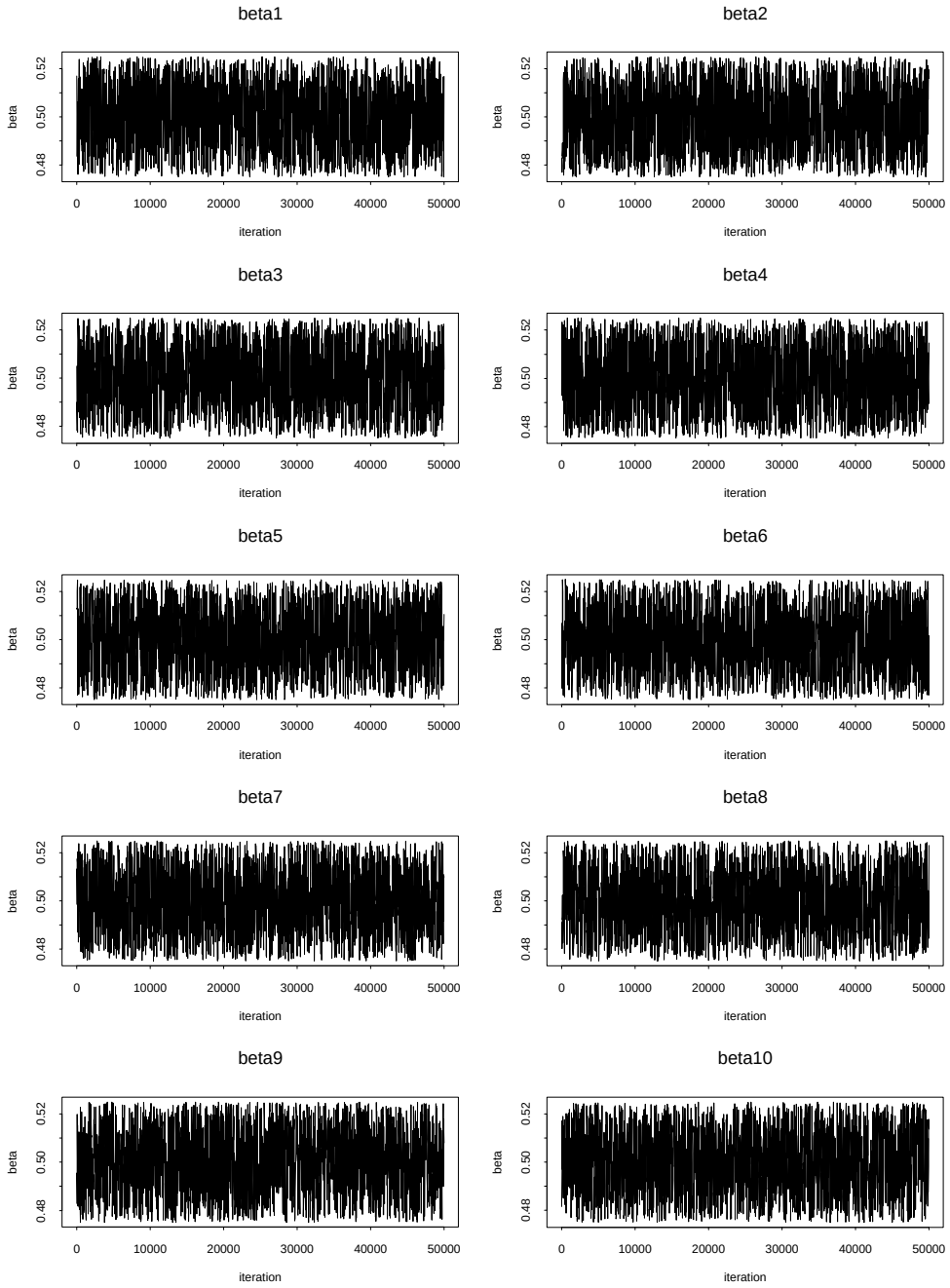


Figure 4.10: MCMC trajectories for $\beta_1, \ldots, \beta_{10}$ (corresponding to the axis for latitude) for the earthquake data analysis. These results are for Analysis 1, Section 4.1.2.

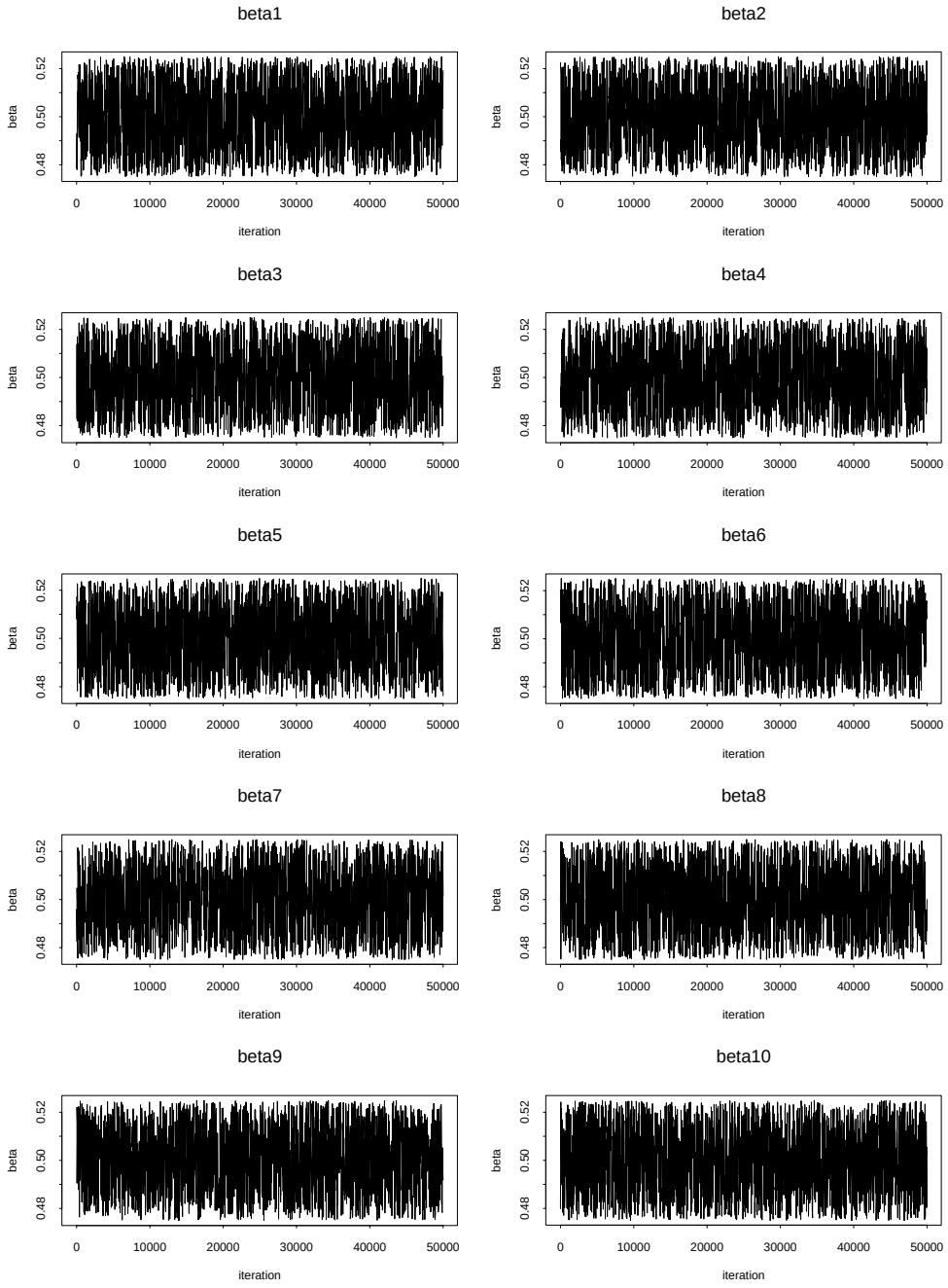


Figure 4.11: MCMC trajectories for $\beta_1, \ldots, \beta_{10}$ (corresponding to the axis for latitude) for the earthquake data analysis. These results are for Analysis 1, Section 4.1.2.

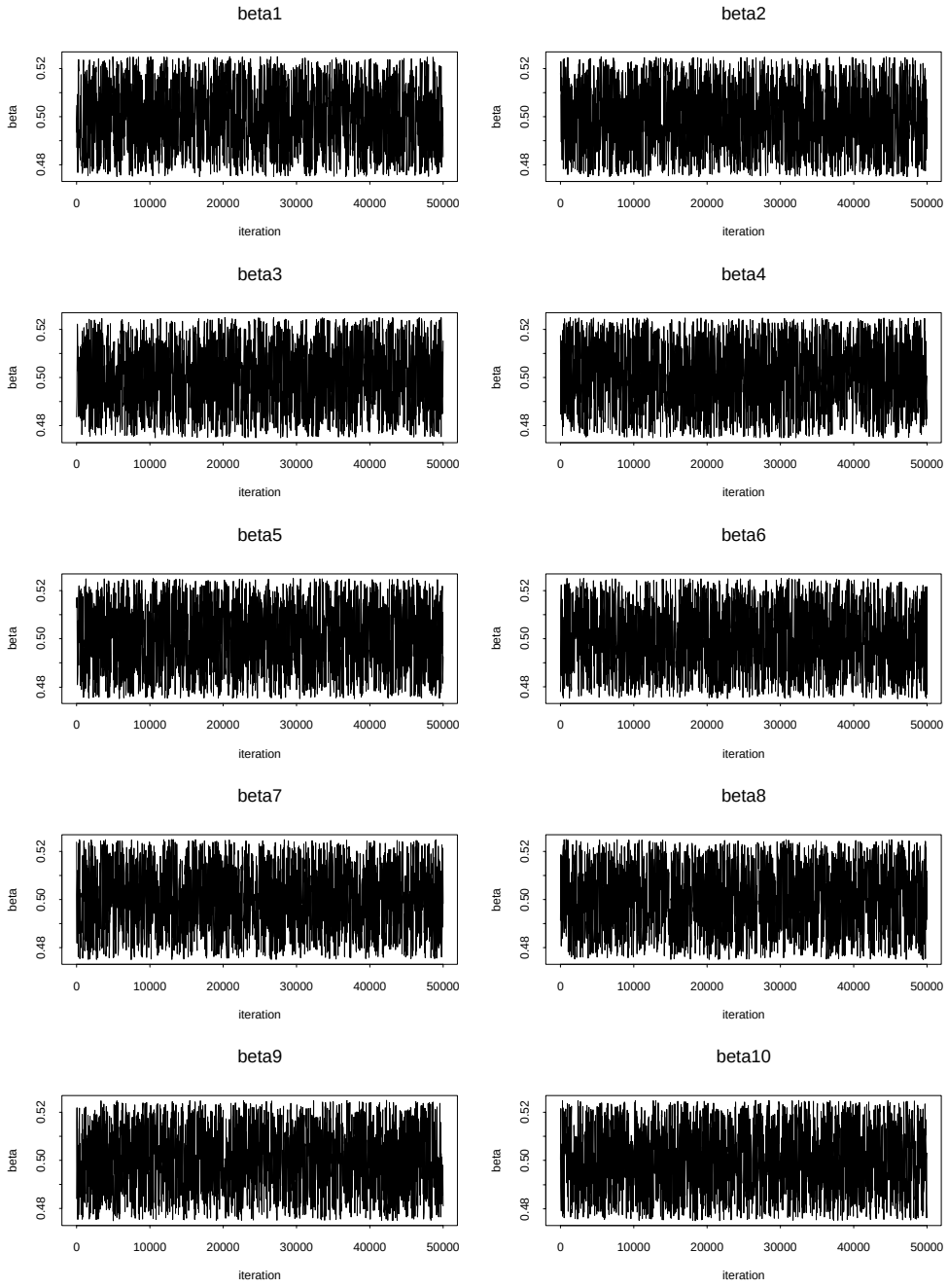


Figure 4.12: MCMC trajectories for $\beta_1, \ldots, \beta_{10}$ (corresponding to the axis for latitude) for the earthquake data analysis. These results are for Analysis 1, Section 4.1.2.

Latitude		Longitude		Magnitude	
Parameter	Estimate	Parameter	Estimate	Parameter	Estimate
β_1	0.9999	β_1	0.9999	β_1	1.0000
β_2	0.9999	β_2	1.0007	β_2	1.0019
β_3	1.0012	β_3	1.0017	β_3	1.0000
β_4	1.0001	β_4	1.0000	β_4	1.0004
β_5	0.9999	β_5	0.9999	β_5	0.9999
β_6	1.0006	β_6	0.9998	β_6	0.9998
β_7	0.9998	β_7	0.9998	β_7	1.0000
β_8	0.9998	β_8	1.0001	β_8	1.0005
β_9	1.0010	β_9	1.0012	β_9	1.0000
β_{10}	0.9998	β_{10}	1.0001	β_{10}	0.9998

Table 4.6: BOA Output — Corrected Scale Reduction Factors from the Brooks, Gelman and Rubin diagnostic. These results are for Analysis 1, Section 4.1.2.

diagnostic ranged from 0.943940 to 1.066471, which again indicate that nothing that would suggest lack of convergence is being detected. Overall, the analysis of the MCMC output supports convergence.

4.2 Imputation of Missing Data: 1993-1994 College Tuition Data

The purpose of this analysis is to illustrate output from an analysis using randomized trees in which missing data are imputed. The source for the data set used in this analysis is StatLib:

`http://www.stat.cmu.edu/datasets`

The data set consists of data on tuition and faculty compensation for U.S. colleges and universities which come from two sources — U.S. News & World Report's Guide to Americas Best Colleges and the American Association of University Professors (AAUP) 1994 Salary Survey which appeared in the March-April 1994 issue of

Academe. These data sets were the focus of the 1995 American Statistical Association Statistical Graphics Exposition Contest. $n = 1283$ colleges and universities are listed in the data set. While the data set contains several variables, just two, tuition and faculty compensation, are focused on here for ease of exposition. The variable, faculty compensation, is missing for 162 of these observations. Figure B.1 shows the marginal histograms of tuition and faculty compensation, and Figure B.2 shows a scatterplot of the joint distribution of the variables, omitting missing observations. As is clear from the graph, there is a bimodal distribution for tuition and faculty compensation, due to a mixture of public and private universities, which are both in the sample.

50000 simulations were drawn via Gibbs sampling from the joint posterior predictive distribution of a randomized Polya tree updated to 10 levels; the hyperparameter τ is set to 0.05, while the prior Dirichlet parameters at level m are set to $0.1m^2$. The 162 missing values for faculty compensation are imputed as described in Section 2.5 of Chapter 2. Figures B.2 show a two-dimensional histogram of the simulated values for tuition and faculty compensation, as well as a scatterplot of 5000 sub-sampled simulated values. Figure B.5 shows some replications of the missing data across all observations. The histograms of the replications pictured there are roughly of the same shape, and are in agreement with the histogram of the faculty compensation data in Figure B.1. Figure B.6 displays histograms of a sample of the 162 simulated posterior distributions for the missing faculty compensation observations; the title on each histogram gives the observed value of tuition for that observation. As the two variables are fairly correlated, which is clear from the scatterplot of the data in Figure B.2, it would be expected that as tuition increased, the posterior distribution for faculty compensation should be centered at higher values. Examination of the imputed values for faculty compensation (Figure B.6) supports this expectation.

4.2.1 Convergence of Markov Chain Monte Carlo Simulation

Figures 4.13–4.14 display the trajectories of β_1 from the missing data analysis. From all appearances, the trajectories appear to be moving about reasonably well throughout the space, and do not appear to get stuck at local modes nor do the chains appear to be mixing slowly, which would warrant consideration of the possibility of non-convergence. The acceptance rates of all of the $\{\beta_i\}_{i=1}^n$ range from $30-83\%$, with practically all β_i's having acceptance rates around $70-80\%$. Convergence diagnostics were applied to the MCMC trajectories from one analysis. On the Heidelberger-Welch stationarity test, all trajectories passed. The Raftery-Lewis dependence factors are all close to 1.0, which suggests convergence. The auto- and cross-correlations were all basically zero, which is a good sign as well. Trajectories were also examined for the imputed missing data values; trajectories looked normal, and the diagnostics did not indicate any serious problems.

4.3 Computational Time

Table 4.7 shows the computational time for several example analyses of the earthquake data. This Table illustrates how long the posterior predictive simulation and conditional predictive simulations take for a DEC Personal Workstation (433 MHz). Results for implementing randomized trees of five different levels – levels 6, 8, 10, 12, and 15 – are presented, and the number of observations is varied – for some analyses, $n = 2178$ and for others, $n = 1089$.

Because running time is linear with respect to number of iterations, the per second CPU time is displayed. It is clear that the bulk of the computational effort is due to the conditional predictive simulation, and that computational time depends greatly on how many levels there are in the tree; for the analysis with six levels, the computational times in both the right hand and left-hand columns are basically the

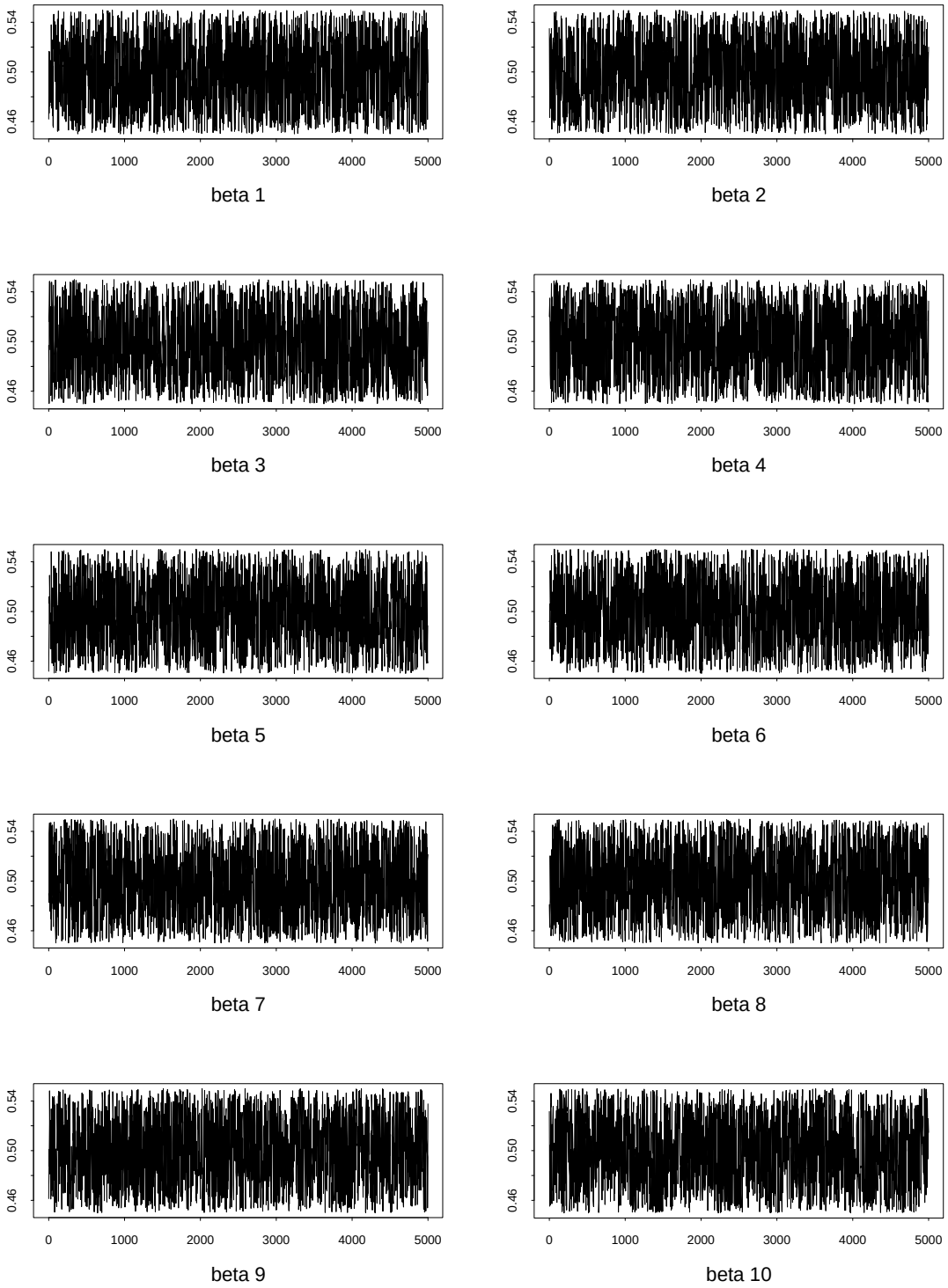


Figure 4.13: MCMC trajectories for $\beta_1, \ldots, \beta_{10}$ (corresponding to the axis for tuition) for the college and university data analysis.

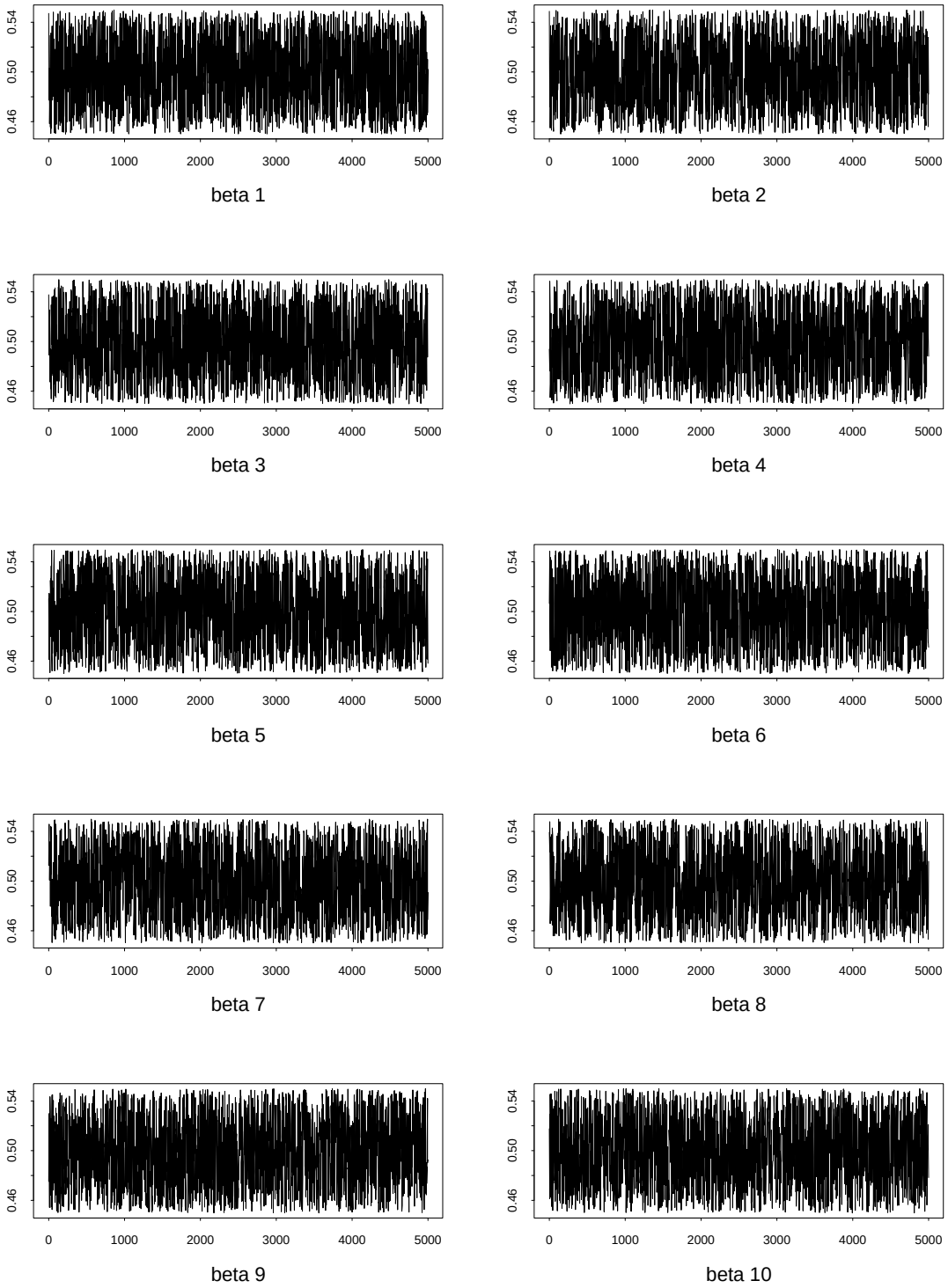


Figure 4.14: MCMC trajectories for $\beta_1, \ldots, \beta_{10}$ (corresponding to the axis for faculty compensation) for the college and university data analysis.

	Posterior predictive simulation only		With three conditional predictive simulations	
levels	n=2178	n=1089	n=2178	n=1089
6	1.33	0.68	1.35	0.66
8	1.80	1.06	2.11	1.14
10	2.27	1.17	5.31	4.19
12	3.02	1.48	51.30	50.20
15	3.66	1.88	3256.00	3230.00

Table 4.7: Per iteration CPU time (seconds) on a DEC Personal Workstation (433MHz).

same, while for fifteen levels, the computational time for the conditional predictive simulation increases dramatically. The analyses presented in this Chapter correspond to the case with a tree of ten levels, and $n = 2178$ observations. The per iteration CPU time is 5.31 seconds for these analyses. Clearly, the future work on computational time will pertain to cutting back the time required of the conditional predictive simulation scheme.

Chapter 5

SUMMARY AND EXTENSIONS

5.1 Summary

The goal of this research is to explore nonparametric modeling with Polya trees in multidimensional Euclidean space. This dissertation achieves this goal by applying the Polya tree framework to multidimensional Euclidean spaces, and by developing randomized trees as an alternative to Polya trees. This is the first methodological and applied work in this area, though the fundamental theory has been in place for years.

Two key contributions of this dissertation are as follows. First, this dissertation serves as a first step to fully understanding Polya trees can be used for nonparametric modeling of distributions via binary perpendicular recursively partitioning a support of interest. The results of Chapter 4 show that the methods developed in Chapter 2 can accommodate conditional predictive distribution simulation, missing data imputation, and analysis of categorical data. This research provides a foundation for further development and implementation of Polya tree priors for prediction and inference in multidimensional spaces.

Second, the development, implementation and computation of randomized Polya

trees (Chapter 3) has been shown to reduce the effect of the partition on posterior inference. The randomized Polya tree approach is useful in smoothing discontinuities in predictive distributions and eliminating this critical drawback to using Polya tree priors.

The Polya tree prior is also very appealing computationally, as direct simulation from the predictive distribution is possible. This benefit is lost, however, for randomized Polya trees, in which the tree itself becomes a random variable which must be numerically integrated out of the joint posterior predictive distribution via Markov chain Monte Carlo. Despite this, computational appeal is still not lost, as the tree structure itself provides an efficient way to process information on numerous observations.

Data analyses employing the multivariate randomized tree has yielded promising results. The effect of the smoothing mechanism of the randomized tree succeeds in accounting for partition dependence. The analysis of earthquake data presented in Chapter 4 shows examples of how multivariate randomized Polya trees can be used for estimation and prediction when data appear in a non-standard form, such as multimodal or when modes take on non-standard shapes such as data appearing along fault lines, or in isolated locations.

5.2 Extensions

5.2.1 Hyperparameter Learning for Randomized Trees

As the choice of the hyperparameter τ in the mixture prior for parameters $\{\beta_{i_j}\}$ in the randomized tree formulation is relatively ad-hoc, it is appealing to consider formally learning about the values of this hyperparameter. Prior specifications for τ are under current investigation. Let M = number of levels in the tree, N = sample size, and V = number of variables per observation, the conditional density of the $\{\beta_{i_j}\}$ given

τ takes on the following form:

$$\begin{aligned} p(\{\beta_{i_j}\}|\tau) &= \prod_{i=1}^{N}\prod_{m=1}^{M}\prod_{v=1}^{V} U(0.5-\tau, 0.5+\tau) \\ &= [1/(2\tau)]^{M\times N\times V+1} \end{aligned}$$

Suppose a prior distribution is specified for the hyperparameter: $p(\tau) \propto \tau^{-2} \quad \tau \in (a, 0.5)$, where $a \in (0, 0.5)$ but close to 0; this is necessary as $lim_{\tau\to 0} p(\tau) = \infty$.

This implies that the posterior for τ is

$$p(\tau|\{\beta_{i_m}\}) \propto (1/\tau^{J+2}) I\{\tau \in \{\tau_{max}, 0.5\}\}$$

where $\tau_{max} = \underset{\{all\ i,m,v\}}{max}\{|\beta_{i_{m_v}} - 0.5|\}$, $J = N \times M \times V - \sum_{i,m,v} z_{i_{m_v}}$, and $F(\tau) = [a^{-(J+1)} - \tau^{-(J+1)}]/[a^{-(J+1)} - 2^{J+1}]$.

More work is needed on how to effectively model this smoothing parameter. One consideration should be to realize that smoothing at each level of the tree might be effectively modeled by specifying a smoothing parameter for each level. As smoothing at lower levels of the tree might be more interesting than at higher levels, this could reduce the potential number of hyperparameters in the model. Each observation could be given its own smoothing parameter; as one imagines observations in different regions of the space should affect the smoothing mechanism differently, this could be a sound solution. However, this does introduce $N \times M$ new parameters into the model, which may or may not be prohibitive, depending on memory allocation constraints.

5.2.2 Nonparametric Inference on Dependence Structure

The flexibility of the randomized tree framework should be of interest in problems of nonparametric learning about dependence structures not specifically addressed in

this dissertation. Modifications may or may not need to be made to the assumption that the partition is created based on binary recursive perpendicular splits of the axes of a hypercube. As pointed out in Chapter 2, under certain conditions marginal distributions of multivariate Polya tree priors are Polya trees. It would be interesting to study what would happen if the perspective is reversed, to see whether the product of two distributions that each follow marginal Polya tree priors could yield a joint distribution worthy of study. Dependence structure in the time series framework is worthy of study as well.

Nonparametric Time Series

In the time series framework, Müller *et al.* (1997) develop Bayesian mixture models for nonlinear, non-normal autoregressive time series, in which the set of possible mixture model components is assumed to follow a distribution that arises from a Dirichlet process. It would be interesting to explore how a multivariate (randomized) Polya tree prior would compare to the MDP approach; one question is: could local structure be modeled better with randomized tree /Polya tree than with DP mixtures of parametric kernels?

Ruggeri and West (1999) consider an AR(1) process and model dependence between x_t and x_{t-1} via a Polya tree prior on the real line. Positive (or negative), but close to linear, autocorrelation is explicitly modeled by counting the number of identical digits from the top of the tree downward between the binary expansion of x_t and x_{t-1}.

The PT framework shows promise for nonparametric exploration of $\Re^K$ and could build upon works by these authors, addressing concerns such as locality/multiresolution and modeling positive and/or negative autocorrelation in one model.

In this section, a Polya tree nonparametric time series framework for an AR(1)

process is proposed. This framework should be generalizable for cases with covariates and process orders greater than 1, via implementation of results found in Chapter 2.

Model

Assume an AR(1) process. Express the joint density of random variables $X_t (t = 1, \ldots, T)$ as:

$$f(X_T, X_{T-1}, \ldots, X_1) = f(X_1) \prod_{t=2}^{T} f(X_t | X_{t-1}).$$

Model the observation pairs $(X_t, X_{t-1}) \sim F$ where F comes from a Polya tree in $\Re^2$ *a priori*, with the partitioning scheme as in Figure 2.1. Let f be the pdf of F, and $f_1(y) = \int f(y, x)dx$ be the univariate margin of the first element. The implied AR(1) transition density is $f(x_t|x_{t-1}) = f(x_t, x_{t-1})/f_1(x_{t-1})$. Then, based on the observed data, the likelihood function for F may be written formally as

$$p(X|F) \propto \frac{f_1(x_1) \prod_{t=2}^{T} f(x_t, x_{t-1})}{\prod_{t=2}^{T} f_1(x_{t-1})} = \frac{\prod_{t=2}^{T} f(x_t, x_{t-1})}{\prod_{t=3}^{T} f_1(x_{t-1})} \tag{5.1}$$

Updating Scheme

The numerator in Equation 5.1 is similar to that arising in bivariate random samples, but the denominator poses a challenge. To update F, the collection $\mathcal{Y}$ must be updated sequentially. As this is done level by level in the tree, the results below correspond to any $Y = (Y_0, Y_1, Y_2, Y_3) \in \mathcal{Y}$ for any corresponding urn $B \in \Pi$. Let ϵ_t be the indicator of which child urn the pair (x_t, x_{t-1}) falls (Figure 2.1), and define $\delta_j(\epsilon) = 1$ if $\epsilon = j$ and 0 otherwise. Let e_j be the vector with 1 as the value for component j and 0 otherwise. Define $\Delta_t = \sum_{i=2}^{t} \sum_{j=1}^{4} e_j \times \delta_j(\epsilon_t)$ to be the vector of length 4 corresponding to a count of how many observations fall in the cell corresponding to ϵ_t

at time t for the bivariate Polya tree; for example, if $(x_1, x_2) \in B_0$ and $(x_2, x_3) \in B_1$, then $\Delta_1 = (1, 0, 0, 0)$ and $\Delta_2 = (1, 1, 0, 0)$. Thus, $\alpha = (\alpha_0, \alpha_1, \alpha_2, \alpha_3) \in \mathcal{A}$ is updated to $\alpha + \Delta_t$ at time t. Also, α_j and Δ_{t_j} are the j^{th} components of α and Δ_t, respectively.

At time $t \geq 3$:

$$\begin{aligned} p(Y|x_{t-1}, \ldots, x_1) &\sim Dirichlet(\alpha + \Delta_{t-1}) \\ p(Y|x_t, x_{t-1}, \ldots, x_1) &\propto p(Y|x_{t-1}, \ldots, x_1) \frac{f(x_t, x_{t-1}|Y)}{f_1(x_{t-1}|Y)} \\ &\propto \frac{Y_0^{a_0+\Delta_{t_0}} Y_1^{a_1+\Delta_{t_1}} Y_2^{a_2+\Delta_{t_2}} Y_3^{a_3+\Delta_{t_3}}}{(Y_0+Y_1)^{\delta_1(\epsilon_t)+\delta_2(\epsilon_t)} (Y_2+Y_3)^{\delta_3(\epsilon_t)+\delta_4(\epsilon_t)}} \end{aligned}$$

Generally,

$$p(Y|X) \propto \frac{Dirichlet(Y|\alpha + \Delta_t)}{(Y_0+Y_1)^{c_1}(Y_2+Y_3)^{c_2}}$$

where c_1 is the number of (x_t, x_{t-1}) pairs in $B_0 \cup B_1$ and c_2 the number in $B_2 \cup B_3$.

This initial development suggests some potential, though full development of this framework is pending.

5.2.3 Computation

While tree-based computation algorithms are quite efficient, there is always be room for improvement. This might be a worthwhile area of research if the Polya or randomized tree will be applied to a very large problem. This will be tricker for the randomized tree than for the Polya tree, given the partitions are not known beforehand and must be introduced as parameters.

Application of this methodology to higher-dimensional spaces will inevitably be hindered by the curse of dimensionality – local neighborhoods are empty and non-empty neighborhoods are not local. Perhaps an adaptive partitioning scheme would

remedy the problem in some cases – only partition the interesting regions of a space and ignore the rest. This would entail formal learning about where the data should be partitioned. One potential approach could be to update the tree a varying number of levels throughout various regions of the space, depending on how "interesting" the space is, or to exploit the multiresolution nature of the tree in some way.

Appendix A

GRAPHS FROM EARTHQUAKE ANALYSES OF CHAPTER 4

This Appendix contains the graphs of the conditional predictive distribution simulations, based on the earthquake data, that were described in Chapter 4. Figures A.1–A.3 show the conditional predictive simulations based on the Polya tree prior. Figures A.4–A.21 display simulations for location given various values magnitude, as denoted on each Figure. Figures A.22–A.29 display the output from simulations of the conditional predictive distributions of location given various values of depth.

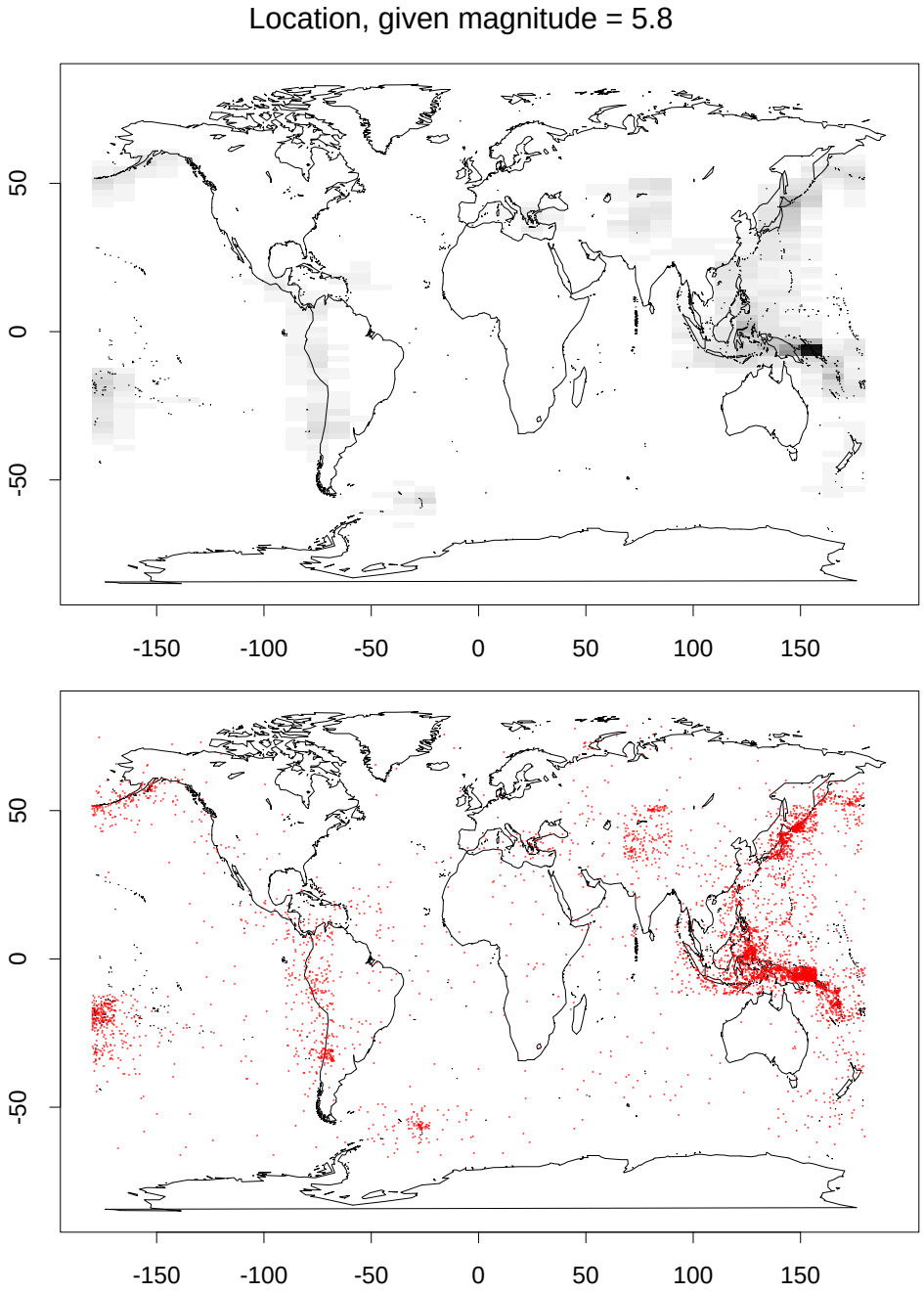


Figure A.1: Section 4.1.1: Conditional predictive distribution of location given magnitude = 5.8. $\alpha = 0.1m^2$ and G =Uniform. Polya tree prior.

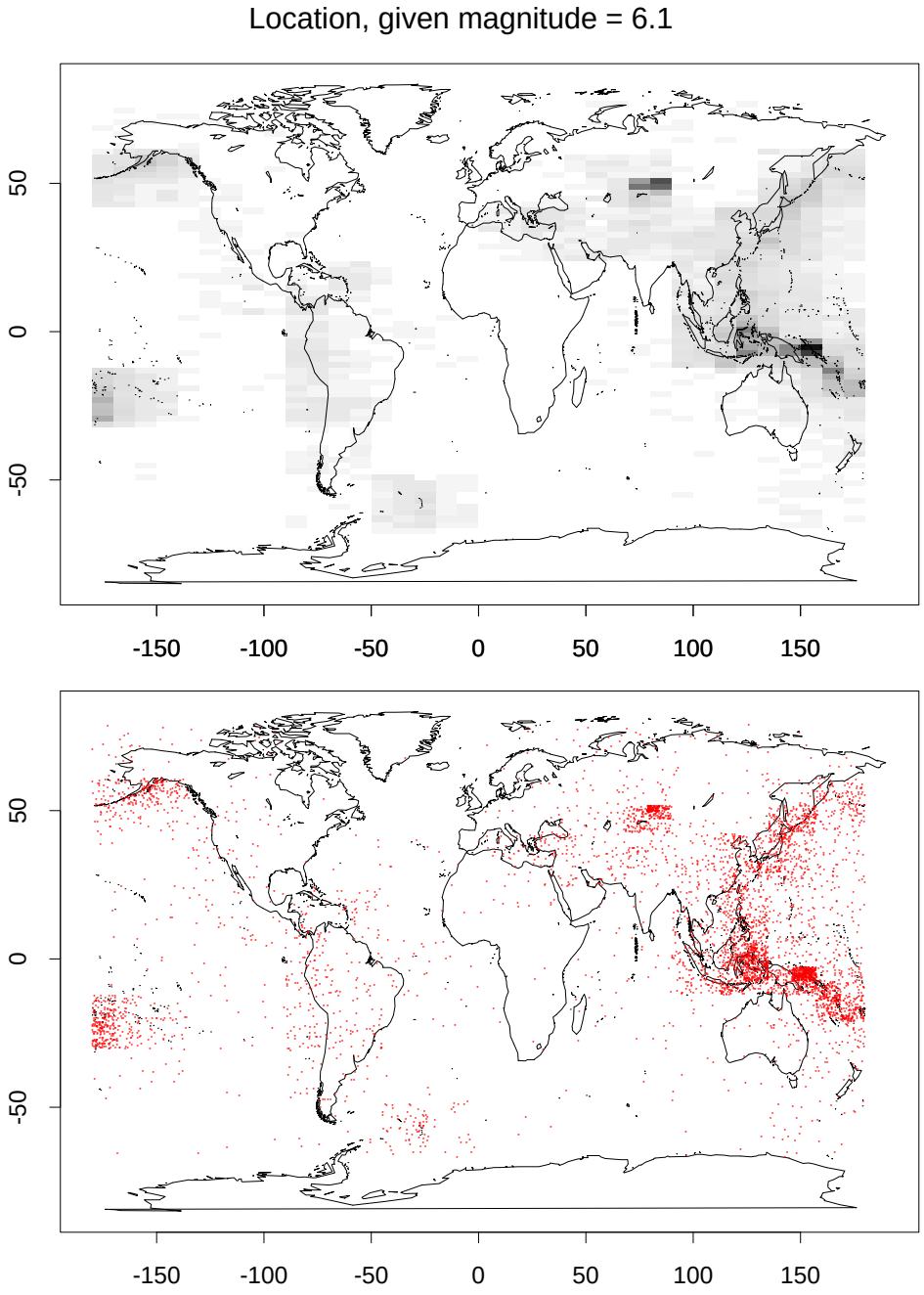


Figure A.2: Section 4.1.1: Conditional predictive distribution of location given magnitude = 6.1. $\alpha = 0.1m^2$ and G =Uniform. Polya tree prior.

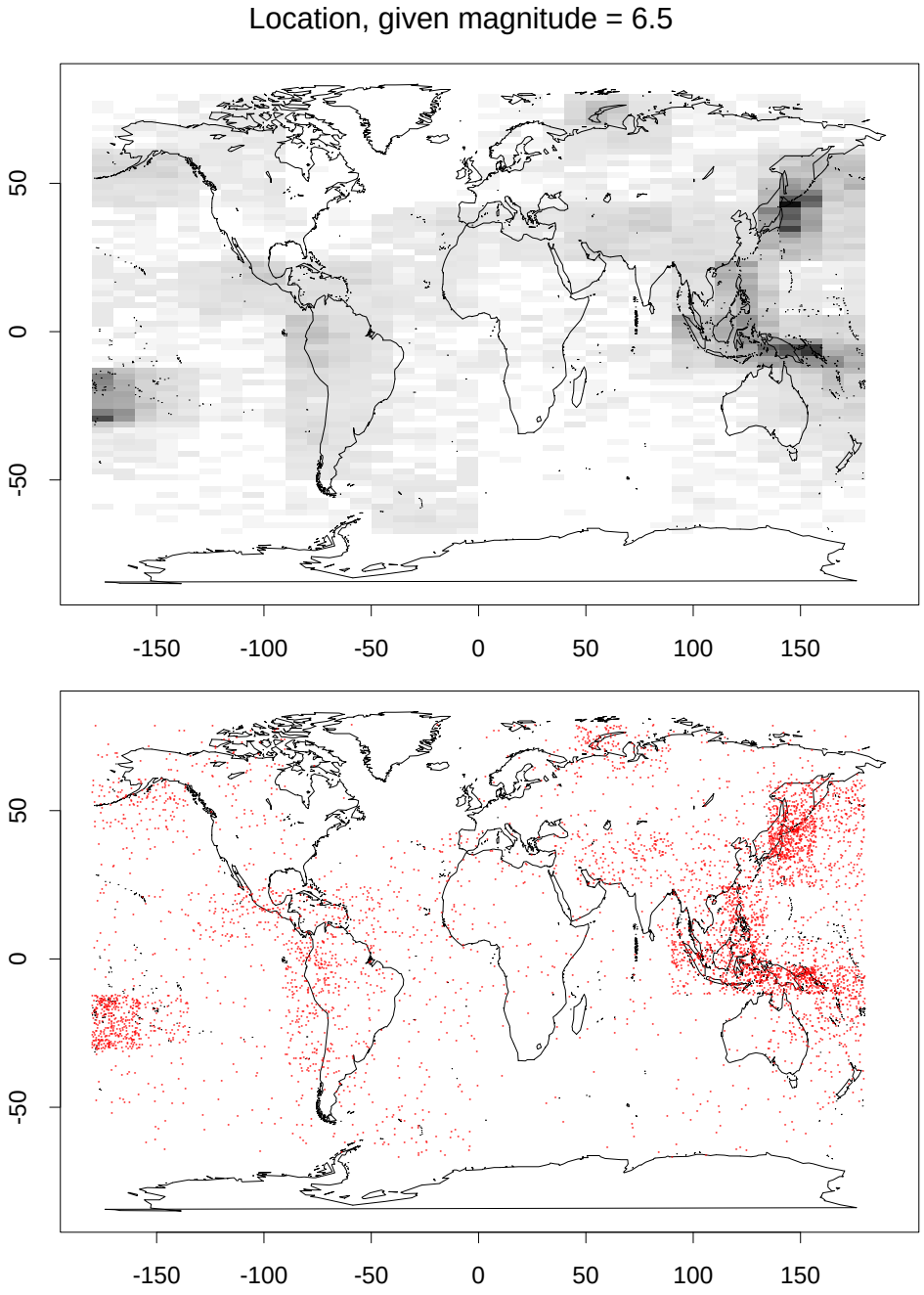


Figure A.3: Section 4.1.1: Conditional predictive distribution of location given magnitude = 6.5. $\alpha = 0.1m^2$ and G =Uniform. Polya tree prior.

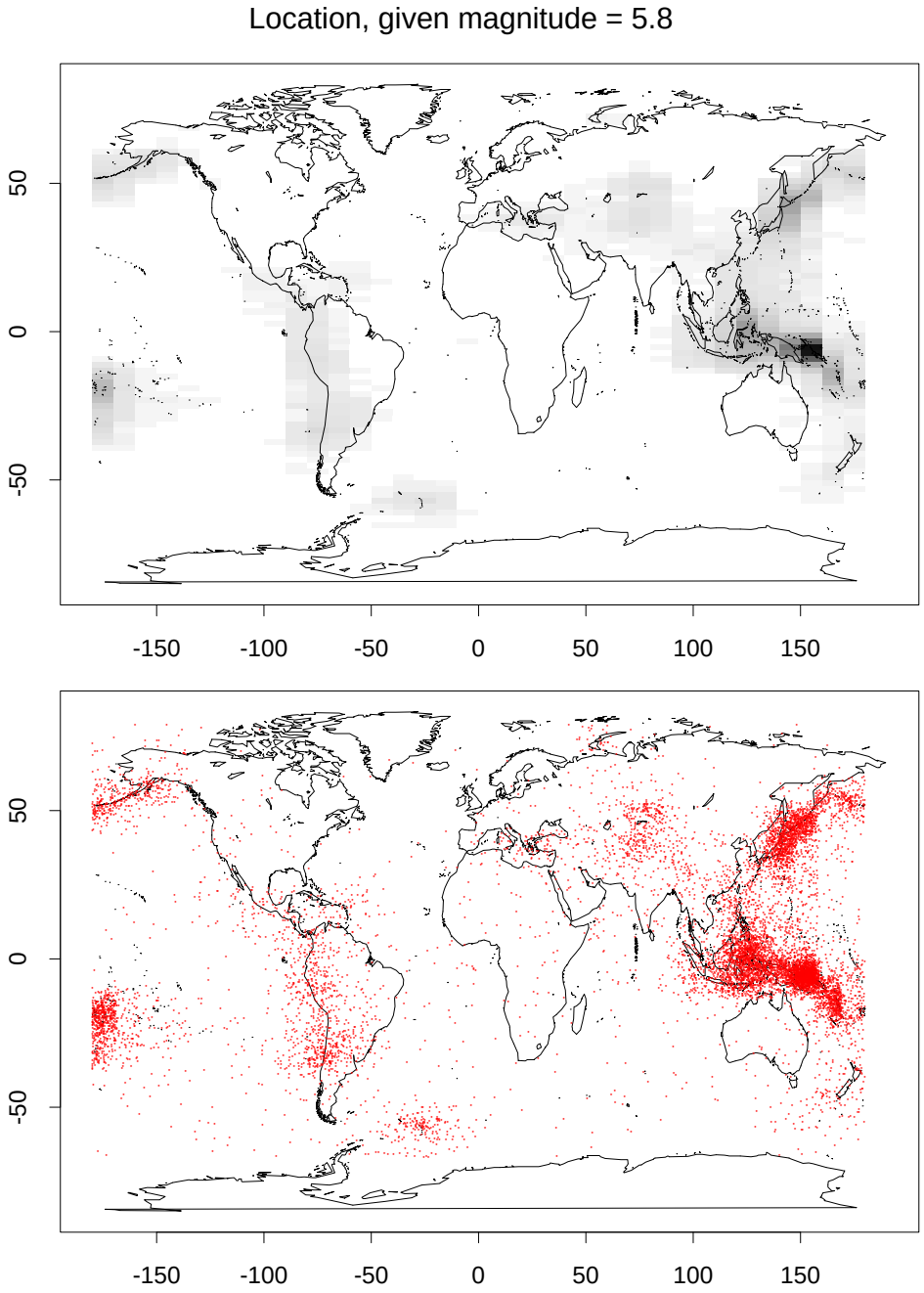


Figure A.4: Section 4.1.2 – Analysis 1: Conditional predictive distribution of location given magnitude = 5.8. $\alpha = 0.1m^2$, $\tau = 0.025$, and G =Uniform

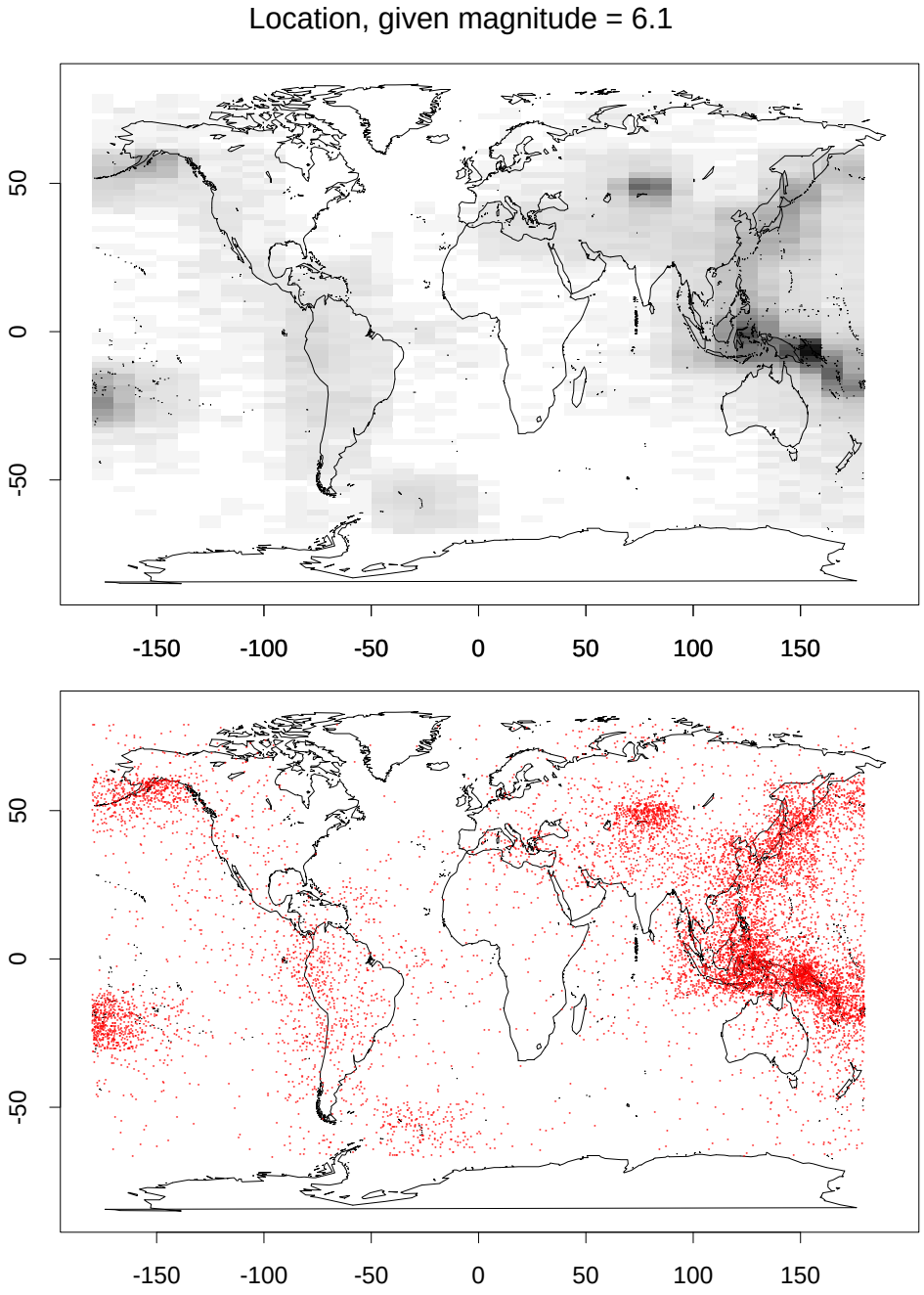


Figure A.5: Section 4.1.2 – Analysis 1: Conditional predictive distribution of location given magnitude = 6.1. $\alpha = 0.1m^2$, $\tau = 0.025$, and G =Uniform

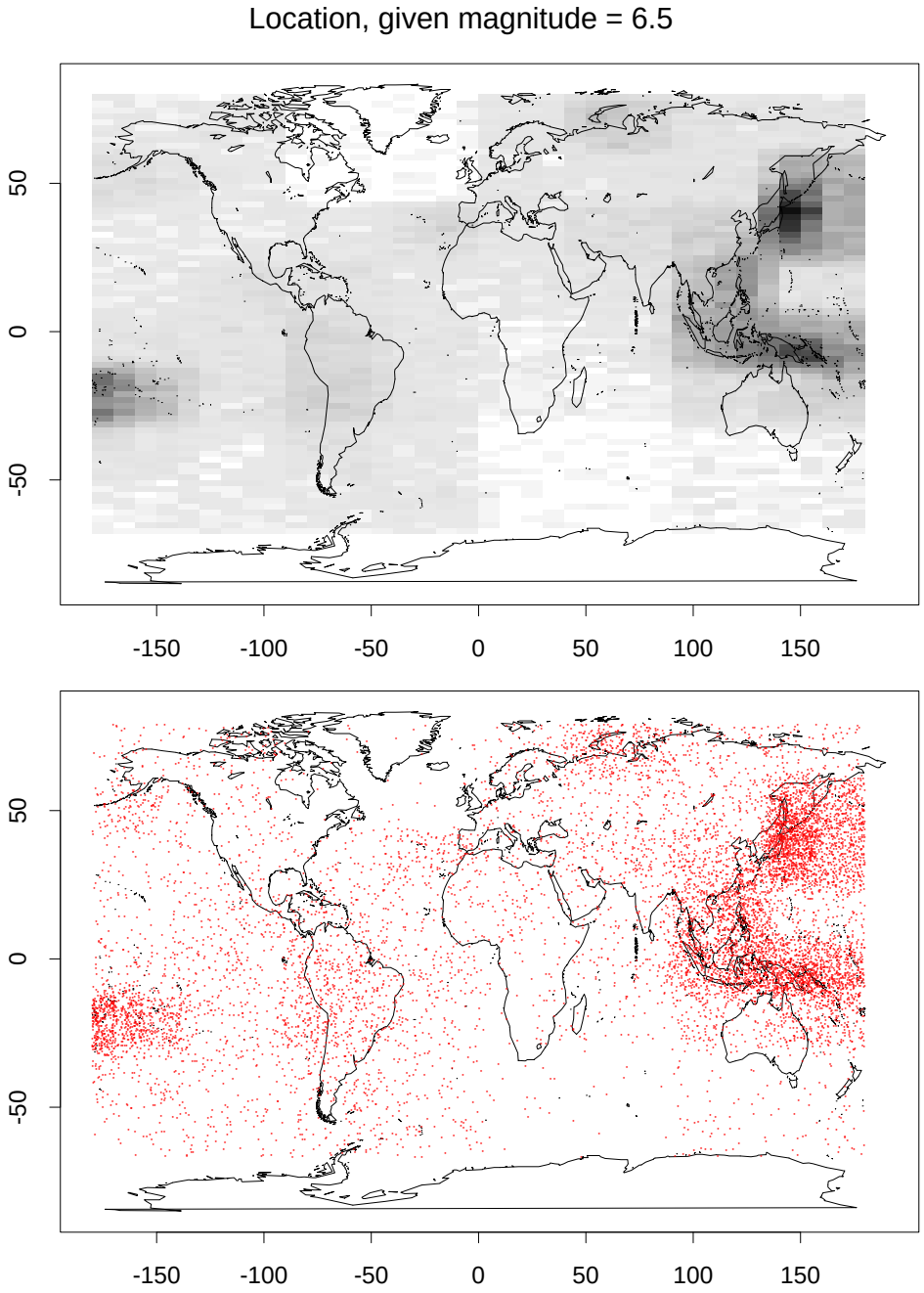


Figure A.6: Section 4.1.2 – Analysis 1: Conditional predictive distribution of location given magnitude = 6.5. $\alpha = 0.1m^2$, $\tau = 0.025$, and G =Uniform

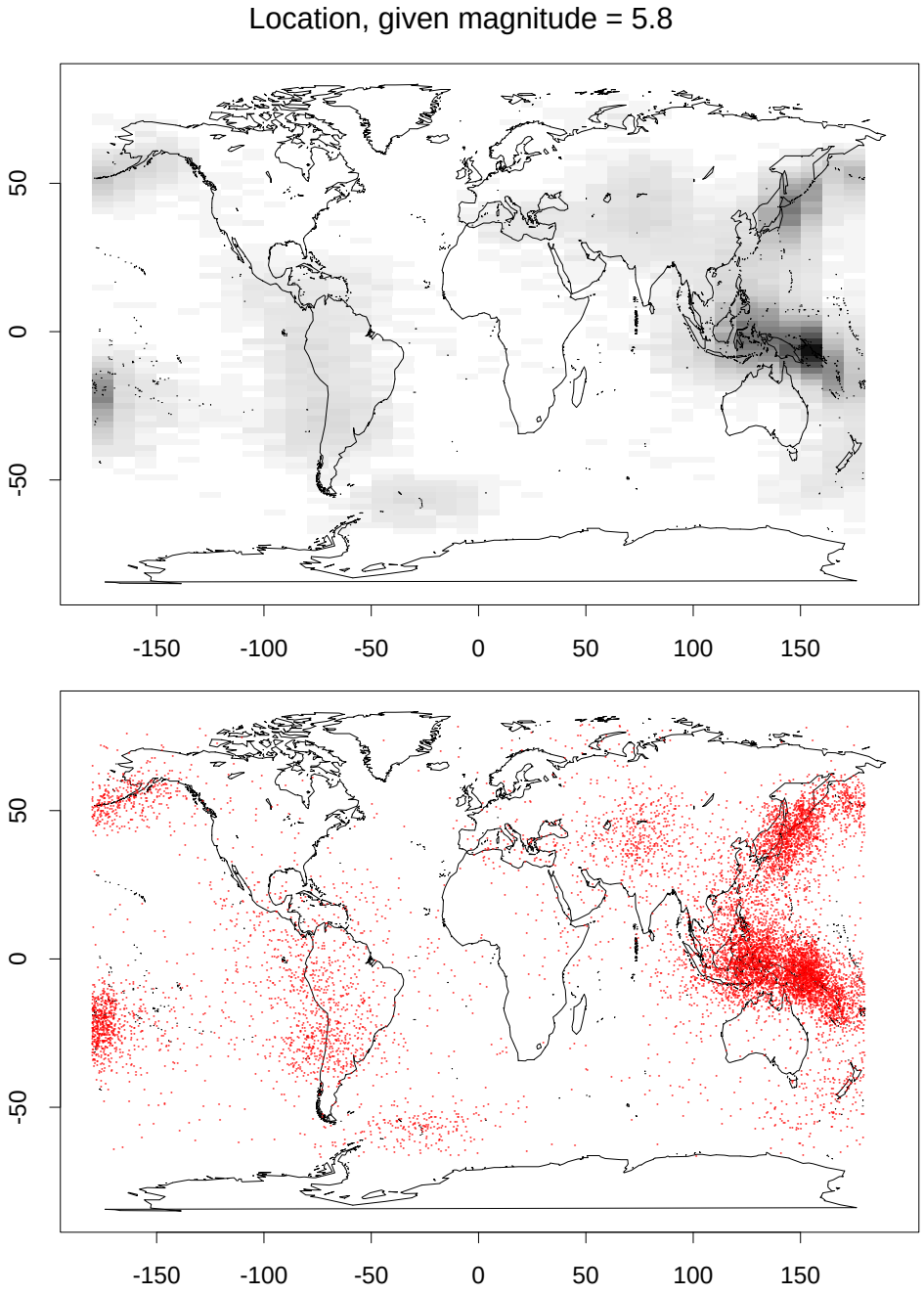


Figure A.7: Section 4.1.2 – Analysis 2: Conditional predictive distribution of location given magnitude = 5.8. $\alpha = 0.1m^2$, $\tau = 0.05$, and G =Uniform

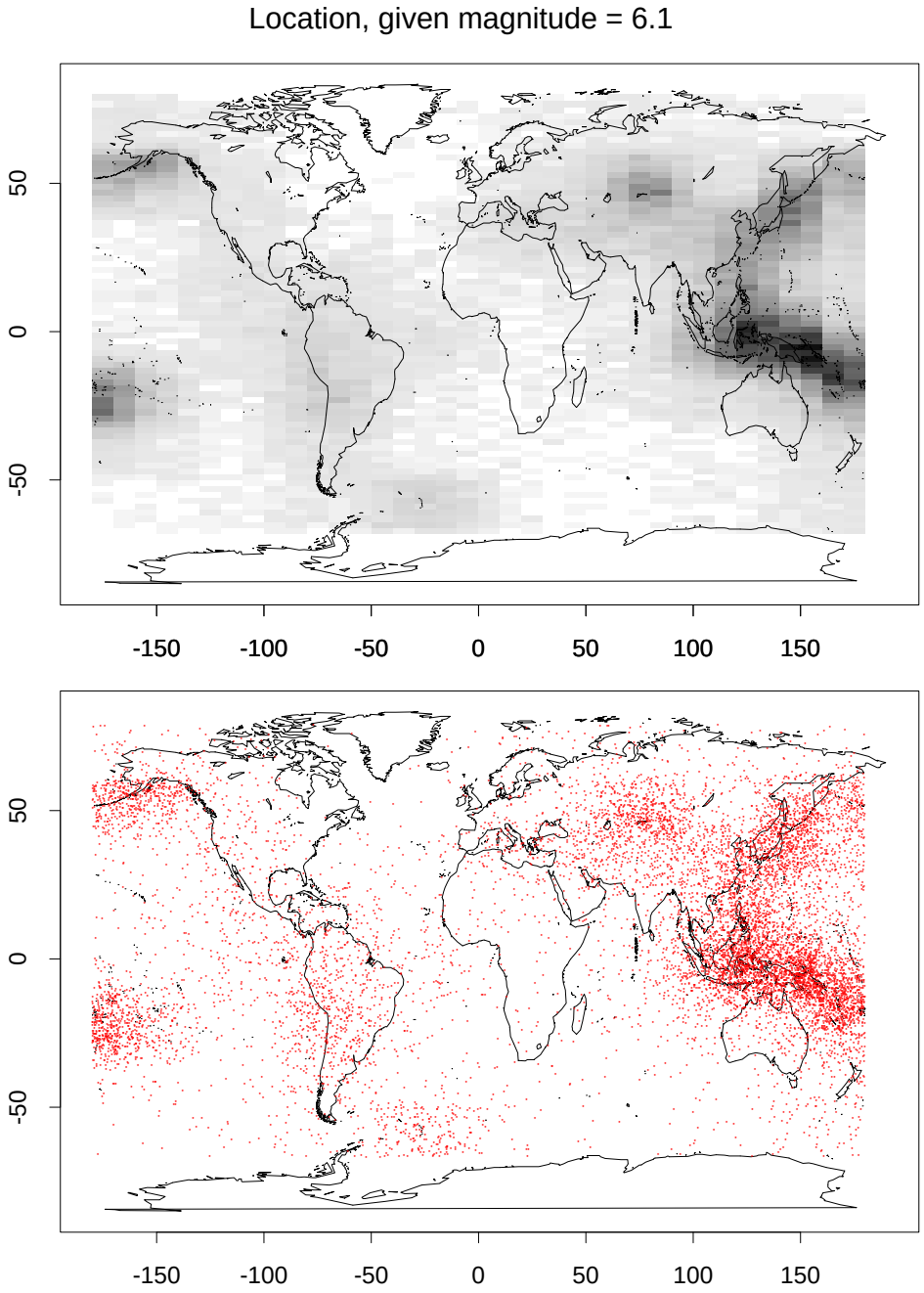


Figure A.8: Section 4.1.2 – Analysis 2: Conditional predictive distribution of location given magnitude = 6.1. $\alpha = 0.1m^2$, $\tau = 0.05$, and G =Uniform

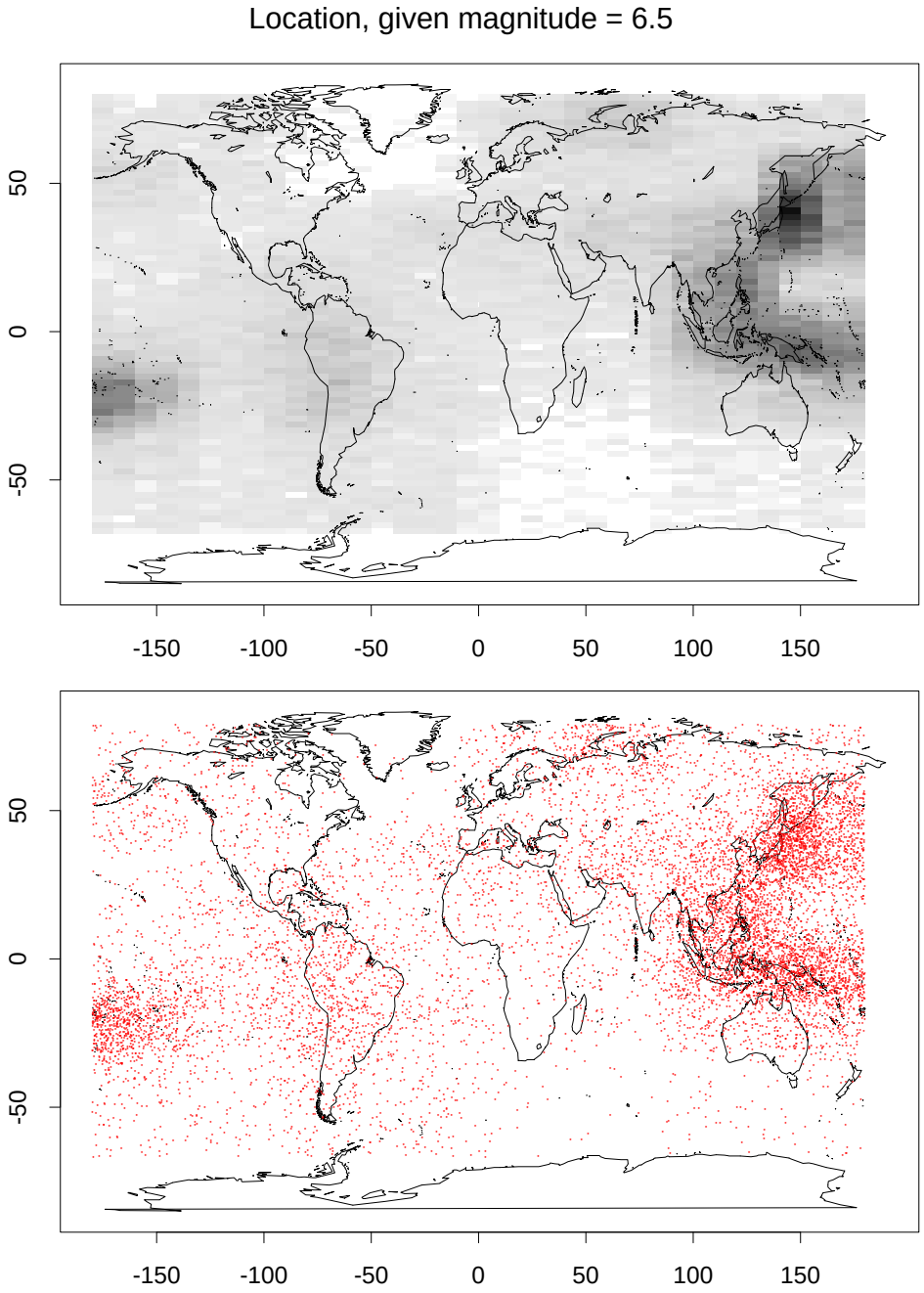


Figure A.9: Section 4.1.2 – Analysis 2: Conditional predictive distribution of location given magnitude = 6.5 $\alpha = 0.1m^2$, $\tau = 0.05$, and G =Uniform

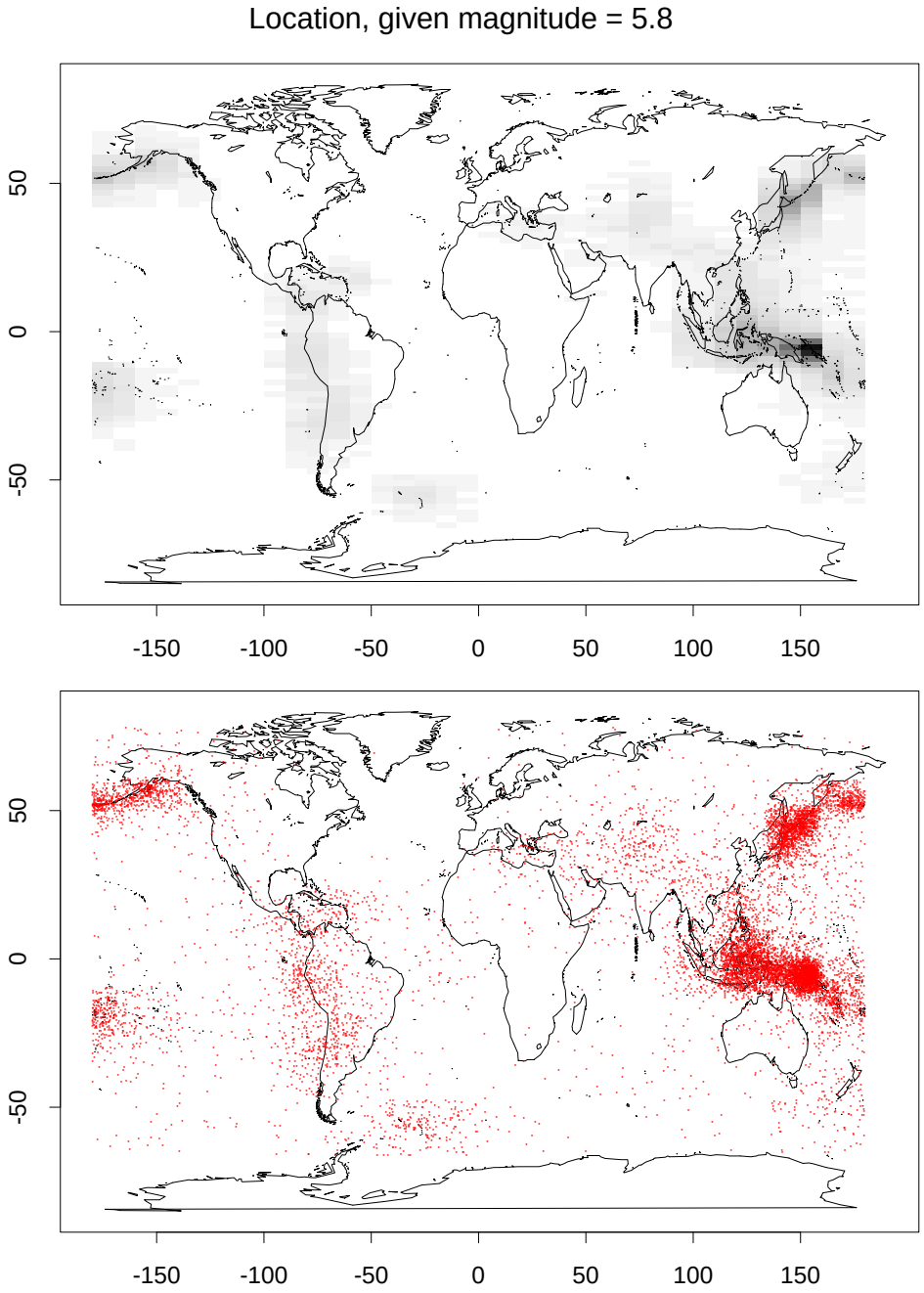


Figure A.10: Section 4.1.2 – Analysis 3: Conditional predictive distribution of location given magnitude = 5.8. $\alpha = 0.1m^2$, $\tau = 0.025$, G=empirical cdf

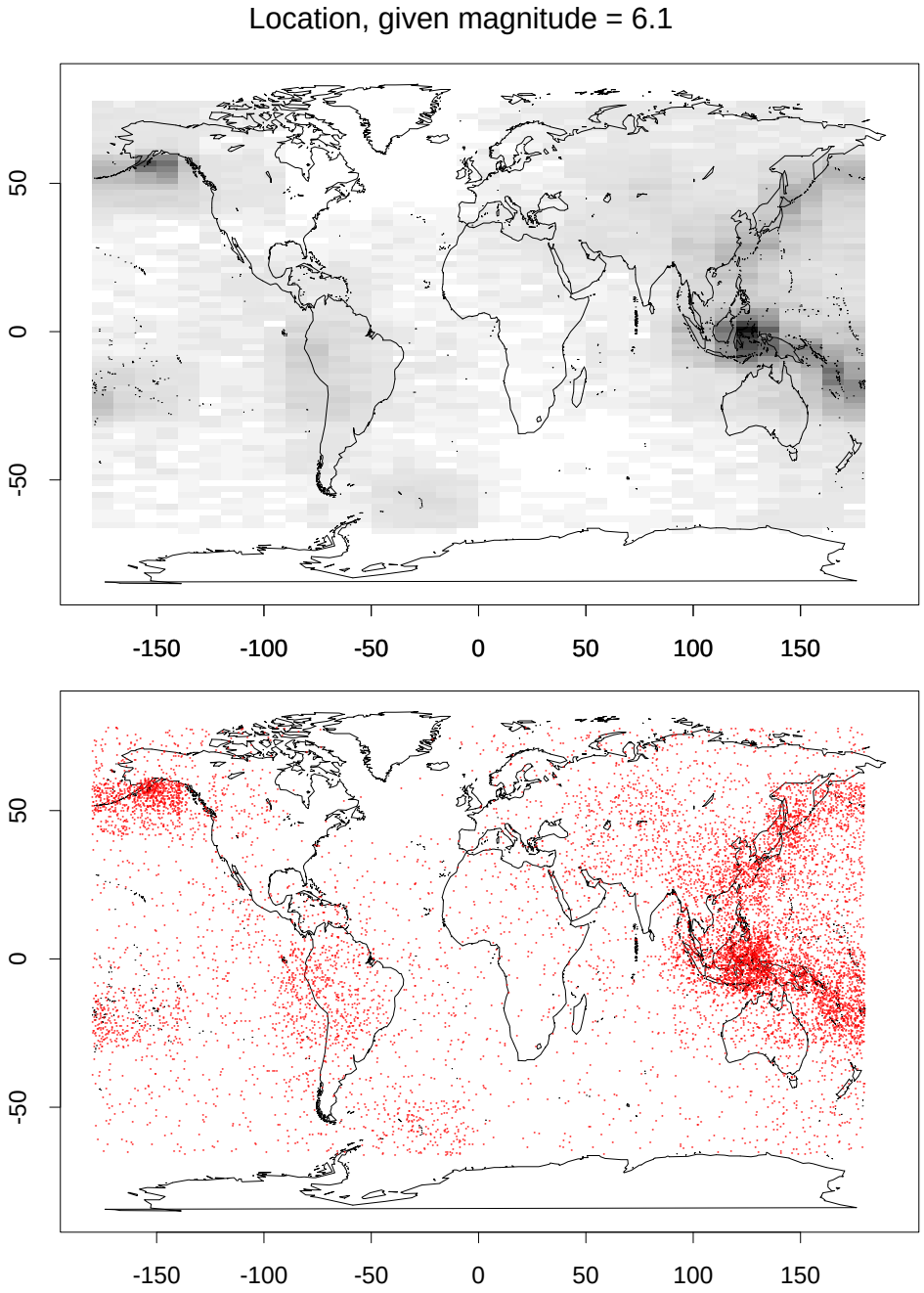


Figure A.11: Section 4.1.2 – Analysis 3: Conditional predictive distribution of location given magnitude = 6.1. $\alpha = 0.1m^2$, $\tau = 0.025$, G=empirical cdf

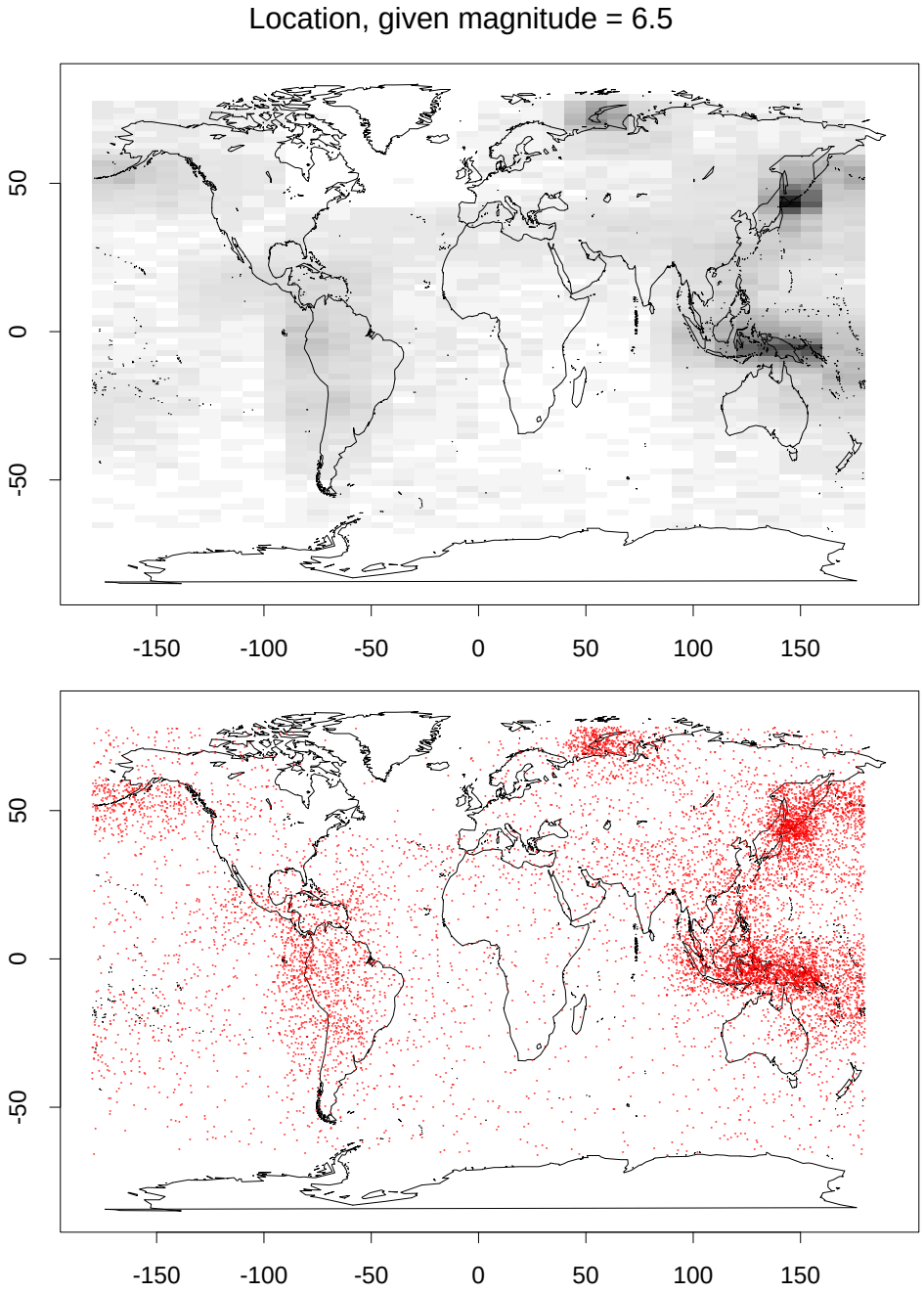


Figure A.12: Section 4.1.2 – Analysis 3: Conditional predictive distribution of location given magnitude = 6.5 $\alpha = 0.1m^2$, $\tau = 0.025$, G=empirical cdf

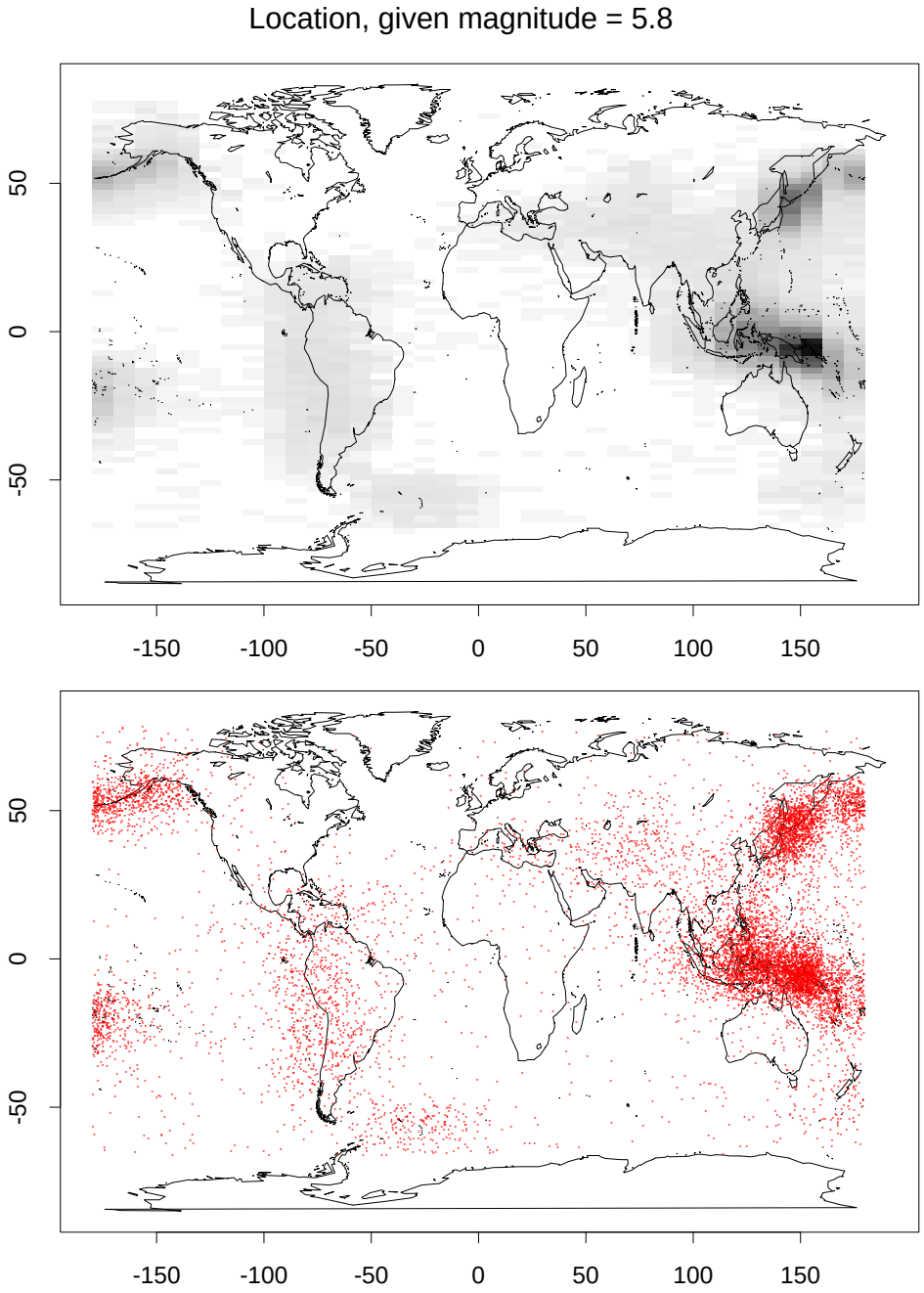


Figure A.13: Section 4.1.2 – Analysis 4: Conditional predictive distribution of location given magnitude = 5.8. $\alpha = 0.1m^2$, $\tau = 0.05$, G=empirical cdf

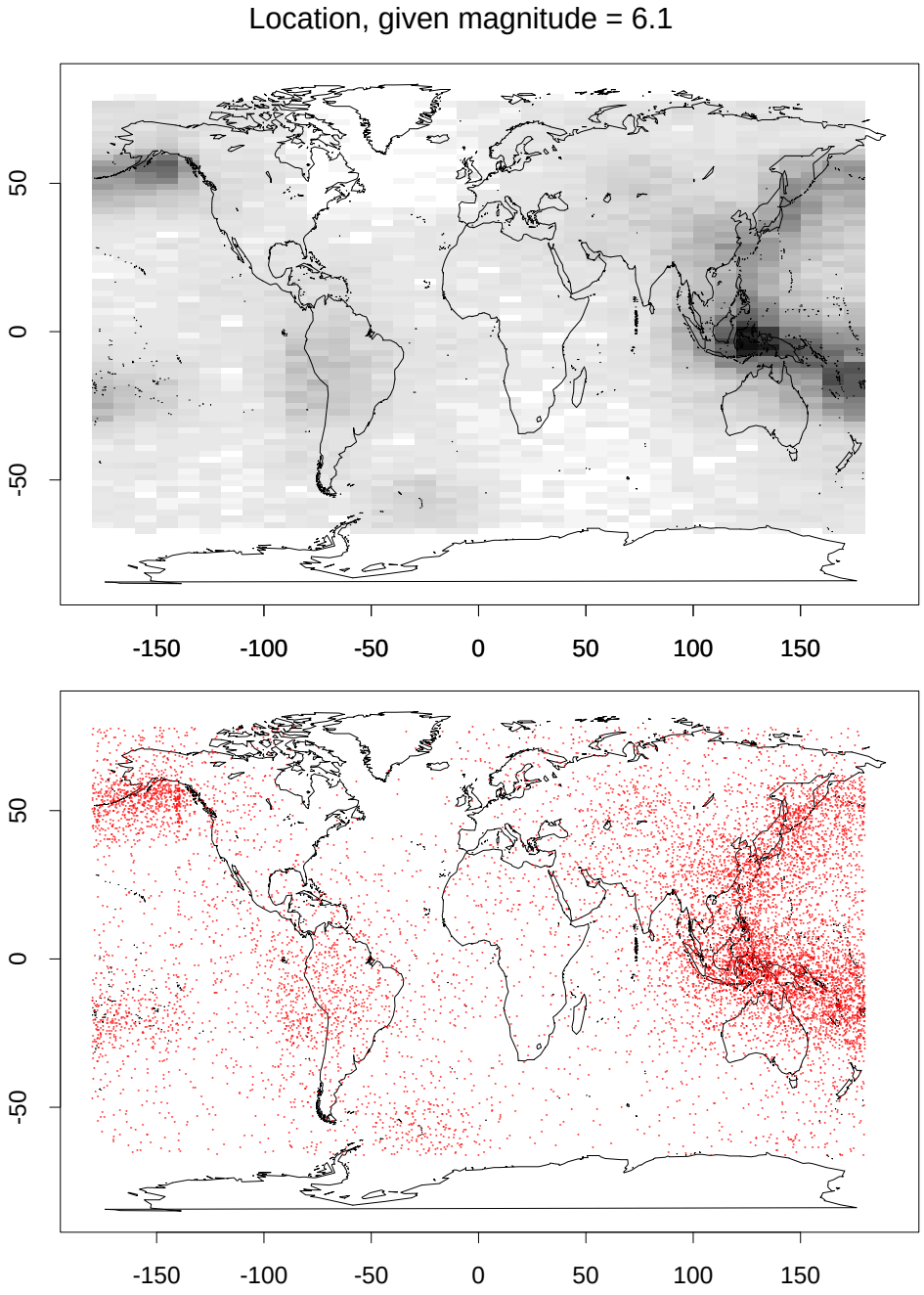


Figure A.14: Section 4.1.2 – Analysis 4: Conditional predictive distribution of location given magnitude = 6.1. $\alpha = 0.1m^2$, $\tau = 0.05$, G=empirical cdf

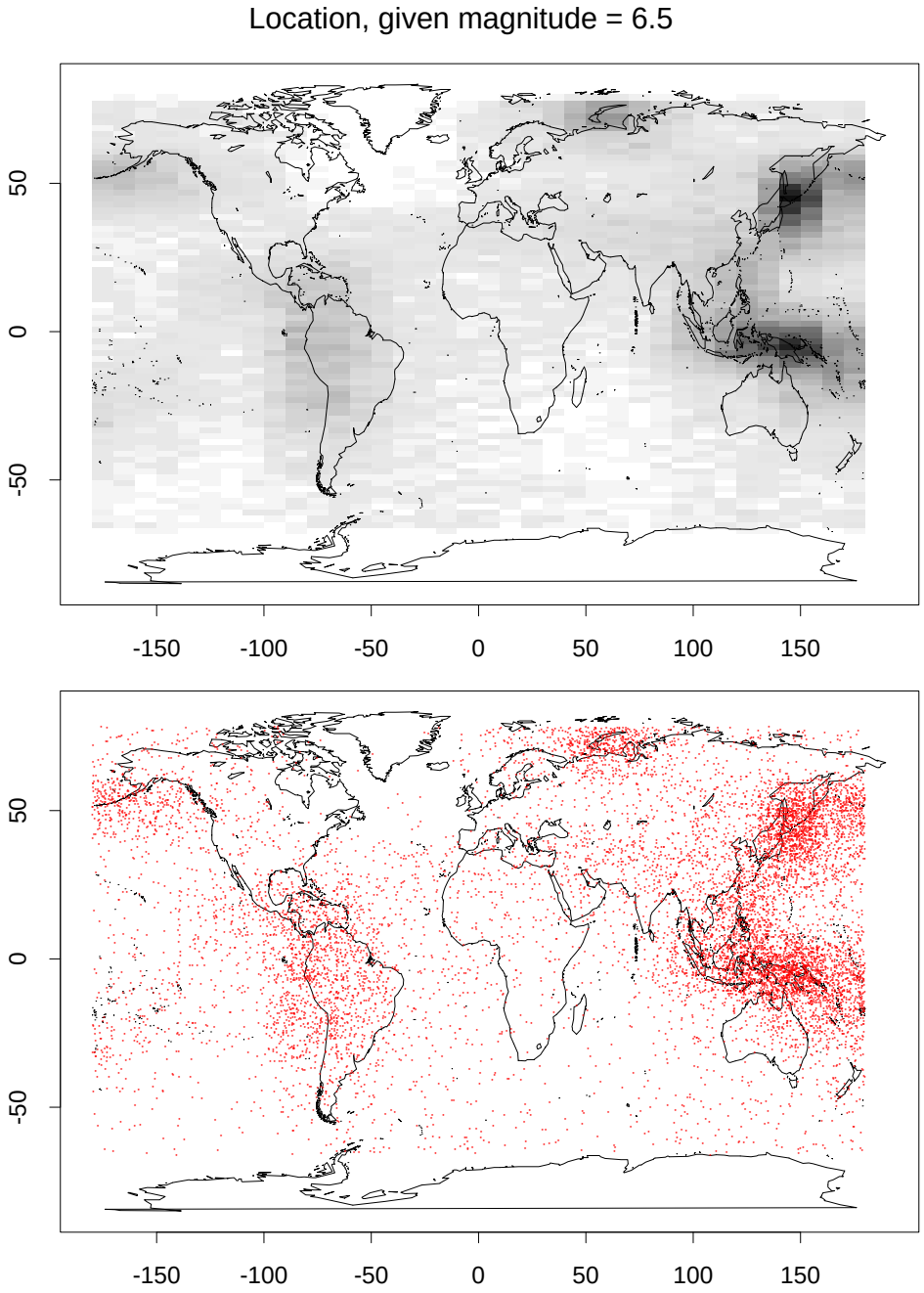


Figure A.15: Section 4.1.2 – Analysis 4: Conditional predictive distribution of location given magnitude = 6.5 $\alpha = 0.1m^2$, $\tau = 0.05$, G=empirical cdf

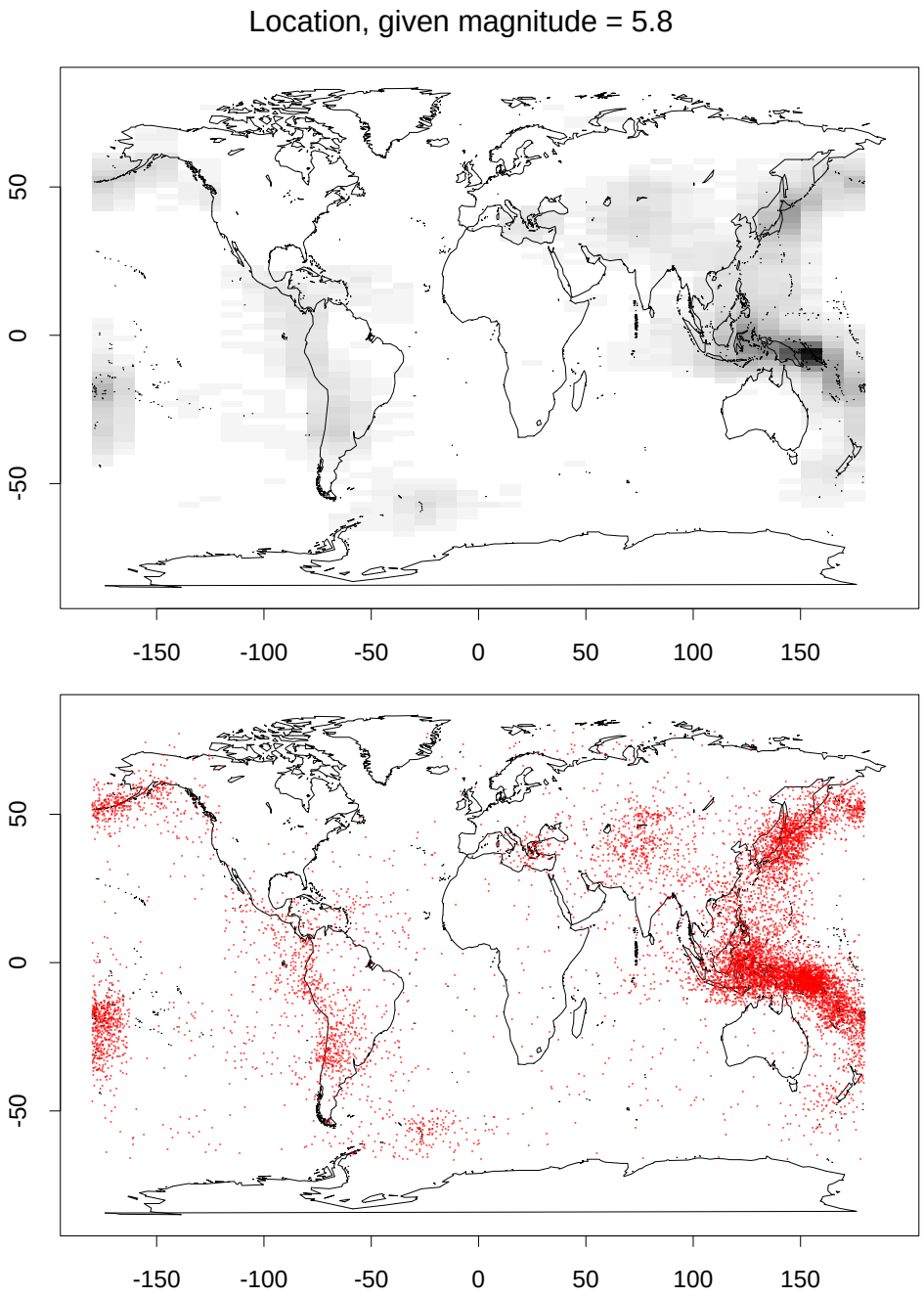


Figure A.16: Section 4.1.2 – Analysis 5: Conditional predictive distribution of location given magnitude = 5.8. $\alpha = 0.1m^2$, $\tau = 0.025$, G=Uniform. Longitude in (-120,240).

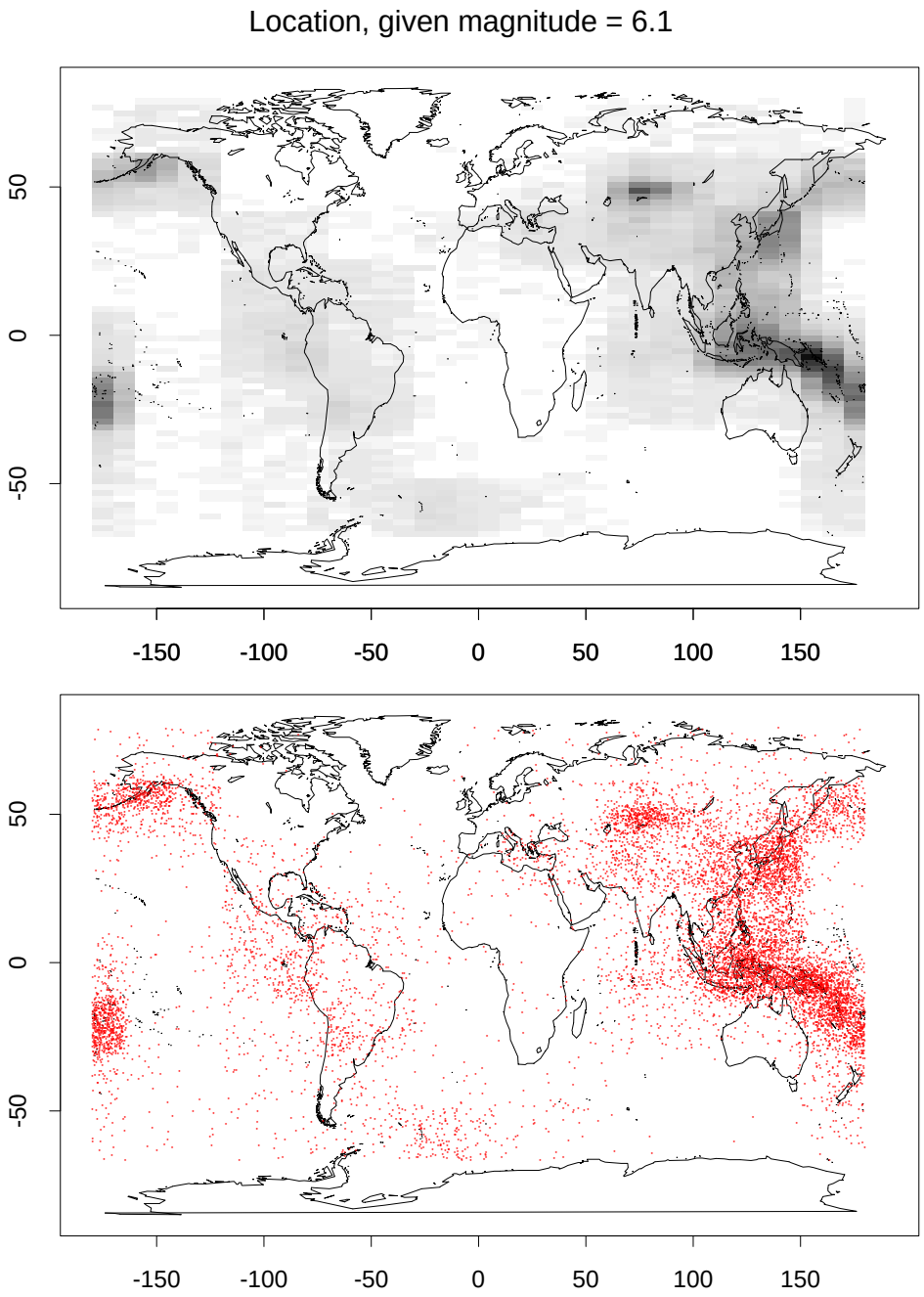


Figure A.17: Section 4.1.2 – Analysis 5: Conditional predictive distribution of location given magnitude = 6.1. $\alpha = 0.1m^2$, $\tau = 0.025$, G=Uniform. Longitude in (-120,240).

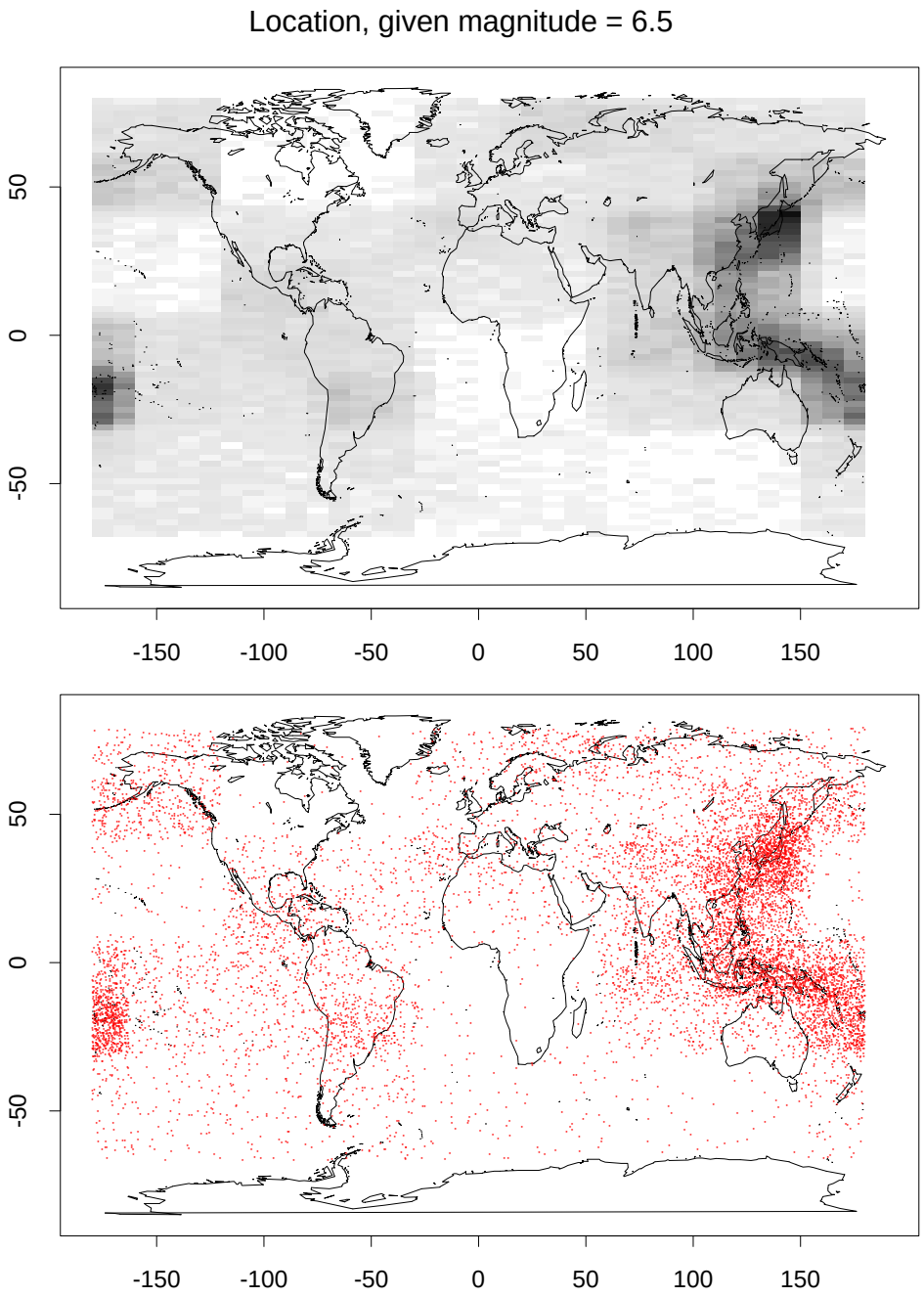


Figure A.18: Section 4.1.2 – Analysis 5: Conditional predictive distribution of location given magnitude = 6.5 $\alpha = 0.1m^2$, $\tau = 0.025$, G=Uniform. Longitude in (-120,240).

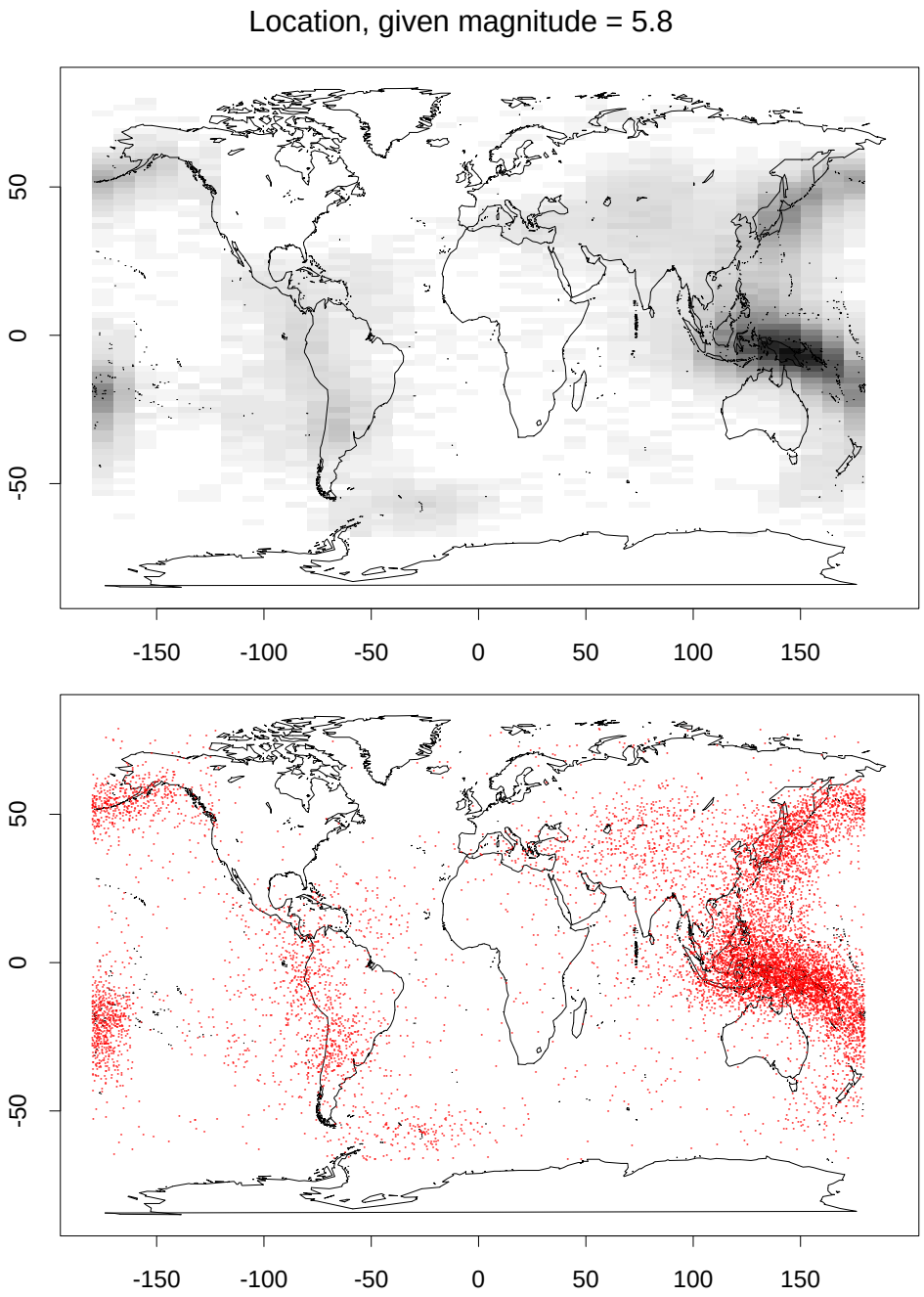


Figure A.19: Section 4.1.2 – Analysis 6: Conditional predictive distribution of location given magnitude = 5.8. $\alpha = 0.1m^2$, $\tau = 0.05$, G=Uniform. Longitude in (-120,240).

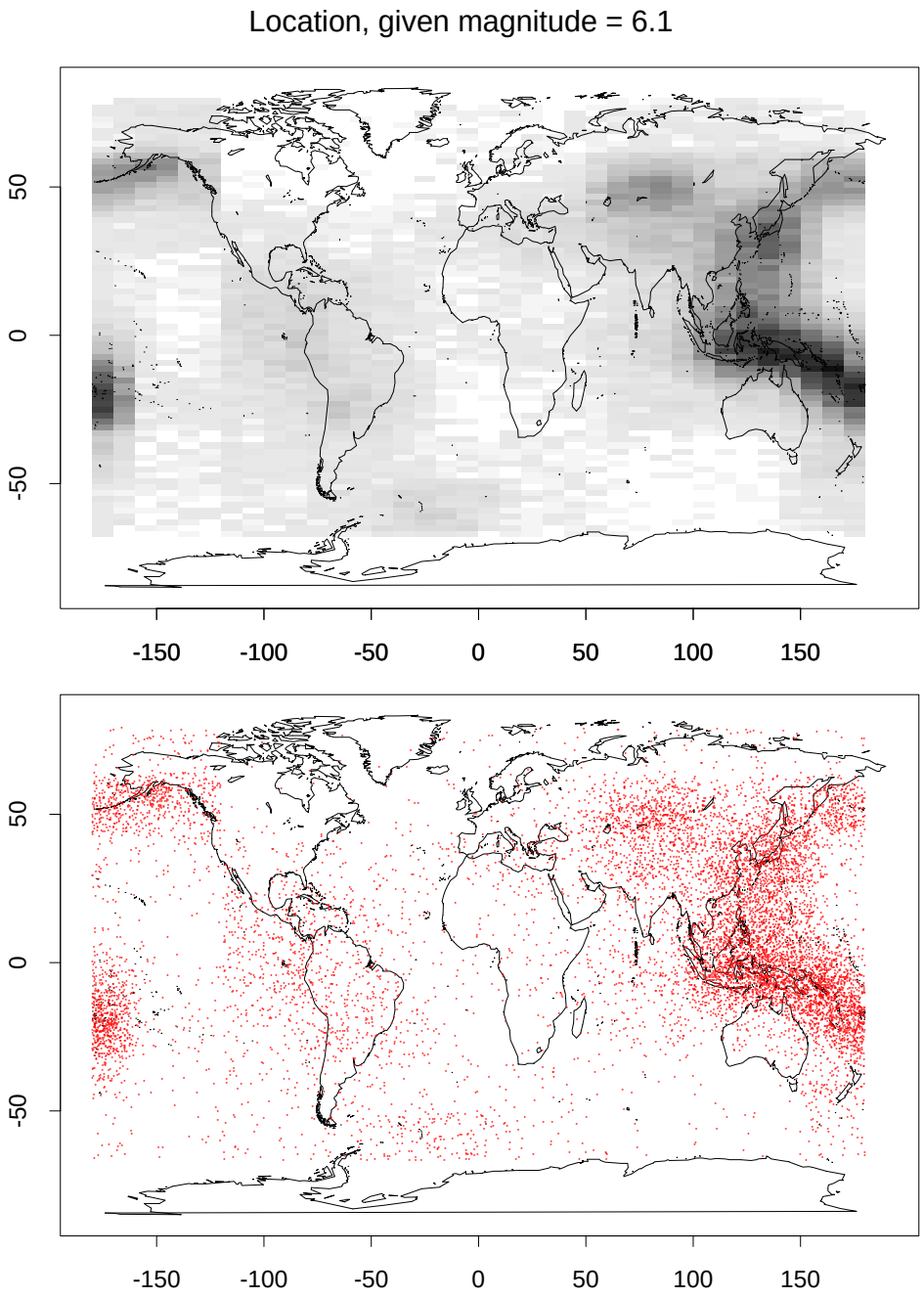


Figure A.20: Section 4.1.2 – Analysis 6: Conditional predictive distribution of location given magnitude = 6.1. $\alpha = 0.1m^2$, $\tau = 0.05$, G=Uniform. Longitude in (-120,240).

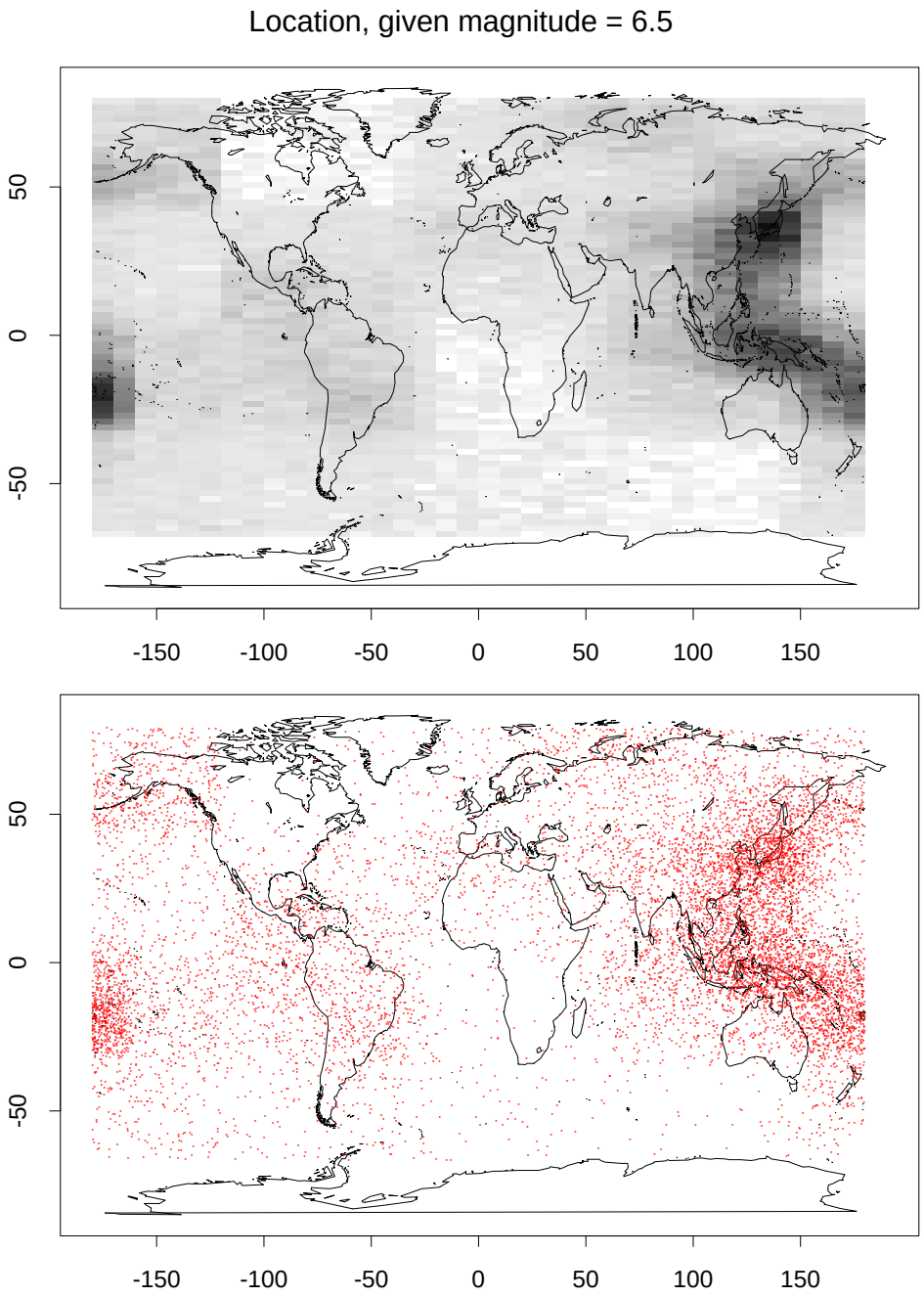


Figure A.21: Section 4.1.2 – Analysis 6: Conditional predictive distribution of location given magnitude = 6.5 $\alpha = 0.1m^2$, $\tau = 0.05$, G=Uniform. Longitude in (-120,240).

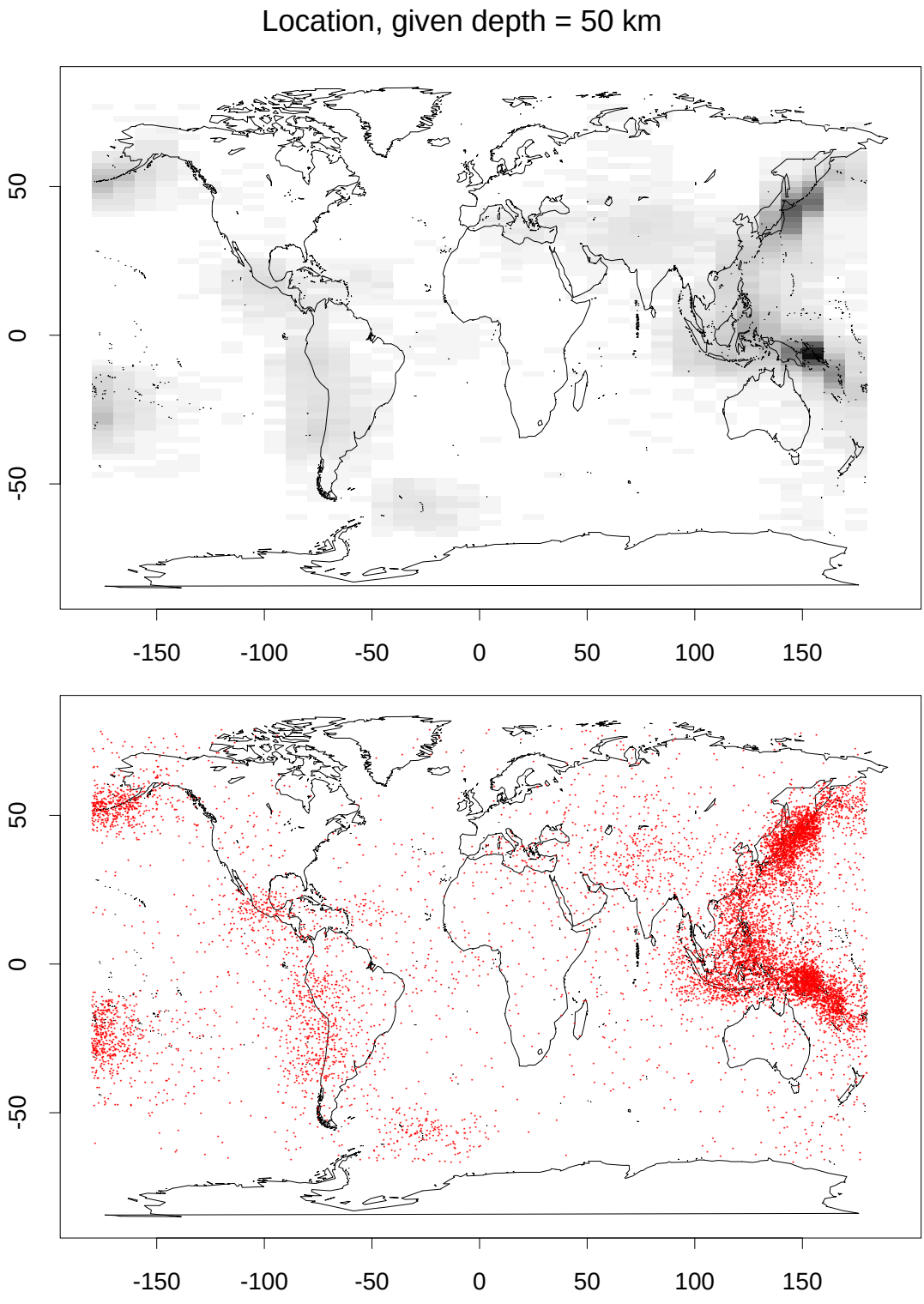


Figure A.22: Section 4.1.3: Conditional predictive distribution of location given depth = 50 km. $\alpha = 0.1m^2$, $\tau = 0.025$ and G =Uniform.

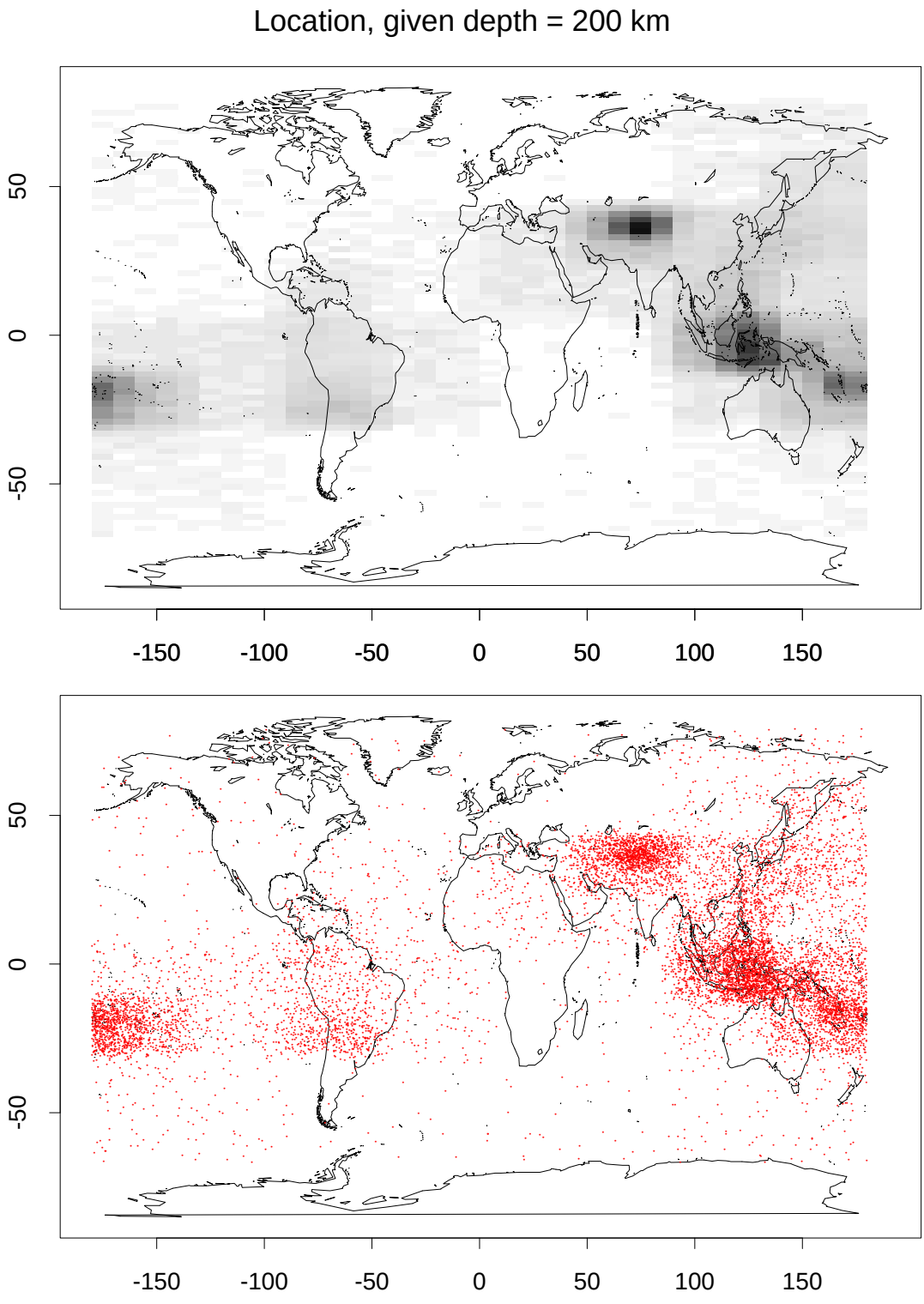


Figure A.23: Section 4.1.3: Conditional predictive distribution of location given dpeth = 200 km. $\alpha = 0.1m^2$, $\tau = 0.025$, and G =Uniform.

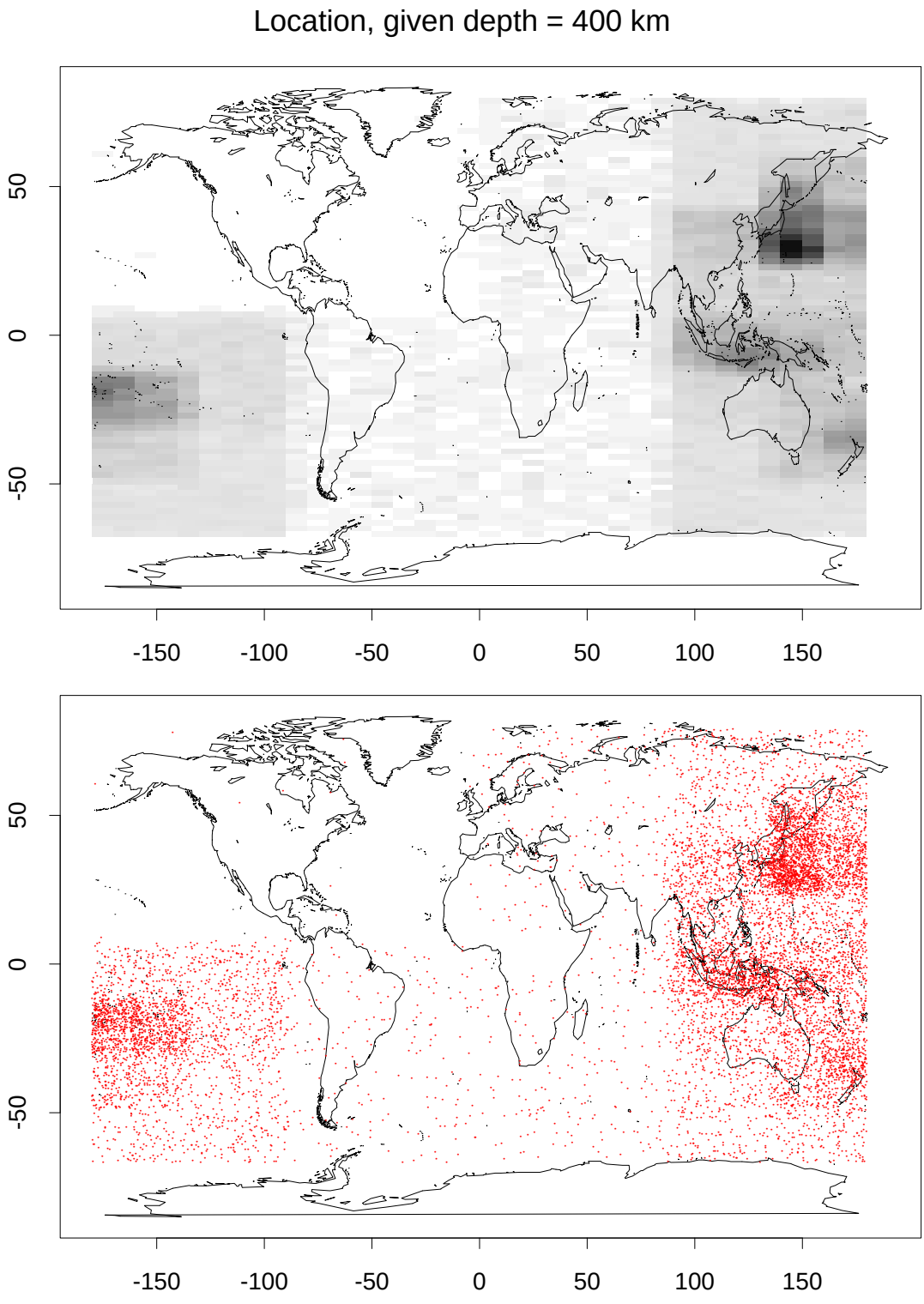


Figure A.24: Section 4.1.3: Conditional predictive distribution of location given depth = 400 km. $\alpha = 0.1m^2$, $\tau = 0.025$ and G =Uniform.

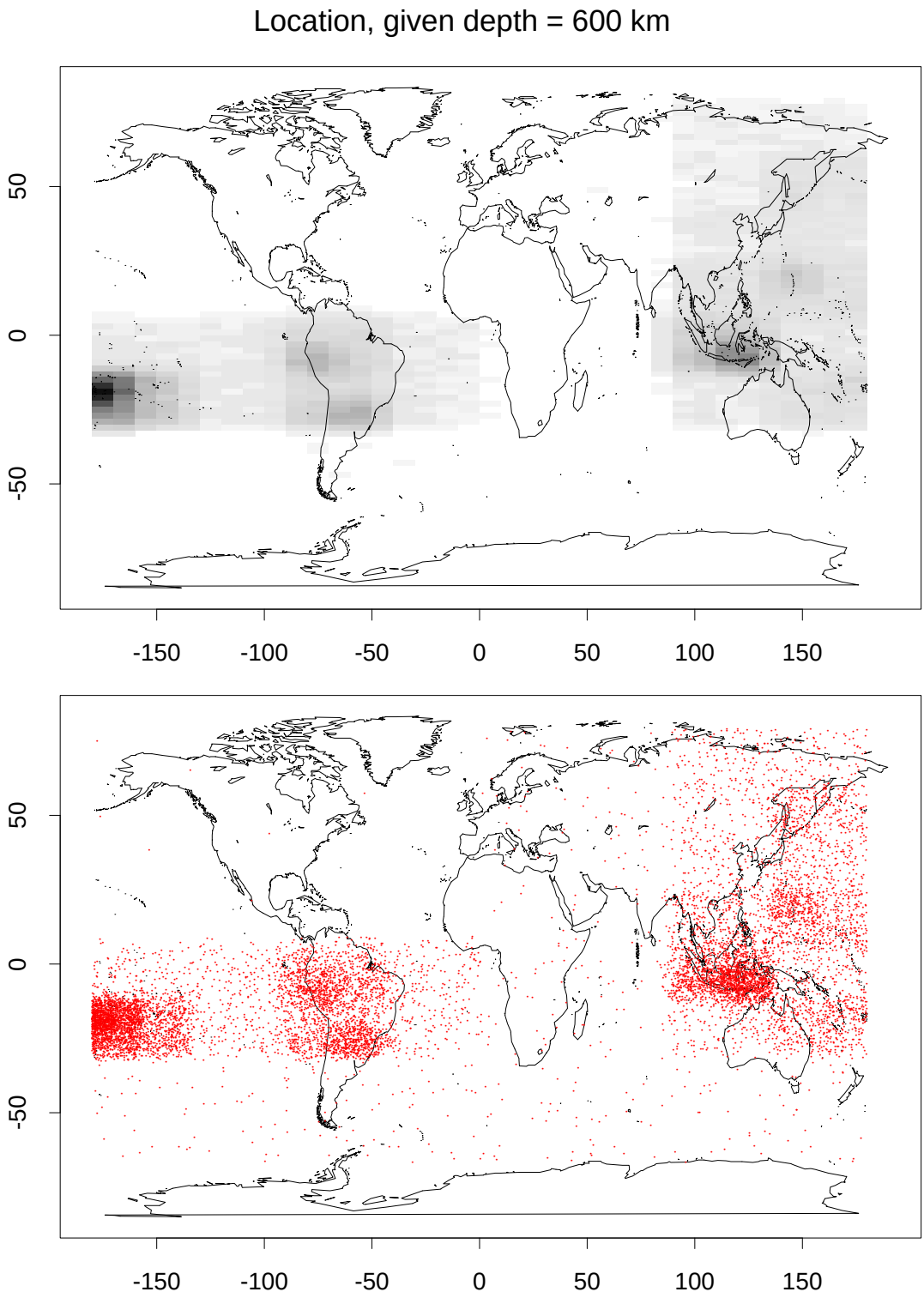


Figure A.25: Section 4.1.3: Conditional predictive distribution of location given depth = 600 km. $\alpha = 0.1m^2$, $\tau = 0.025$, and G =Uniform.

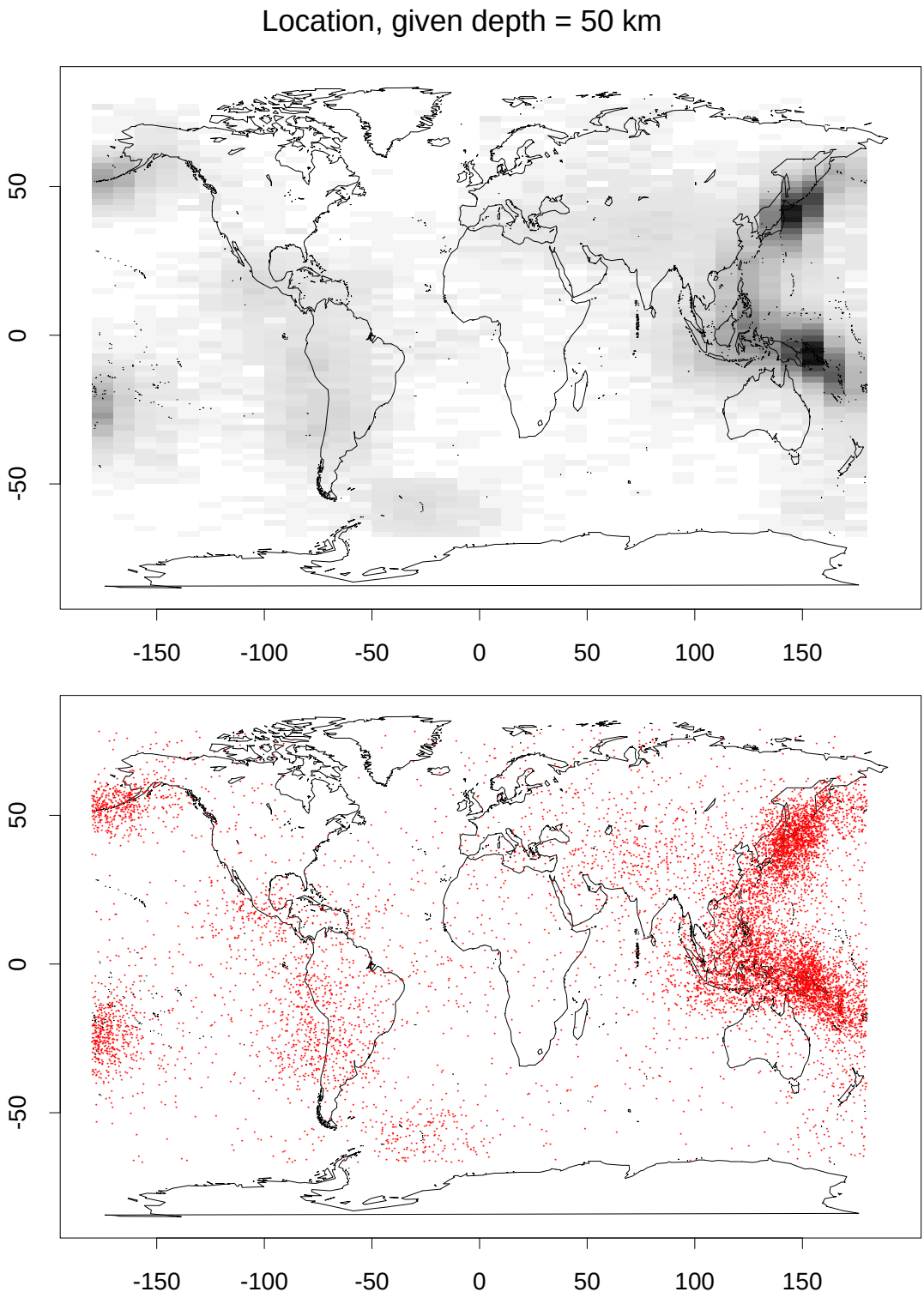


Figure A.26: Section 4.1.3: Conditional predictive distribution of location given depth = 50 km. $\alpha = 0.1m^2$, $\tau = 0.05$ and G =Uniform.

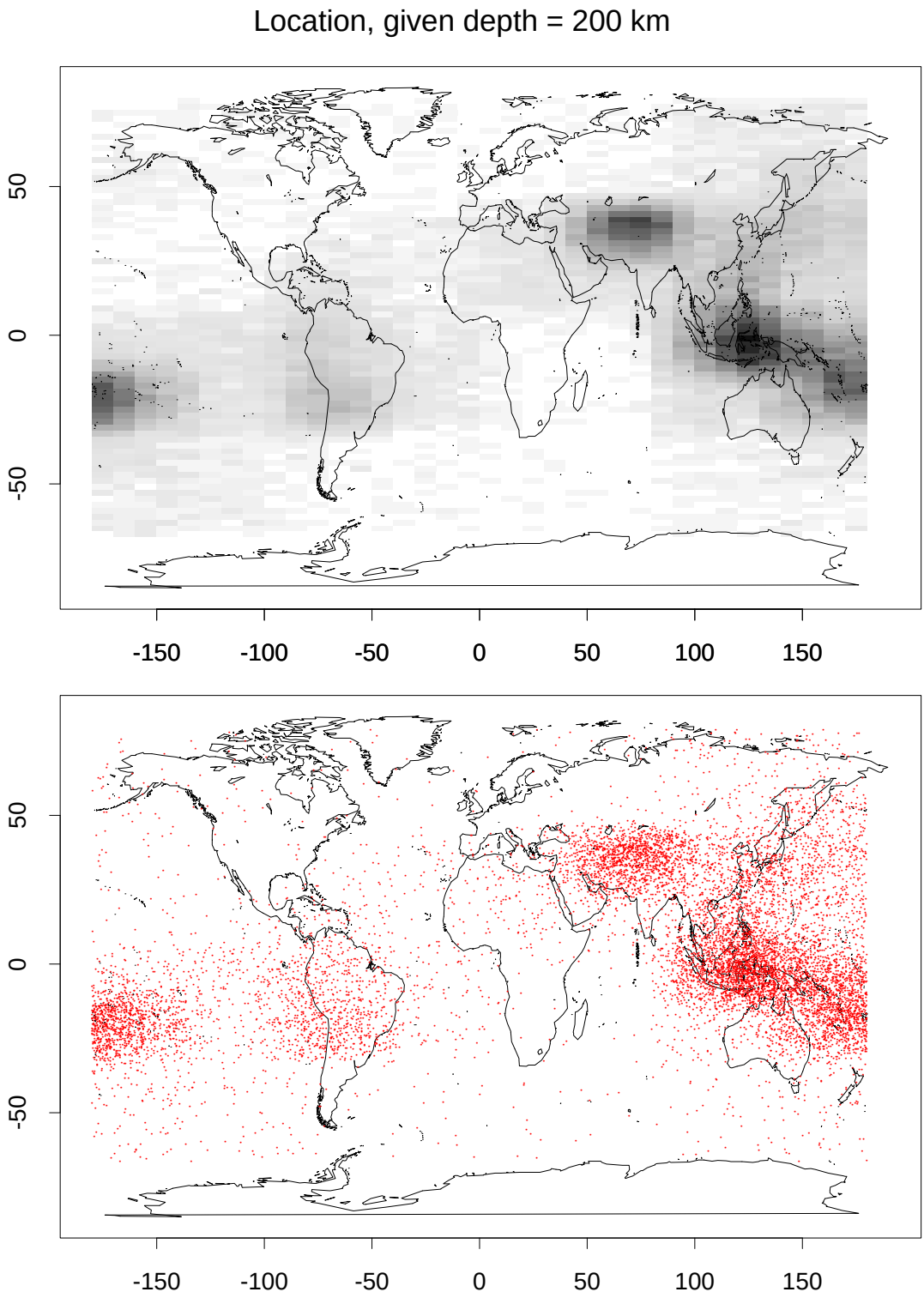


Figure A.27: Section 4.1.3: Conditional predictive distribution of location given dpeth = 200 km. $\alpha = 0.1m^2$, $\tau = 0.05$, and G =Uniform.

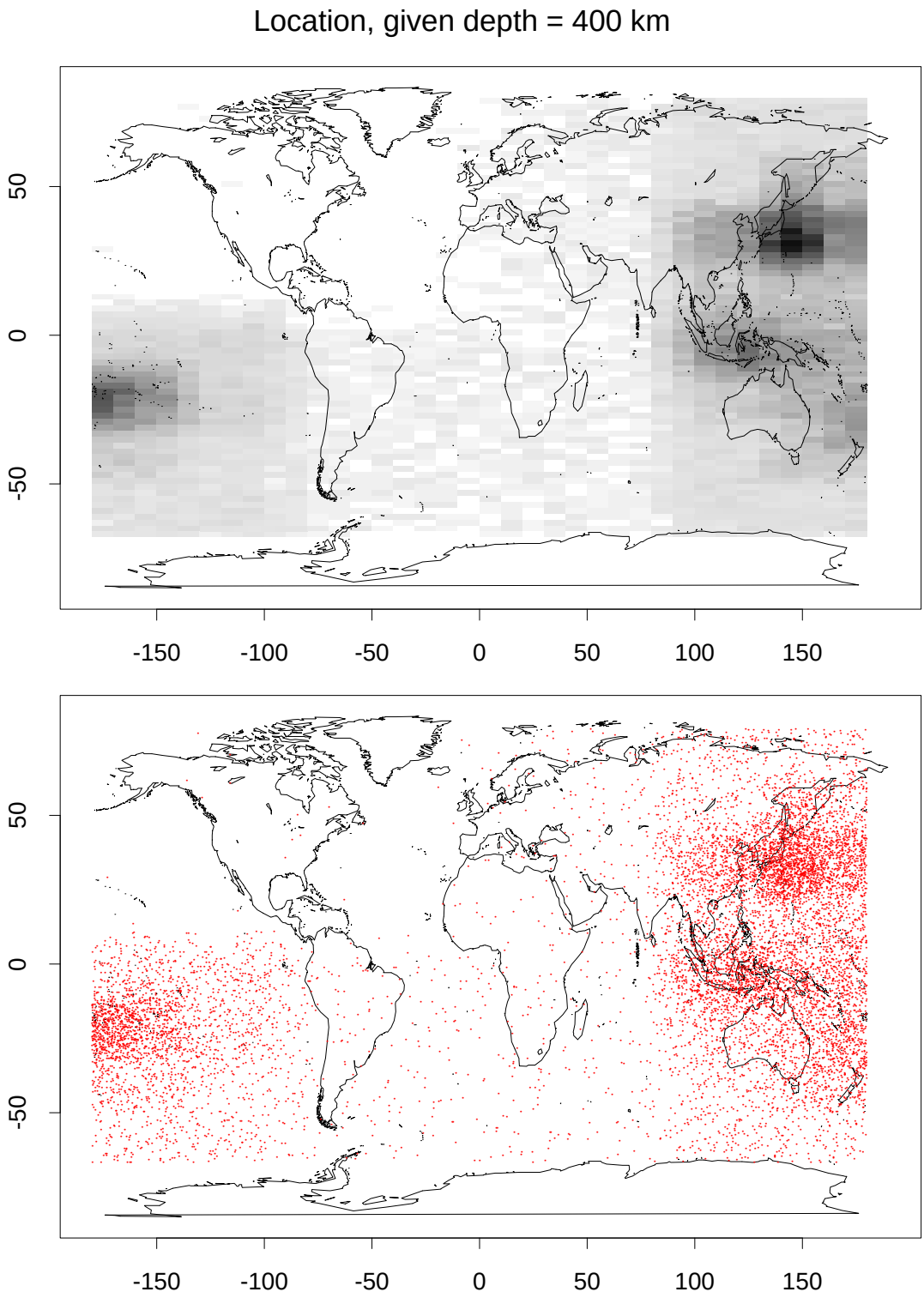


Figure A.28: Section 4.1.3: Conditional predictive distribution of location given depth = 400 km. $\alpha = 0.1m^2$, $\tau = 0.05$ and G =Uniform.

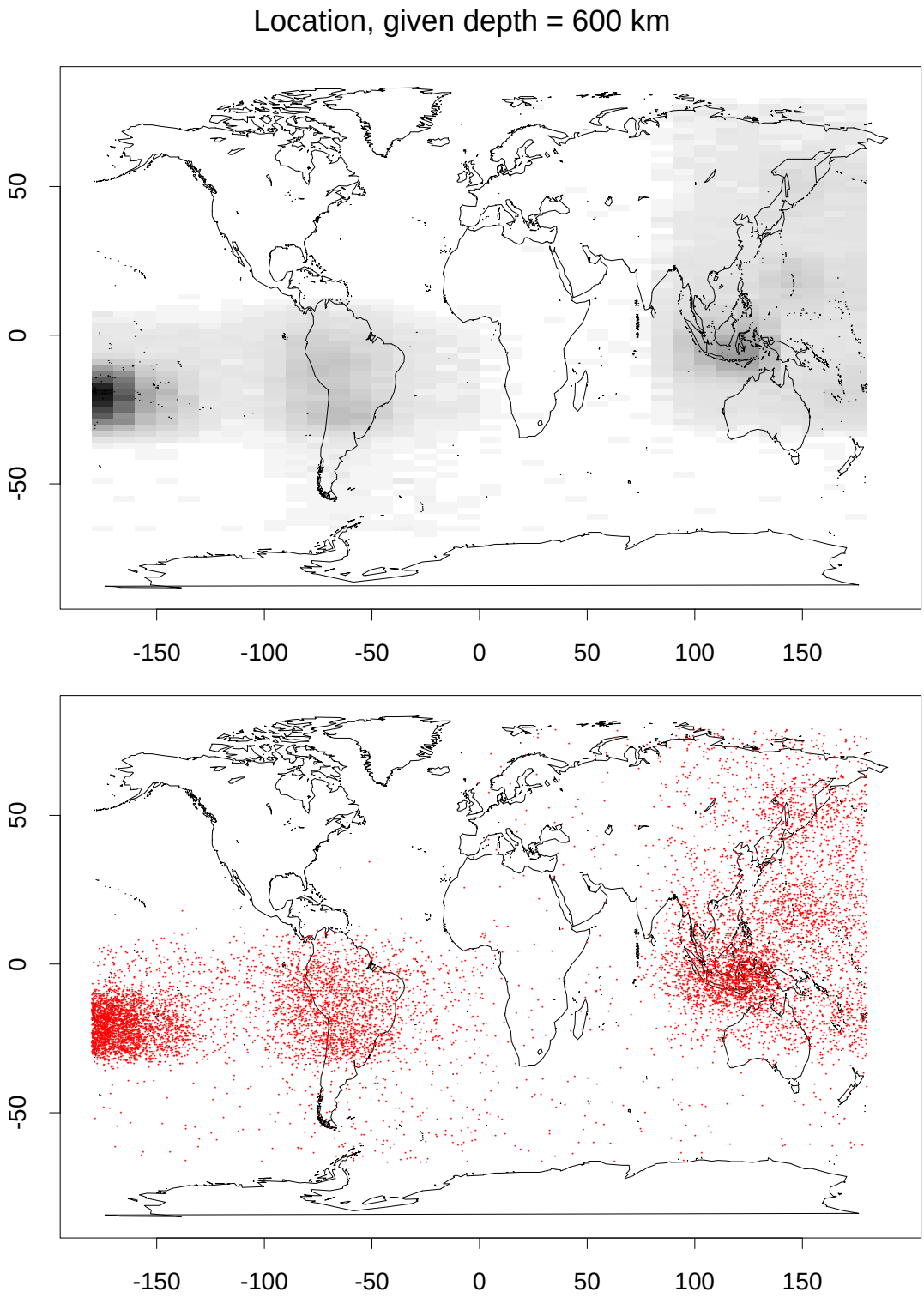


Figure A.29: Section 4.1.3: Conditional predictive distribution of location given depth = 600 km. $\alpha = 0.1m^2$, $\tau = 0.05$, and G =Uniform.

Appendix B

GRAPHS FROM MISSING DATA IMPUTATION EXAMPLE OF CHAPTER 4

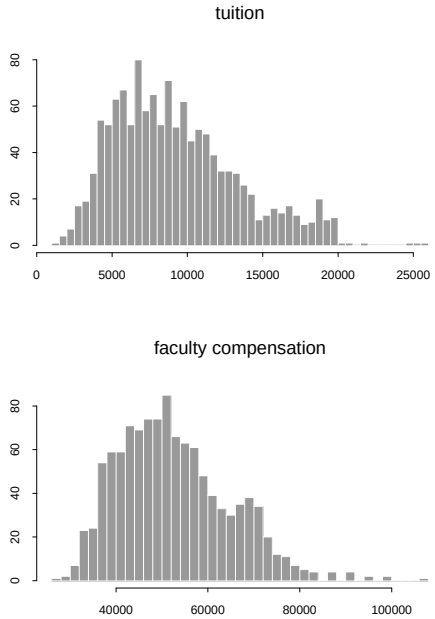


Figure B.1: Histograms of data: tuition and faculty compensation.

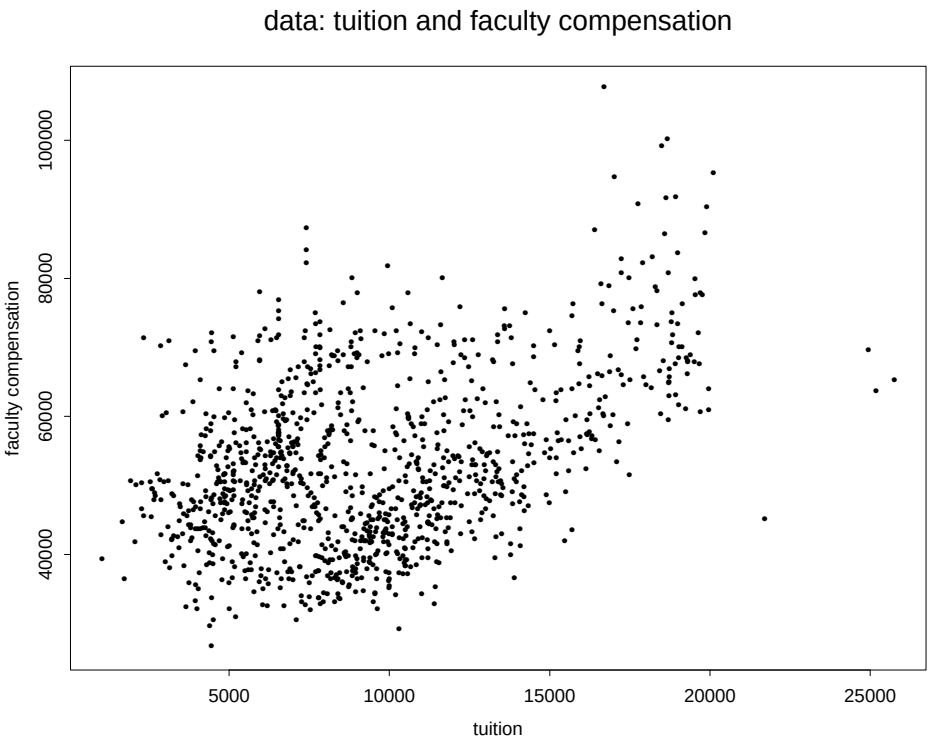


Figure B.2: Scatterplot of data: tuition and faculty compensation.

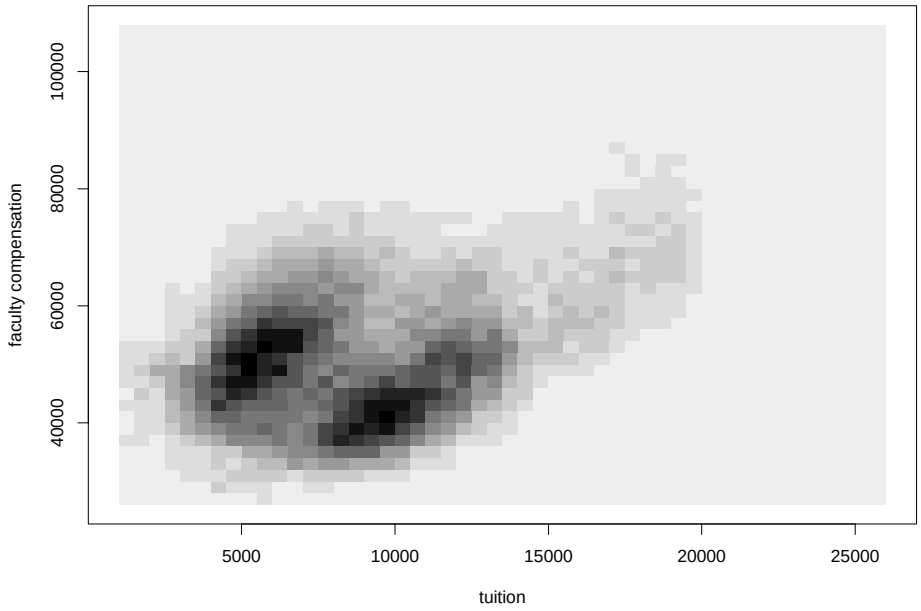


Figure B.3: Simulation of 50000 draws from the posterior predictive distribution for tuition and faculty compensation.

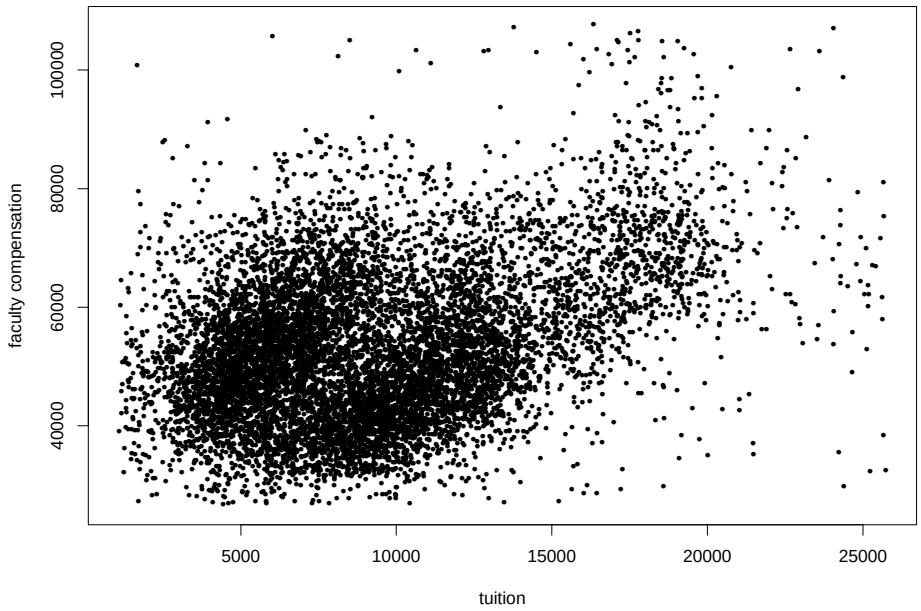


Figure B.4: Subsample of 5000 of 50000 MCMC simulations of the posterior predictive distribution for tuition and faculty compensation

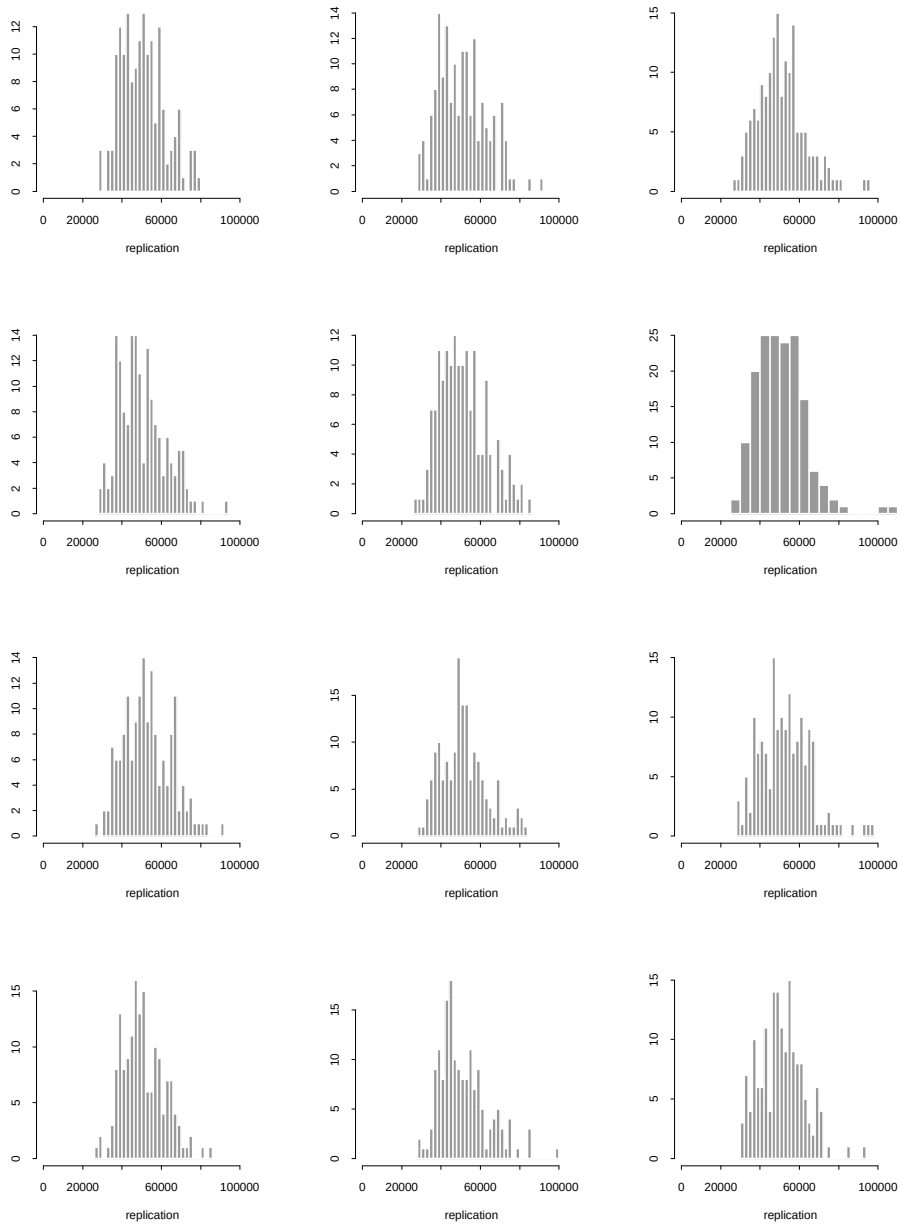


Figure B.5: Twelve randomly-selected replications of $n = 162$ missing values of faculty compensation

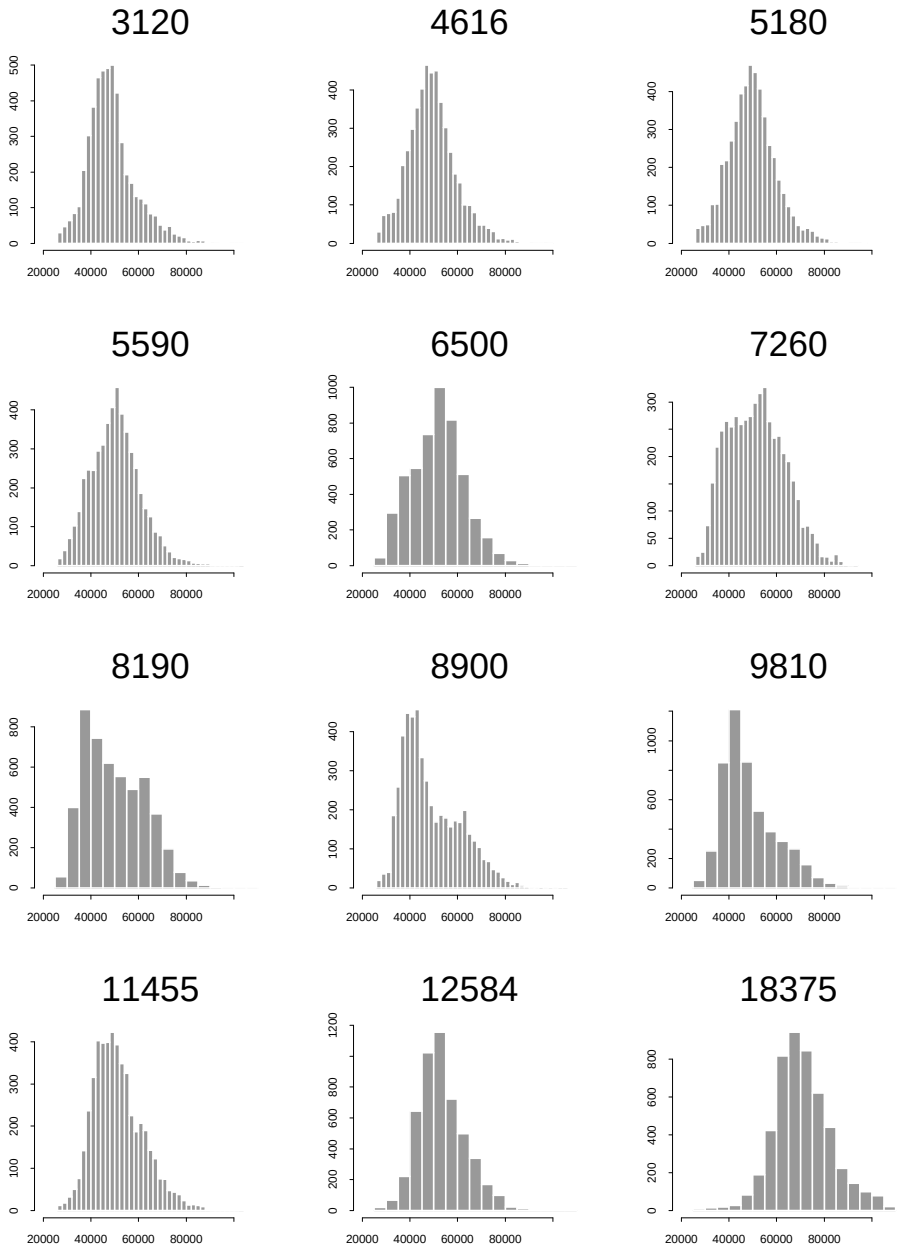


Figure B.6: Imputed Distributions for Twelve Missing Values of faculty compensation. Histograms are labeled with values for tuition

Appendix C

COMPUTATION

Tree structures are very flexible, and can enable efficient computation and storage of information. Thorough overviews of tree algorithms are given by (Aho *et al.*, 1974) and (Cormen *et al.*, 1994). All computations presented in this chapter were programmed in C. A description of some programming strategies follows.

C.1 Structure of Tree

The structure of a tree is given in Figure C.1. Reference to Figure C.1 will be made as definitions are set forth below. **Nodes** are represented by the squares. In the previous chapters, "urns" or "partition elements" constitute the nodes of a tree. The **root** node is the top-most node in the tree, which is labeled $B_\emptyset$, represents the support, Ω, in the Polya tree and randomized tree framework. The **links** are the line segments connecting the nodes. Every node, except the root node, has a **parent** node. **Sibling** nodes have the same parent. Siblings B_0 and B_1 have the same parent, $B_\emptyset$, and are **child** nodes of $B_\emptyset$. Nodes without children are **external** nodes; $B_0, B_{01}, B_{10}, B_{11}$ are external nodes; all others are **internal** nodes. A **NULL** node is one that does not exist; B_{00}, which would appear as a left-hand child of B_0, is NULL. A **subtree** rooted at node B is a tree consisting of node B and its descendants (child nodes, and in

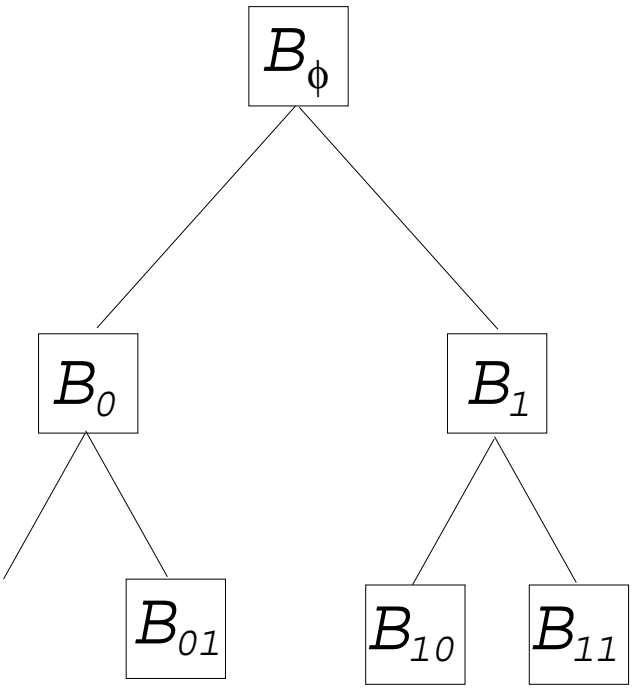


Figure C.1: Basic structure of a tree

turn child nodes of the child nodes, etc.).

The distance of the node from the root node plus 1 is the **level** of the tree. The **degree** of a node is the number of children it has; B_0 is of degree 1 whereas the other internal nodes are of degree 2. The tree **height** is the maximum number of links between the root and external nodes plus 1; the tree in Figure C.1 is of height 2.

C.2 Employing the Tree Structure in Computation

It must be decided which information should be encoded into the tree prior to its creation. In the Polya tree, each node represents a partition element, or an urn. Parameters describing the tree, namely $\mathcal{A}$ and node level, must be stored in the nodes. To enable tree traversal for parameter updating and information extraction,

each node must contain **pointers** to child and parent nodes; pointers are memory addresses indicating where an object can be found in computer memory. The pointers serve as signposts that provide a sense of direction in the tree. Each node must contain the following information:

- Pointer to parent node
- Pointers to child nodes
- A set of $\alpha \in \mathcal{A}$ parameters for the node
- Level in tree of node

The C programming language allows for user-defined **structures** to be used as data types. For the implementation of algorithms presented in this thesis, new data types for trees and nodes were created. The code that was developed for the analyses presented for this thesis is based on software written by Michael Lavine (Lavine, 1999). For example, the structure `URN` is created by declaring the following in a header file:

```
struct URN {
     int              level;
     double           * alpha;
     struct URN       * parent;
     struct URN       * child[N];
};
```

where $*$ `variable` represents a pointer to `variable`. The urn contains an array of N pointers to N child nodes, a pointer to the parent node, and a pointer to a parameter vector α. To access an element (*e.g.*, `level`) of a variable `urn` of type URN, call `level` by "`urn`$\rightarrow$`level`"in the code.

C.3 Moving About the Tree

Moving about the tree is possible once the parent/child and other node information is encoded. There are two general reasons to traverse the tree:

- To update information stored in the nodes

  Since nodes are created whenever data exist in partition elements represented by the nodes, creation of the tree coincides with the first aim. Posterior updates of $\mathcal{A}$ also occur in this step.

- To extract information from the nodes

  Following creation of a tree and updating of its nodes, simulation of the joint posterior distribution can occur. Algorithms for both follow.

C.3.1 Simultaneous Creation of Tree and Posterior Updating of Parameters

An illustration of the computing algorithm follows for a Polya tree. Let N be the number of observations, M be the maximum depth, or height, of the tree. Let root be the root node of the tree, and urn.ctr be an index of the nodes. Let $\Theta_i(v) = (\theta_i(1,v), \ldots, \theta_i(m,v))$ be the M-length 0/1 vector corresponding to $x(i,v)$ – variable v of observation i. Let cut.pt be the cut point at which to split the node to create child nodes.

Algorithm for simultaneous tree creation and posterior updating of parameters

for i in 1:N{

read in observation x_i

```
set urn.ctr = root
for j in 1:M{
  index = 0
  if urn.ctr is NULL
     create urn.ctr
  for k in 1:V {
     compute cut.pt(k) for urn.ctr
     if x(i,k) ≤ cut.pt(k)
        set θ_i(j,k) = 0
     else
        set θ_i(j,k) = 1
     index = index + 2^{K-k-1} θ_i(j,k)
     }
     increment urn.ctr→ α(index + 1) by 1
  set urn.ctr = urn.ctr→child(index +1)
  }
}
```

At each level m of the tree, let c_{km} be the cut point of the urn along axis k. For all k, if $x_k > c_{km}$, set $\theta_{km} = 1$; else $\theta_{km} = 0$. Update the components of α corresponding to Θ_m. Descend to level $m+1$ to the child urn indexed by $\sum_{k=1}^{K} \theta_{km} 2^{K-k}$, and repeat until the bottom of the tree is reached.

Nodes are created only when an observation falls in a region of the space corresponding to the partition element; other nodes remain NULL, and the prior distribution is assumed for the partition elements represented by those nodes. Creation of a node means memory is set aside for the node, and values for α, level, and pointers to parent and child nodes are entered into the newly-created node object.

C.3.2 Posterior Predictive Simulation

For a given tree structure, a point along its support can be drawn from the joint posterior predictive distribution the following way:

Algorithm for simulation from joint posterior predictive distribution

Update the tree as previously described. Given the updated tree, simulate a draw from the predictive distribution by simulating a new observation x_{n+1} by simulating a new $\epsilon \leftrightarrow \Theta$ at each level m, based on the multinomial probability vector:

$$Pr(\epsilon_1, \ldots, \epsilon_{2^K}) = \frac{Y_{\varepsilon_{m-1}\bullet}}{\sum_{\epsilon=0}^{2^K-1} Y_{\varepsilon_{m-1}\epsilon}} \tag{C.1}$$

Then, descend to the next level $m+1$ of the tree by selecting the ϵ^{th} child urn.

```
set urn.ctr = root
  for j in 1:M {
    set index = 0
     for k in 1:V {
       if urn.ctr is not NULL {
            compute predictive probability of θ_{n+1}(j,k) = 0 via vector urn.ctr→ α
```

draw $\theta_{n+1}(j,k) = 0$ or 1 based on above step

}

else if urn.ctr is NULL

draw $\theta_{n+1}(j,k) = 0$ or 1 based on assumed prior values of α

index = index + $2^{K-j-1}\theta_{n+1}(j,k)$

}

if urn.ctr→child(index + 1) not NULL

set urn.ctr = urn.ctr→child(index + 1)

}

for k in 1:V, compute $x_{n+1}(k) = f(\Theta_{n+1}(:,k))$

The last step in the above scheme, computation of $x_{n+1}(k)$, can be evaluated via Equation 3.2; for the Polya tree prior described in Chapter 2, assume all $\{\beta\}$ in Equation 3.2 are equal to 0.5.

Conditional Predictive Distribution Simulation and Missing Data Imputation

Given the updated tree, simulation from the conditional predictive distribution of $X_1, \ldots, X_j$ given $X_{j+1}, \ldots, X_K$ (or imputation of missing components $X_1, \ldots, X_j$ given observed components $X_{j+1}, \ldots, X_K$ is similar in spirit to simulation from the joint posterior distribution, with the following modifications:

a) the multinomial probability vector in Equation C.1 is now of length 2^j rather than 2^K (where j is the dimension of $X_1, \ldots, X_j$) and it represents only those urns which are allowed given knowledge of $X_{j+1}, \ldots, X_K$.

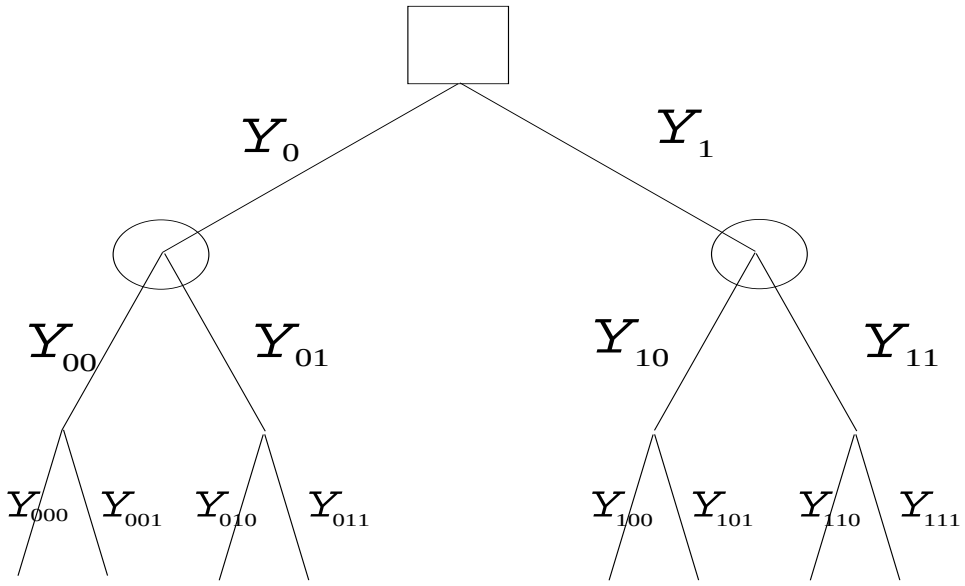


Figure C.2: Subtrees of tree are rooted at the circle nodes

b) The probability calculation includes an additional term to allow for the "tail" of $X_{j+1}, \ldots, X_K$.

At each level c, the factor can be computed by summing the probability of all possible subtrees induced by $\{\theta^{(l)}_{c+1:m}\}_{l=1}^{j}$. For example, suppose $c = 1$, corresponding to the (square) root node in Figure C.2. The probabilities of the two corresponding subtrees at the circled nodes can be computed via recursive traversal of the tree as $Y_{00}(Y_{000} + Y_{001}) + Y_{01}(Y_{010} + Y_{011})$ and $Y_{10}(Y_{100} + Y_{101}) + Y_{11}(Y_{110} + Y_{111})$.

Simulation Scheme for the Conditional Predictive Distribution and Missing Data Imputation

At level c:

loop over $i = 1, \ldots, 2^j$ possible values for $\{\Theta^{(l)}\}_{l=1}^{j}$ (call the value $(\{\Theta^{(l)}\}_{l=1}^{j})_i$)

{

if urn $\neq$ NULL

p(i) = probability of the urn corresponding to $\{\Theta^{(l)}\}_{l=1}^{j})_i$ given $\{\Theta^{(l)}\}_{l=j+1}^{K}$

else use default prior values to compute p(i)

compute d = probability induced by subtree for $(\Theta_1, \ldots, \Theta_j)_i$ given $\{\Theta^{(l)}\}_{l=j+1}^{K}$

compute p(i) = p(i) $\times$ d

Select the child urn from a $Multinomial(1; \mathbf{p})$ and continue to next level }

At the end of the loop, a draw from the conditional predictive distribution or an imputation of missing data will result.

C.4 Dynamic Memory Allocation

With the Polya tree, the above simulations can be repeated any number of times with the same tree. For the randomized tree of Chapter 3, the tree itself is a random variable that is numerically integrated with respect to via Markov Chain Monte Carlo (Equation 3.12). This means a new tree must be created at each iteration of the sampling scheme. Each tree can be quite memory consuming (each α parameter is of length 2^K in K-dimensional space), so the memory allocated to the tree must be freed before a new tree can be created. The tree is freed recursively — starting at the root node, loop over all $i = 1, \ldots, 2^K$ child nodes. At each child i, check to see whether its children need to be freed. A sketch of the freechild recursive function is below, for which the input is "urn":

function freechild(urn)

for i in 1 to 2^K

if(urn→child(i) $\neq$ NULL)

freechild(urn→child(i))

freeurn(urn→child(i))

freeurn(urn)

Bibliography

Aho, A. V., Hopcroft, J. E. and Ullman, J. D. (1974) *The Design and Analysis of Computer Algorithms.* Addison-Wesley.

Andreev, A. and Arjas, E. (1996) A note on histogram approximation in Bayesian density estimation. In *Bayesian Statistics 5. Proceedings of the Fifth Valencia International Meeting on Bayesian Statistics* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 487–490. Oxford University Press.

Antoniak, C. E. (1974) Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, **2,** 1152–1174.

Best, N., Cowles, M.K and Vines, K. (1995) CODA: Convergence diagnosis and output analysis software for Gibbs sampling output. MRC Biostatistics Unit, Institute of Public Health, Cambridge University.

Billingsley, P. (1995) *Probability and Measure (Third Edition).* Wiley.

Blackwell, D. (1973) Discreteness of Ferguson selections. *The Annals of Statistics*, **1,** 356–358.

Blackwell, D. and MacQueen, J. B. (1973) Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, **1,** 353–355.

Brooks, S.P. and Gelman, A. (1998) General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, **7,** no. 4, 434–455.

Cormen, T. H., Leiserson, C. E. and Rivest, R. L. (1994) *Introduction to Algorithms.* MIT Press and McGraw-Hill.

DeGroot, M. H. (1986) *Probability and Statistics (Second Edition).* Addison-Wesley.

Dubins, L.E. and Freedman, D.A. (1967) Random distribution functions. In *Proceedings of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability* (eds L.M. LeCam and J. Neyman), pp. 183–214. Univ. of California Press, Berkeley/Los Angeles.

Escobar, M. D. and West, M. (1995) Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, **90,** 577–588.

Fabius, J. (1964) Asymptotic behavior of Bayes' estimates. *The Annals of Mathematical Statistics*, **35,** 846–856.

Ferguson, T. S. (1973) A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, **1,** 209–230.

Ferguson, T. S. (1974) Prior distributions on spaces of probability measures. *The Annals of Statistics*, **2,** 615–629.

Fienberg, S. E., Steele, R. J. and Makov, U. E. (1996) Statistical notions of data disclosure avoidance and their relationship to traditional statistical methodology: Data swapping and loglinear models. In *Proceedings of the Bureau of the Census 12th Annual Research Conference*, pp. 87–105. Bureau of the Census, Washington, DC.

Freedman, D. A. (1963) On the asymptotic behavior of Bayes' estimates in the discrete case. *The Annals of Mathematical Statistics*, **34,** 1386–1403.

Frohlich, C. and Davis, S.D. (1990) Single-link cluster analysis as a method to evaluate spatial and temporal properties of earthquake catalogues. *Geophysical Journal International*, **100,** 19–32.

Gelfand, A. E. and Smith, A. F. M. (1990) Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, **85,** 398–409.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (Ed) (1996) *Markov Chain Monte Carlo in Practice*. Chapman & Hall.

Hartigan, J.A. (1996) Bayesian histograms (with Discussion). In *Bayesian Statistics 5. Proceedings of the Fifth Valencia International Meeting on Bayesian Statistics* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 211–222. Oxford University Press.

Johnson, N.L. and Kotz, S. (1976) *Distributions in Statistics: Continuous Multivariate Distributions*. Wiley.

Kraft, C. H. (1964) A class of distribution function processes which have derivatives. *Journal of Applied Probability*, **1,** 385–388.

Kuo, L. (1986) Computations of mixtures of Dirichlet processes. *SIAM Journal on Scientific and Statistical Computing*, **7,** 60–71.

Lavine, M. (1992) Some aspects of Polya tree distributions for statistical modelling. *The Annals of Statistics*, **20,** 1222–1235.

Lavine, M. (1994) More aspects of Polya tree distributions for statistical modelling. *The Annals of Statistics*, **22,** 1161–1176.

Lavine, M. (1996) Discussion of 'Bayesian approaches to non- and semiparametric density estimation' by N.L. Hjort. In *Bayesian Statistics 5. Proceedings of the Fifth Valencia International Meeting on Bayesian Statistics* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), p. 250. Oxford University Press.

Lavine, M. (1998) Another look at conditionally Gaussian Markov random fields. Technical Report. ISDS, Duke University, Discussion Paper: 98-04.

Lavine, M. (1999) Personal communication.

Lavine, M. and Lozier, S. (1998) A Markov random field spatio-temporal analysis of ocean temperature. Technical Report. ISDS, Duke University, Discussion Paper: 97-11.

Little, R. J. A. and Rubin, D. B. (1987) *Statistical Analysis With Missing Data.* Wiley.

Lo, A. Y. (1984) On a class of Bayesian nonparametric estimates: I. Density estimates. *The Annals of Statistics*, **12,** 351–357.

Mauldin, R. D., Sudderth, W. D. and Williams, S. C. (1992) Polya trees and random distributions. *The Annals of Statistics*, **20,** 1203–1221.

Mauldin, R.D. and Williams, S.C. (1990) Reinforced random walks and random distributions. In *Proceedings of the American Mathematical Society*, vol. 110, pp. 251–258. American Mathematical Society.

Métivier, M. (1971) On the construction for random measures almost surely absolutely continuous with respect to a given measure (French: Sur la construction de mesures aléatoires presque sûrement absolument continues par rapport à une mesure donnée). *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, **20,** 332–334.

Müller, P., West, M. and MacEachern, S. (1997) Bayesian models for non-linear autoregressions. *Journal of Time Series Analysis*, **18,** 593–614.

Ruggeri, F. and West, M. (1999) Time series and Polya trees. Unpublished manuscript, Institute of Statistics and Decision Sciences, Duke University, Durham NC.

Schervish, M. J. (1995) *Theory of Statistics.* Springer-Verlag.

Simonoff, J. S. (1996) *Smoothing Methods in Statistics.* Springer-Verlag.

Smith, B.J. (1999) Bayesian output analysis program (BOA), version 0.4.2 for UNIX S-Plus. Department of Biostatistics, School of Public Health, University of Iowa.

STAT/LIBRARY, IMSL (1984) Fortran subroutines for statistical analysis. IMSL, Inc.

Tanner, M. A. (1996) *Tools for Statistical Inference (Third Edition).* Springer-Verlag.

Tierney, L. (1994) Markov chains for exploring posterior distributions (with Discussion). *The Annals of Statistics*, **22,** 1701–1728.

Walker, S. G. and Mallick, B. (1997) Hierarchical generalized linear models and frailty models with Bayesian nonparametric mixing. *Journal of the Royal Statistical Society, Series* B, **59,** 845–860.

West, M. (1992) Modelling with mixtures (with Discussion). In *Bayesian Statistics 4. Proceedings of the Fourth Valencia International Meeting* (eds J.O. Berger, J.M Bernardo, A.P. Dawid and A.F.M. Smith), pp. 503– 519. Oxford University Press.

Wolpert, R.L. and Lavine, M. (1995) Markov random field priors for univariate density estimation. In *Bayesian Robustness* (eds J.O. Berger, E. Betro, E. Moreno, L.R. Pericchi, F. Ruggeri and G. Salinetti), pp. 253–270. IMS Press (Hayward, CA).

Biography

Susan M. Paddock was born in Minneapolis, Minnesota on January 11, 1972. She received her B.A. in Biostatistics and Mathematics from the University of Minnesota in the Twin Cities, Minnesota in December 1994, graduating summa cum laude, Phi Beta Kappa. She received her M.S. in Statistics from Duke University in Durham, NC in May 1997. She has co-authored one refereed publication, one refereed conference proceeding, and two technical reports.