Computational Issues in Using Bayesian Hierarchical Methods for the Spatial Modeling of fMRI Data

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Kuo-Jung Lee

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

August, 2010

Acknowledgements

I deeply thank my advisor Galin Jones, for his continuous support in the Ph.D. program. Galin was always there to listen and to give advice. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal and helped me complete the writing of this dissertation as well as the challenging research that lies behind it.

Besides my advisor, I would like to thank the rest of my thesis committee: Peihua Qiu, Hui Zou and Sudipto Banerjee. They asked me good questions, gave insightful comments, and reviewed my dissertation. A special thank goes to Glen Meeden who taught me how to give a good presentation and brought out the good ideas in me.

I thank my family: my parents, my wife, and my daughter. Without their encouragement and constant support, I could not have finished this dissertation.

Abstract

One of the major objectives of fMRI (functional magnetic resonance imaging) studies is to determine which areas of the brain are activated in response to a stimulus or task. To make inferences about task-specific changes in underlying neuronal activity, various statistical models are used such as general linear models (GLMs). Frequentist methods assessing human brain activity using data from fMRI experiments rely on results from the theory of Gaussian random fields. Such methods have several limitations.

The Bayesian paradigm provides an attractive framework for making inference using complex models and bypassing the multiple comparison problems. We propose a Bayesian model which not only takes into account the complex spatio-temporal relationships in the data while still being computationally feasible, but gives a framework for addressing other interesting questions related to how the human brain works. We study the properties of this approach and demonstrate its performance on simulated and real examples.

Contents

Acknowledgements **i**

Abstract **i**

List of Tables **v**

List of Figures **vi**

1 Introduction **1**

1.1 Motivation 1

1.2 Stroop Test and Stroop Data 2

1.2.1 Stroop Test 2

1.2.2 Stroop Data 3

1.3 Overview of fMRI Technique 4

1.4 Overview of Statistical Analysis for fMRI Data 5

1.4.1 General Linear Model 6

1.4.2 Hierarchical Bayes model 7

1.4.3 Markov Chain Monte Carlo Algorithms 9

2 Background **11**

2.1 Basic Physical Principles of MRI 11

2.2 BOLD fMRI 20

2.3 fMRI Experiment Designs 22

2.4 Preparing fMRI Data for Statistical Analysis 23

2.4.1 Slice Timing Correction . . . 25
2.4.2 Motion Correction . . . 26
2.4.3 Coregistration . . . 26
2.4.4 Spatial Normalization . . . 26
2.4.5 Spatial and Temporal Smoothing . . . 28
2.4.6 Segmentation . . . 28

3 Markov Chain Monte Carlo **29**
3.1 Markov Chain Monte Carlo . . . 30
3.1.1 Markov Chain . . . 31
3.1.2 Metropolis-Hastings Algorithm . . . 32
3.1.3 Gibbs Sampling Algorithm . . . 32
3.2 Monte Carlo Error . . . 33
3.2.1 Batch Means . . . 34
3.2.2 Overlapping Batch Means . . . 35
3.2.3 Stopping the simulation . . . 35

4 Normalizing Constant Estimation **37**
4.1 Monte Carlo Algorithms . . . 39
4.1.1 Importance Sampling . . . 40
4.1.2 Umbrella Sampling . . . 41
4.1.3 The Wang-Landau Algorithm . . . 41
4.1.4 The Modified Wang-Landau Algorithm . . . 45
4.1.5 Path Sampling . . . 48
4.1.6 Single-Variable Exchange Algorithm . . . 49
4.1.7 Approximate Bayesian Computation . . . 50
4.2 Simulation Study . . . 51
4.3 Conclusion . . . 54

5 Bayesian Hierarchical Methods for the Spatial Modeling of fMRI Data **60**
5.1 Spatial Bayesian Variable Selection Models . . . 62
5.1.1 Ising Prior and Zellner's g-Prior . . . 63

5.1.2 Posterior density . . . 65
5.1.3 Bayesian Inference via MCMC Sampling . . . 66
5.2 Parameter Estimation . . . 68
5.3 Neighborhood Structures . . . 69
5.4 Threshold . . . 69
5.5 Two-Stage Estimation Procedure . . . 70
5.6 Simulation Study . . . 70
5.7 Stroop Data . . . 74
5.8 Gaussian Conditional Autoregressive Models . . . 95
5.8.1 Bayesian Inference via MCMC Sampling . . . 100
5.8.2 Real Data Analysis . . . 102

References **113**

Appendix A. R Packages **121**
A.1 R-package 'NCIsing': Estimation of the Normalizing Constant in an Ising Model . . . 121
A.2 R-package 'fMRI.SpBVS': Spatial Bayesian Variable Selection for fMRI Time Series Data. . . . 139

List of Tables

4.1 The estimate of θ from seven different algorithms. 52

5.1 The average Monte Carlo estimates of θ and corresponding Monte Carlo standard errors (MCSE) based on 10,000 iteration over 10 simulated data. . 74

5.2 The accuracy are estimated based on 10 simulated data 74

5.3 The estimate of θ and the corresponding MCSE given in the parenthesis . . 77

List of Figures

1.1 The area of parietal lobe in the brain (Taken from Centre for Neuro Skills). . 3
2.1 Spin: A proton rotates around its own axis. Procession: The axis of spin itself wobbles around the main axis of the magnetic field. 12
2.2 Before a magnetic field is applied, the protons are randomly oriented. 13
2.3 When placed in the strong magnetic field, all the protons align in either a parallel or anti-parallel with the direction of the magnetic field (B_0). 14
2.4 A schematic illustration of an application of a radio frequency to magnetization vector. When a RF pulse is applied to the magnetization vector, its direction changes. The radio frequency power of the pulse is proportional to the flip angle through which the spins are tilted under its influence. . . . 15
2.5 A conceptual overview of T_2 decay. 15
2.6 A conceptual overview of T_1 recovery. 16
2.7 The signal is induced in the receiver coil over time. 17
2.8 Slice Selection: A particular slice with the precession frequency of the spins matching FR pulse is imaged. 18
2.9 Frequency Encoding: The use of a magnetic gradient to image two objects at different location. 18
2.10 Phase Encoding: All spins have the same precessional frequency before applying the phase encoding gradient. When the phase encoding gradient is applied, some spins will be precessing faster than others. When it is turned off, all spins have the same frequency again, but different phase. . . 19
2.11 Contrast Mechanisms: The BOLD Effect (Taken from University of Oxford FMRIB Centre Department of Clinical Neurology). 21

2.12 The typical schematic representation of fMRI BOLD hemodynamic response. 23
2.13 (A) A block design. A serial of the same trial is given in each time interval. (B) An event-related design. Tasks are randomly repeated in each block. (C) The stimulus is randomly presented over time. 24
2.14 Slice's acquisition time are different for whole brain, so the BOLD signal sampled at different layers of the brain is at different time points 25
2.15 Effects of head motion on fMRI. The numerical intensity values for the voxels within the red box are shown on the right side. The top one (A) is the image in the red box and corresponding numerical values before head motion. The bottom one (B) is the image in the red box and corresponding numerical values after head motion. 27
4.1 (a) is the logarithm of normalizing constant, (b) the trace plot, (c) the histogram , and (d) the autocorrelation function based on 10,000 simulated samples. 55
4.2 (a) is the logarithm of normalizing constant, (b) the trace plot, (c) the histogram , and (d) the autocorrelation function based on 10,000 simulated samples. 56
4.3 (a) is the logarithm of normalizing constant, (b) the trace plot, (c) the histogram , and (d) the autocorrelation function based on 10,000 simulated samples. 57
4.4 (a) is the logarithm of normalizing constant, (b) the trace plot, (c) the histogram , and (d) the autocorrelation function based on 10,000 simulated samples. 58
4.5 (a) is the trace plot, (b) the histogram , and (c) the autocorrelation function based on 10,000 simulated samples. 59
4.6 (a) is the trace plot, (b) is the histogram , and (c) is the autocorrelation function based on 10,000 simulated samples. 59
5.1 Voxels labeled X are neighbors of the voxel V 69
5.2 Histograms and trace plots for θ in different models for a simulated data. . 73
5.3 Histograms and trace plots for each θ when assuming independence in error terms. 78

5.4 Histograms and trace plots for each θ when assuming AR(1) dependence in error terms. 79
5.5 Predicting activation when performing "InkOnly" task obtained by using Bayesian approach with independent error terms. 80
5.6 Predicting activation when performing "Congruence" task obtained by using Bayesian approach with independent error terms. 80
5.7 Predicting activation when performing "Interference" task obtained by using Bayesian approach with independent error terms. 81
5.8 Predicting activation when performing "Ink Only" task obtained by using Frequentist approach with independent error terms. 81
5.9 Predicting activation when performing "Congruence" task obtained by using Frequentist approach with independent error terms. 81
5.10 Predicting activation when performing "Interference" task obtained by using Frequentist approach with independent error terms. 82
5.11 Predicting activation when performing "Ink Only" task obtained by using Bayesian approach with AR(1) dependence for error terms. Two-stage estimation approach is used to estimate the posterior probabilities. 82
5.12 Predicting activation when performing "Congruence" task obtained by using Bayesian approach with AR(1) dependence for error terms. Two-stage estimation approach is used to estimate the posterior probabilities. 82
5.13 Predicting activation when performing "Interference" task obtained by using Bayesian approach with AR(1) dependence for error terms. Two-stage estimation approach is used to estimate the posterior probabilities. 83
5.14 Predicting activation when performing "Ink Only" task obtained by using Frequentist approach with AR(1) dependence for error terms. 83
5.15 Predicting activation when performing "Congruence" task obtained by using Frequentist approach with AR(1) dependence for error terms. 83
5.16 Predicting activation when performing "Interference" task obtained by using Frequentist approach with AR(1) dependence for error terms. 84

5.17 Predicting activation from the frequentist approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the first trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 86
5.18 Predicting activation from the frequentist approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the second trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 87
5.19 Predicting activation from the frequentist approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the third trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 88
5.20 Predicting activation from the Bayesian approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the first trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 89
5.21 Predicting activation from the Bayesian approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the second trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 90
5.22 Predicting activation from the frequentist approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the third trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 91

5.23 Predicting activation from the frequentist approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the first trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 92
5.24 Predicting activation from the frequentist approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the second trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 93
5.25 Predicting activation from the frequentist approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the third trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 94
5.26 Predicting activation from the Bayesian approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the first trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 95
5.27 Predicting activation from the Bayesian approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the second trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 96
5.28 Predicting activation from the Bayesian approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the third trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 97

5.29 Predicting activations corresponding to different tasks obtained from the frequentist approach assuming independent error terms. 104
5.30 Predicting activations corresponding to different tasks obtained from the frequentist approach assuming AR(1) dependence in the error terms. 104
5.31 Predicting activation from the Bayesian variable selection approach with spatial relationship modeled by an Ising distribution. 105
5.32 Predicting activation from the Bayesian variable selection approach with spatial relationship modeled by the Gaussian autoregression model (CAR1). 105
5.33 Predicting activation from the Bayesian variable selection approach with spatial relationship modeled by the Gaussian autoregression model (CAR2). 106
5.34 Predicting activation from the Bayesian approach when performing "Ink Only", "Congruence," and "Interference" by using Ising distribution to model spatial correlation and assuming independence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 107
5.35 Predicting activation from the Bayesian approach when performing "Ink Only", "Congruence," and "Interference" by using Ising distribution to model spatial correlation and assuming AR(1) dependence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 108
5.36 Predicting activation from the Bayesian approach when performing "Ink Only", "Congruence," and "Interference" by using CAR2 to model spatial correlation and assuming independence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 109
5.37 Predicting activation from the Bayesian approach when performing "Ink Only", "Congruence," and "Interference" by using CAR2 to model spatial correlation and assuming AR(1) dependence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 110

5.38 Predicting activation from the frequentist approach when performing "Ink Only", "Congruence," and "Interference" assuming independence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 111
5.39 Predicting activation from the frequentist approach when performing "Ink Only", "Congruence," and "Interference" assuming AR(1) dependence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task. 112
A.1 Given a $\theta = 0.3$, 32×32 2D Ising images are generated using the perfect sampling with different algorithms. 123
A.2 32×32 2D Ising images with different values of θ generated using the perfect sampling. 125
A.3 Estimate of logarithm of the normalizing constant, trace plot, histogram and ACF plots for the sample generated using path sampling. 128
A.4 Estimate of logarithm of the normalizing constant, trace plot, histogram and ACF plots for the sample generated using the modified Wang Landau algorithm. The plot of "Estimate of log Z" shows the corresponding estimate of logarithm of the normalizing constant at different values θ in [0, 1] up to a constant. Only the last 1000 sample points are shown in the trace plot. 131
A.5 Estimate of logarithm of the normalizing constant, trace plot, histogram and ACF plots for the sample generated using the modified Wang Landau algorithm. The plot of "Estimate of log Z" shows the corresponding estimate of logarithm of the normalizing constant at different values θ in [0, 1] up to a constant. Only the last 1000 sample points are shown in the trace plot. 134
A.6 Trace plot, histogram and ACF plots for the simulated sample using single variable exchange algorithm. 136
A.7 Trace plot, histogram and ACF plots for the simulated sample using the approximate Bayesian computation algorithm. 138

Chapter 1

Introduction

1.1 Motivation

Alzheimer's disease (AD) is a fatal brain disease that gradually destroys a person's memory and eventually causes loss of ability to learn and to carry out daily activities such as talking, eating, or going to the bathroom. It is estimated that more than 5 million Americans may have AD making it one of the major health problems in the United States [1]. No current cure of this disease is available, but several treatments have been developed to help patients maintain mental function and manage behavioral symptoms. More importantly, early diagnosis of AD is critical for adequate treatment and care which could slow, delay, or prevent the disease. pathologic changes most likely start years or even decades before the manifestation of clinical symptoms in AD. This long asymptomatic phase of AD provides a potential period for early therapeutic interventions to slow and perhaps ultimately prevent the progression to AD.

Functional magnetic resonance imaging (fMRI) offers a unique in vivo way to investigate the functionality of a human brain and hence might provide early diagnostic options. Most fMRI studies in AD published to date have focused on particular areas such as the medial temporal lobe (MTL), including hippocampus and neighboring parahippocampal cortices, a region critical for memory function. This area implicated early in the course of AD. Moreover, several fMRI studies investigate alteration of brain activation patterns in different regions in asymptomatic individual from autopsy-confirmed late-onset familial

AD cases and matched controls [2], [3], and [4]. These studies used a variety of visually presented stimulus to be memorized, including word-pair-associate and geometric shapes. They found the activation in response to these memory paradigms significantly different in these two different groups. These findings suggest that functional brain imaging combined with behavioral tests can identify preclinical changes that may predict AD. Therefore, fMRI has become a prevailing tool to examine alterations in brains function related to the earliest symptoms of AD possibly before development of significant irreversible structural damage.

1.2 Stroop Test and Stroop Data

1.2.1 Stroop Test

To better understand the neural activity of individuals at risk for AD we therefore implemented a block fMRI version of the Stroop task. The Stroop task [5] has been used for many years as test that exploits the conflicts between one well-learned or automatic behavior (e.g. reading) and a decision rule that requires this behavior to be inhibited. It is interference in the completion of a task caused by one area of the brain dominating and inhibiting the response of other functional areas. Many previous behavioral studies have established the features of the Stroop task that produce cognitive interference. Furthermore, recent neuroimaging studies [6], [7], [8], [9], [10], [11] have indicated that several brain regions are involved in the performance of the Stroop task, although these imaging studies do not all agree on which brain areas are most centrally involved in resolving Stroop inference.

In this fMRI experiment, we have data collected on over 200 patients participating in a study of AD progression at Johns Hopkins University. All subjects are older, generally well-educated, and healthy. None of them have any clinically diagnosed neurologic disorders or Huntington's disease, that would impact our modeling decisions. However, some of the subjects are at high risk for AD. Each subject performed the following Stoop task: Subjects are shown words and subsequently asked to press a button corresponding to the color ink when a word is shown in the scanner. There are three different types of tasks:

- Ink only - the word is XXXX(i.e. not really a word at all)

- Congruence - the word is the color of the ink. For example, the word might the **blue** written in **blue** letters.
- Interference - the word is a different color from the color of the ink. For example, the word might be **blue** written in **red** ink.

There is an important cognitive mechanism involved in this task, specifically, directed attention. Since most people are very proficient at reading words it takes effort to ignore them and concentrate on the color. This test is a standard measure in neurophysiological assessment for measuring cognitive processing. Accordingly, the Stroop interference design is supposed to activate the parietal lobe in the brain as shown in Figure 1.1. The parietal lobe is associated with cognition, information processing, spatial orientation, speech and visual perception.

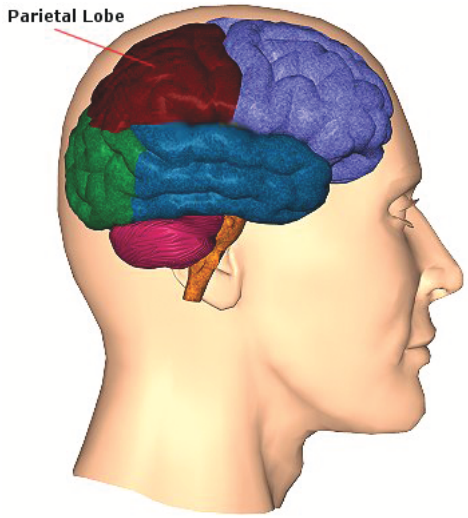


Figure 1.1: The area of parietal lobe in the brain (Taken from Centre for Neuro Skills).

1.2.2 Stroop Data

The images are collected while the subject performs the Stroop test inside the scanner. Subjects participating in this experiment should not have metal in their heads, shouldn't

be claustrophobic or pregnant, and be able to lie still for up to 2 hours. High-resolution 3D images with each voxel's size $2mm^3$ were collected while the subject in the scanner performed the Stroop test. In each run, images are taken approximately every 2 seconds for up to 2 hours. Each image consists of data on $128 \times 128 \times 24$ voxels which translates to slightly more than 3MB of storage of double-precision real numbers and there are often more than 200 images in each session. There may be several runs and there may be many subjects in each session. Because we know the general region of the brain where activation should occur we do not need to store the entire image and hence the images have been trimmed so that the dimension is $79 \times 95 \times 68$ in each time point. However, these studies may contain 200+GB of data.

The goal of this study is to address two major scientific questions: Which regions of the brain are activated and does the activation pattern change over time when subjects at-risk for AD perform the Stoop test? The latter is of interest since a subject may get better at the Stroop test with practice, the activation of the brain during this task might change over time.

1.3 Overview of fMRI Technique

FMRI is a non-invasive technique useful for measuring human brain activity in terms of the changes in the ratio of oxygenated to deoxygenated blood measured via the hemodynamic response. When a brain area becomes active in response to a particular task or stimulus, it consumes more oxygen which causes changes in blood oxygenation. In other words, an increase in blood flow to active brain regions results in a higher concentration of oxygenated blood and therefore results in the changes of oxygenated and deoxygenated blood levels. This remarkable feature of brain metabolism is tightly linked to local neural activity. This phenomenon can be used to detect brain activity with an MRI machine and is known as blood oxygenation level dependent (BOLD) fMRI.

The BOLD signal in a voxel is actually coming from the total amount of both deoxygenated hemoglobin and noise arising from a variety of sources. Several studies [12] and [13] have shown an early negative response due to a transient increase in the amount of deoxygenated hemoglobin. After a dip of 1 to 2 s duration, the metabolic demands of

active neurons over baseline levels result in an increased inflow of oxygen in blood. More oxygen is supplied to the area than is demanded, and this gives rise to a decrease in the amount of deoxygenated hemoglobin within the voxel. In general, the signal increases above baseline at about 2s following the onset of neuronal activity, peaking at about 6s for a short-duration stimulus. If the neuronal activity is extended across a block of time, the peak may be similarly extended into a plateau. This delay and blurring is modeled by a hemodynamic response function (HRF). Hence, the stimulus function is then convolved with the assumed or modeled HRF to give the assumed BOLD response, in turn, as a column vector of a design matrix in linear regression model. Thus one can predict the BOLD fMRI signal change that could result from any arbitrary pattern of neural activity. Consequently, in terms of these prediction of BOLD fMRI, we can identify the neuronal activity triggered by performing a task in a human brain.

This BOLD fMRI has gained popularity for studying the human brain in the last several years because fMRI has considerably higher spatial and temporal resolution compared to other imaging techniques like PET and there is minimal known risk.

1.4 Overview of Statistical Analysis for fMRI Data

One of the major objectives of fMRI studies is to determine which areas of the brain are activated in response to a stimulus or task. To make inferences about task-specific changes in underlying neuronal activity, various statistical models are used such as general linear models (GLMs) [14]. In the GLM framework, the fMRI time series in each voxel is modeled independently by a linear combination of several regressors corresponding to some experimental effects. The error term in GLM is assumed to be a stochastic process such as an autoregressive model of small order to account for the temporal correlation arising from the fact that stimuli are presented continuously or periodically over time.

1.4.1 General Linear Model

A typical GLM of the time-series of an individual voxel in 3D volume data set may be expressed as

$$\mathbf{y}_v = \mathbf{X}_v\boldsymbol{\beta}_v + \boldsymbol{\varepsilon}_v, \quad \varepsilon_v \sim \mathcal{N}(0, \sigma_v^2\Lambda_v), \tag{1.1}$$

where $\mathbf{y}_v = (y_{v,1}, y_{v,2}, \ldots, y_{v,T_v})$ is a vector of fMRI signals at each voxel position v, $v = 1, 2, \ldots, N$, where T_v is the number of time points in voxel v. The $\mathbf{X}_v$ is a design matrix which is the stimulus convolved with the HRF, $\boldsymbol{\beta}_v = (\beta_{v,1}, \ldots, \beta_{v,p})$ is a time-invariant parameter vector of p regression coefficients corresponding to effects of stimuli at voxel v. The error, $\boldsymbol{\varepsilon}_v$, is assumed to be normally distributed with covariance matrix as $\sigma_v^2\Lambda_v$ which specifies the temporal correlation between observations.

To see if voxels are activated in response to a given task, hypothesis testing is then performed in the usual manner. Suppose we are more interested in detecting whether the neuron has a response to the stimulus given. An effect of interest may be compared to a null hypothesis of no effect using a contrast vector, c, to describe the effect of interest. Specifically the effect of interest is modeled by $c\boldsymbol{\beta}_v$, and estimated by $c\hat{\boldsymbol{\beta}}_v$, where $\hat{\boldsymbol{\beta}}_v$ is a point estimate of $\boldsymbol{\beta}_v$ given by

$$\hat{\boldsymbol{\beta}}_v = (\mathbf{X}_v^T\Lambda^{-1}\mathbf{X}_v)^{-1}\mathbf{X}_v^T\Lambda_v^{-1}\mathbf{y}_v$$

The test statistic is

$$T = \frac{c\hat{\boldsymbol{\beta}}_v}{\sqrt{\mathrm{Var}(c\hat{\boldsymbol{\beta}}_v)}},$$

and can be shown that for any constant c the ratio follows a Student's t distribution. We test whether or not there is any evidence for the effect using this t-test to test the null hypothesis that $c\hat{\boldsymbol{\beta}}_v = 0$.

On the other hand, we may wish to detect differences between a set of stimuli. In other words, we are interested in some difference between the stimuli rather than in comparing each stimulus with a baseline. This can be done by using an appropriate contrast $k \times p$ matrix $\mathbf{c}$, then an F-statistic is defined as

$$F = \frac{\hat{\boldsymbol{\beta}}_v^T\mathbf{c}[\mathrm{Var}(\mathbf{c}')\hat{\boldsymbol{\beta}}_v]^{-1}\mathbf{c}^T\hat{\boldsymbol{\beta}}_v}{k}$$

where k is the number of contrasts.

Doing this for each voxel yields t- or F-statistics which together form a statistical parametric mapping (SPM). To obtain the activation map of a human brain, the next step is to threshold the t- or F-statistics at a given overall error rate. This leads to a problem of multiplicity. A popular way to solve this problem is to use Gaussian random field (GRF) theory [15] and [16]. It can be applied to threshold the image to identify which parts of the brain were activated, but it is limited because the technique is based on the assumption of a stationary Gaussian random field which is often not satisfied in fMRI settings, and what is more, most of current methods for analyzing fMRI brain images ignore at least one of spatial or temporal relationship between observations in fMRI data sets. The neglect of either spatial or temporal correlation in the model might lead to a seriously biased conclusion. Furthermore, currently available approaches mostly focus on localizing activated areas of the human brain when a cognitive task is performed. This does not allow us to make more complex inference beyond localization of activation of neurons.

The drawbacks of the existing models for analyzing fMRI brain data motivate us to create a new statistical methodology. The main goal of this thesis is the building of a new statistical methodology which more adequately represents the complex spatio-temporal relationships in fMRI brain data sets than do the currently available methods. We start using the model proposed by [17] for fMRI brain data and then extend the model to have Bayesian hierarchical structure, allowing for temporal dependence and using Markov random fields (MRFs) to incorporate spatial dependence in the data.

1.4.2 Hierarchical Bayes model

The Bayesian paradigm provides an attractive framework for making inference using complex models and to overcome the multiple comparison problem. We propose a Bayesian model which not only takes into account the complex spatio-temporal relationships in the data while still being computationally feasible, but gives a framework for addressing other interesting questions related to how the human brain works.

In 2007, Smith *et al* [17] proposed a spatial Bayesain variable selection method which allows automatic detection of the activation of a voxel corresponding to a given effect using indicator variables with an Ising prior. This prior is a binary Markov random field

(MRF) and is useful for spatially smoothing the indicator variables representing whether or not the variable is zero or nonzero in each regression coefficient of each voxel. In their work, they assumed the error term is a multivariate normal such that the voxels in different times are independent. However, those voxels near another in time tend to have similar values giving rise to the temporal autocorrelation. Thus, repeated measurements at the same brain location over time are not independent. To account for this dependence, we assume the error terms have AR(1) dependence. In other words, the element of the covariance matrix Λ_v in (1.1) for the position (i, j) is $\Lambda_v(i, j) = \rho_v^{|i-j|}$.

In the analysis of fMRI data, one often wishes to detect the activation of a certain voxel with respect to a stimulus presented. Therefore, the binary random variable is introduced $\boldsymbol{\gamma}_v = (\gamma_{v,1}, \ldots, \gamma_{v,p})$ to indicate whether the voxel is activated by a sequence of input stimuli. In this model, the indicators are placed in the coefficients to indicate if the voxel is activated corresponding to the given tasks. The coefficient $\beta_{v,i}$ is equal to zero if $\gamma_{v,i} = 0$ which means no effect on voxel v is caused by the corresponding experimental task i given, and $\beta_{v,i}$ is nonzero if $\gamma_{v,i} = 1$. So the model in (1.1) can be written as

$$\mathbf{y}_v = \mathbf{X}_v(\boldsymbol{\gamma_v})\boldsymbol{\beta}_v(\boldsymbol{\gamma_v}) + \boldsymbol{\varepsilon}_v, \quad \varepsilon_v \sim \mathcal{N}(0, \sigma_v^2\Lambda_v).$$

This model is constructed assuming that the main goal is to detect areas of activation in the human brain when the same task is performed in p different task periods. The temporal and spatial relationship in the data are modeled simultaneously.

FMRI brain data sets usually contain an enormous number of observations, for instance, our fMRI brain data set having $79 \times 95 \times 68 \times 465 = 237,308,100$ observations. The large amount of data acquired from a single fMRI experiment puts a large computational burden on any advanced model. In addition, there is no available software that can handle so much data and the complex models we use in this thesis. We must write our own code to do inference.

The use of models we proposed for analysis of fMRI brain data is computationally intensive. To make the model more computationally attractive, we develop a two-stage approach to identify activated areas in the human brain. Although it not as complete as the procedure of estimation of parameters, it is a reasonable alternative especially since minimizing the computing time for estimation is major goal. Hence, the activation map is generated from full spectrum of posterior inference created through a Markov chain

Monte Carlo (MCMC) scheme.

1.4.3 Markov Chain Monte Carlo Algorithms

MCMC is essentially a general method to approximately generate the sample from the target distribution. A sequence of dependent samples generated from the target distribution using an MCMC algorithm, provided the sample is large enough, is used to estimate feature of the posterior density by taking those draws and forming the relevant sample-based estimates. For example, the sample average of the sampled draws would be our simulation-based estimate of the posterior mean, while the quantiles of the sampled output would be estimates of the posterior quantiles.

Fundamentally, a Markov chain is artificially created in MCMC algorithms to have the target distribution as its stationary distribution from which we wish to simulate. The two most popular algorithms to create a Markov chain are the Gibbs sampler [18] and the Metropolis-Hastings algorithms [19]. Other MCMC algorithms are usually variations of these two algorithms. However, a key issue in the successful implementation of MCMC algorithms is the number of runs until the chain approaches stationarity. In other words, when do we terminate the MCMC algorithms and accept that the generated samples truly represent the stationary distribution of Markov chain? Or how long of iteration should be? Various convergence tests are proposed to assess whether stationarity has presumably been reached. Furthermore, the greater the dimensionality of the problem, the more complicated the problem of assessing convergence can not be implemented easily in practice, because they either are very model dependent or require an infeasible number of iterations. Some of those MCMC convergence diagnostics are reviewed later in chapter 3.

In this paper, we begin with brief review of the basic physical principles of generating MRI signals and summary of major features of the fMRI techniques and fMRI brain data in chapter 2. Chapter **??** introduces a general linear model, the most popular technique applied to fMRI data. It also includes a description of t- and F-test to detect the activation of areas in the human brain. To take account of the temporal and spatial dependence and heterogeneity of the mean and variance structures across the brain, we then extend the model to have Bayesian hierarchical structure in chapter 5. Chapter 3 summarizes know facts about MCMC algorithms which are major tools in analyzing complex Bayesian

models and review some MCMC convergence diagnostics. The estimation of normalizing constants is considered in Chapter 4. We conduct a simulation study to compare the models and apply our models to the real fMRI data in chapter 5.

Chapter 2

Background

A basic understanding of the physiological and physical mechanism of fRMI signal change provides a foundation for the design and interpretation of fMRI studies of cognition. FMRI is a non-invasive technique useful for imaging human brain activity based on change of blood flow in active brain areas. The increase of the blood flow in these areas exceeding the consumption of oxygen results in the decrease of proportion of deoxyhemoglobin (dHb) in blood. The reduction of proportion of dHb can make image intensity strengthen since dHb is paramagnetic substance interfering with a magnetic field. Therefore, dHb (sometimes called endogenous contrast enhancing agent) is the source of fMRI signal and the physiological foundation of fMRI technology. This chapter introduces some basic principles of fMRI and how to prepare the fMRI data set in preprocessing steps for the subsequent statistical analysis. For more information, the interested readers should refer to [20], [21], and [22].

2.1 Basic Physical Principles of MRI

Having knowledge of how an MR scanner operates and generates signals is useful in discussing fMRI in its applications to psychology and medicine. Additionally, fMRI relies on a core set of physical principles to generate the brain images. The purpose of this section is to present a general description of physical principles of signal generation in the MR scanner. For more detail of MRI physics, please refer to [21].

FMRI uses the magnetic properties of hydrogen and its interaction with both a large external magnetic field and radiowave to produce a highly detailed image of a human brain. Hydrogen exists in water and fat and a human body is mainly composed of these two types of molecules. Because of their abundance in a human body and having a significant magnetic moment, the torque exerted on a magnet, hydrogen atoms are the most commonly-imaged nuclei in MRI. The nucleus of the hydrogen atom contains a single proton. Under normal conditions, it possesses a significant magnetic moment which cause spin about its axis; meanwhile, the spinning proton changes in orientation of the axis orbiting around the direction of the applied field B_0, see Figure 2.2.

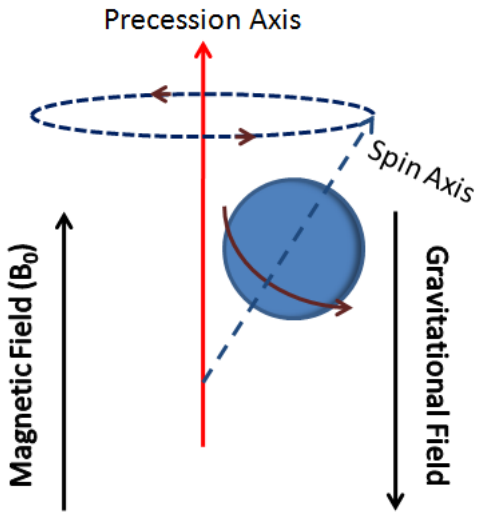


Figure 2.1: Spin: A proton rotates around its own axis. Procession: The axis of spin itself wobbles around the main axis of the magnetic field.

Without an external magnetic field, the protons are arbitrarily arranged as shown in Figure 2.2 and tend to cancel out each other. As a result, the net magnetization denoted by M_0, the sum of all magnetic moments from spins of different orientations, is zero. If they are, however, placed in strong magnetic field, all protons will be separated into two groups. One of them will align in a parallel and the other will align in anti-parallel with the direction of the magnetic field, but the individual spins do not align exactly parallel

to the applied field, but at an angle to it given in Figure 2.3. Protons lining up in the parallel orientation are said to be in the low energy state. On the contrary, protons lining up in the anti-parallel orientation are said to be in the high energy state. In addition to aligning with magnetic field (B_0), protons precess at some frequency, which is described by the Larmor equation [23]. Increase in B_0 will increase the precessional frequency and, contrarily, decrease in B_0 will decrease the precessional frequency. The net magnetization (M_0) directly proportional to the strength of magnetic field (B_0), so the greater the intensity of the magnetic filed, the better the signal is.

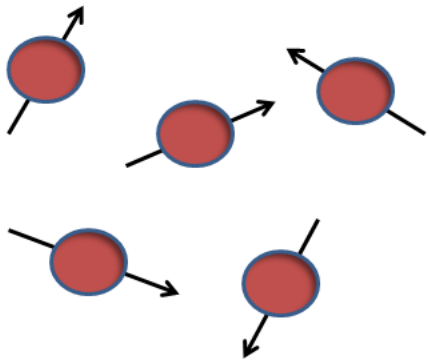


Figure 2.2: Before a magnetic field is applied, the protons are randomly oriented.

When protons are exposed to radio frequency (RF) energy, protons will absorb the energy so that the net magnetization is rotated to a certain angle away from the B_0 axis. This angle is referred to as the flip angle and is proportional to the duration and amplitude of the RF pulse. The process of sending energy to protons is called excitation. The sufficient energy to produce a 90° flip of the net magnetization is called 90-degree pulse, resulting in the net magnetization (M_0) reaching x-y plane, see Figure 2.4. When we apply a pulse of RF energy into the tissue at the Larmor frequency, all individual spins start to precess in phase.

Upon termination of the RF pulse, the protons return to their original alignment, parallel or anti-parallel to the magnetic field, and energy is emitted in the form of a weak

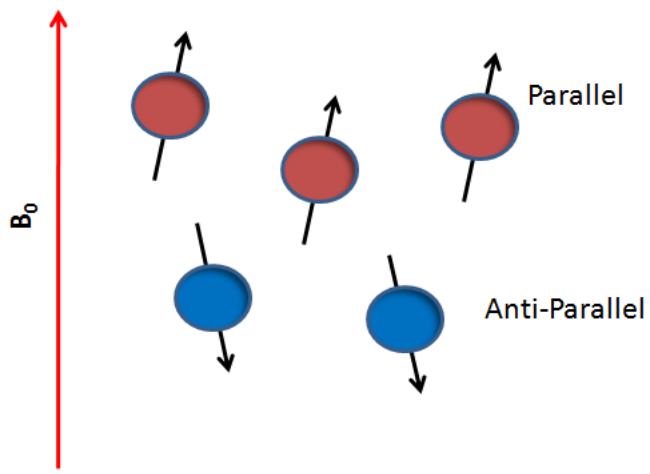


Figure 2.3: When placed in the strong magnetic field, all the protons align in either a parallel or anti-parallel with the direction of the magnetic field (B_0).

RF signal. The process of all protons returning to the original energy state when the RF pulse is turned off is called relaxation. There are two types of relaxation. The signal fades as the individual spins contributing to the net magnetization lose their phase coherence, making the vector sum equal to zero. This is called transverse relaxation or spin-lattice relaxation and the corresponding time of the decay is T_2 shown in Figure 2.5. On the other hand, the protons slowly release the energy and re-align with the main magnetic field in the low energy state. This is called longitudinal relaxation or spin-spin relaxation and the time constant associated with this process is called T_1 given in Figure 2.6. Detection and analysis of the emitting RF signal provides insight into the chemical composition of the material. This process of alternating absorption and emission of RF energy by the material is termed magnetic resonance (MR).

At the end of the applied RF pulse, a current or signal induced by the changes in intensity of net magnetization of protons in x-y plane and z axis can be detected by a receiver coil. The signal intensity diminishes rapidly (within a few hundred milliseconds) as the higher energy state (the antiparallel state) is depopulated and the nuclei return to

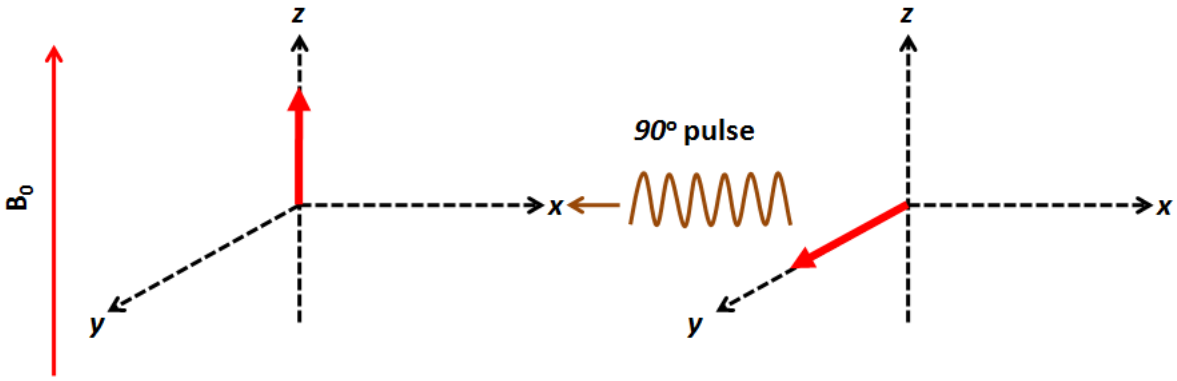


Figure 2.4: A schematic illustration of an application of a radio frequency to magnetization vector. When a RF pulse is applied to the magnetization vector, its direction changes. The radio frequency power of the pulse is proportional to the flip angle through which the spins are tilted under its influence.

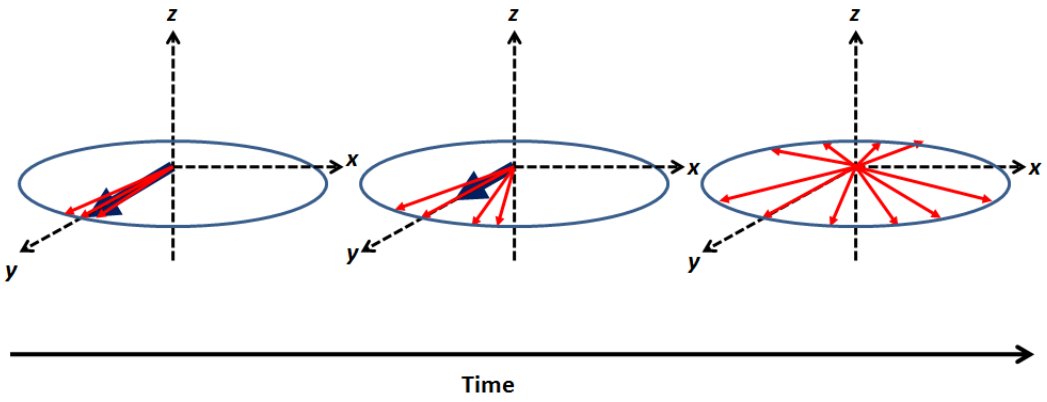


Figure 2.5: A conceptual overview of T_2 decay.

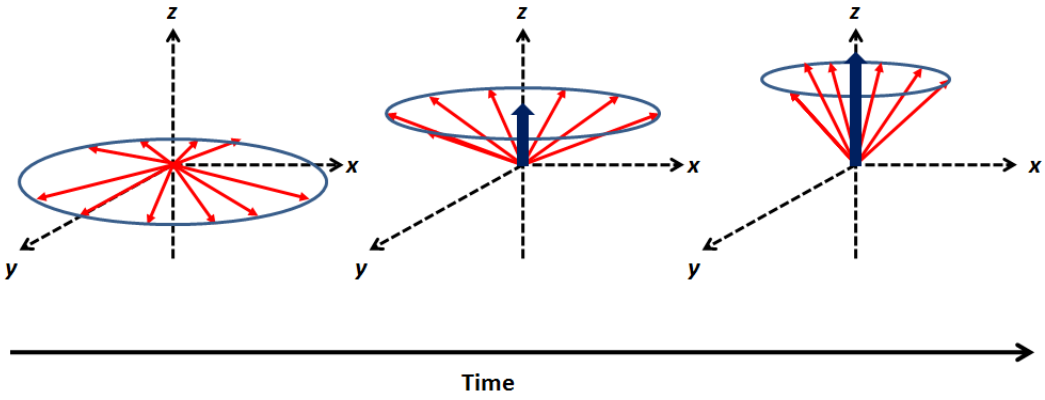


Figure 2.6: A conceptual overview of T_1 recovery.

their original energy state. The waveform of this signal is an exponentially damped sine wave and is called the free induction decay (FID), see Figure 2.7 and it is main signal measured by MRI technique.

Since the relaxation of T_1 and T_2 appears differently in different tissues, researchers can use the property to image particular tissues in human brain. For example, from a clinical perspective, the white matter appears bright in a T_1 image but dark in a T_2 image. In addition, to image the structure of different positions or functional change in the human brain, different magnetic fields (gradients) are intentionally applied. By applying linear changes in magnetic field (or gradients) at various times and various orientations we can artificially change the resonant frequency of the spins so that it is spatially dependent. Three main magnetic gradients used are slice selection, frequency encoding, and phase encoding.

Slice Selection

Here we consider how only a finite section or slice of anatomy can be preselected by the scanner. First we will discuss how an axial image is acquired. In this case we perform slice selection along the z-direction: a gradient in this direction is turned on such that it acts symmetrically about the center of the scanner. Since the magnetic gradient can cause the different frequencies of protons in different positions, when an RF pulse is given,

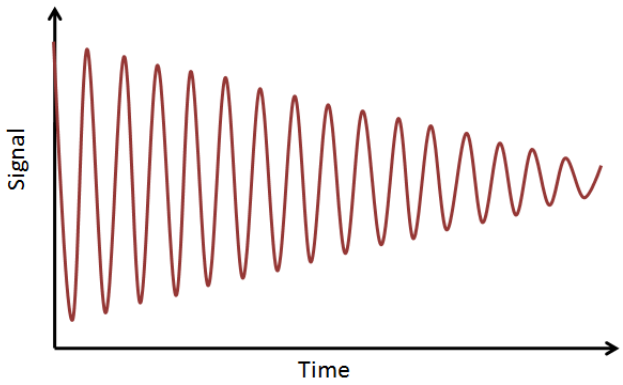


Figure 2.7: The signal is induced in the receiver coil over time.

although a whole brain is exposed to it pulse, only the same frequency in the plane will absorb the energy (resonance) so that the image of interest can be acquired, see Figure 2.8. The slice thickness or position can be varied by using different gradient strengths or RF bandwidths.

Frequency Encoding

Once the signal from the slice has been isolated, the remaining two in-plane dimensions need to be encoded (in this case the x and y directions). One of the directions is encoded by changes of frequency. A gradient is turned on in the x direction. The center of the slice remains unaltered and the frequency is smaller in the left but larger in the left so that columns of pixels from left-to-right are discriminated in terms of frequency differences, see Figure 2.9.

Phase Encoding

It can be shown that a gradient applied in the y-direction to change frequency in this dimension would not be sufficient to uniquely ascribe frequency to each column and row of pixels. For the last dimension the signal is encoded in terms of phase. In general, a number of gradients are needed to create phase changes from row-to-row so that the Fourier transformation is provided with enough information to fully encode the final image. What is more straightforward to understand is how gradients can alter phase as well

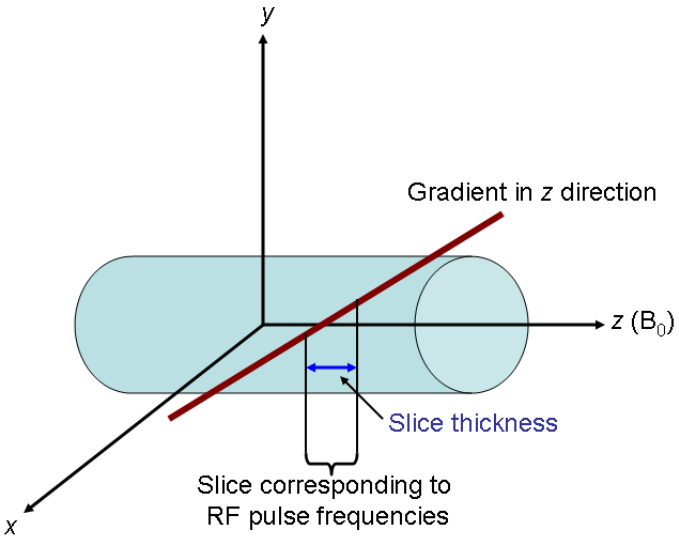


Figure 2.8: Slice Selection: A particular slice with the precession frequency of the spins matching FR pulse is imaged.

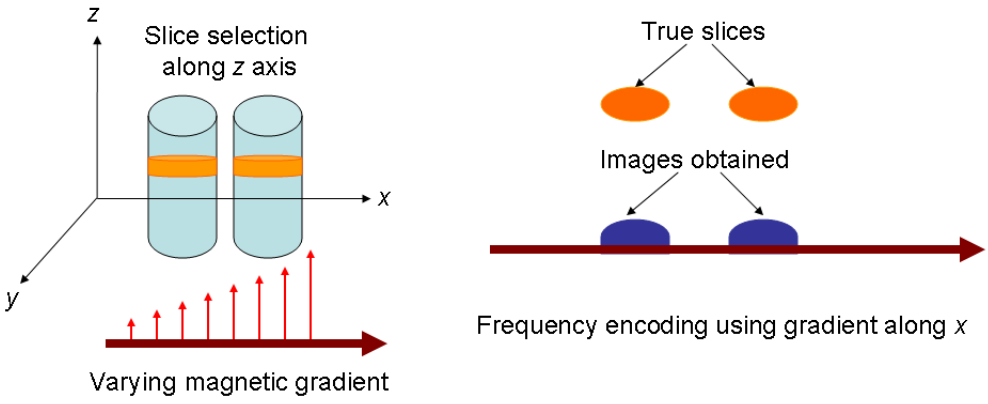


Figure 2.9: Frequency Encoding: The use of a magnetic gradient to image two objects at different location.

as frequency. Clearly having applied a gradient, some spins will be precessing faster than others. Once the gradient is removed, the resonant frequency is the same as it was before for all the spins. However, the spins will now be out of phase with each other, see Figure 2.10.

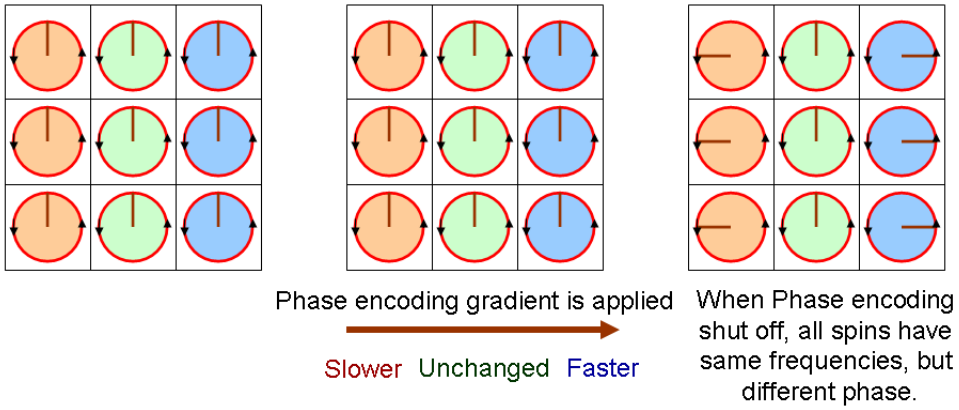


Figure 2.10: Phase Encoding: All spins have the same precessional frequency before applying the phase encoding gradient. When the phase encoding gradient is applied, some spins will be precessing faster than others. When it is turned off, all spins have the same frequency again, but different phase.

The technology of MRI is based on the flexible application of multiple RF pulses and multiple gradients, synchronized precisely. The pulse sequence diagram indicates where a given slice of human brain is imaged, and how the resulting signals are preferentially selected to obtain information of deoxyhemoglobin in venous blood flow.

As with any imaging modality, the key variable in producing a meaningful image is contrast. The signal measured at one point in space or time must be higher or lower than the signal at another point, and the variation in signal intensity across the image should systematically follow some variable of interest. In brain mapping, the ultimate aim is neural activity. In order to measure local brain activity with MRI, one must exploit a chain of indirect linkages from neural activity to changes in brain physiology and metabolism

and finally to changes in the magnetic properties of substances within the brain.

Anything that causes a change in the MRI signal from a given voxel relative to other voxels at the same time is source of contrast in the image. The density of protons in a given voxel is one such source of contrast. More commonly, it is the variation in rates of relaxation from voxel to voxel that generates contrast in the image.

2.2 BOLD fMRI

A wide variety of techniques which produce various contrasts have been developed for detecting changes in brain physiology. FMRI is an advanced technique for imaging brain activity related to a specific task or sensory process. However, fMRI do not record the response that we really want– activity of neurons. Instead, we observe changes in blood oxygenation which are indirectly caused by the neuronal activity. This remarkable feature of brain metabolism that blood flow and energy metabolism are tightly linked to local neural activity allows us to detect neuronal activity. When neurons are active in the brain, there is an increase in blood flow and blood volume in that region of activity. MRI can be used to detect the change in blood flow directly, then in turn to detect the neuronal activity. Next we introduce how an MRI can be used to study the brain activity.

In practice, functional imaging techniques rely on the BOLD phenomenon. Ogawa *et al* [24] at AT&T Bell Laboratories made possible mapping a functional brain using the venous blood oxygenation level-dependent magnetic resonance imaging (MRI) contrast in an experiment of rat brain studies during global stimulation at 7 Tesla (T). BOLD works because deoxygenated hemoglobin is paramagnetic while oxygenated hemoglobin is diamagnetic. When a brain area is more active, it consumes more oxygen which causes changes in blood oxygenation. An increase in blood flow to active brain regions results in a higher concentration of oxygenated blood and therefore results in an increased MRI signal in the active region as seen in Figure 2.11. Therefore, fMRI can be used to produce an activation map showing which parts of the brain are involved in a particular mental process.

The BOLD signal in a voxel is coming from the total amount of deoxygenated hemoglobin and noise arising from a variety of sources. Several studies [12] and [13] have shown

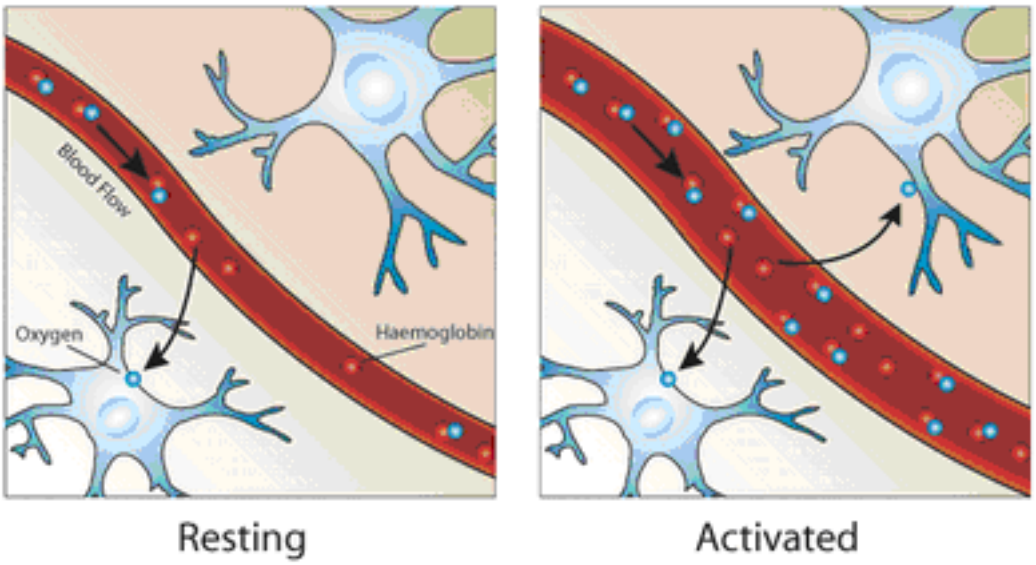


Figure 2.11: Contrast Mechanisms: The BOLD Effect (Taken from University of Oxford FMRIB Centre Department of Clinical Neurology).

an early negative response due to a transient increase in the amount of deoxygenated hemoglobin. After a dip of 1 to 2 s duration, the metabolic demands of active neurons over baseline levels result in an increased inflow of oxygen in blood. More oxygen is supplied to the area than is demanded, and this gives rise to a decrease in the amount of deoxygenated hemoglobin within the voxel. This change in the MR signal triggered by the increased neuronal activity is called the hemodynamic response. Since the shape of the hemodynamic response may differ with the properties of the evoking stimulus and, presumably, with the underlying neuronal activity, determining the exact relation between the neuronal events that trigger the hemodynamic response and the shape of it, however, is complicated by their differing dynamics. In general, the signal increases above baseline at about 2s following the onset of neuronal activity, peaking at about 6s for a short-duration stimulus, see Figure 2.12. This maximum is known as the peak of the hemodynamic response. If the neuronal activity is extended across a block of time, the peak may be similarly extended into a plateau. This delay and blurring is modeled by a hemodynamic response function (HRF).

The stimulus is then convolved with the assumed or modeled HRF to give the assumed BOLD response as a column vector of design matrix in linear regression model. Consequently, one can predict the BOLD fMRI signal change that could result from any arbitrary pattern of neural activity.

2.3 fMRI Experiment Designs

Experimental design plays a very important role in making inference about brain activation by a particular cognitive operation. Before an fMRI time-series is acquired, we need an experimental design to make the statistical analysis more powerful for detecting evoked changes in neural activity. There are two main types of experimental designs in fMRI, block design and event-related design. Figure 2.13 schematically shows the difference between the two designs. In a block design, a series of trials in one condition is presented during a discrete epoch of time. An epoch is the cycle of blocked trial and rest. Block designs are very popular because they are easy to program and will generally be more sensitive to detecting activations. On the other hand, in an event-related design, an

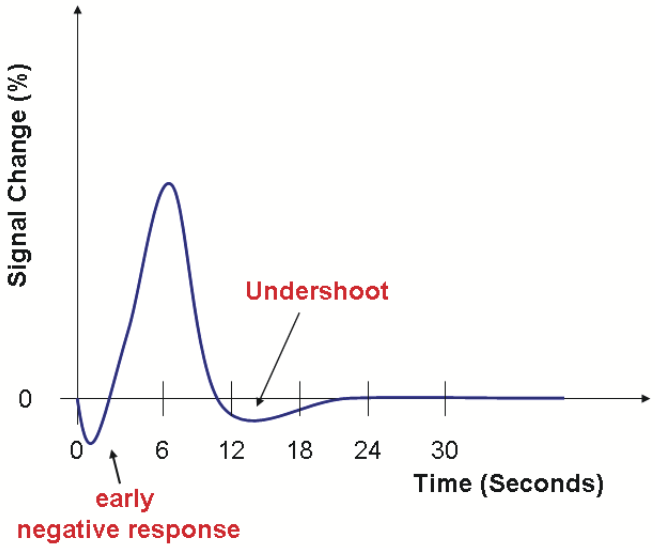


Figure 2.12: The typical schematic representation of fMRI BOLD hemodynamic response.

individual trial is separated by as little as a few seconds in each block or stimuli are presented randomly over time. An event-related design may better characterize the BOLD response because it allows the possible removal of task-correlated signal change.

Once the experimental design is decided, the design matrix is obtained from the convolution of hemodynamic response function (HRF) with external stimulus. In chapter **??**, we'll give an introduction to modeling the response to the stimulus.

2.4 Preparing fMRI Data for Statistical Analysis

A number of preprocessing steps must be performed prior to the statistical analysis of fMRI data due to physical and psychological effects not due to the BOLD response. These steps are taken to remove extraneous sources of variability in order to enhance power in statistical analysis. This allows better detection of the spatially extended signal within or across subjects and to reverse displacements of the data in time or space that may have occurred during acquisition. In the real world, it is impossible to remove all of the

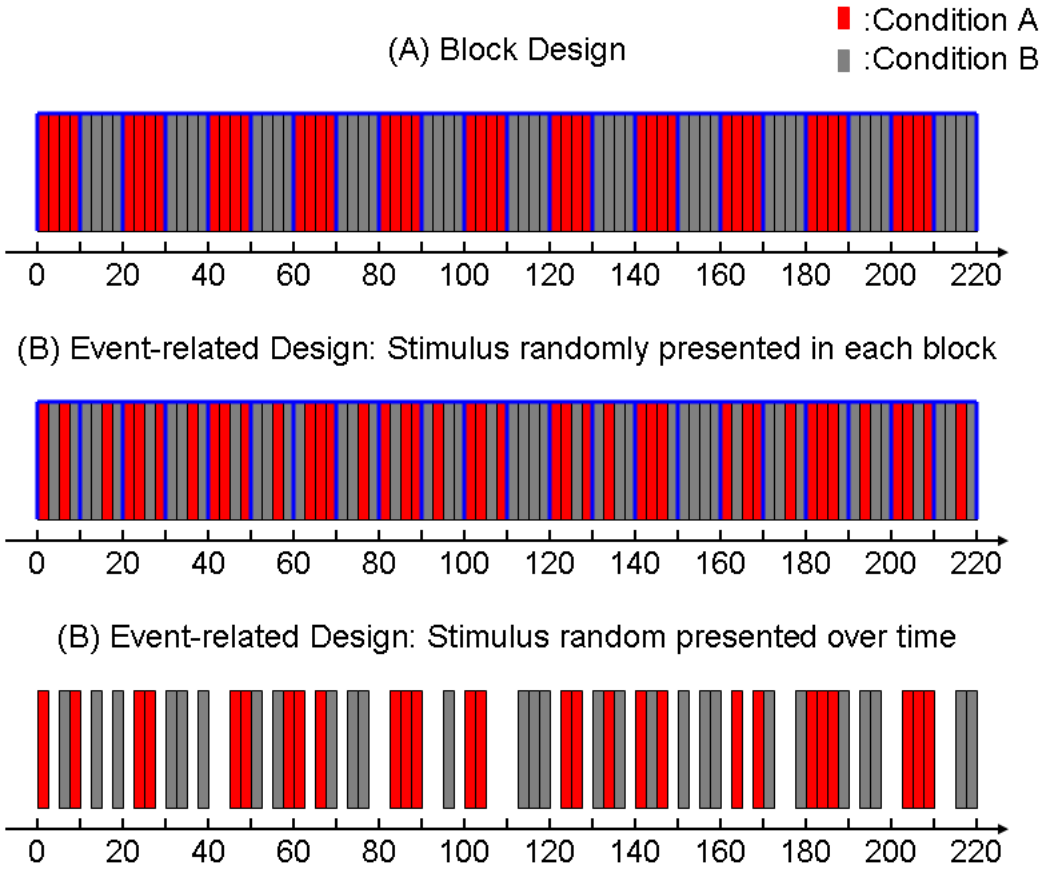


Figure 2.13: (A) A block design. A serial of the same trial is given in each time interval. (B) An event-related design. Tasks are randomly repeated in each block. (C) The stimulus is randomly presented over time.

unwanted noise of various types. Here we only touch upon a small number of approaches that are commonly applied in the preprocessing steps.

2.4.1 Slice Timing Correction

A whole brain image in fMRI is composed of slices collected during one repetition time (TR). Only one slice is formed at a time, so a whole brain image consists of slices spread out over few seconds. Accordingly, the BOLD fMRI responses within a brain will appear to exhibit time-delayed phenomena corresponding to a neural event occurring simultaneously on different slices as shown in Figure 2.14. In our statistical analysis we will model each voxel's time-series with an assumption that data for each time point was taken at the beginning of the corresponding volume's scan time. In other words, we assume the whole brain image is captured at the same time for each time point. However, different points in the volume were scanned at slightly different times, so the model fitting is not optimal. A way to adjust the time series is to apply a slice acquisition correction by shifting the time series of the value at each voxel. This operation is referred to as slice timing correction and is achieved by temporal interpolation during preprocessing.

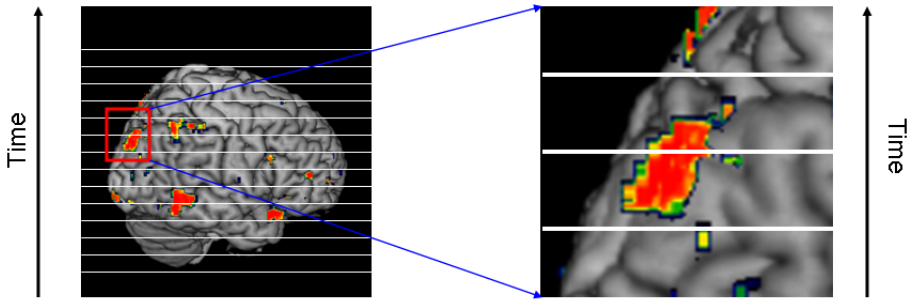


Figure 2.14: Slice's acquisition time are different for whole brain, so the BOLD signal sampled at different layers of the brain is at different time points

2.4.2 Motion Correction

The subject's motion poses a severe problem for the analysis of functional data. During an fMRI session, functional time series are acquired using ultrafast MR sequences sensitive to the BOLD response. But small head movements masking relatively small BOLD signal changes or the position of the brain within the functional images will vary over time which means any particular voxel's time series does not refer to the same point in the brain. This has strong consequences on the following statistical analysis. A variety of methods are utilized to minimize head motion. For example it is good experimental practice to use suitable restraining devices and to train the subjects to hold still. In spite of the use of physical constraints, head movements cannot be completely eliminated during an fMRI session. The general method of treating head movements is done by realigning the image of the brain obtained at each point in time back to the first image acquired at the start of the scanning session.

2.4.3 Coregistration

Some clinical application of fMRI is to study functionality in a particular anatomy of human brain; however, functional magnetic resonance imaging (fMRI) is often highly distorted or reference anatomical data sets are acquired in separate sessions for each subject, so it is necessary to coregister the functional to undistorted anatomical images. In other words, we align functional and structural data for individual subjects. Spatial coregistration includes computing a transformational matrix specifying the transformation parameters and applying it to the data of interest to ensure an accurate coregistration between anatomical and functional data. More information about coregistration please refer to [25] and [21].

2.4.4 Spatial Normalization

It is known that the configuration of the brain is different from person to person. In multivariate statistical analysis, a hypothesis test of whether or not a certain area of the brain is active corresponding to a stimulus in a group of patients, requires us to identify that same area of the brain across subjects. There are a variety of sophisticated methods to achieve

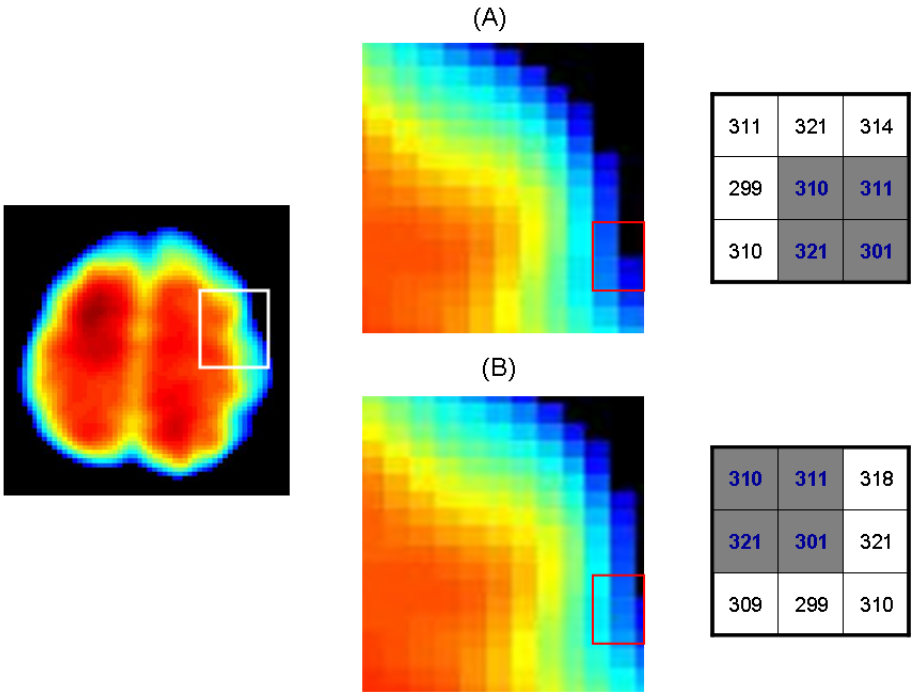


Figure 2.15: Effects of head motion on fMRI. The numerical intensity values for the voxels within the red box are shown on the right side. The top one (A) is the image in the red box and corresponding numerical values before head motion. The bottom one (B) is the image in the red box and corresponding numerical values after head motion.

this, one of which is to align the anatomical structure of the brain of different subjects to match a template brain within a standard defined space. This approach improves inference about whether or not the response of some functional region is stimulated by a task across subjects.

2.4.5 Spatial and Temporal Smoothing

In neuroimaging, filters are used to remove uninteresting variation in the data that can be safely attributed to noise sources, while preserving the signal of interest. Two main reasons to use spatial and temporal smoothing for fMRI data in space prior to statistical analysis are to increase signal-to-noise (SNR) ratio and to validate the common assumption in statistical analysis [26] that t-statistics in search regions are smooth isotropic random fields. The noise is the unavoidable random variation in image intensity which is present even when no stimulation is applied. Spatial smoothing can reduce the noise level but retain the underlying signal. Also it can control the false-positive rate while a lot of t-statistic are performed in GLM because of smoothness of image reducing independent statistical tests to allow less-stringent control over that t value is considered a significant result [27], [28] and [29].

2.4.6 Segmentation

Segmentation is an important process that helps in identifying objects of the image, for example, white matter and gray matter in a human brain. A high quality tissue segmentation can enhance the ability to investigate the brain function. There main algorithms, classification-based [30], region-based [31], contour-based approaches [32], have been developed and widely applied to automatic tissue segmentation for 3D MR images [33] and [34]. In general, this is a very difficult and time-consuming process.

Chapter 3

Markov Chain Monte Carlo

Fitting with a GLM to fMRI data typically requires advanced numerical or simulation tools. We take the opportunity of this requirement to review the Markov chain Monte Carlo (MCMC) methods. The strong appeal of MCMC approaches is that it is rather universal in its formulation as well as in its use.

To compute posterior quantities, the ideal method is exact numerical evaluation. However, the numerical integration approach may not always be applicable, in particular, when a Bayesian model is complicated or when the dimension of integration is high. Once analytical evaluation is not available, simulation-based approaches can be applied. One computational method for inference in Bayesian analysis is MCMC. This chapter gives a summary of MCMC algorithms which are major tools in analyzing complex Bayesian models. MCMC is a class of algorithms for generating observations from a target distribution, π, known up to a normalizing constant. The basic idea behind MCMC algorithms is to artificially create a Markov chain whose stationary distribution is exactly π. We review the most two popular approaches, the Gibbs sampler and the Metropolis-Hastings algorithms, to constructing such chain. Any other MCMC algorithms are the generalization of these two algorithms and descriptions can found in [35] and [36]. Although MCMC algorithms offer a way to generate samples from the target distribution and enable estimation in complex models through simulation, a common question asked is are the samples representative of the target distribution of interest, in other words, how do we know when to stop the simulation. Several stopping rules to terminate the simulation

have been proposed. We summarize the results presented in [37] and[38].

3.1 Markov Chain Monte Carlo

The general objective in Monte Carlo simulation is the use of stimulated random numbers to estimate some functional of a probability distribution or to estimate some characteristics of a random variable X. In practice, the quantity of interest is oftenan expectation

$$E_\pi g = E_\pi[g(X)] = \int g(x)\pi(dx), \tag{3.1}$$

where π is the probability distribution, either discrete or continuous, and g is a real-valued function such that the expectation exists. This integral above cannot be always calculated analytically and numerically. As a result, to estimate the expected value, a sample $\{X_1, X_2, \ldots, X_n\}$ is generated from π to approximate (3.1) by the empirical average

$$\bar{g}_n = \frac{1}{n}\sum_{i=1}^{n} g(x_i).$$

By the Strong Law of Large Number (SLLN), $\bar{g}_n$ converges almost surely to $E_\pi[g(X)]$. However, direct simulation from π is sometimes impossible due to the complexity or high dimension of π. Fortunately, it is possible to sample the complex distribution π by a set of methods that are called MCMC methods. MCMC methods have had an enormous practical impact since the 1970's as they provide a large scope for statistical modeling. They have proven to be effective in areas such as spatial statistics, image analysis, Bayesian statistics, operations research, economics, and many others.

MCMC is essentially a general method to approximately generate the sample from the target distribution. Fundamentally, a Markov chain is artificially created to have the target distribution as its stationary distribution from which we wish to simulate. Once a sequence of dependent samples is generated from the target distribution, provided the sample is large enough, not only the integral (3.1) but also features of the posterior density can be evaluated by the relevant sample-based estimates. For example, the sample average of the sampled draws would be our simulation-based estimate of the posterior mean, while the quantiles of the sampled output would be estimates of the posterior quantiles. These estimates would converge to the posterior quantities under certain regularity

conditions as the simulation size becomes large. In short, the problem of computing an intractable integral is reduced to the problem of sampling the posterior density. Next, we briefly review some concepts of general Markov chain theory before the most two popular MCMC algorithms are introduced. More extensive uses of MCMC methods are discussed in [39], [35] and [36].

3.1.1 Markov Chain

Let $\mathcal{X} = \{X_1, X_2, X_3, \ldots\}$ be a discrete-time Markov chain on a general state space X and let $\mathcal{B}$ denote the associated Boreal σ-algebra. Then let $P(x, dy)$ denote the associated Markov transition kernel; that is, for $x \in \mathsf{X}$ and $A \in \mathcal{B}$,

$$P(x, A) = P(X_{i+1} \in A | X_i = x).$$

For $n \in \mathbb{N} := \{1, 2, 3, \ldots\}$, let $P^n(x, dy)$ denote the n-step Markov transition kernel; that is, for $x \in \mathsf{X}$, $A \in \mathcal{B}$, and $i \in \mathbb{N}$,

$$P^n(x, A) = P(X_{n+i} \in A | X_i = x).$$

For simplicity, we will often assume that the probability $P(x, \cdot)$ has a conditional density, $k(\cdot|x)$, with respect to Lebesgue measure so that

$$P(x, A) = \int k(u|x)du. \tag{3.2}$$

We will call k a Markov transition density. Further, if there exists a density π such that

$$\pi(x) = \int_{\mathsf{X}} k(x|y)\pi(y)dy, \tag{3.3}$$

then π is called the stationary or invariant density for the Markov chain $\mathcal{X}$. The basic idea behind the stationarity in (3.2) is that if y is drawn from π and apply the Markov transition kernel, $P(x, dy)$, resulting in the transition $x \to y$, then the marginal density of x is also π. It turns out that when X_1 can be drawn from π, then the sequence $\mathcal{X}$ is a dependent sample generated from π, that is, the chain is stationary.

There may be, however, more one stationary distribution for a Markov chain. If a Markov chain does not have a unique stationary distribution, it is useless for MCMC. A Markov chain will have a unique stationary distribution if it is aperiodic, irreducible and positive Harris recurrent.

1. Aperiodicity means that given that you are in the state, there is no periodic pattern to when your can return.

2. Irreducibility means that all states communicate with each other. In other words, there is a positive probability that the chain to visit all states.

3. Recurrence means if the chain is run forever, each state will be visited infinitely often.

For more detail of this conditions, the interest read should refer to [40].

Notably, MCMC methods entail constructing an aperiodic, irreducible and recurrent Markov chain $\mathcal{X}$ satisfying 3.3 and then simulating $\mathcal{X}$ for a finite number of steps, say n, and using $\bar{g}_n$ to estimate $E_\pi g$.

The popularity of MCMC methods results from the ease with which such an $\mathcal{X}$ can simulated. Two most popular approaches to construct a Markov chain are Gibbs sampling and Metropolis-Hastings sampling. We give a brief summary of the two algorithms.

3.1.2 Metropolis-Hastings Algorithm

We outline the MH algorithm to draw a sample from the target distribution π on the state space X. Suppose the current state is x, that is $X_j = x$. Then a candidate value y generate from a proposal distribution Q with density q such that supp(q)$\supseteq\mathsf{X}$ and accept the transition to y with probability

$$\alpha(x,y) = \begin{cases} \min\left\{\frac{\pi(x)q(x|y)}{\pi(y)q(y|x)}, 1\right\}, & \pi(x)q(y|x) > 0; \\ 1, & \pi(x)q(y|x) = 0. \end{cases}$$

Note that the candidate distribution $q(\cdot|x)$ depends on the current state of the chain. If the candidate draw is accepted, we set $X_{j+1} = y$; otherwise, we reject the candidate, y, and let $X_{j+1} = x$. Suitable choices for Q and rates of acceptance could achieve high efficiency of a Metropolis-based Markov chain and they have been studied extensively in [36].

3.1.3 Gibbs Sampling Algorithm

The Gibbs sampling algorithm is another way to generate a Markov chain whose stationary distribution is $\pi(x)$ and a particular case of MH. It could update a single component

at a time and each component may be univariate or multivariate. This algorithm was introduced by [41] in the context of image processing. The value of the Gibbs algorithm was demonstrated by [42] for a range of problems in Bayesian analysis.

Let $\pi(\boldsymbol{x})$ be the p-dimension target distribution with $\boldsymbol{x}= (x_1, x_2, \ldots, x_p)$. Suppose we're interested in generating a random sample from $\pi(\boldsymbol{x})$. Provided an initial value $\boldsymbol{x}^{[0]} = (x_1^{[0]}, x_2^{[0]}, \ldots, x_p^{[0]})$, the Gibbs algorithm at the nth step proceeds as follows:

- Generate $x_1^{[n]} \sim \pi(x_1|x_2^{[n-1]}, x_3^{[n-1]}, \ldots, x_p^{[n-1]})$
- Generate $x_2^{[n]} \sim \pi(x_2|x_1^{[n]}, x_3^{[n-1]}, \ldots, x_p^{[n-1]})$

  $\vdots$
- Generate $x_p^{[n]} \sim \pi(x_p|x_1^{[n]}, x_2^{[n]}, \ldots, x_{p-1}^{[n]})$

The Markov transition kernel of this chain is

$$k(\boldsymbol{x}|\boldsymbol{y}) = \pi(y_1|x_2, x_3, \ldots, x_p)\pi(y_2|y_1, x_3, \ldots, x_p)\cdots\pi(y_p|y_1, y_2, \ldots, y_{p-1}).$$

It can be shown that it satisfies the stationarity 3.3 and the Gibbs chain is irreducible and aperiodic. Given the n is large enough, $\boldsymbol{x}^{[n]}$ converges in distribution to random variable from $\pi(\boldsymbol{x})$.

3.2 Monte Carlo Error

The two approaches provide a way to simulate the sample from the π so that we can estimate characteristics of it. However, an obvious question is when we should stop the simulation? That is, how large should n be? Or, when is $\bar{g}_n$ a good estimate of $E_\pi g$? It is difficult to decide when to terminate the MCMC algorithms to accept that the simulated samples are truly representative of the stationary distribution of the Markov chain. An approach to evaluate the quanlity of the estimate is to provide the associated Monte Carlo standard error (MCSE) of $\bar{g}_n$,

$$\begin{aligned}\sigma_g^2 = \mathrm{Var}\,(\bar{g}_n) =& \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathrm{cov}\,\{g(X_i), g(X_j)\} \\ =& \mathrm{var}_\pi\{g(X_1)\} + 2\sum_{i=2}^{\infty}\mathrm{cov}_\pi\{g(X_1), g(X_i)\} \qquad (3.4)\end{aligned}$$

The issue we study here is how to consistently estimate σ_g^2 and measure the accuracy of the resulting estimate such that proper stopping rules can be obtained and then applied to terminate MCMC algorithms in a general setting.

Estimating σ_g^2 is an important issue in MCMC output analysis since the estimate can be used to decide when to terminate the simulation or assess the reliability of the current point estimate $\bar{g}_n$. Obviously, the estimate of $\bar{g}_n$ is seldom equal to the true quantity of interest, $E_\pi g$. In fact, there is certainly some distance between $\bar{g}_n$ and $E_\pi g$ and this difference $\bar{g}_n - E_\pi g$ is the Monte Carlo error. However, we can assess this error by estimating the variance from the asymptotic distribution of $\bar{g}_n$.

Let $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ be a Markov chain generated from a target distribution using one of the previous MCMC algorithms. Assume a Markov chain central limit theorem (CLT) exists for $\bar{g}_n$, for any initial distribution,

$$\sqrt{n}(\bar{g}_n - E_\pi g) \xrightarrow{d} N(0, \sigma_g^2) \quad \text{as } n \to \infty \tag{3.5}$$

where σ_g^2 is defined in (3.4). Calculating the asymptotic variance σ_g^2 is very difficult since the sample is not independent. We introduce two techniques for estimating the variance: batch mean, overlapping batch means. For other approaches the interested reader is referred to [43], [44] and [38]

3.2.1 Batch Means

Batch means (BM) is one method used to compute MCSEs. Although there is a vast literature on sophisticated methods for computing MCSEs for MCMC output, batch means has the advantage of being easy to implement and it appears to work reasonably well in practice.

Suppose a single run of length n generated from a target distribution by using the previous algorithm and then output is split into several batches a, each consisting of b observations, in other words, $n = ab$. In fact, we could have different number of observation in different batches. For ease of explanation, we assume each batch has equal number of observations, b, and then the mth batch mean is defined

$$\bar{Y}_m = \frac{1}{b} \sum_{i=(m-1)b+1}^{m \cdot b} g(X_i),$$

for any $m = 1, \ldots a$ and the corresponding batch mean estimate of σ_g^2 is given by

$$\hat{\sigma}^2_{\mathrm{BM}} = \frac{b}{a-1} \sum_{m=1}^{a} \left(\bar{Y}_m - \bar{g}_n\right)^2 . \tag{3.6}$$

In general, the estimator (3.6) is not a consistent estimator of σ_g^2. However, Jones *et al* [38] shows that if the batch size and the number of batches are allowed to increase as the overall length of the simulation, it may be possible to obtain consistency. In addition, a consistent batch mean method can provide an asymptotically valid confidence interval for $E_\pi g$ given by

$$\bar{g}_n \pm t_{a-1} \frac{\hat{\sigma}_{\mathrm{BM}}}{\sqrt{n}}$$

where t_{a-1} is the appropriate quantile from a studentąęs t distribution with $a-1$ degrees of freedom. Notice that burn-in is not required to implement BM.

3.2.2 Overlapping Batch Means

In this section, we consider the use of estimators based on overlapping batches, as in Meketon and Schmeiser [45]. Overlapping batch mean (OLBM) is just a generalization of BM. The mth batch mean is defined as

$$\bar{g}_m = \frac{1}{b} \sum_{i=k}^{b+m-1} X_i$$

for any $m = 1, 2, \ldots, n-b+1$. OLBM averages across all batches and gets the estimate of σ_g^2

$$\hat{\sigma}^2_{\mathrm{OBM}} = \frac{nb}{(n-b)(n-b+1)} \sum_{k=1}^{n-b+1} (\bar{g}_k - \bar{g}_n)^2 .$$

Empirically, OLBM seemed like a big improvement over BM. The asymptotic properties of OLBM are given by Flegal and Jones [43]. They showed $\hat{\sigma}^2_{\mathrm{OBM}}$ converges to σ_g^2 in probability.

3.2.3 Stopping the simulation

An important issue in the use of MCMC-based methods is the assessment of the MC error. Namely, it is important to determine how long a Markov chain must be run in order to

produce a good estimate of the parameters of interest. We consider the consistent batch means method [38] where the standard error based on the batch means method is used to determine a stopping time for an MCMC simulation.

The basic idea is that determine a desired level of accuracy in terms of MCSE for the estimates. For example, suppose we want to report estimates with three significant figures. One way of achieving this goal is through evaluation of the Monte Carlo error estimated by MCSE. MCSEs are periodically calculated for the parameters using consistent batch means as described above. Stop the chain when all MCSEs attain desired levels. Note that one should make sure that the chain is run for a minimum initial length (depending on the problem, observed autocorrelations etc.) so that the standard error estimates themselves are not too unreliable.

We outline the fixed-width method proposed by [38]. Suppose we want to estimate the parameter of interest within ± 0.01. Usual frequentist notions of the desired width of $(1-\alpha)\%$ confidence intervals for the parameters can help determine the desired MCSE. The first confidence interval will be calculated at 10,000 iteration in the chain. If the maximum half-width was greater than 0.01, then 1,000 iterations were added to the chain before checking again. Formally, a simulation was terminated when

$$t^*_{\alpha/2}\frac{\hat{\sigma}_g}{\sqrt{n}} + 0.01I(n < 10,000) < 0.01$$

where $t^*_{\alpha/2}$ is the appropriate critical value leaving an area of $(1-\alpha/2)$ to the right tail of the t-distribution, $I(\cdot)$ is an indicator function, and $\hat{\sigma}_g$ is an estimate of σ_g. Ensuring the resulting confidence intervals are asymptotically valid requires a strongly consistent estimator of σ_g^2. We could use the BM method to estimate it.

Chapter 4

Normalizing Constant Estimation

In some statistical models [17], it possibly needs to generate a sample from a distribution where there are additional parameter-dependent normalization terms necessary to be evaluated. Considering sampling from the posterior distribution

$$p(\theta|\mathbf{y}) = f(\mathbf{y}|\theta)\varphi(\theta), \tag{4.1}$$

where $\varphi(\theta)$ is a prior distribution and $f(\mathbf{y}_0|\theta)$ takes the form

$$f(\mathbf{y}|\theta) = \frac{1}{Z(\theta)} \exp\{-\theta D(\mathbf{y})\}, \tag{4.2}$$

where θ is the reciprocal of the temperature, $Z(\theta)$ is the normalizing constant (or partition function) dependent on θ and not available in closed form, and $D(\mathbf{y})$ is the energy function. The distribution of the form in (4.2) is called Boltzmann distribution in physics. To make inference on θ, a bunch of random samples is necessary to be simulated from (4.1). In general, a standard MCMC algorithm can be used to draw correlated samples from (4.1) in the following way

Initialization: Choose an arbitrary starting value $\theta^{[0]}$.
Iteration: $t+1$ $(t \geq 0)$.

1. Given $\theta^{[t]}$, generate $\tilde{\theta}$ from a transition kernel $q(\cdot|\theta^{[t]})$.
2. Calculate the acceptance prbability

$$\begin{aligned}\alpha &= \min\left\{1, \frac{p(\tilde{\theta}|\mathbf{y})q(\theta^{[t]}|\tilde{\theta})}{p(\theta^{[t]}|\mathbf{y})q(\tilde{\theta}|\theta^{[t]})}\right\} \\ &= \min\left\{1, \frac{Z(\theta^{[t]})}{Z(\tilde{\theta})}\frac{f(\mathbf{y}|\tilde{\theta})\varphi(\tilde{\theta})q(\theta^{[t]}|\tilde{\theta})}{f(\mathbf{y}|\theta^{[t]})\varphi(\theta^{[t]})q(\tilde{\theta}|\theta^{[t]})}\right\}.\end{aligned}$$

3. With probability α, accept $\tilde{\theta}$ and set $\theta^{[t+1]} = \tilde{\theta}$; otherwise reject $\tilde{\theta}$ and set $\theta^{[t+1]} = \theta^{[t]}$.

In the above algorithm, we would need to be able to evaluate values of $Z(\theta)$ at different values. In other words, we have to calculate the ratio, r, of normalizing constants:

$$r = \frac{Z(\theta^{[t]})}{Z(\tilde{\theta})}. \tag{4.3}$$

The problem of evaluating the ratio of normalizing constants is referred to a doubly-intractable distribution [46]. As a result of that $Z(\theta)$ is, however, not easily evaluated in many statistical models, such as spatial models, Bayesian hierarchical models and models for incomplete data, Monte Carlo simulation from the posterior distribution is problematic. Therefore, a wide range of Monte Carlo algorithms have been proposed for circumventing this problem to alleviate the computational intensity. These include the umbrella sampling [47], the Wang-Landau algorithm [48], approximate Bayesian computation (ABC) [49] and the path sampling [41] methods. The primary interest in this chapter is to compare the accuracy and speed of these algorithms in estimating the normalizing constants and then apply them to analyze the fMRI data.

The remainder of this chapter is organized as follows. Section 4.1 review the Monte Carlo algorithms. A simulated study to compare the accuracy and speed of algorithms is given in Section 4.2. Section 4.3 gives the conclusion.

4.1 Monte Carlo Algorithms

In this section, we review several algorithms which are applicable to estimate the intractable normalizing constant in the probability distribution. First, a widely used algorithm, importance sampling, is briefly introduced in section 4.1.1. The drawback of the importance sampling is that when the instrumental distribution is not close to the target distribution, this method gives a terrible estimate. An alternative algorithm is the umbrella sampling in principle similar to the importance sampling, but it uses mixture distributions to cover the range of the target distribution in order to have a better estimation. The algorithm is briefly described in section 4.1.2.

The Wang-Landau (WL) algorithm is a flat-histogram Monte Carlo method performing random walks in the sample space to obtain a close estimation of the density states iteratively. With the estimation of the density states, consequently the normalizing constant can be determined. This algorithm is presented in section 4.1.3.

Zhang and Ma [50] modified the WL algorithm to estimate the normalizing constant directly. This method offers a more general and flexible framework for handling various types of sample spaces (ensembles), especially one in which computation of the density of states is not convenient. Section 4.1.4 gives the description of the algorithm.

Another way to approximate the normalizing constant is path sampling coined by [51] in terms of one dimensional integrals. Gelman and Meng [51] provided a very detailed analysis and application of the path sampling to statistical as well as physical models. We give a brief review in section 4.1.5.

The previous algorithms are designed to estimate the normalizing constant so that it can be used in the Monte Carlo simulation. Two algorithms, single variable exchange [46] and approximate Bayesian Computation (ABC) [49], were developed to generate the sample from the target distribution without estimating the normalizing constant in simulation. But the single variable exchange algorithm needs an exact sample [52] and [53] from the target distribution and ABC needs the choice of metric function to measure the distance of two datasets. Although an exact sample can be generated using the perfect sampling, it is very computationally demanding. Furthermore, a challenging problem in using the ABC algorithm is how to choose a good metric. The algorithms are outlined in sections 4.1.6 and 4.1.7, respectively.

4.1.1 Importance Sampling

To attain an estimate of (4.3), the importance sampling method is often used. Importance sampling [36] is a technique most noted for its ability to draw samples from an alternative distribution and reweigh with an importance $w(\mathbf{y})$ when drawing from $f(\mathbf{y}|\theta)$ is impractical.

Suppose a distribution $g(\mathbf{y})$ has the same support as $f(\mathbf{y}|\theta)$ and are easily simulated. In addition, the normalizing constant in (4.2) can be written as

$$Z(\theta) = \int \exp\{-\theta D(\mathbf{y})\} d\mathbf{y} = \int h(\mathbf{y}|\theta)\lambda(d\mathbf{y}) = \int \frac{h(\mathbf{y}|\theta)}{g(\mathbf{y})} g(\mathbf{y})\lambda(d\mathbf{y}) = E\left[\frac{h(\mathbf{y}|\theta)}{g(\mathbf{y})}\right],$$

where λ is a measure and $h(\mathbf{y}|\theta) = \exp\{-\theta D(\mathbf{y})\}$. Suppose a random sample $\{\mathbf{y}^{[1]}, \ldots, \mathbf{y}^{[T]}\}$ can be easily generated from $g(\mathbf{y})$, then the Monte Carlo estimate of $Z(\theta)$ is

$$\hat{Z}(\theta) = \frac{1}{T}\sum_{t=1}^{T} \frac{h(\mathbf{y}^{[t]}|\theta)}{g(\mathbf{y}^{[t]})}. \tag{4.4}$$

We carry out this approach to obtain the estimates of $Z(\theta^{[t]})$ and $Z(\tilde{\theta})$, respectively, then an estimate of the ratio (4.3) is

$$\hat{r} = \frac{\sum_{t=1}^{T} \frac{h(\mathbf{y}^{[t]}|\theta^{[t]})}{g(\mathbf{y}^{[t]})}}{\sum_{t=1}^{T} \frac{h(\mathbf{y}^{[t]}|\tilde{\theta})}{g(\mathbf{y}^{[t]})}} = \frac{\hat{Z}(\theta^{[t]})}{\hat{Z}(\tilde{\theta})}.$$

Once the estimate of ratio of normalizing constants is obtained, an MCMC algorithm is applied to generate a sample from (4.1). The advantage of this method is that only a random sample is generated from $g(\mathbf{y})$ and we can use the sample in each iteration within MCMC procedures. One's choice of distribution from which to draw the sample will, however, affect the quality of the Monte Carlo estimator. If the $g(\mathbf{y})$ is far from the target distribution, it will give a poor estimate. Rather than generating the sample only from one distribution, the umbrella sampling algorithm tries to generate the sample from a mixture of given distributions.

4.1.2 Umbrella Sampling

In order to cover all the important parts of the sample space it is quite natural to simulate from a mixture. Then, a basic choice of the instrumental distribution is

$$g(\mathbf{y}) = \sum_{j=1}^{M} \omega(\theta_i) l(\mathbf{y}; \theta_i)$$

for some roughly uniform choice of the weights $\omega(\theta_i)$ and distributions $l(\mathbf{y}; \theta_i)$. The idea of using mixtures of distributions as importance sampling distribution was proposed by [47], called umbrella sampling.

The umbrella sampling algorithm proceeds as follows:

Initialization: Decide a weight $\omega(\theta)$ on a set of $\{\theta_1, \ldots, \theta_M\}$. Choose an arbitrary starting value $\theta^{[0]}$.
Iteration: $t + 1$ $(t \geq 0)$.

1. Generate a random sample $\mathbf{y}^{[t]}$ from $l(\mathbf{y}; \theta^{[t]})$.
2. Randomly choose $\theta^{[t+1]} = \theta_i$ in terms of the weight $\omega(\theta)$.

Suppose $\{\mathbf{y}^{[0]}, \ldots, \mathbf{y}^{[T]}\}$ is a Monte Carlo sample generated from the above procedure. Then the law of large numbers implies, for any θ, that

$$\hat{Z}(\theta) = \frac{1}{T} \sum_{i=0}^{T} \frac{f(\mathbf{y}^{[t]}|\theta^{[t]})}{\sum_{j=1}^{M} \pi(\theta_j) l(\mathbf{y}^{[t]}|\theta_j)} \to Z(\theta),$$

as the Monte Carlo sample size T is large enough. Next, we could evaluate the ratio r of the normalizing constants as before.

4.1.3 The Wang-Landau Algorithm

A different approach, the Wang-Landau (WL) algorithm, was recently proposed in [54]. It originally is designed to calculate the density of states in statistical physics by performing a random walk in energy space. It has already been shown not only to be quite powerful but also to have quite wide applicability. Liang [55] generalized the WL algorithm in

statistics for Monte Carlo computation. This method is related in spirit to the multicanonical Monte Carlo [56] and [57] and umbrella sampling techniques [47] and their variations such as the flat histogram Monte Carlo, [58] and [59]. It also has the merit of greater simplicity and, unlike other methods, it is rather straightforward to implement and is, hence, potentially much more useful.

Without loss of generality, we assume $\mathbf{y}$ is a discrete random variable with a probability density given in (4.2). In physics, several quantities of interest are, for example, the internal energy

$$E_\theta(\mathbf{y}) = \sum_{\mathbf{y}} D(\mathbf{y}) f(\mathbf{y}|\theta)$$

and free energy

$$F_\theta(\mathbf{y}) = -\theta \ln Z(\theta),$$

both of which are relative to the normalizing constant

$$Z(\theta) = \sum_{\mathbf{y}} \exp\{-\theta D(\mathbf{y})\},$$

where the sum in $\mathbf{y}$ runs over all possible states (or configurations). No exact solutions are known for this summation, except for very simple cases, which are not adequate for most applications of interest.

In practice, it is impossible for a computer to handle this summation. A number of computational difficulties prohibit the application of the models to a wider class of problem. Therefore, the problem of calculating the normalizing constant in a computationally amenable way is of great interest. As a result, several Monte Carlo simulation techniques are developed to generate the sample from (4.2) to estimate the normalizing constant. They, however, often suffer from slow mixing of the Markov chain which occurs particularly in the phase transition. Slow mixing reduces the effective number of samples and sometimes leads to wrong results sensitive to initial state of the Markov chain. To overcome this problem, Wang and Landau [54] introduced a flat-histogram algorithm that simulates a biased random walk in energy space, systematically estimate the density of states, U_ξ, which is the number of state with energy ξ of the system defined by

$$U_\xi = \#\{\mathbf{y} : D(\mathbf{y}) = \xi\}.$$

Since the classical normalizing constant can be written as a sum over all energy levels instead of over all states. That is, we can rewrite it in a different, but equivalent, form

$$Z(\theta) = \sum_{\mathbf{y}} \exp\{-\theta D(\mathbf{y})\} = \sum_{\xi} U_{\xi} \exp\{-\theta\xi\},$$

and the corresponding probability density of ξ is

$$f(\xi|\theta) = \frac{U_{\xi}}{Z(\theta)} \exp\{-\theta\xi\}.$$

The WL algorithm [60] is a flexible, powerful, iterative algorithm to directly estimate U_{ξ} instead of trying to extract samples from the probability distribution produced by "standard" Monte Carlo simulations. It offers substantial advantages over existing approaches since it estimates directly the density of states U_{ξ}. With an accurate estimate of U_{ξ} for all energies, one can calculate the normalizing constant by summing over ξ running over all the existing energy levels.

To estimate the U_{ξ}, we could simply perform the simple random walk on the energy space. But the main difficulty of doing a simple random walk to determine U_{ξ} is that the walk would spend most of its time visiting the same energy values over and over again and would not reach the values of U_{ξ} less probable. The idea of the WL algorithm is to do a biased random walk in energy space by flipping single spins at random and accepting the changes with a probability that is proportional to the reciprocal of the density of states. In this way energy values that would be visited often using a simple random walk would be visited less often because they have a larger density of state. Finally, all the energy levels will be equally visited and a flat histogram is generated for the energy distribution to determine the convergence.

Analogous to the Metropolis algorithm, the WL algorithm accepts the new energy ξ' with probability

$$\alpha = \min\left\{1, \frac{U_{\xi}}{U_{\xi'}}\right\}.$$

The action is to create an equal probability of visiting each energy level in the system. In other words, the number of visits in different energy levels should be uniformly distributed. Therefore, there is an important quantity introduced, the visit histogram $H(\xi)$, in the algorithm to record where the random walk has been. In each Monte Carlo step, the

histogram of visited sites $H(\xi)$ is updated based on the accepted state and, in addition, the estimated density of states is changed by

$$\ln \hat{U}_\xi = \ln \hat{U}_\xi + \ln \eta,$$

where η is the modification factor. The evolution continues until the histogram becomes reasonably flat.

Surely, it is impossible to obtain a perfectly flat histogram. To test if histogram is flat, we first calculate the average number of entries in the histogram

$$h = \frac{1}{M} \sum_{i=1}^{M} H(\xi_i)$$

where M is the number of energy levels. Then we say the histogram is flat if

$$H(\xi_i) \geq \epsilon \times h \quad i = 1, \ldots M,$$

where ϵ is taken between 0.75 and 0.95 depending on the problem. When the flatness of $H(\xi)$ meets a criterion set in advance, the modification factor η will then be replaced by a new value $\sqrt{\eta}$, and histogram $H(\xi)$ will be reset to 0 and a fresh histogram is accumulated and tested for flatness while $\hat{U}_\xi$ retained unchanged. The process is repeated until η is very small, say, $\ln \eta < 10^{-9}$, and the estimated $\hat{U}_\xi$ is considered final.

The procedure of the WL algorithm is given below

Initialization: Let $H(\xi) = 0$ and $\hat{U}_\xi = 1$ for all possible energy levels, and $\ln \eta = 1$.
Iteration: $t + 1$ ($t \geq 0$).

1. Compute the present energy $\xi^{[t]}$ and choose a spin at random and make a trial flip and compute the energy ξ' and accept $\xi^{[t+1]} = \xi'$ with probability

$$\alpha = \min\left\{1, \frac{\hat{U}_{\xi^{[t]}}}{\hat{U}_{\xi'}}\right\},$$

   otherwise $\xi^{[t+1]} = \xi^{[t]}$.

2. Update the current value of $\hat{U}_\xi$ corresponding to $\xi^{[t+1]}$ by multiplying a modification factor η . In practical simulation, in order to meet the precision limits that a compute can handle, $\ln U_\xi$ is often used rather than $\hat{U}_\xi$. Therefore,

$$\ln \hat{U}_\xi = \ln \hat{U}_\xi + \ln \eta.$$

3. Update the the number of visits in the corresponding energy ξ

$$H(\xi) = H(\xi) + 1.$$

4. Repeat the Step 1-3 until a flat histogram is obtained. Reset $H(\xi) = 0$ and decrease $\eta = \sqrt{\eta}$.

5. Once η less than a prespecified amount, then exit and the final $\hat{U}_\xi$ is obtained. Otherwise repeat Step 1-4.

The algorithm does not satisfy detailed balance condition until the modification factor η decreases to zero. See [61] for some recent effort toward understanding and updating the WL algorithm.

4.1.4 The Modified Wang-Landau Algorithm

An obstacle to further exploitation of the WL approach is that the discrete representation of U_ξ on an energy space causes the number of levels to increase extensively with the system size. If our goal is purely to estimate the normalizing constant, Zhang and Ma

[50] developed an algorithm that better serves our purpose. The modified algorithm uses the ideas from both the simulated tempering algorithm and the WL algorithm and allows us to compute the normalizing constant directly by taking two types of trial moves: an energy move and a temperature move. Before each Monte Carlo step, a fixed probability is used to determine which type of move the system takes.

In an energy move, as usual, a standard Metropolis Monte Carlo is applied to change a spin at fixed temperature. On the other hand, in temperature move, randomly choose a temperature, say, $\tilde{\theta}$, and accept the chosen temperature with probability

$$\alpha = \min\left\{1, \frac{Z(\theta)\exp\{-\tilde{\theta}\xi\}}{Z(\tilde{\theta})\exp\{-\theta\xi\}}\right\},$$

where ξ is the present energy, $Z(\theta)$ and $Z(\tilde{\theta})$ are the normalizing constants at current temperature θ and a trial temperature $\tilde{\theta}$, respectively. Here, since $Z(\theta)$ are unknown, we substitute $Z(\theta)$ by an estimate $\hat{Z}(\theta)$.

How to obtain an estimate $\hat{Z}(\theta)$ of $Z(\theta)$? After each MC step either updating energy or temperature, the estimated normalizing constant at the present temperature is multiply by a factor η, that is,

$$\ln \hat{Z}(\theta) = \ln \hat{Z}(\theta) + \ln \eta.$$

By repeating the above procedure, it can be shown that the estimated normalizing constant can eventually converge within certain fluctuations proportional to $\sqrt{\ln \eta}$.

The procedure of the algorithm is given below

Set up a series of inverse temperatures $\{\theta_1, \ldots, \theta_M\}$ of interest. Choose the probability p of a trial temperature change.
Initialization: Let $\theta^{[0]} = \theta_i$, $\hat{Z}(\theta_i) = 1$ for all $i = 1, \ldots M$, and $\ln \eta = 1$.
Iteration: $t + 1$ $(t \geq 0)$

1. Compute the energy of the current state $\xi^{[t]}$.
2. Choose a move: With the probability p to have a temperature move.
   (a) Temperature Move: Randomly choose a temperature, say, θ_j, and accept the temperature $\theta^{[t+1]} = \theta_j$ with probability
   $$\alpha = \min \left\{ 1, \frac{\hat{Z}(\theta^{[t]}) \exp\{-\theta^{[t+1]} \xi^{[t]}\}}{\hat{Z}(\theta^{[t+1]}) \exp\{-\theta^{[t]} \xi^{[t]}\}} \right\}.$$
   (b) Energy Move: Perform a number of Monte Carlo moves per spin.
3. Whichever type of change chosen, update the normalizing constant at current temperature by
   $$\ln \hat{Z}(\theta) = \ln \hat{Z}(\theta) + \ln \eta.$$
4. Reduce η as in the WL algorithm, that is, $\eta = \sqrt{\eta}$.
5. Stop the simulation when η less than is prespecified amount.

This algorithm yields the logarithm of the normalizing constants up to a constant at the temperatures of interest. If the current estimate of the normalizing constant for a particular temperature is too low, the algorithm will favor moves toward that temperature, thus eventually approaching the correct value for the normalizing constant. Sometimes the simulation does not accurately compute the low temperature values for Z because not enough time was spent there. One way to tell is to check if $\ln Z$ is a monotonically increasing function of temperature. Thus, it is important to do several independent runs.

4.1.5 Path Sampling

An alternative the previous algorithm is thermodynamic integration. This method, also called path sampling, was explained in greater detail in [41] and [62].

Without loss of generality, we suppose $\theta_1 > \theta_0$. Our goal is to obtain a log-ratio of the normalizing constants

$$\ln r = \ln \frac{Z(\theta_0)}{Z(\theta_1)}. \tag{4.5}$$

We take the derivative of $\ln Z(\theta)$ with respective to θ, then

$$\begin{aligned}
\frac{\partial \ln Z(\theta)}{\partial \theta} &= \frac{1}{Z(\theta)} \frac{\partial Z(\theta)}{\partial \theta} \\
&= \frac{1}{Z(\theta)} \frac{\partial}{\partial \theta} \int h(\mathbf{y}|\theta) \lambda(d\mathbf{y}) \\
&= \frac{1}{Z(\theta)} \int \frac{\partial h(\mathbf{y}|\theta)}{\partial \theta} \lambda(d\mathbf{y}) \\
&= \int \frac{h(\mathbf{y}|\theta)}{Z(\theta)} \frac{1}{h(\mathbf{y}|\theta)} \frac{\partial h(\mathbf{y}|\theta)}{\partial \theta} \lambda(d\mathbf{y}) \\
&= \int \frac{\ln h(\mathbf{y}|\theta)}{\partial \theta} f(\mathbf{y}|\theta) \lambda(d\mathbf{y}) \\
&= E\left[\frac{\partial \ln h(\mathbf{y}|\theta)}{\partial \theta} \right].
\end{aligned}$$

Integration over $[\theta_0, \theta_1]$, we have

$$\begin{aligned}
\ln r &= \ln Z(\theta_0) - \ln Z(\theta_1) \\
&= \int_{\theta_0}^{\theta_1} \frac{\partial \ln Z(\theta)}{\partial \theta} d\theta \\
&= \int_{\theta_0}^{\theta_1} E\left[\frac{\partial \ln h(\mathbf{y}|\theta)}{\partial \theta} \right] d\theta.
\end{aligned}$$

An important feature of above equation is that one can use MCMC methods to yield a sample from the unnormalized density $h(\mathbf{y}|\theta)$ and then $E\left[\frac{\partial \ln h(\mathbf{y}|\theta)}{\partial \theta}\right]$ can be estimated as average over this sample. We repeated the procedure for a series of values of θ regularly spaced over $[\theta_0, \theta_1]$. Once $E\left[\frac{\partial \ln h(\mathbf{y}|\theta)}{\partial \theta}\right]$ is obtained, we can use numerical integral based on the B-spline interpolation in the sample expectations to produce an estimate of (4.5).

4.1.6 Single-Variable Exchange Algorithm

All previous algorithms are required to evaluate the ratio of normalizing constants in each Monte Carlo simulation, but it is hard as well as time consuming. Murray *et al*. [46] proposed an algorithm to avoid evaluating the ratio; however, it is necessary to generate an exact sample from the target distribution. Murray called it the single-value exchange algorithm. The perfect sampling method [52] and [53] is a way to obtain an observation exactly from target distribution. The algorithm proceeds by going far enough into the past that chain values started from all the different points of the state space and then stop when all coalesce at time 0. The time to coalescence in the algorithm can be significant large. In addition, the perfect sampling method is considerably hard to implement in many practical application.

The single-variable exchange algorithm is simpler than the single auxiliary variable algorithm [63]. Murray *et al* give an intuitive argument of how it works in practice without rigorous proof. The algorithm proceeds as follows:

Initialization: Let $\theta^{[0]}$ be the initial state.
Iteration: $t+1$ $(t \geq 0)$.

1. Generate $\tilde{\theta}$ from a proposal distribution, $q(\cdot|\theta^{[t]})$.
2. Generate an auxiliary variable, $\tilde{\mathbf{y}}$, from
$$f(\mathbf{y}|\tilde{\theta}) = \frac{1}{Z(\theta)} \exp\{-\tilde{\theta} D(\mathbf{y})\}.$$
3. Calculate the acceptance probability
$$\alpha = \min\left\{1, \frac{q(\theta^{[t]}|\tilde{\theta})\pi(\tilde{\theta}) f(\mathbf{y}|\tilde{\theta})}{q(\tilde{\theta}|\theta^{[t]})\pi(\theta^{[t]}) f(\mathbf{y}|\theta^{[t]})} \times \frac{f(\mathbf{y}|\theta^{[t]})}{f(\tilde{\mathbf{y}}|\tilde{\theta})}\right\}.$$
4. With probability α, accept $\tilde{\theta}$ and set $\theta^{[t+1]} = \tilde{\theta}$; otherwise reject $\tilde{\theta}$ and set $\theta^{[t+1]} = \theta^{[t]}$.

4.1.7 Approximate Bayesian Computation

Despite several methods developed to provide ways of simulating the sample from the target distribution with intractable constants, all of them require a lot computational effort to estimate the normalizing constants. To avoid directly evaluation of the normalizing constants, approximate Bayesian computation in Markov chain Monte Carlo was developed by [64].

Approximate Bayesian computation (ABC) algorithms provide a way of simulating a random sample from the target distribution. In the ABC-MCMC algorithm, a candidate parameter $\tilde{\theta}$ drawn from a proposal distribution, and a random sample $\tilde{\mathbf{y}}$ is generated from the likelihood function conditional on $\tilde{\theta}$. If the simulated data $\tilde{\mathbf{y}}$ and observed data $\mathbf{y}$ are sufficiently close, then $\tilde{\theta}$ is accepted. To measure the closeness between $\tilde{\mathbf{y}}$ and $\mathbf{y}$, we need to define a metric function ρ. When $\rho(\tilde{\mathbf{y}}, \mathbf{y})$ is less than a tolerance ϵ, it can say $\tilde{\mathbf{y}}$ and $\mathbf{y}$ are close enough. In some cases where the data is complicated, it is usually difficult to define a suitable metric function for full data. Instead, we can define a metric function on summary statistics $S(\mathbf{y})$ which can capture the same information contained in the full data $\mathbf{y}$.

The algorithm proceeds as follows:

Initialization: Choose an arbitrary starting value $\theta^{[0]}$.
Iteration: $t+1$ $(t \geq 0)$.

1. A candidate $\tilde{\theta}$ is generated from a proposal distribution $q(\cdot|\theta^{[t]})$.
2. Generate a data $\tilde{\mathbf{y}}$ from $f(\cdot|\tilde{\theta})$.
3. Calculate the acceptance probability

$$\alpha = \min\left\{1, \frac{\pi(\tilde{\theta})q(\tilde{\theta}|\theta^{[t]})}{\pi(\theta^{[t]})q(\theta^{[t]}|\tilde{\theta})} 1\left(\rho\left(S(\tilde{\mathbf{y}}), S(\mathbf{y})\right) \leq \epsilon\right)\right\}$$

4. With probability α, accept $\tilde{\theta}$ and set $\theta^{[t]} = \tilde{\theta}$; otherwise reject $\tilde{\theta}$ and set $\theta^{[t+1]} = \theta^{[t]}$

This method requires the selection of a suitable metric ρ as well as a choice of ϵ. When

$\epsilon \to \infty$, the sample is generated from the prior, and when $\epsilon \to 0$, the sample is from the target distribution. The choice of ϵ reflects the interplay between computability and accuracy. For a given ρ and ϵ, the samples are independent and identically distribution as $p\left(\theta|\rho\left(S(\tilde{\mathbf{y}}), S(\mathbf{y})\right) \leq \epsilon\right)$. If the summary statistics S are near-sufficient and ϵ is small, then $p\left(\theta|\rho\left(S(\tilde{\mathbf{y}}), S(\mathbf{y})\right) \leq \epsilon\right)$ should be reasonable approximate to $p(\theta|\mathbf{y})$. The big issue of the method is how to choose the metric ρ.

To improve the efficiency of the ABC-MCMC algorithm in terms of acceptance rate, ABC partial rejection control (ABC-PRC) is proposed by [65].

4.2 Simulation Study

In order to compare the accuracy and speed of the algorithms described above, we perform a Monte Carlo simulation for an Ising model with free boundary condition on a 32×32 square-lattice E. Let $\mathbf{y} = \{y_1, \ldots, y_n\}$ be data on E with y_i is represented by a binary indicator, $+1$ and -1 and assume a uniform prior $\varphi(\theta)$ on θ. The posterior distribution is characterized by

$$p(\theta|\mathbf{y}_0) = \frac{1}{Z(\theta)} \exp\left\{-\theta \sum_{i \sim j} \left[w_{i,j} \times 1(y_i = y_j)\right]\right\} \varphi(\theta), \tag{4.6}$$

where $i \sim j$ denotes the nearest-neighbors between i and j and $w_{i,j}$ is prespecified weights. In this simulation study, we assumed $w_{i,j} = 1$ and θ is restricted over the interval $[0, 1]$.

Let $D(\mathbf{y}) = \sum_{i \sim j} \left[w_{i,j} \times 1(y_i = y_j)\right]$. Then, the normalizing constant can be expressed as

$$Z(\theta) = \left(\sum_{\mathbf{y} \in \Upsilon} \exp\{-\theta \cdot D(\mathbf{y})\}\right)^{-1},$$

where Υ is the number of all possible outcomes in the configuration of $\mathbf{y}$. To simulate a sample from (A.1), the normalizing constant $Z(\theta)$ is necessarily to be evaluated in advance. It's, however, extremely difficult to compute the normalizing constant $Z(\theta)$ because it is required to make a summation over all possible values of $\mathbf{y}$, which involve 2^n terms where n is the number of data points. Therefore, we use the algorithms presented above to evaluate the normalizing constant before performing MH sampler to generate the sample from (A.1).

Given a $\theta = 0.4$, a data $\mathbf{y}$ is generated using perfect sampling [53]. Table (4.1) gives the simulated result based on 10,000 samples. In the umbrella sampling algorithm, the

Table 4.1: The estimate of θ from seven different algorithms.

Algorithm	$\hat{\theta}$	MCSE($\hat{\theta}$)	Acceptance rate	Time Elapse
Umbrella Sampling	0.336	0.008	47.33%	6344 seconds
WL	0.388	0.015	34.35%	30 seconds
Modified WL	0.391	0.015	34.90%	171 seconds
Single Variable Exchange	0.415	0.013	47.56%	6980 seconds
ABC	0.394	0.021	37.73%	9 seconds
Path Sampling	0.393	0.015	35.04%	90 seconds

instrumental distribution is a mixture of independent bernoulli given as

$$g(\mathbf{y}) = \sum_{k=1}^{K} w(\rho_k)\rho_k^x(1-\rho_k)^{n-x} \tag{4.7}$$

for some proportions $0 \leq w(\rho_i) \leq 1$ where $w(\rho_1)+\cdots+w(\rho_K) = 1$ and $x = \sum_{i=1}^{n} I(y_i = 1)$.

1. Generate a sequence of data from (4.7), say $\{\mathbf{y}^{[1]}, \ldots, \mathbf{y}^{[N]}\}$, as follows

   (a) Generate $\rho^{[i]}$ with
   $$p(\rho^{[i]} = \rho_k) \propto w(\rho_k)$$

   (b) Generate $\mathbf{y}^{[\mathbf{i}]} = \{y_1^{[i]}, \ldots, y_n^{[i]}\}$ given $\rho^{[i]}$ generated from Step 1(a),
   $$p(\mathbf{y}^{[\mathbf{i}]}) = p(\{y_1^{[i]}, \ldots, y_n^{[i]}\}) = \prod_{i=1}^{n} \rho^{y_i^{[i]}}(1-\rho)^{1-y_i^{[i]}} = \rho^x(1-\rho)^{n-x}.$$

2. Evaluate the normalizing constant in terms of data generated from Step 1. For any θ, we have
   $$Z(\theta) \approx \frac{1}{N}\sum_{i=1}^{N} \frac{\exp\left\{-\theta D(\mathbf{y}^{[i]})\right\}}{\sum_{k=1}^{K} w(\rho_k)\rho_k^x(1-\rho_k)^{n-x}}.$$

   (a) Generate parameter θ using MH algorithm.

i. Generate $\theta \sim p(\theta|\theta^{[t]})$, for instance, $p(\theta|\theta^{[t]})$ is taken as

$$(1-\vartheta)N\left(\theta^{[t]}, (2.38)^2\sigma\right) + \vartheta N\left(\theta^{[t]}, \sigma\right)$$

where σ is the current empirical estimate of standard error based on the run so far and ϑ is a small positive constant (we take $\vartheta = 0.05$) [66].

ii. After a fixed number of simulation, say m_1, we estimate the parameter θ, $\hat{\theta}$ by

$$E(\theta|y) \approx \frac{1}{m_1}\sum_{i=1}^{m_1}\theta_i.$$

(b) Adapting the proportion $w(\rho_i)$, put more weight around $\hat{\theta}$, then go to Step 1. Here is the proposal to update the proportion

$$w(\theta_i) = \frac{\exp\{-|\theta_i - \hat{\theta}|\}}{\sum_{i=1}^{n}\exp\{-|\theta_i - \hat{\theta}|\}}$$

From this simulation study, except the Umbrella sampling method, the other give a quite good estimate of θ. Several importance things should be noticed:

1. The umbrella sampling gives a very unstable estimate of θ in our simulation study. It probably over- or underestimates θ in different simulations. A mixture of Bernoulli distributions is chosen for computational convenience in this simulation study. Perhaps, the other distributions can be used to improve the estimation.

2. In the modified WL and path sampling algorithm we chose 50 points from $[0, 1]$ of θ to equally partition the interval into 50 spaces.

3. In the single variable exchange algorithm, one needs patience to get an exact sample from the target distribution either by the coupling from the past algorithm or by Fill's interruptible algorithm. Here we limit the value of θ within [0, 0.5] in such a way that the algorithm is not terminated with memory of a computer running out in the simulation.

4. ABC methodology needs the least computational effort among these algorithm because it is unnecessary to expensively evaluate the likelihood within the simulation, but it is very difficult to choose a metric ρ. The promising solution to the issue is using the partial rejection control approach within ABC.

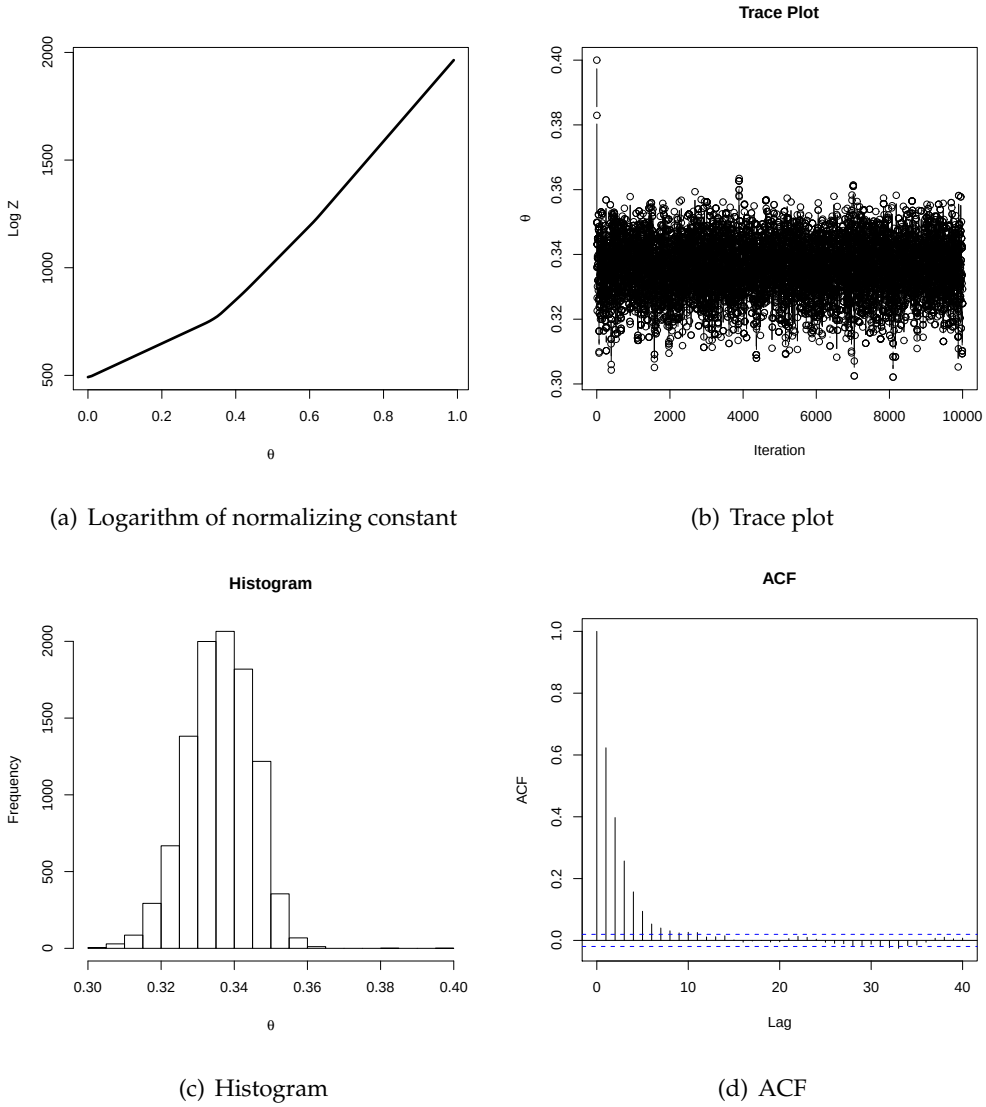
An estimate of the logarithm of the normalizing constant using the umbrella sampling, WL, modified WL and path sampling algorithms is shown in Figures 4.1(a), 4.2(a), 4.3(a), and 4.4(a). The umbrella sampling is pretty unreliable in estimating the normalizing constant.

The mixing rate are almost same for the umbrella sampling, WL, modified WL and path sampling algorithms in terms of the autocorrelation graphs given in Figures 4.1(d), 4.2(d), 4.3(d), and 4.4(d). The ABC and SVE algorithms produce bad mixing rates shown in Figures 4.5(d) and 4.6(d).

The distributions of data generated from the simulation are given in in Figures 4.1(c), 4.2(c), 4.3(c), 4.4(c), 4.5(c) and 4.6(c). The trace plots shown in Figures 4.1(b), 4.2(b), 4.3(b), 4.4(b),4.5(b), and 4.6(b) provide a visual inspection of convergence. They are quite stable.

4.3 Conclusion

In this chapter, we compare the speed and accuracy of different algorithms to estimate the spatial coefficient θ. In terms of the simulation study, path sampling and modified Wang-Landau algorithms sever our purpose not only to estimate the normalizing constants but also to extract the sample from target distribution. The umbrella sampling is a easy and fundamental way to estimate the normalizing constant, but it is very computationally intensive and gives unreliable estimate. Although ABC seems very promising to reduce the computation cost without evaluation of the normalizing constants, it is extremely hard to select a good metrics ρ in complex or high-dimension models so that the method is difficult to be implemented. The SVE algorithm needs to use the perfect sampling to generate an exact sample while avoiding estimating the normalizing constants. The use of perfect sampling is not straightforward such that SVE algorithm is not quite applicable in practice. In addition, the challenge of analysis of fMRI data is how to alleviate the computational effort since the size of the data is usually very huge. Therefore, we consider path sampling and modified Wang-Landau algorithms in the following analysis of fMRI data.



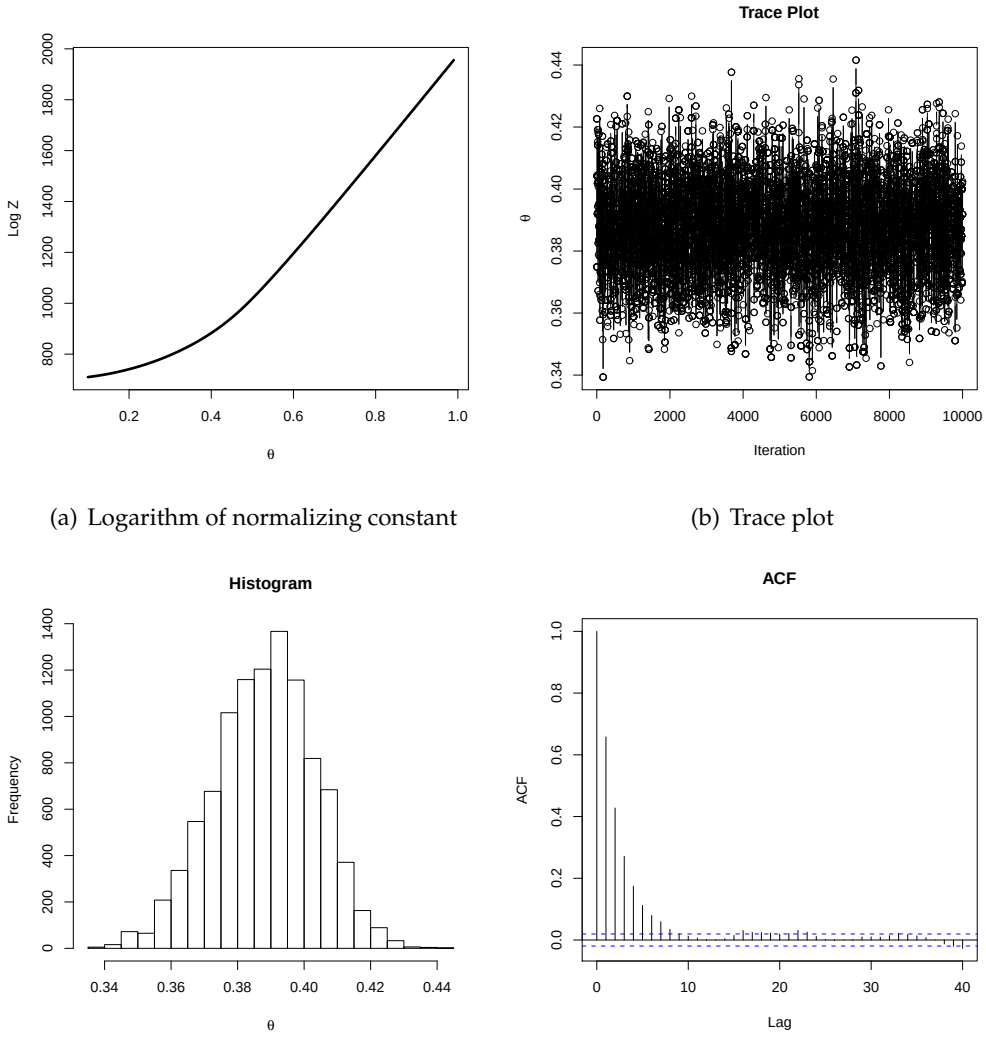
(a) Logarithm of normalizing constant

(b) Trace plot

(c) Histogram

(d) ACF

Figure 4.1: (a) is the logarithm of normalizing constant, (b) the trace plot, (c) the histogram , and (d) the autocorrelation function based on 10,000 simulated samples.



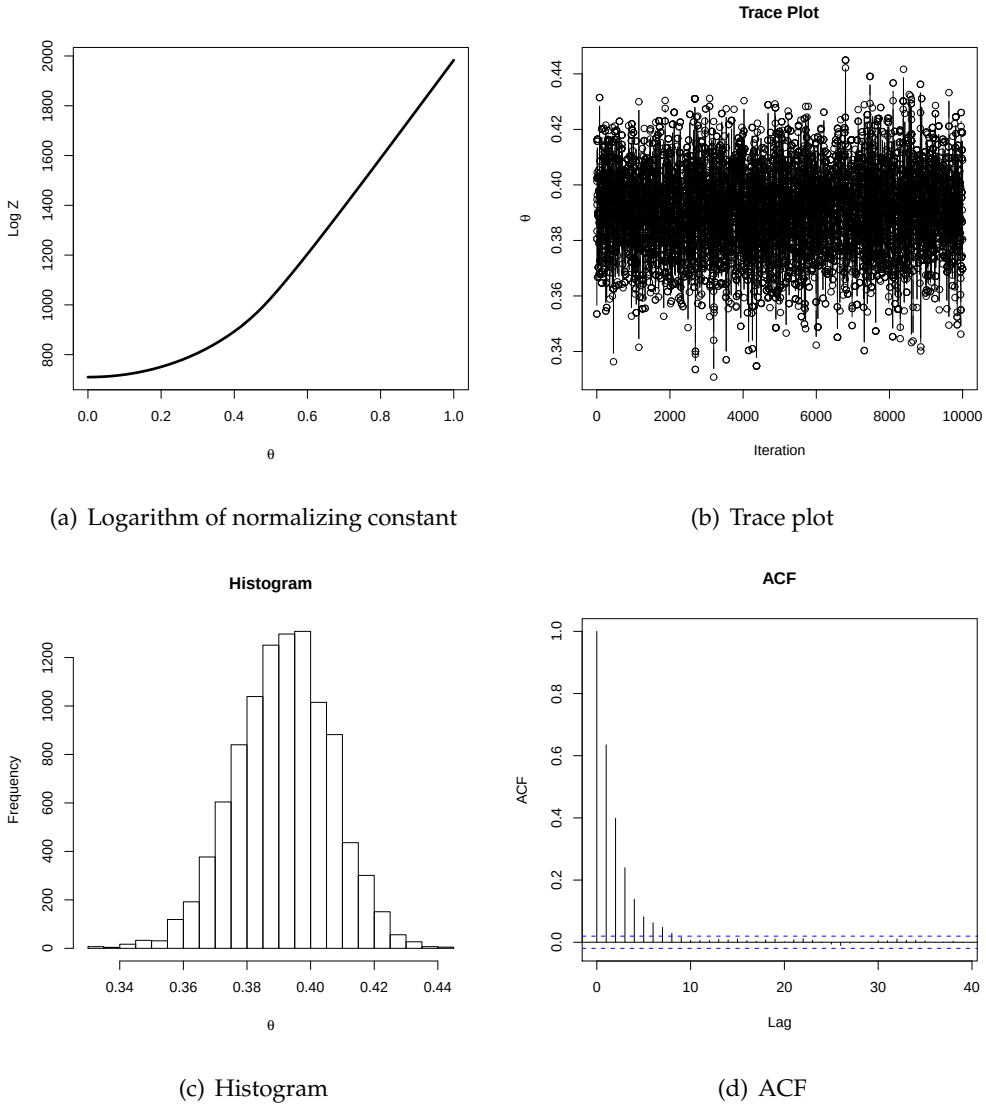
(a) Logarithm of normalizing constant

(b) Trace plot

(c) Histogram

(d) ACF

Figure 4.2: (a) is the logarithm of normalizing constant, (b) the trace plot, (c) the histogram , and (d) the autocorrelation function based on 10,000 simulated samples.



(a) Logarithm of normalizing constant

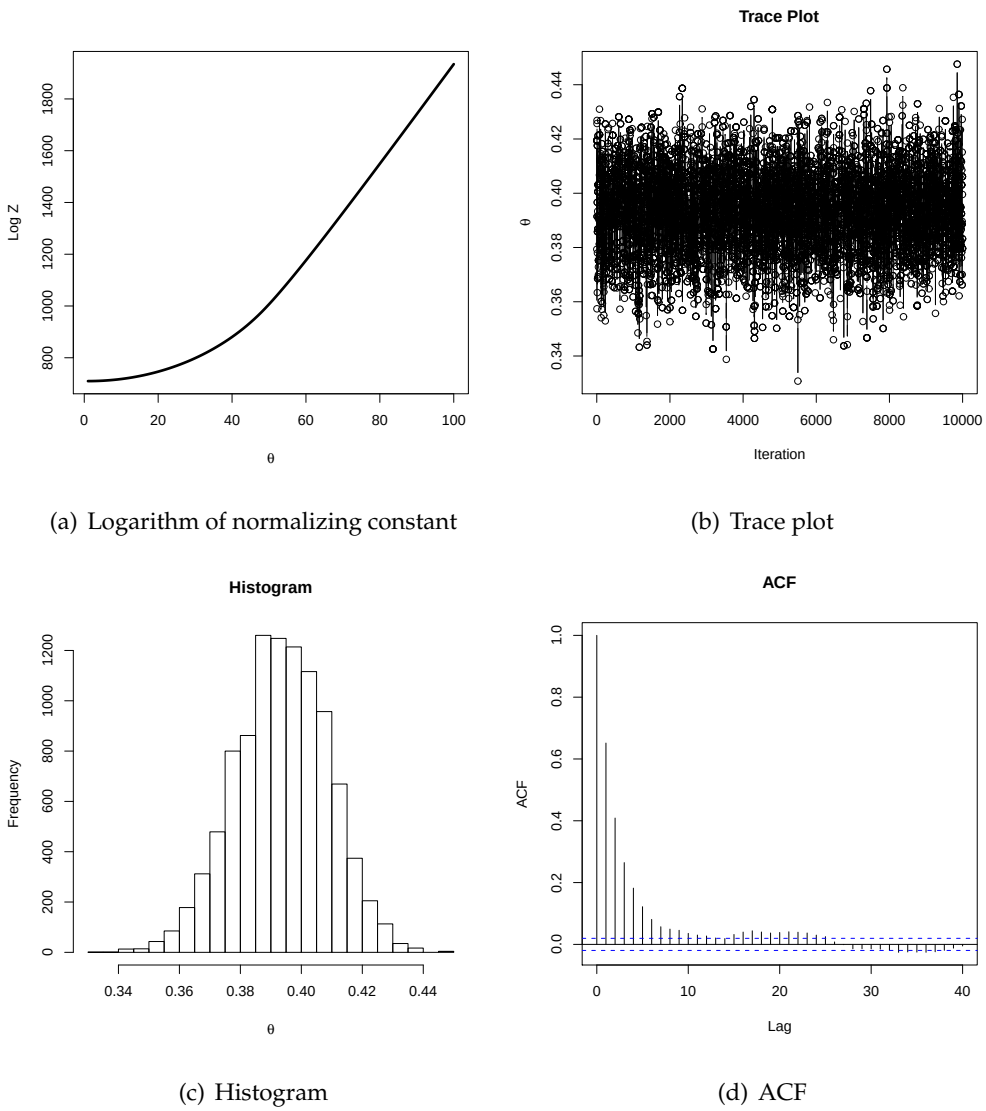
(b) Trace plot

(c) Histogram

(d) ACF

Figure 4.3: (a) is the logarithm of normalizing constant, (b) the trace plot, (c) the histogram , and (d) the autocorrelation function based on 10,000 simulated samples.

Path Sampling



(a) Logarithm of normalizing constant

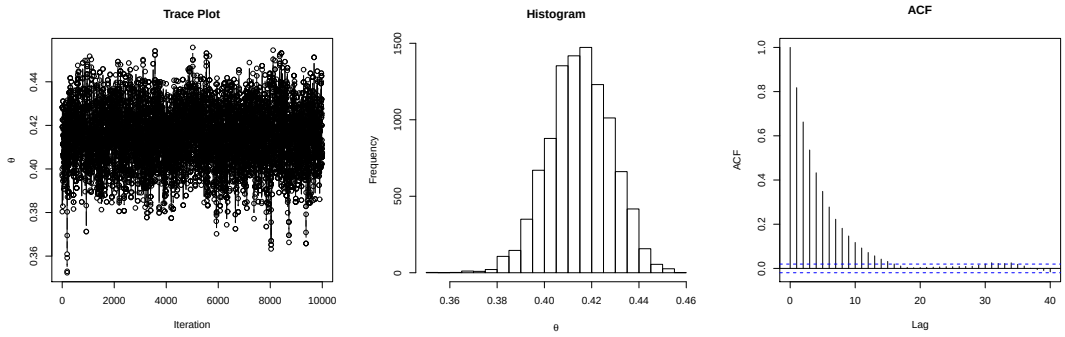
(b) Trace plot

(c) Histogram

(d) ACF

Figure 4.4: (a) is the logarithm of normalizing constant, (b) the trace plot, (c) the histogram , and (d) the autocorrelation function based on 10,000 simulated samples.

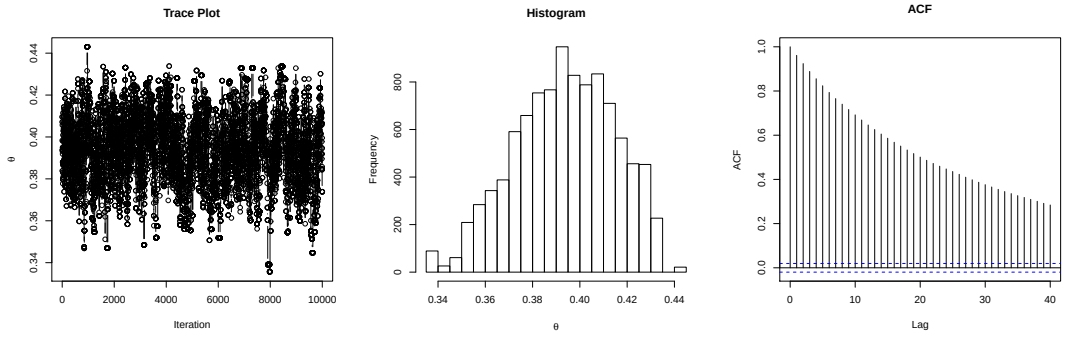
Single Variable Exchange Algorithm



(a) Trace plot (b) Histogram (c) ACF

Figure 4.5: (a) is the trace plot, (b) the histogram , and (c) the autocorrelation function based on 10,000 simulated samples.

Approximate Bayesian Computation



(a) Trace plot (b) Histogram (c) ACF

Figure 4.6: (a) is the trace plot, (b) is the histogram , and (c) is the autocorrelation function based on 10,000 simulated samples.

Chapter 5

Bayesian Hierarchical Methods for the Spatial Modeling of fMRI Data

One of the major objectives of fMRI studies is to determine which areas of the brain are activated in response to a stimulus or task. To make inferences about task-specific changes in underlying neuronal activity, various statistical models are used such as general linear models (GLMs) [14]. In the GLM framework, the fMRI time series in each voxel is modeled independently by a linear combination of several regressors corresponding to some experimental effects. The error term in a GLM is assumed to be a stochastic process such as an autoregressive model of small order to account for the temporal correlation arising from the fact that stimuli are presented continuously or periodically over time. Hypothesis testing is then performed in the usual manner. In other words, voxels are tested to see if they are activated in response to a given task. This yields a large number of t- or F-statistics which together form a statistic parametric mapping (SPM). Use of SPM leads to a problem of multiplicity, so the next step is to threshold the t- or F-statistics while trying to maintain a given overall error rate. At a given level of significance, Gaussian random field (GRF) theory [15] and [67] is applied to threshold the image to determine which parts of the brain were activated. However, this method is limited because the result is based on the assumption of a stationary Gaussian random field theory which relies on several assumptions including the following: the image has the same parametric distribution at each spatial location, the point spread function has two derivatives at the origin, sufficient

smoothness to justify application of continuous RF theory, and a sufficiently high threshold for the asymptotic results to be accurate [15]. Often those assumption are not satisfied in fMRI settings.

The Bayesian paradigm provides an appropriate framework for making inference using complex models and to overcome the multiple comparison problems. It also constitutes a natural but rigorous theory for combining prior and experimental information. Here, we review some Bayesian approaches in the analysis of fMRI data. Genovese [68] modeled the baseline drift nonparametrically with regression splines. For the activation profile, he proposed a nonlinear parametric model for shaped parameters varying from voxel to voxel and being estimated jointly with all other parameters. The exploitation of spatial characteristics in fMRI data has only recently begun to be investigated. Spatial correlation occurs because all of the voxels are in the brain of a single subject and voxels that are closed to each other in the brain might be correlated in their activation pattern. Therefore, Gossl et al [69] proposed a separable spatio-temporal model to simultaneously incorporate temporal and spatial dependencies between voxels directly. The spatial dependencies were characterized using Conditional Autoregression (CAR) or Markov random field (MRF) priors. Woolrich et al [70] used the space-time simultaneously specified autoregressive (STSAR) model to analyze the fMRI data by considering the temporal and spatial correlation simultaneously. They assumed a STSAR model with different orders of AR dependence both spatially and temporally. The model can automatically select the order of AR in each voxel; however, it is very computationally intensive. Penny et al [71] proposed a fully Bayesian analysis with spatial priors defined over regression coefficients of a GLM, using Gaussian Markov random fields, and the errors are modeled as an autoregressive process. They used the variational Bayes (VB) approach to reduce the computational intensity. Quiros et al [72] used a Bayesian spatiotemporal model to analyze block-design BOLD fMRI data where Gaussian MRF priors are assumed only on activation voxels to model the spatially varying response level of activated regions. However, the spatial correlation is likely more complicated than if it were based on simple physical distances. That is, it is also likely that voxels distant from each other can co-activate. As one example, voxels in the different language processing areas might be highly correlated, even if they aren't in physical proximity. Bowman [73] incorporated a functional

defined distance metric into a parametric structure for spatial correlation within a region of interest (ROI) and included temporal correlation between scans.

In this chapter, we seek to extend the work of Smith et al [17] to incorporate the spatial with temporal characteristics using Bayesian hierarchical models. In their model, to alleviate the computation effort, they did not take into account the spatial correlation between observations over time in voxels. We simply assume an autoregression model in error term to model temporal correlation and assign a spatial prior over the regression coefficients to model the spatial correlation. The Bayesian approach combines both the likelihood function from data along with assuming a distribution of errors and the prior probability of activation. These priors can enter as known values or can be estimated from the data, provided we have observed multiple instances of the effect we are interested in. The purpose of this chapter is to describe a spatial Bayesian variable selection approach with different priors to detect the activation of neurons in response to a stimulus. An MCMC sampling scheme is also provided.

5.1 Spatial Bayesian Variable Selection Models

The main goal of the analysis fMRI data is often to localize the brain activity triggered by a specific stimulus. To achieve the goal, we utilize the spatial Bayesian variable selection method to allow automatic detection of the activation of a voxel corresponding to a given effect using indicator variables. In most standard GLM approaches, each voxel is analyzed separately with little consideration of the spatial correlation among neighboring voxels. To incorporate the spatial dependence in fMRI data, Smith *et al* [17] introduced a spatial MRF prior, Ising prior, placed on indicator variables. This prior is a binary Markov random field (MRF) and is useful for spatially smoothing the indicator variables representing whether or not the variable is zero or nonzero in each regression coefficient of each voxel. In their work, they assumed the error term is a multivariate normal such that the voxels in different times are independent. However, those voxels near another in time tend to have similar values gives rise to the temporal autocorrelation. Thus, repeated measurements at the same brain location over time are not independent. To account for this dependence, we assumed the error terms have AR(1) dependence. Other temporal

dependence structure could be used, but we focus on AR(1) here because it is sensible and it is very convenient from a computational perspective.

A linear regression model with AR(1) dependence to model the image intensity, $\boldsymbol{y}_v = (y_{v,1}, y_{v,2}, \ldots, y_{v,T_v})$, at voxel v, $v = 1, \ldots, N$, is given by

$$\boldsymbol{y}_v = \mathbf{X}_v \boldsymbol{\beta}_v + \boldsymbol{\varepsilon}_v, \quad \varepsilon_v \sim \mathcal{N}(0, \sigma_v^2 \Lambda_v), \tag{5.1}$$

where T_v is the number of time points in voxel v, $\mathbf{X}_v$ is a design matrix, and $\boldsymbol{\beta}_v = (\beta_{v,1}, \ldots, \beta_{v,p})$ is a vector of p regression coefficients. We assume the error, $\boldsymbol{\varepsilon}_v$, is normally distributed with covariance matrix as $\sigma_v^2 \Lambda_v$ where the element of Λ_v in the position (i, j) is $\Lambda_v(i, j) = \rho_v^{|i-j|}$.

In the analysis of fMRI data, one often wishes to detect the activation of a certain voxel with respect to a stimulus presented. Smith [17] introduced binary random variables $\boldsymbol{\gamma}_v = (\gamma_{v,1}, \ldots, \gamma_{v,p})$ to indicate whether the voxel is activated by a sequence of input stimuli. In the spatial Bayesian variable selection model, the indicators are placed on the coefficients to indicate if the voxel is activated corresponding to the given tasks. That is, the coefficient $\beta_{v,i}$ is equal to zero if $\gamma_{v,i} = 0$ and $\beta_{v,i}$ is nonzero if $\gamma_{v,i} = 1$. The zero of $\gamma_{v,i}$ implies no effect on voxel v is caused by the corresponding experimental task i. Therefore, the model in (5.1) can be written as

$$\boldsymbol{y}_v = \mathbf{X}_v(\boldsymbol{\gamma_v}) \boldsymbol{\beta}_v(\boldsymbol{\gamma_v}) + \boldsymbol{\varepsilon}_v, \quad \varepsilon_v \sim \mathcal{N}(0, \sigma_v^2 \Lambda_v).$$

Next, the conventional types of priors used are introduced. A prior for $\boldsymbol{\gamma}_v$ is considered Ising distribution. We will consider another prior, Gaussian conditional autoregression (CAR) models, in section 5.8.

We assign the proper priors for each parameter in the model. The assignment of priors is hierarchical in the Bayesian spatial variable selection model. We introduce the priors next.

5.1.1 Ising Prior and Zellner's g-Prior

The prior of the indicator variables we consider here is a binary spatial Ising prior which is a popular binary Markov random filed prior in image analysis. Let $\boldsymbol{\gamma}_{(j)} = \{\gamma_{1,j}, \ldots, \gamma_{N,j}\}$

be the vector of indicator variables for regression j over locations $\{1, 2, \ldots, N\}$ and set $p(\boldsymbol{\gamma})=\Pi_{j=1}^{p} p(\boldsymbol{\gamma}_{(j)})$, where

$$p(\boldsymbol{\gamma}_{(j)}) \propto \exp\left\{\sum_{\upsilon=1}^{N} \alpha_{\upsilon,j}(\gamma_{\upsilon,j}) + \sum_{i\sim k} \theta_{i,k,j}\omega_{i,k} I(\gamma_{\upsilon,j} = \gamma_{k,j})\right\}. \tag{5.2}$$

Here, $\sim$ denotes a neighborhood relation, i.e., if υ is a neighbor of k then k is a neighbor of υ. It is in (5.2) that we can encode prior beliefs about the properties of underlying fMRI images. Specifically, $\sum_{\upsilon=1}^{N} \alpha_{\upsilon,j}(\gamma_{\upsilon,j})$ labeled as the "external field" can incorporate anatomical prior information. In most applications, $\sum_{\upsilon=1}^{N} \alpha_{\upsilon,j}(\gamma_{\upsilon,j})$ is a linear combination of parameters, i.e.,

$$\sum_{\upsilon=1}^{N} \alpha_{\upsilon,j}(\gamma_{\upsilon,j}) = \sum_{\upsilon=1}^{N} \alpha_{\upsilon,j}\gamma_{\upsilon,j}.$$

The spatial interaction effect of the elements of $\boldsymbol{\gamma}_{(j)}$ for all neighboring sites is given by

$$\sum_{\upsilon\sim k} \theta_{\upsilon,k,j}\omega_{\upsilon,k} I(\gamma_{\upsilon,j} = \gamma_{k,j}).$$

The neighborhood structure can be defined by the user. Moreover, $\omega_{\upsilon,k}$ are prespecified constants that allow us to weigh the interaction between neighboring locations on lattices υ and k and $\theta_{\upsilon,k,j}$ is the positive parameter to represent the strength of the interaction between υ and k. In the application of the Ising model, usually a single interaction parameter of $\boldsymbol{\gamma}_{(j)}$ is used. As a result, we can let $\theta_{\upsilon,j,k} = \theta_j$ and (5.2) can be written as

$$p(\boldsymbol{\gamma}_{(j)}) \propto \exp\left\{\sum_{\upsilon=1}^{N} \alpha_{\upsilon,j}\gamma_{\upsilon,j} + \theta_j \sum_{i\sim k} \omega_{i,k} I(\gamma_{\upsilon,j} = \gamma_{k,j})\right\}. \tag{5.3}$$

As for the prior of $\boldsymbol{\beta}_{\upsilon}(\boldsymbol{\gamma}_{\upsilon})$ for a particular voxel, υ, we consider the Zellner's g-prior [74] distribution given by

$$\boldsymbol{\beta}_{\upsilon}(\boldsymbol{\gamma}_{\upsilon})|\boldsymbol{y}_{\upsilon}, \sigma_{\upsilon}^2, \boldsymbol{\gamma}_{\upsilon} \sim \mathcal{N}(\hat{\boldsymbol{\beta}}_{\upsilon}(\boldsymbol{\gamma}_{\upsilon}), T_{\upsilon}\sigma_{\upsilon}^2[\mathbf{X}_{\upsilon}'(\boldsymbol{\gamma}_{\upsilon})\Lambda_{\upsilon}^{-1}\mathbf{X}_{\upsilon}(\boldsymbol{\gamma}_{\upsilon})]^{-1}), \tag{5.4}$$

where

$$\hat{\boldsymbol{\beta}}_{\upsilon}(\boldsymbol{\gamma}_{\upsilon}) = [\mathbf{X}_{\upsilon}'(\boldsymbol{\gamma}_{\upsilon})\Lambda_{\upsilon}^{-1}\mathbf{X}_{\upsilon}(\boldsymbol{\gamma}_{\upsilon})]^{-1}\mathbf{X}_{\upsilon}'(\boldsymbol{\gamma}_{\upsilon})\Lambda_{\upsilon}^{-1}\boldsymbol{y}_{\upsilon}. \tag{5.5}$$

when given $\boldsymbol{\gamma}_{\upsilon}$.

The g-prior is data-based because of the mean of $\boldsymbol{\beta}_\upsilon$ depends on $\boldsymbol{y}_\upsilon$. A valuable feature for using this prior is that a closed form of $p(\boldsymbol{y}_\upsilon|\boldsymbol{\gamma}_\upsilon)$ can be obtained which speed up posterior evaluation and MCMC exploration. This prior has been extensively discussed in the literature on Bayesian variable selection methodologies; see [75], [76], [77], and [78].

For σ_υ^2, we chose a standard invariant prior

$$p(\sigma_\upsilon^2) \propto \frac{1}{\sigma_\upsilon^2}. \tag{5.6}$$

We also place a uniform prior on $\boldsymbol{\theta}$, i.e., $p(\boldsymbol{\theta}) \propto \Pi_{j=1}^{p} I(0 < \theta_j < \theta_{\text{max}})$, where θ_{max} is a predetermined maximum. The prior for ρ_υ is assumed to be uniformly distributed between -1 and 1.

5.1.2 Posterior density

For convenience, we denote $\boldsymbol{\rho}$=$(\rho_1, \ldots, \rho_N)$, $\boldsymbol{\theta}$=$(\theta_1, \ldots \theta_p)$ and $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_N^2)$, respectively. To choose prior densities, we will assume $\boldsymbol{\theta}$, $\boldsymbol{\rho}$, and $\boldsymbol{\sigma}^2$ a *a priori* independent , $\boldsymbol{\gamma}$ conditionally independent, and independence across voxels.

After the specification of all priors, the joint posterior density of $\boldsymbol{\beta}(\boldsymbol{\gamma})$, $\boldsymbol{\sigma}^2$, $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\theta}$, given the sample, $\boldsymbol{y}$, can be obtained as follows

$$\begin{aligned}
&\pi(\boldsymbol{\beta}(\boldsymbol{\gamma}), \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\theta}, \boldsymbol{\sigma^2}|\boldsymbol{y}) \\
&\propto p(\boldsymbol{y}|\boldsymbol{\beta}(\boldsymbol{\gamma}), \boldsymbol{\gamma}) \pi(\boldsymbol{\beta}(\boldsymbol{\gamma})|\boldsymbol{\gamma}) \pi(\boldsymbol{\gamma}|\boldsymbol{\theta}) \pi(\boldsymbol{\theta}) \pi(\boldsymbol{\rho}) \pi(\boldsymbol{\sigma^2}) \\
&\propto \prod_{\upsilon=1}^{N} \frac{1}{|\sigma_\upsilon^2 \Lambda_\upsilon|^{1/2}} \exp\left\{ -\frac{1}{2\sigma_\upsilon^2} \left[\boldsymbol{y}_\upsilon - \mathbf{X}_\upsilon(\boldsymbol{\gamma_\upsilon})\boldsymbol{\beta}_\upsilon(\boldsymbol{\gamma_\upsilon})\right]' \Lambda_\upsilon^{-1} \left[\boldsymbol{y}_\upsilon - \mathbf{X}_\upsilon(\boldsymbol{\gamma_\upsilon})\boldsymbol{\beta}_\upsilon(\boldsymbol{\gamma_\upsilon})\right] \right\} \\
&\times \prod_{\upsilon=1}^{N} \frac{1}{|T_\upsilon \sigma_\upsilon^2 (\mathbf{X}_\upsilon' \Lambda_\upsilon^{-1} \mathbf{X}_\upsilon)^{-1}|^{1/2}} \exp\left\{ -\frac{1}{2T_\upsilon \sigma_\upsilon^2} \left[\boldsymbol{\beta}_\upsilon(\boldsymbol{\gamma}_\upsilon) - \hat{\boldsymbol{\beta}}_\upsilon(\boldsymbol{\gamma}_\upsilon)\right]' \mathbf{X}_\upsilon{}' \Lambda_\upsilon^{-1} \mathbf{X}_\upsilon \left[\boldsymbol{\beta}_\upsilon(\boldsymbol{\gamma}_\upsilon) - \hat{\boldsymbol{\beta}}_\upsilon(\boldsymbol{\gamma}_\upsilon))\right] \right\} \\
&\times \prod_{j=1}^{p} \exp\left\{ \sum_{\upsilon=1}^{N} \alpha_{\upsilon,j} \gamma_{\upsilon,j} + \theta_j \sum_{i \sim k} \omega_{i,k} I(\gamma_{\upsilon,j} = \gamma_{k,j}) \right\} \\
&\times \prod_{\upsilon=1}^{N} \frac{1}{\sigma_\upsilon^2}.
\end{aligned}$$

Consider a specific voxel, υ, after integrating out $\boldsymbol{\beta}_\upsilon(\boldsymbol{\gamma}_\upsilon)$ and σ_υ^2, we have the joint posterior distribution is

$$\pi(\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\theta}|\boldsymbol{y}) \propto \prod_{\upsilon=1}^{N} p(\gamma_\upsilon, \theta_\upsilon, \rho_\upsilon|y_\upsilon) \prod_{j=1}^{p} \exp\left\{\sum_{\upsilon=1}^{N} \alpha_{\upsilon,j}\gamma_{\upsilon,j} + \theta_j \sum_{i\sim k} \omega_{i,k} I(\gamma_{\upsilon,j} = \gamma_{k,j})\right\}, \quad (5.7)$$

where

$$p(\boldsymbol{\gamma}_\upsilon, \rho_\upsilon|\boldsymbol{y}_\upsilon) \propto |\Lambda_\upsilon|^{-1/2}(1+T_\upsilon)^{-q_\upsilon/2} S(\boldsymbol{\gamma}_\upsilon, \rho_\upsilon)^{-T_\upsilon/2}, \quad (5.8)$$

$$q_\upsilon = \sum_{j=1}^{p} \gamma_{\upsilon,j}, \quad (5.9)$$

and

$$S(\boldsymbol{\gamma}_\upsilon, \rho_\upsilon) = \left[\boldsymbol{y}_\upsilon - \mathbf{X}(\boldsymbol{\gamma}_\upsilon)_\upsilon \hat{\boldsymbol{\beta}}_\upsilon(\boldsymbol{\gamma}_\upsilon)\right]' \Lambda_\upsilon^{-1} \left[\boldsymbol{y}_\upsilon - \mathbf{X}(\boldsymbol{\gamma}_\upsilon)_\upsilon \hat{\boldsymbol{\beta}}_\upsilon(\boldsymbol{\gamma}_\upsilon)\right]. \quad (5.10)$$

This is a very useful closed form for evaluating the posterior means of indicator variables $\boldsymbol{\gamma}_\upsilon$, temporal dependence ρ_υ and corresponding amplitude $\boldsymbol{\beta}_\upsilon$.

To calculate the posterior quantities of interest to make statistical inference, we need to use MCMC algorithms to draw the sample from (5.7). We introduce the sampling procedure in next section.

5.1.3 Bayesian Inference via MCMC Sampling

In order to make inference about model parameters, we need to integrate over high-dimensional probability distributions. Markov chain Monte Carlo (MCMC) methods are very helpful for solving our problems. MCMC is Monte Carlo integration using Markov chains. It draws samples from the required distribution by running a cleverly constructed Markov chain for a long time and then forms sample averages to approximate expectations. The Gibbs sampler and Metropolis-Hastings (MH) algorithms are the basic ways of constructing those chains. A great advantage of the Gibbs sampler and the MH algorithms is the ease of implementation. Excellent references on the methodology have been provided in the chapter 3. The MCMC algorithm for drawing the sample from $\pi(\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\theta}|\boldsymbol{y})$ defined at (5.7) proceeds as follows.

Step 1 Generate $\boldsymbol{\gamma}$ given $\boldsymbol{\theta}$, and $\boldsymbol{\rho}$, using single-site sampling for the binary variable from the decomposition

$$P(\gamma_{\upsilon,j}|\boldsymbol{\gamma}_{-\upsilon,j},\boldsymbol{\theta},\rho_\upsilon,\boldsymbol{\gamma}_\upsilon) \propto P(\boldsymbol{y}_\upsilon|\boldsymbol{\gamma}_\upsilon,\rho_\upsilon)P(\gamma_{\upsilon,j}|\boldsymbol{\theta},\boldsymbol{\gamma}_{-\upsilon,j}), \tag{5.11}$$

where $\boldsymbol{\gamma}_{-\upsilon,j}$ is a vector of binary indicators excluding $\gamma_{\upsilon,j}$ and

$$p(\boldsymbol{y}_\upsilon|\boldsymbol{\gamma}_\upsilon,\rho_\upsilon) \propto |\Lambda_\upsilon|^{-1/2}[S(\boldsymbol{\gamma}_\upsilon,\rho_\upsilon)]^{-T_\upsilon/2}(1+T_\upsilon)^{-q_\upsilon/2}.$$

For notational simplicity, we denote $L(\gamma_{\upsilon,j}) = p(\boldsymbol{y}_\upsilon|\boldsymbol{\gamma}_\upsilon,\rho_\upsilon)$ and $\pi(\gamma_{\upsilon,j}) = P(\gamma_{\upsilon,j}|\boldsymbol{\theta},\boldsymbol{\gamma}_{-\upsilon,j})$. Then, sequentially generating a candidate $\gamma^*_{\upsilon,j}$ from

$$\pi(\gamma_{\upsilon,j}=1) = \frac{1}{1+g_{\upsilon,j}},$$

where

$$g_{\upsilon,j} = \exp\left\{-\alpha_{\upsilon,j} + \theta_j\left(\sum_{k\in\delta_\upsilon}\omega_{k,\upsilon}(1-2\gamma_{k,j})\right)\right\}$$

for $\upsilon = \{1,2,\ldots,N\}$ and $j = \{1,2,\ldots,p\}$ with acceptance probability $\phi(\upsilon,j)$ given by

$$\phi(\upsilon,j) = \min\left\{\frac{L(\gamma^*_{\upsilon,j})}{L(\gamma_{\upsilon,j})},1\right\}.$$

The advantage to using this approach to generate the sample $\boldsymbol{\gamma}$ is that the computation required will be reduced because only if a switch in $\gamma_{\upsilon,j}$ is proposed does one need to calculate the acceptance probability. That is, suppose a sample is generated say $\gamma^*_{\upsilon,j} = \gamma_{\upsilon,j}$, then we immediately accept it and set $\gamma^{\text{new}}_{\upsilon,j} = \gamma^*_{\upsilon,j}$, otherwise,

$$\gamma^{\text{new}}_{\upsilon,j} = \begin{cases} \gamma^*_{\upsilon,j}, & \text{if } \phi(\upsilon,j) < u; \\ 1-\gamma^*_{\upsilon,j}, & \text{if } \phi(\upsilon,j) \geq u, \end{cases}$$

where u a sample from $U(0,1)$.

Step 2 Generate ρ_υ for $\upsilon = 1,\ldots,N$ via MH algorithm with proposal distribution $U[0,1]$, given $\boldsymbol{\gamma}$ and $\boldsymbol{y}$, where

$$p(\rho_\upsilon|\boldsymbol{\gamma},\boldsymbol{y}) \propto |\Lambda_\upsilon|^{-1/2}[S(\boldsymbol{\gamma}_\upsilon,\rho_\upsilon)]^{T_\upsilon/2}.$$

Step 3 Generate θ_j for $j = 1, 2 \ldots p$ via the MH algorithm from

$$p(\theta_j|\boldsymbol{\gamma}, \boldsymbol{y}, \theta_{-j}) \propto p(\boldsymbol{\gamma}_{(j)}|\theta_j) I(0 < \theta_j < \theta_{\max}) \\ \propto C_j^{-1}(\theta_j, \boldsymbol{\alpha}_j) \exp\left\{\theta_j \sum_{\upsilon \sim k} \omega_{\upsilon,k} I(\gamma_{\upsilon,j} = \gamma_{k,j})\right\} \times I(0 < \theta_j < \theta_{\max}) \tag{5.12}$$

where

$$C_j(\theta_j, \boldsymbol{\alpha}_j) = \left[\sum_{\boldsymbol{\gamma}_{(j)}} \exp\left\{\sum_{\upsilon=1}^{N} \alpha_{\upsilon,j}\gamma_{\upsilon,j} + \theta_j \sum_{\upsilon \sim k} \omega_{\upsilon,k} I(\gamma_{\upsilon,j} = \gamma_{k,j})\right\}\right]$$

is the normalizing constant of the density $p(\boldsymbol{\gamma}_{(j)}|\theta_j)$ which can be estimated by the path sampling or the Wang-Laudau algorithms ; see chapter 4. The proposal distribution is $\mathcal{N}(\tilde{\theta}, \tilde{\sigma}^2)$ where $\tilde{\theta}$ is the sample from previous step and $\tilde{\sigma}$ will be tuned so that the acceptance rate is about 30-40%.

5.2 Parameter Estimation

The inference in parameters of interest in Bayesian variable selection models is based on Bayesian model averaging (BMA). This approach estimates the quantities of interest through a weighted average of all possible models in the model space M. The weights depend on the how much the data support each model as measured by the posterior probabilities on models. In our setting, the model is different at voxel υ with different value of $\boldsymbol{\gamma}_\upsilon$. In (5.13), the posterior mean of $\boldsymbol{\beta}_\upsilon$ can be expressed as weight of model-specific quantities $E(\boldsymbol{\beta}_\upsilon|\boldsymbol{\gamma}_\upsilon, \boldsymbol{y})$. The estimates of the $\boldsymbol{\beta}_\upsilon$ and $p(\gamma_{\upsilon,j} = 1|\boldsymbol{y})$ are given as follows. Suppose there is a sample $\{(\boldsymbol{\gamma}^{[1]},\boldsymbol{\rho}^{[1]}), (\boldsymbol{\gamma}^{[2]},\boldsymbol{\rho}^{[2]}), \ldots, (\boldsymbol{\gamma}^{[K]},\boldsymbol{\rho}^{[K]})\}$ generated from the posterior distribution using MCMC, respectively. The posterior quantity of most interest $p(\gamma_\upsilon = 1|\mathbf{y})$ can be estimated via Rao-Blackwellisation by

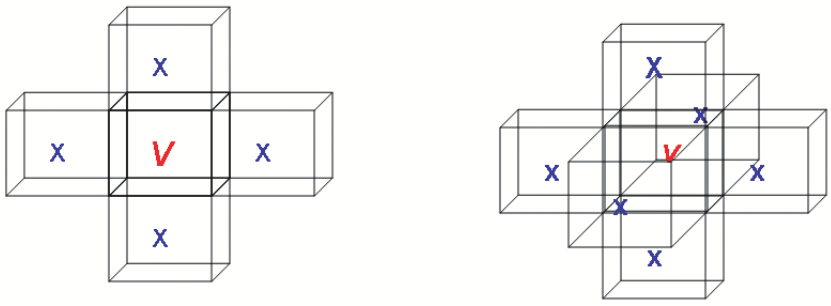
$$E(\boldsymbol{\beta}_\upsilon|\boldsymbol{y}) = \sum_{\boldsymbol{\gamma}_\upsilon} E(\boldsymbol{\beta}_\upsilon|\boldsymbol{\gamma}_\upsilon, \boldsymbol{y}) p(\boldsymbol{\gamma}_\upsilon|\mathbf{y}) \approx \frac{1}{K} \sum_{k=1}^{K} \hat{\boldsymbol{\beta}}_\upsilon(\boldsymbol{\gamma}_\upsilon^{[k]}). \tag{5.13}$$

and

$$P(\gamma_{v,j}=1|\boldsymbol{y}) = \int P(\gamma_{v,j}=1|\rho_v,\boldsymbol{\gamma}_{-v,j},\boldsymbol{y})P(\rho_v|\boldsymbol{y})P(\boldsymbol{\gamma}_{-v,j}|\boldsymbol{y})d\rho_v d\boldsymbol{\gamma}_{-v,j}$$
$$\approx \frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{\gamma}_{v,j}=1|,\boldsymbol{\gamma}_{-v,j}^{[k]}=1,\rho_v^{[k]},\boldsymbol{y}). \tag{5.14}$$

5.3 Neighborhood Structures

The first-order neighborhood structure shown in Figure (5.1) is used in this analysis. A two dimensional neighborhood structure is given in Figure 5.1(a) and a the three dimensional neighborhood structure in 5.1(b). Essentially, only nearest adjacent neighbors are considered.



(a) 2-dimension neighborhood structure (b) 3-dimension neighborhood structure

Figure 5.1: Voxels labeled X are neighbors of the voxel V

5.4 Threshold

In the analysis of fMRI BOLD imaging data, the most important step is to detect if voxels are activated by stimulus and to create the maps of activation. In order to construct an activation map, we have to specify a threshold and when a voxel with a significance

value above a given threshold is considered activated by task. The selection of a threshold value in the classical frequentist method is introduced by [15] and [16]. In Bayesian analysis, we must still choose a threshold. To produce activation maps, it is crucial to decide a threshold after we obtained the estimate of posterior quantity of $p(\gamma_{v,j} = 1|\boldsymbol{y})$. Whether a voxel is activated by performing task j or not in terms of whether the posterior probability $p(\gamma_{v,j} = 1|\boldsymbol{y})$ given in (5.14) is greater a prespecified threshold. Unlike the classical methods that use the Gaussian random field to find a threshold, [17] set the threshold to be 0.8722. An individual voxel, if $p(\gamma_v = 1|\boldsymbol{y}) > 0.8722$, is categorized as active, otherwise as inactive. The motivation for choosing this value is that $-2\log\left[(1 - p(\gamma_{v,j} = 1|\boldsymbol{y}))/p(\gamma_{v,j} = 1|\boldsymbol{y})\right]$ is on the same scale as a likelihood ratio statistic and is distributed approximately χ^2_1. Setting the p-value equal 0.05, the critical value of χ^2_1 is 3.841, which implies the posterior probability $p(\gamma_{v,j} = 1|\boldsymbol{y}) = 0.8722$ at this critical value. In the following analysis, 0.8722 is used as a threshold to acquire activation maps.

5.5 Two-Stage Estimation Procedure

In MCMC sampling procedure, it is necessary to evaluate Λ_v^{-1} and $S(\boldsymbol{\gamma}_v, \rho_v)$ in order to simulate the ρ_v for each voxel. However, calculating both of these require a substantial computational effort. To avoid evaluating them in each iteration, we present a two-stage estimation approach to reduce the computational intensity. In stage I, each ρ_v is estimated using frequentist approaches such maximum likelihood (ML) or restricted maximum likelihood (REML) estimations. Given the estimate, we assume the covariance matrix is known so that Λ_v and $S(\boldsymbol{\gamma}_v, \rho_v)$ are evaluated once in simulation. That is, the data is pre-whitened in this stage. In stage II, as usual, we generate the $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ iteratively. The stage-II deals with spatial coefficient between voxels.

5.6 Simulation Study

To validate if the model established previously can ideally identify the activation and to ascertain the performance and the ability of the sampling scheme to estimate model parameters as well, a Monte Carlo simulation is conducted. In this simulation study,

we assume the value of the spatial coefficient equal to 0.7, that is, $\theta = 0.7$. Given θ, a 30×30 activated-inactivated square image $\boldsymbol{\gamma}$ is generated using perfect sampling [53], where 1 represents activation and 0 inactivation. The perfect sampling algorithm can exactly generate a sample from the target distribution π. The essential idea of it is to find a random past time T such that when we construct sample paths from every point in the state space starting at T, then all paths will have coalesced successfully by time zero. The common value of the path at time zero is a exact sample drawn from target distribution. In terms of this activated-inactivated image, we simulate a time-series data y_υ from general linear models in each voxel υ under the following settings. Let

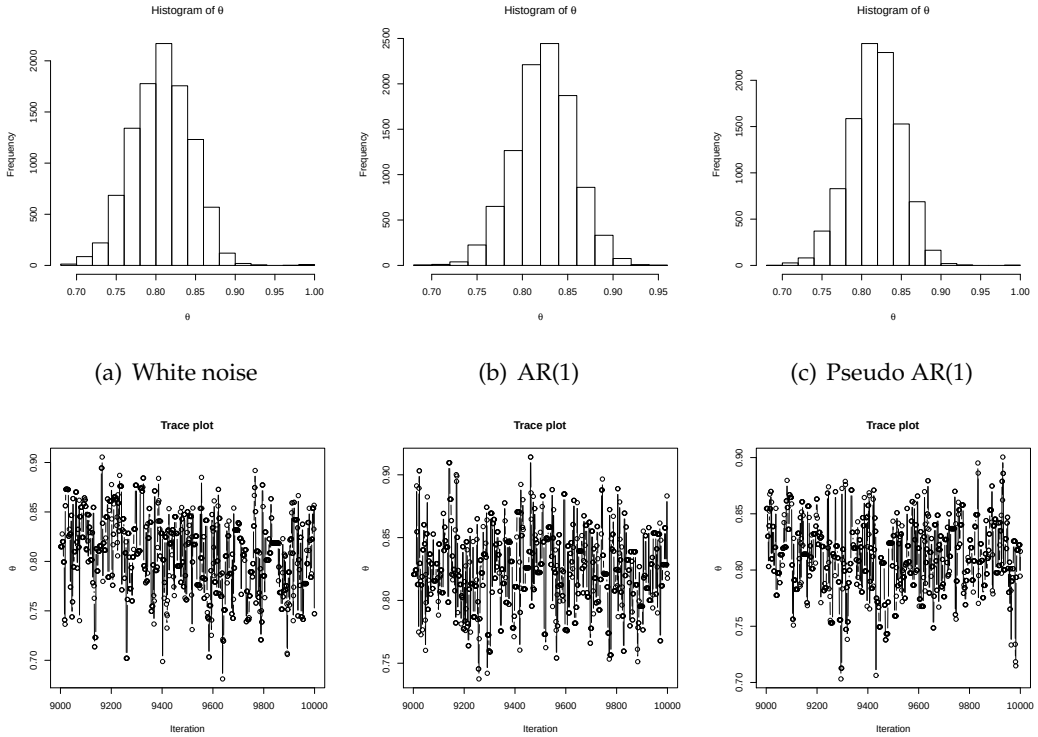
$$\boldsymbol{\beta}_\upsilon = \begin{pmatrix} \beta_{\upsilon,0} \\ \beta_{\upsilon,1} \end{pmatrix},$$

where $\beta_{\upsilon,0}$ represents the baseline level and $\beta_{\upsilon,1}$ describes the amplitude of activation in response to a stimulus at each voxel υ. We use $\gamma_{\upsilon,j}$ to indicate if $\beta_{\upsilon,j}$ is equal to 0 or not. In this simulation, we always assume $\gamma_{\upsilon,0} = 1$ since $\beta_{\upsilon,0}$ models the baseline level in a human brain. On the other hand, $\gamma_{\upsilon,1}$ can be either 0 and 1. When $\gamma_{\upsilon,1} = 1$, we set $\boldsymbol{\beta}'_\upsilon = (300, 5)'$; otherwise $\boldsymbol{\beta}_\upsilon = \beta_{\upsilon,0} = 300$. Here, we take $\beta_{\upsilon,1} = 5$ because the BOLD contrast is fairly small, with activation inducing a signal increase ranging from 1 to 5%. Furthermore, the autoregression coefficient, ρ_υ, is generated from $U(-1, 1)$ and $\sigma_\upsilon = 3$ for each voxel.

Given the previous settings, we simulate 10 data. For each simulated data, the spatial Bayesian variable selection approach is applied to detect the activation and to estimate the spatial coefficient θ and autoregression coefficients ρ_υ based on 10,000 MCMC samples. To evaluate the performance of the spatial Bayesian variable method, we consider the accuracy and false positive rate. The accuracy is defined as the percentages of voxels truly classified. We say a voxel γ_υ is truely classified if the predicted value $\hat{\gamma}_\upsilon$ of γ_υ coincides with γ_υ. We define the false positive rate as the percentage of active voxels falsely identified, that is, concluding voxels are activated when they actually aren't.

Before making inference based on MCMC samples, the output should be analyzed to determine when to stop sampling or if the size of simulated sample is large enough. All stopping rules are only applicable to dealing with the one dimensional problem at present [38], however, in fMRI data analysis, we require diagnostics for high dimension problems.

There are many difficulties and pitfalls to extend one-dimensional MCMC output assessment to high-dimensional settings. For a particular stimulated data, we provide a visual inspection of convergence and MCSE only for estimate of θ. To ensure all MCSEs are less than 0.005, 10,000 points are generated from the posterior distributions to the estimate θ. The histograms of θ generated based on different models are given in Figures 5.2(a), (b) and (c) and the trace plots of last 1,000 simulated points are shown in Figures 5.2(d), (e) and (f). They are quite stable. The average of estimates of θ and corresponding Monte Carlo standard errors (MCSEs) over 10 simulated data are given in Table 5.1. Without considering the temporal correlation, it could cause overestimate the spatial correlation, θ.



(a) White noise (b) AR(1) (c) Pseudo AR(1)

(d) White noise (e) AR(1) (f) Pseudo AR(1)

Figure 5.2: Histograms and trace plots for θ in different models for a simulated data.

Table 5.2 reports the average accuracy and false positive rate over 10 simulated data when two different models are applied to identify the activation. AR1 means we assume the AR(1) dependence for error terms and use regular sampling scheme given in section 5.1.3. That is, ρ_v is generated in each step. Pseudo AR1 means that we also assume AR(1) dependence in error terms but we estimate the ρ first, and then use two-stage approach given in section 5.5 to estimate the posterior quantities. We don't necessarily to generate the ρ in each step. It will save a lot computational effort. Frequentist means a GLM is applied to each voxel and a voxel is classified as activated when the corresponding p-value is less than 0.0001. As revealed in Table 5.2, ignoring the temporal correlation in analysis of fMRI time-series data could cause higher false positive rate, that is, inactivated voxel are falsely classified as activated ones. As result, researchers could make the wrong

Table 5.1: The average Monte Carlo estimates of θ and corresponding Monte Carlo standard errors (MCSE) based on 10,000 iteration over 10 simulated data.

Models	Estimate $\hat{\theta}$	MCSE
AR(1)	0.73	0.0017
Pseudo AR(1)	0.74	0.0015
White Noise	0.78	0.0015

Table 5.2: The accuracy are estimated based on 10 simulated data

Models	White Noise	AR1	Pseudo AR1	Frequentist
Accuracy (%)	91.38	97.38	97.16	89.86
False Positive Rate (%)	13.59	0.45	0.04	0
Time Elapse (second)	77	24001	77	68

conclusion in the following analysis. In addition, in two-state approach, the computation time is reduced dramatically and the performance of classifying the voxels is as good as the approach where each ρ is generated in each iteration. Although the frequentist approach is the fastest to get the activation map, the performance in classification of voxels as activation and inactivation is not as good as the other methods. This might be due to the spatial correlation between voxels not considered in the model.

5.7 Stroop Data

The Stroop task [5] is a psychological test of mental flexibility. It has been used for many years as test that exploits the conflicts between one well-learned or automatic behavior (reading) and a decision rule that requires this behavior to be inhibited. It is interference in the completion of a task caused by one area of the brain dominating and inhibiting the response of other functional areas. Many previous behavioral studies have established

the features of the Stroop task that produce cognitive interference. Moreover, recent neuroimaging studies [6], [7], [8], [9], [10], [11] have indicated that several brain regions are involved in the performance of the Stroop task, although these imaging studies do not all agree on which brain areas are most centrally involved in resolving Stroop interference.

We have data collected on over 200 patients participating in a study of Alzheimer's disease (AD) progression at Johns Hopkins University. All subjects are older, generally well-educated, and healthy. None of them have any clinically diagnosed neurologic disorders or Huntington's disease that would significantly impact our modeling decisions. However, some of the subjects are at high risk for AD. Each subject performs the Stoop task subsequently. In this test, subjects are shown words and asked to press a button corresponding to the color ink when a word is shown in the scanner. There are three different types of tasks:

- Ink only - the word is XXXX (i.e. not really a word at all)
- Congruence - the word is the color of the ink. For example, the word might be **blue** written in **blue** letters.
- Interference - the word is a different color from the color of the ink. For example, the word might be **blue** written in **red** ink.

A question of interest throughout will be what areas of the brain are activated during the performance of the Stroop test. There is an important cognitive mechanism involved in this task, specifically, directed attention. Since most people are very proficient at reading words, it takes effort to ignore them and concentrate on the color. This test is a standard measure in neurophysiological assessment for measuring cognitive processing. Accordingly, the Stroop interference design is supposed to activate the parietal lobe in the brain. The parietal lobe is associated with cognition, information processing, spatial orientation, speech, and visual perception.

Furthermore, another scientific question to be addressed is "Does the activation pattern change over time?" This is interesting since a subject may get better at the Stroop test with practice or their attention may wander and the patient's performance will degrade.

For each subject of an fMRI experiment, the data collected from a single run is a time series of three-dimensional images. In each run, images are taken approximately every 1

second for up to 2 hours, and there are often more than 200 3-dimension (3D) images. In each session, there may be several runs and there may be many subjects. High-resolution 3D images, with each voxel's size $2mm^3$, were collected and each image consists of data on $128 \times 128 \times 24$ voxels which translates to slightly more than 3MB of storage of double-precision real numbers. The entire images have been trimmed so that the dimension is $79 \times 95 \times 68$ in each time point. Thus analyzing this data requires special computational techniques. In this paper, only one subject who has a high risk for AD is analyzed to look for the activation of the brain and the change of activation pattern over time when performing the Stroop test.

Predicting Activation

In order to detect the activity of neurons corresponding to the three different types of tasks performed in the Stroop test, a model we consider is

$$\boldsymbol{y}_\upsilon = \alpha_0 \boldsymbol{z}_0 + \alpha_1 \boldsymbol{z}_1 + \beta_1 \boldsymbol{x}_1 + \beta_2 \boldsymbol{x}_2 + \beta_3 \boldsymbol{x}_3 + \boldsymbol{\varepsilon}_\upsilon \tag{5.15}$$

where $\boldsymbol{y}_\upsilon$ represent a time-series in a particular voxel υ, $\boldsymbol{x}_i$ is a transformed input function, $\boldsymbol{z}_i$ is a vector used to remove low-frequency, stimulus independent, effects, β_i are parameters of interest corresponding to different tasks, "Ink Only", "Congruence", and "Interference", respectively, α_i is nuisance parameter to model the baseline of a brain signal. To account for the temporal correlation arising from the fact that stimuli are presented continuously or periodically over time, the error term in (5.15) is assumed to be a stochastic process such as an autoregressive model of small order. Here we consider AR(1), that is,

$$\varepsilon_\upsilon \sim N(0, \sigma_\upsilon^2 \Lambda_\upsilon),$$

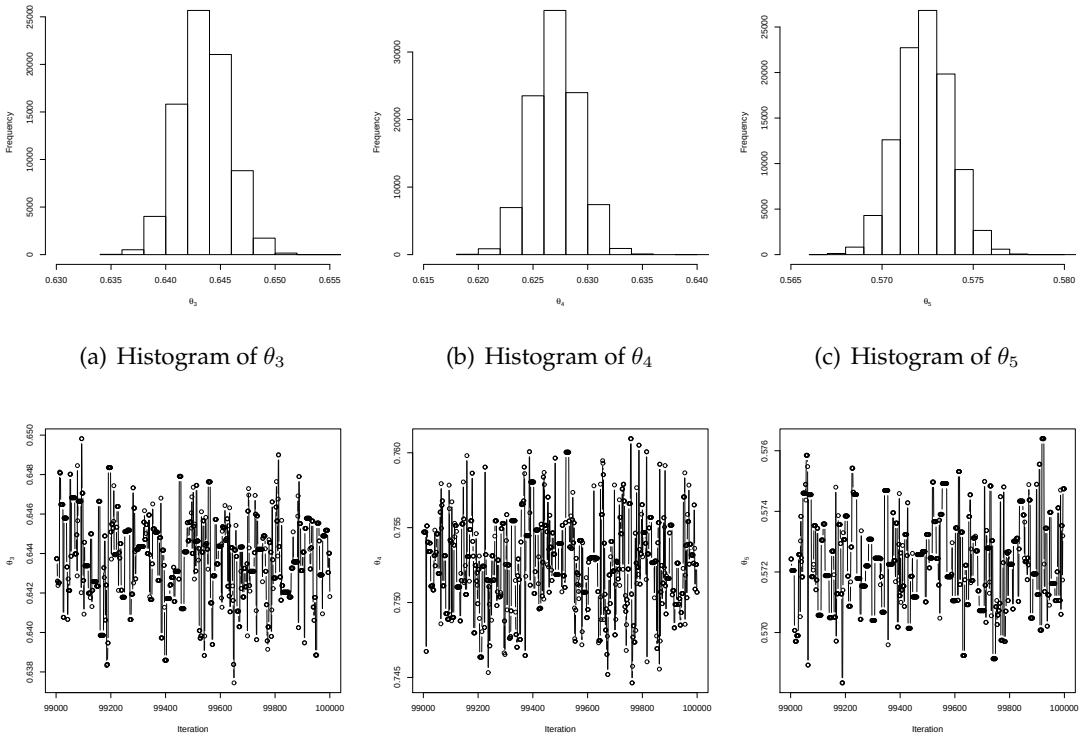
and element in the position (i, j) of Λ_υ is $\Lambda_\upsilon(i, j) = \rho_\upsilon^{|i-j|}$. This could model the temporal correlation within voxel. However, it needs more computational effort. In the simple independence of error terms, the quantity (5.10) only needs to be evaluated once in the stimulation; however, in the AR(1), it should be calculated in each iteration. Therefore, we use the two-stage approach to evaluate the posterior probability of $\gamma_{j,\upsilon}$. A sequence of binary variable $\boldsymbol{\gamma}_\upsilon = \{\gamma_{1,\upsilon}, \gamma_{2,\upsilon}, \gamma_{3,\upsilon}, \gamma_{4,\upsilon}, \gamma_{5,\upsilon}\}$ is used to indicate if the corresponding

Table 5.3: The estimate of θ and the corresponding MCSE given in the parenthesis

Models	θ_2	θ_3	θ_4
White noise	0.6435 (0.00005)	0.6271 (0.00006)	0.5724 (0.00011)
Pseudo AR(1)	0.7476 (0.00006)	0.7520 (0.00006)	0.6181 (0.00007)

parameter is zero or not. In our case, we always assume α_i nonzero so that the sequence becomes $\boldsymbol{\gamma}_v = \{1, 1, \gamma_{3,v}, \gamma_{4,v}, \gamma_{5,v}\}$.

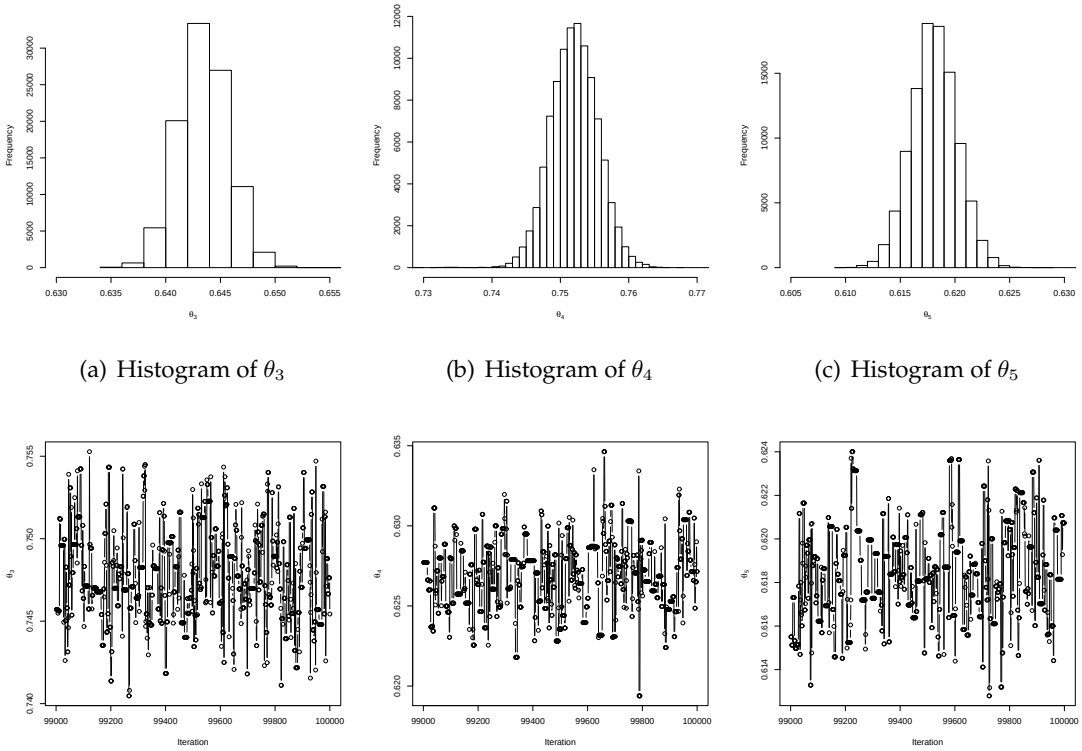
To see the effect of temporal correlation on detecting the activation of a human brain, two models are considered, one of which is assumed to have independence in error terms, the other one is assumed to have AR(1) dependence. A total 100,000 iteration taking about 15 was generated and used to make inference over the parameter $\boldsymbol{\gamma}_v$. Again, MCMC convergence was assessed by both visual inspection on the simulated chains given in Figures 5.3 and 5.4 and calculation of MCSE [38] given in Table 5.3. In the case of independence in error terms, the predicting activations in the brain of a subject when performing corresponding three tasks, Ink Only, Congruence, and Interference, are given in Figures (5.5), (5.6) and (5.7). Compared to Figures (5.8), (5.9) and (5.10) obtained from the frequentist approach using $p < 0.0001$, the activated regions are in general similar. Both techniques could be readily applied to clinical populations for diagnostic or research purpose. The activation voxels slightly group together obtained using Bayes approach where the spatial correlation between voxels is taken into account. On the contrary, the activation maps are more scattered obtained using the frequentist approach. The main reason to explain this is the spatial correlation is considered in the Bayesian model, that is, the closer voxels tend to have similar patters.



(a) Histogram of θ_3 (b) Histogram of θ_4 (c) Histogram of θ_5

(d) Trace plot of θ_3 (e) Trace plot of θ_4 (f) Trace plot of θ_5

Figure 5.3: Histograms and trace plots for each θ when assuming independence in error terms.



(a) Histogram of θ_3 (b) Histogram of θ_4 (c) Histogram of θ_5

(d) Trace plot of θ_3 (e) Trace plot of θ_4 (f) Trace plot of θ_5

Figure 5.4: Histograms and trace plots for each θ when assuming AR(1) dependence in error terms.

The parietal and occipital lobes are activated during the Stroop task, since the parietal lobe accounts for cognition, information processing, and visual perception and the occipital lobe is the visual processing center which is responsible for control of vision and color recognition. Additionally, the activation regions in the right brain are larger than those in the left brain partly because the left and right hemispheres of the brain process information in different ways and we tend to process information using our dominant side. The activation maps suggest that the two different sides of the brain control two different ąğmodesąĺ of thinking.

On the other hand, Figures 5.14, 5.15, and 5.16 show the activation regions of the human brain with assumption of AR(1) dependence in error terms. Compared with the

activation maps obtained from the previous one where we assume error terms are independent, the active areas are smaller here. Based on our simulation study, we found if there is strong temporal correlation in the data, but when a model fails to consider that, the high false positive rate will be high, resulting in more false activation voxels. In addition, it is worth noting that the right parietal lobe is activated in the Stroop task.

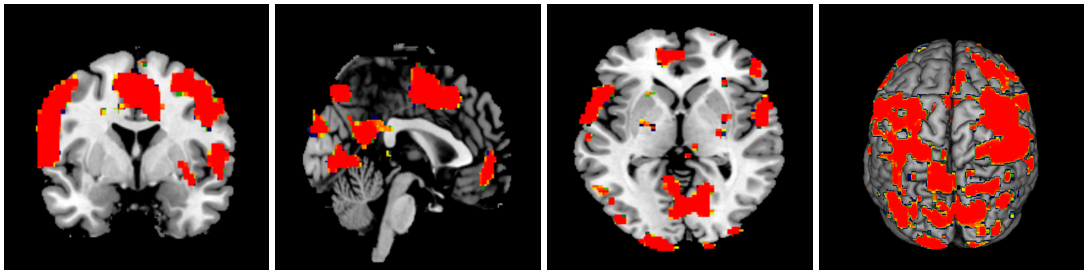


Figure 5.5: Predicting activation when performing "InkOnly" task obtained by using Bayesian approach with independent error terms.

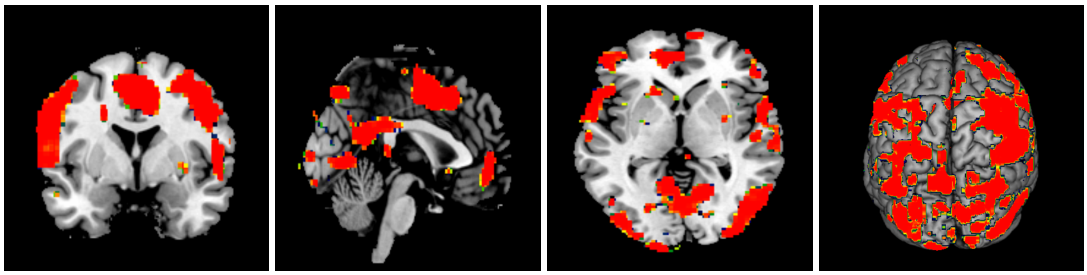


Figure 5.6: Predicting activation when performing "Congruence" task obtained by using Bayesian approach with independent error terms.

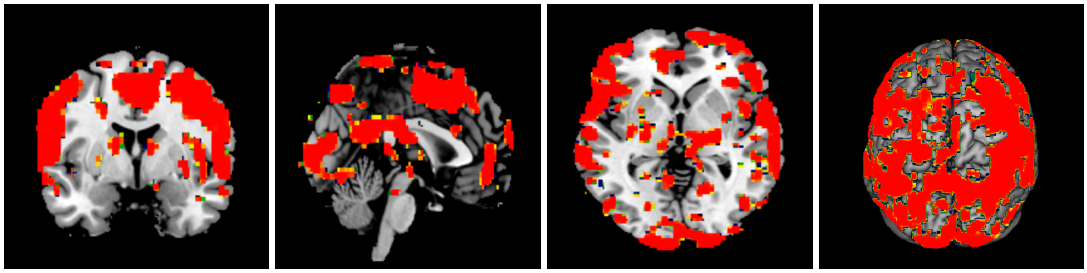


Figure 5.7: Predicting activation when performing "Interference" task obtained by using Bayesian approach with independent error terms.

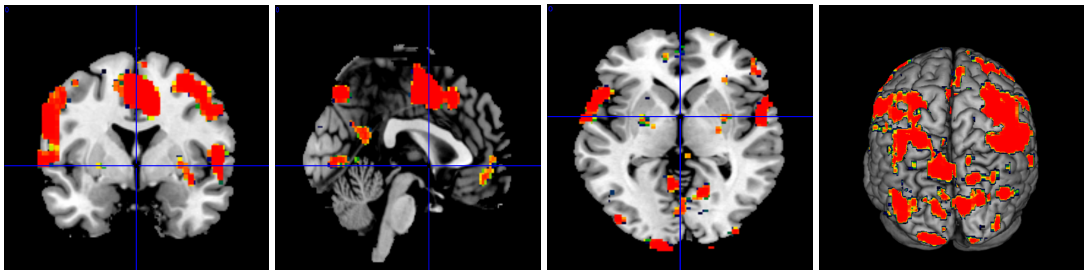


Figure 5.8: Predicting activation when performing "Ink Only" task obtained by using Frequentist approach with independent error terms.

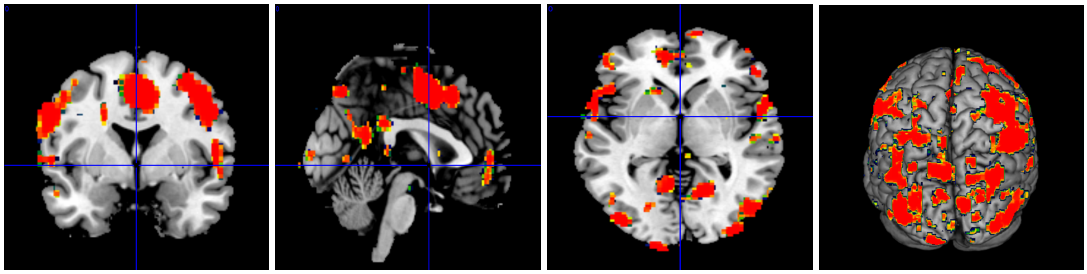


Figure 5.9: Predicting activation when performing "Congruence" task obtained by using Frequentist approach with independent error terms.

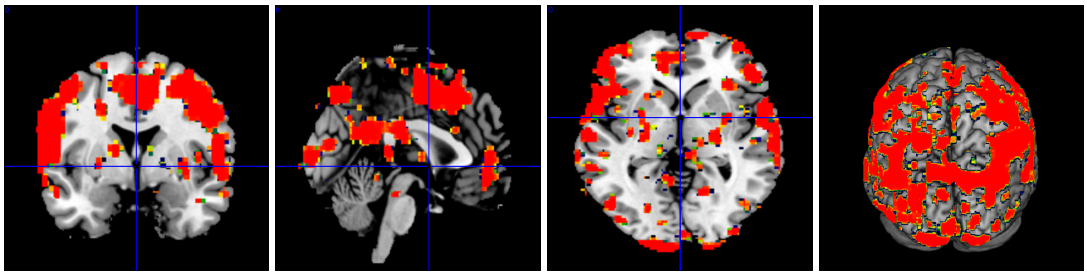


Figure 5.10: Predicting activation when performing "Interference" task obtained by using Frequentist approach with independent error terms.

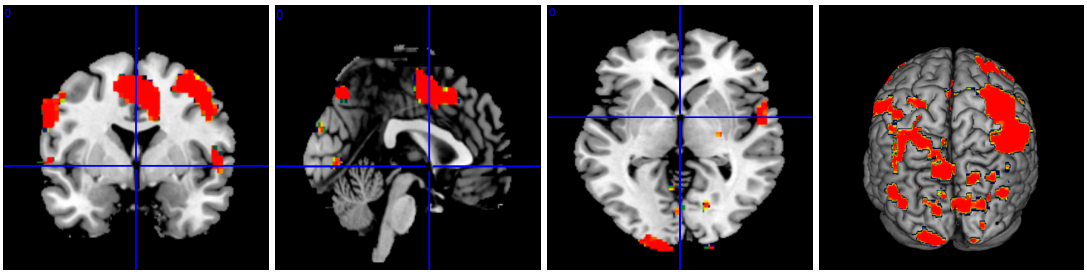


Figure 5.11: Predicting activation when performing "Ink Only" task obtained by using Bayesian approach with AR(1) dependence for error terms. Two-stage estimation approach is used to estimate the posterior probabilities.

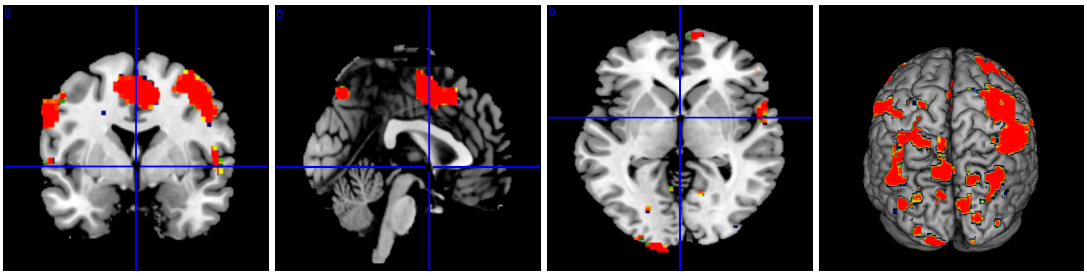


Figure 5.12: Predicting activation when performing "Congruence" task obtained by using Bayesian approach with AR(1) dependence for error terms. Two-stage estimation approach is used to estimate the posterior probabilities.

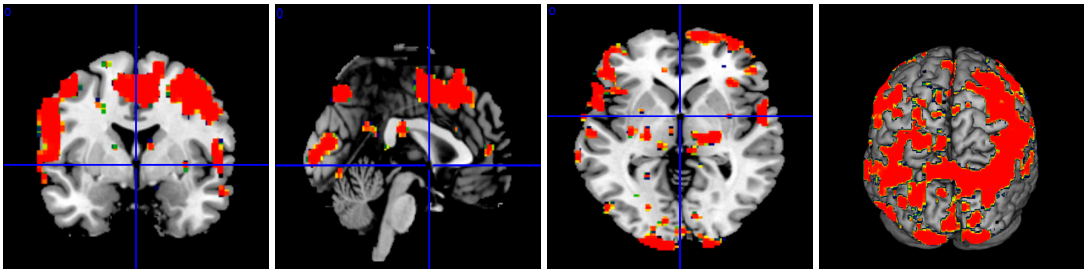


Figure 5.13: Predicting activation when performing "Interference" task obtained by using Bayesian approach with AR(1) dependence for error terms. Two-stage estimation approach is used to estimate the posterior probabilities.

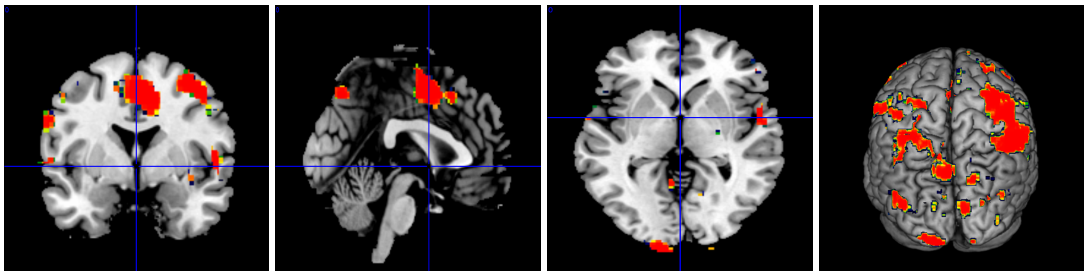


Figure 5.14: Predicting activation when performing "Ink Only" task obtained by using Frequentist approach with AR(1) dependence for error terms.

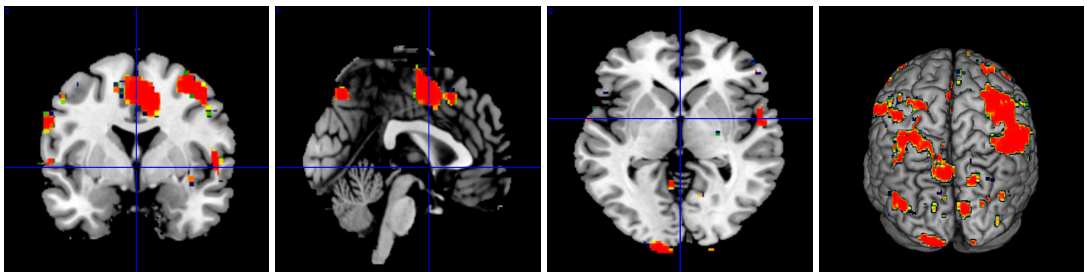


Figure 5.15: Predicting activation when performing "Congruence" task obtained by using Frequentist approach with AR(1) dependence for error terms.

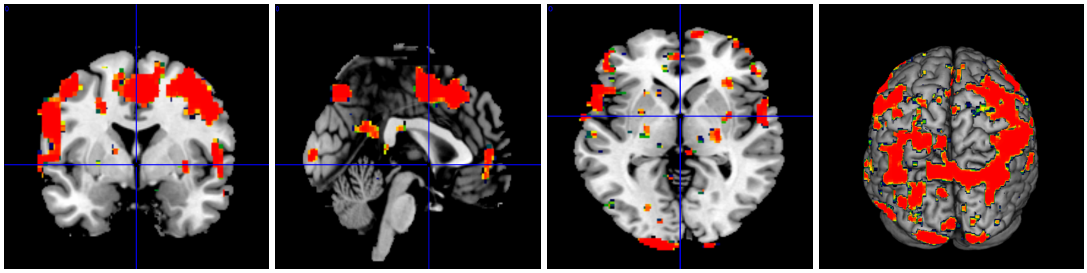


Figure 5.16: Predicting activation when performing "Interference" task obtained by using Frequentist approach with AR(1) dependence for error terms.

Activation Pattern Changing over Time

In our fMRI experiment, the three different tasks are repeated three times. The model in (5.15) fails to serve the goal of studying a possible activation change over time. To study the change of activation pattern, we generalize model in (5.15) as follows

$$\begin{aligned}\boldsymbol{y}_v =& \alpha_0 \boldsymbol{z}_0 + \alpha_1 \boldsymbol{x}_1 \\ &+ \beta_{1,1}\boldsymbol{x}_{1,1} + \beta_{1,2}\boldsymbol{x}_{1,2} + \beta_{1,3}\boldsymbol{x}_{1,3} \\ &+ \beta_{2,1}\boldsymbol{x}_{2,1} + \beta_{2,2}\boldsymbol{x}_{2,2} + \beta_{2,3}\boldsymbol{x}_{2,3} \\ &+ \beta_{3,1}\boldsymbol{x}_{3,1} + \beta_{3,2}\boldsymbol{x}_{3,2} + \beta_{3,3}\boldsymbol{x}_{3,3} + \boldsymbol{\varepsilon}_v, \end{aligned} \tag{5.16}$$

where $\boldsymbol{x}_{i,j}$ is transformed input function of task j in ith trial, $\boldsymbol{z}_i$ is a vector used to remove low-frequency, stimulus independent effects, $\beta_{i,j}$ is the parameter of interest corresponding j task in ith trial, α_i is the the nuisance parameter to model the baseline of a brain signal and $\boldsymbol{\varepsilon}_v$ has a normal distribution, $\mathcal{N}(0, \sigma_v^2 \Lambda_v)$.

A sequence of binary variable

$$\gamma_v = \{\gamma_{1,v}, \gamma_{2,v}, \gamma_{3,v}, \gamma_{4,v}, \gamma_{5,v}, \gamma_{6,v}, \gamma_{7,v}, \gamma_{8,v}, \gamma_{9,v}, \gamma_{10,v}, \gamma_{11,v}\}$$

is used to indicate if the corresponding parameter is zero or not. In our case, we always assume α_i nonzero. Therefore, the sequence becomes

$$\gamma_v = \{1, 1, \gamma_{3,v}, \gamma_{4,v}, \gamma_{5,v}, \gamma_{6,v}, \gamma_{7,v}, \gamma_{8,v}, \gamma_{9,v}, \gamma_{10,v}, \gamma_{11,v}\}.$$

Figures 5.17, 5.18, and 5.19 are the activation maps of the human brain corresponding to performing different tasks in different phases obtained from the frequentist approach where we assume independent error terms and take $p < 0.0001$. On the other, figures 5.23, 5.24, and 5.25 are also obtained from the frequentist approach and take $p < 0.0001$, but we assume error terms have AR(1) dependence. In Bayesian analysis, we use the two-stage approach to estimate the posterior probabilities when assuming AR(1) dependence in error terms. Figures 5.20, 5.21, and 5.22 show the activation maps when assuming independence in error terms and figures 5.26, 5.27, and 5.28 when assuming AR(1) dependence in error terms. All these figures show that the partial lobe is activated during these all tasks in different phases but the occipital lobe is only activated in the third trial and the frontal lobe is slightly activated in the first trial of congruence and interference tasks and in the second trial of ink only task. Temporal lobe is only activated in the third phase, not in first and second phases. As expected, the size of activation areas when performing the "Interference" task is bigger than that when performing the other two. All estimates of posterior probabilities based on 1,000 samples which takes about 70 hours using C++ code in Appendix.

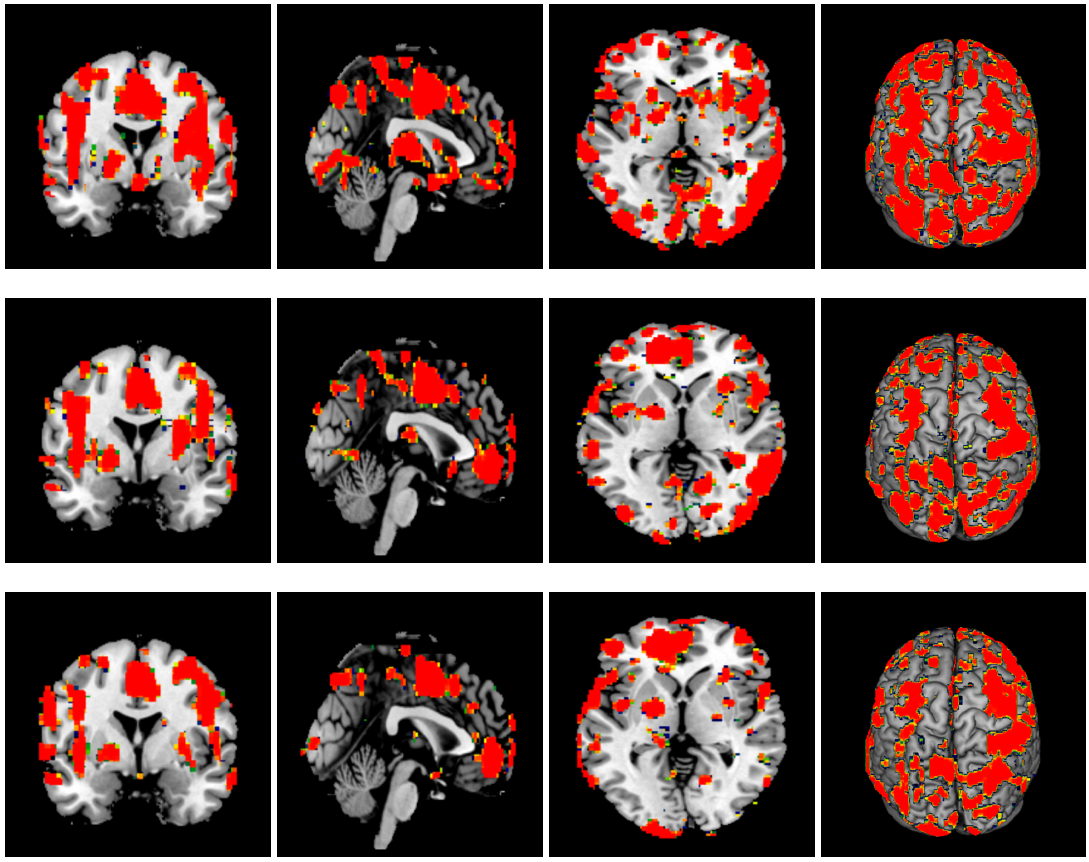


Figure 5.17: Predicting activation from the frequentist approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the first trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

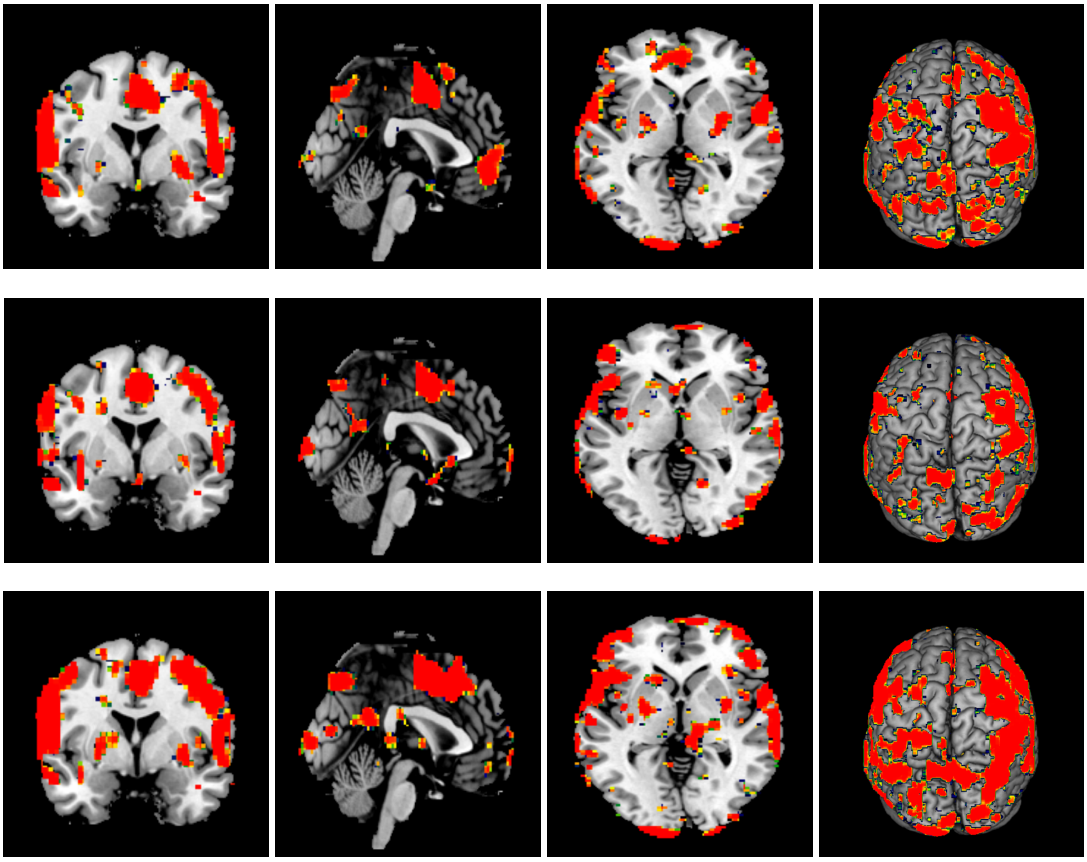


Figure 5.18: Predicting activation from the frequentist approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the second trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

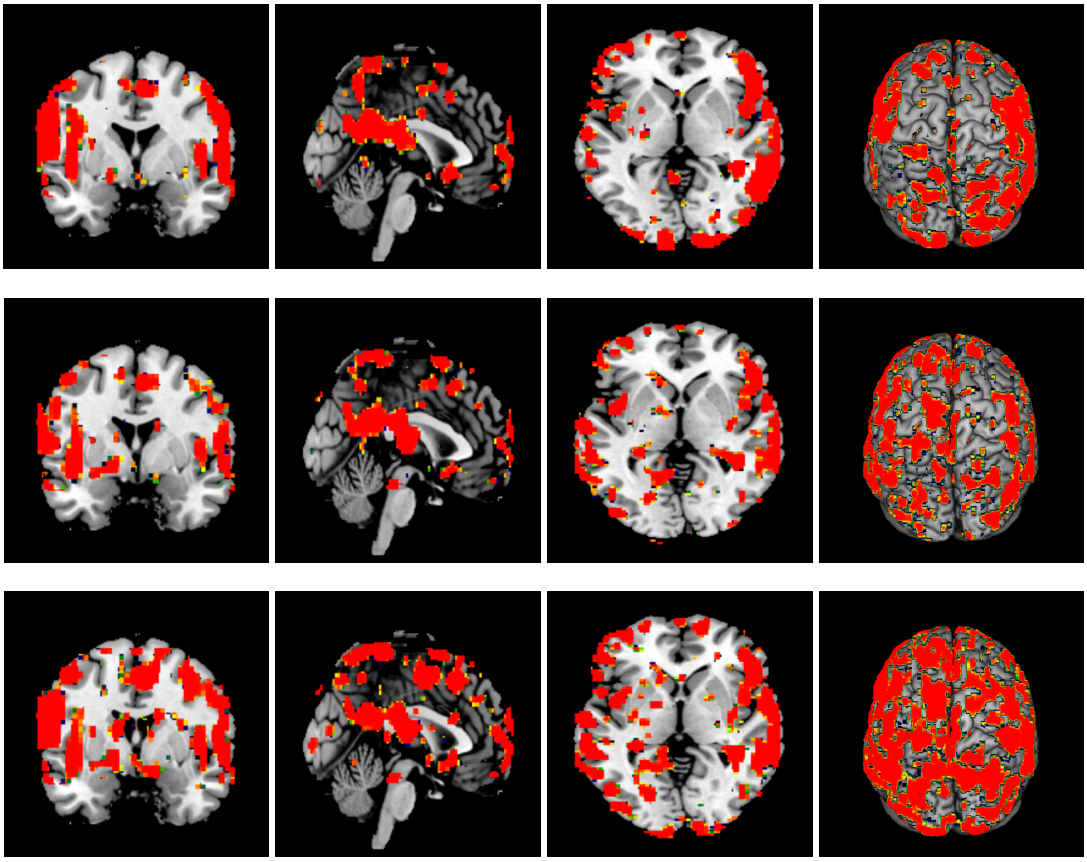


Figure 5.19: Predicting activation from the frequentist approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the third trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

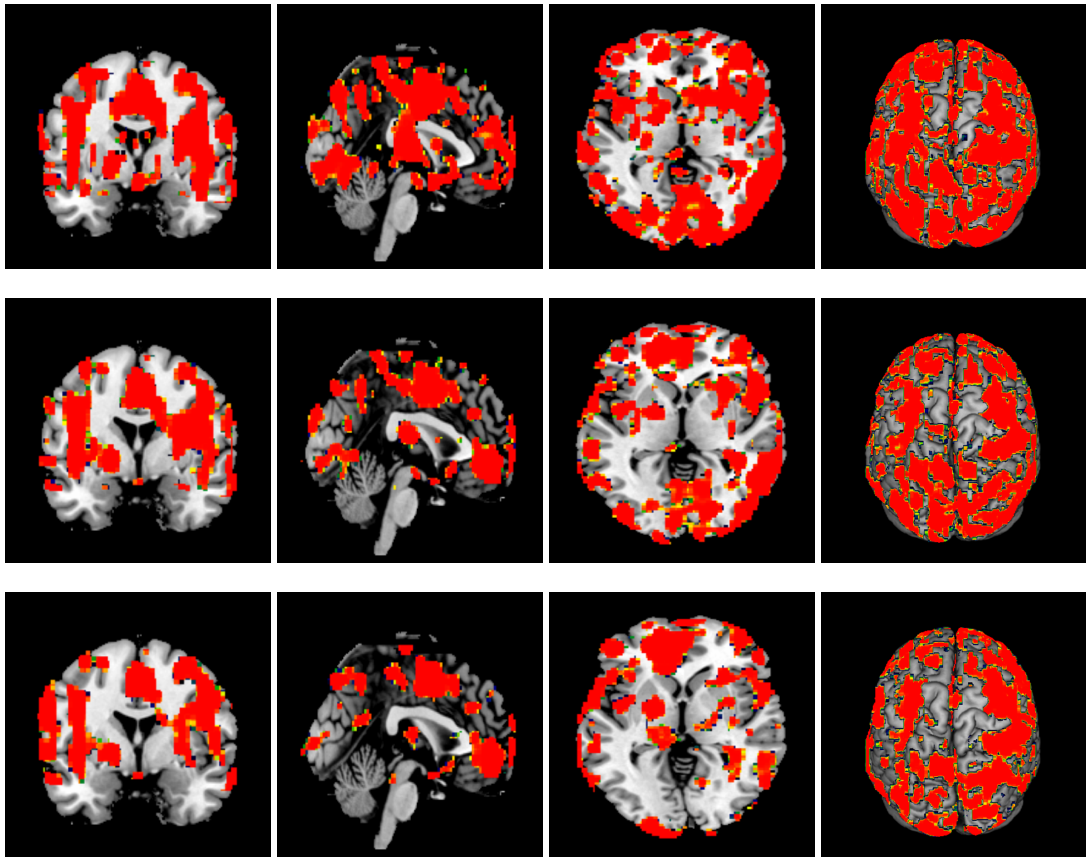


Figure 5.20: Predicting activation from the Bayesian approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the first trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

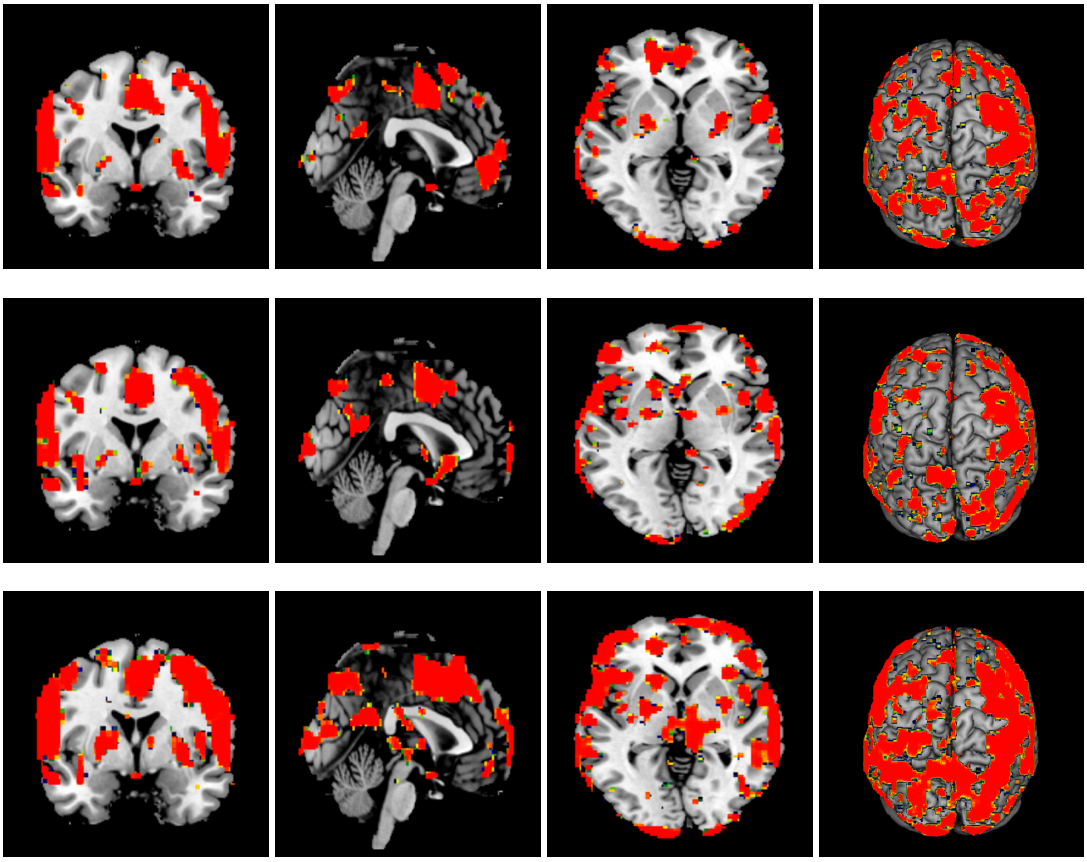


Figure 5.21: Predicting activation from the Bayesian approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the second trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

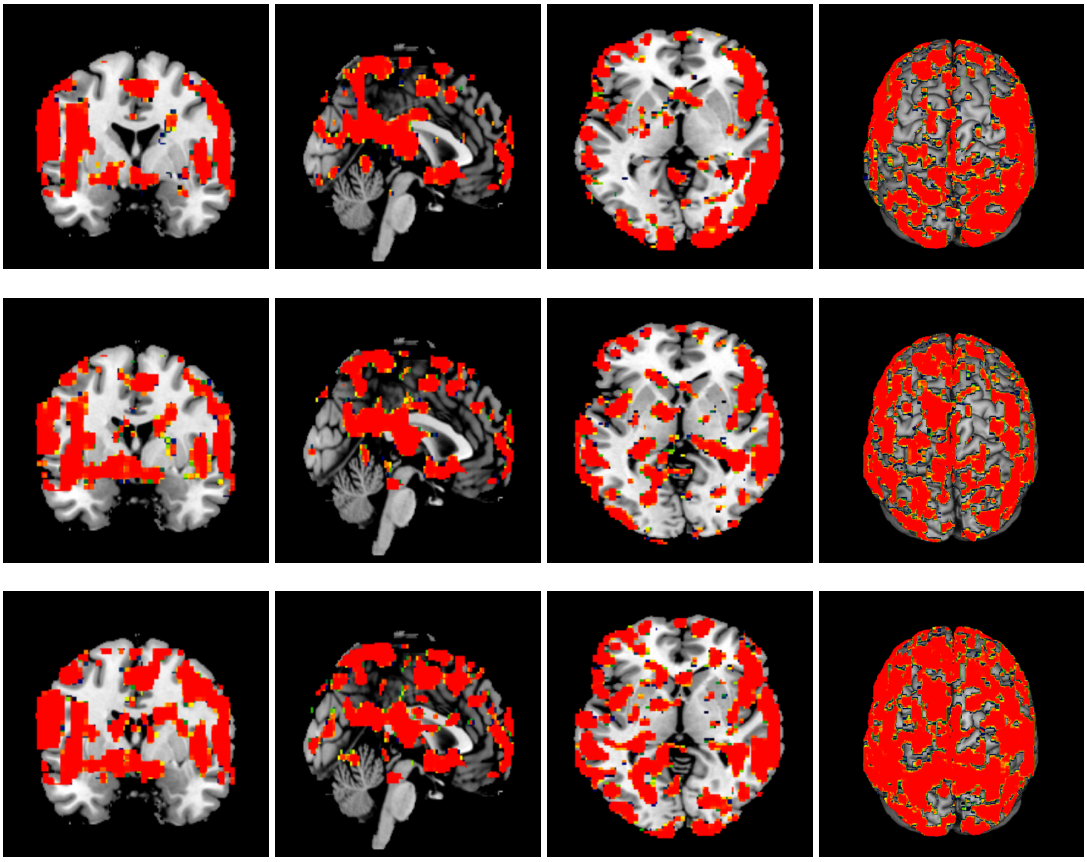


Figure 5.22: Predicting activation from the frequentist approach with assumption of independent error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the third trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

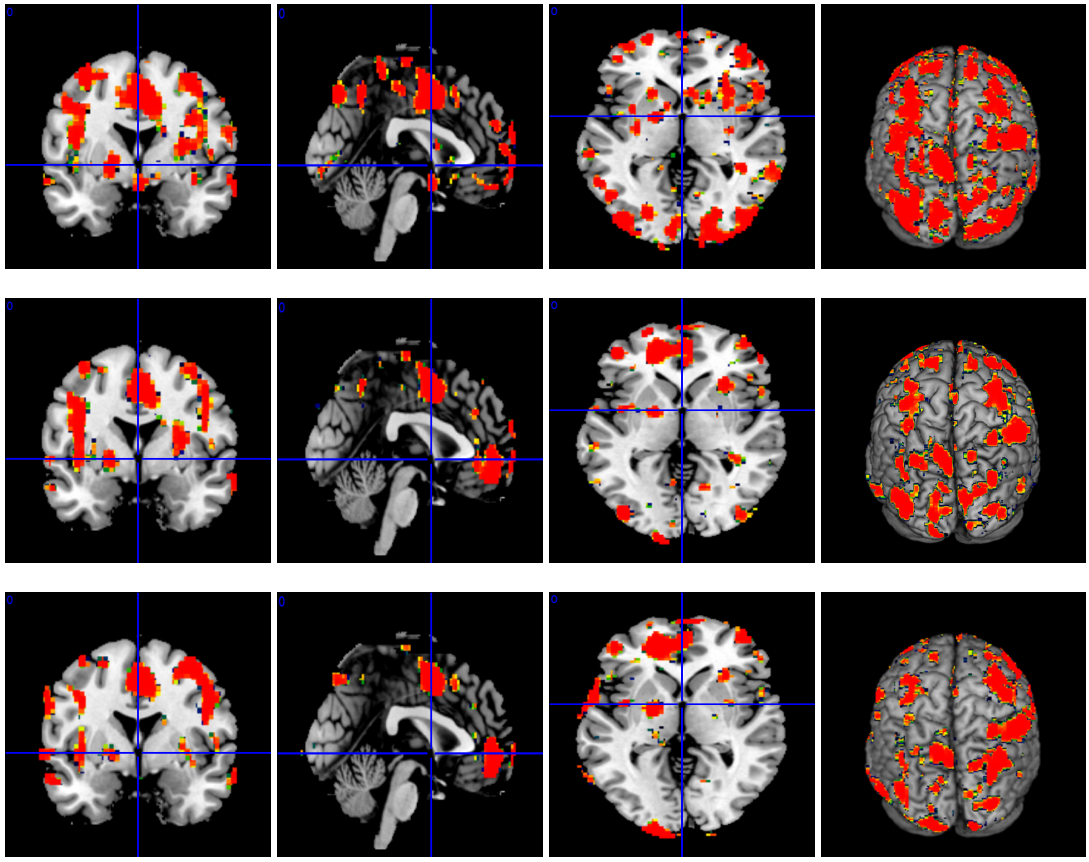


Figure 5.23: Predicting activation from the frequentist approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the first trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

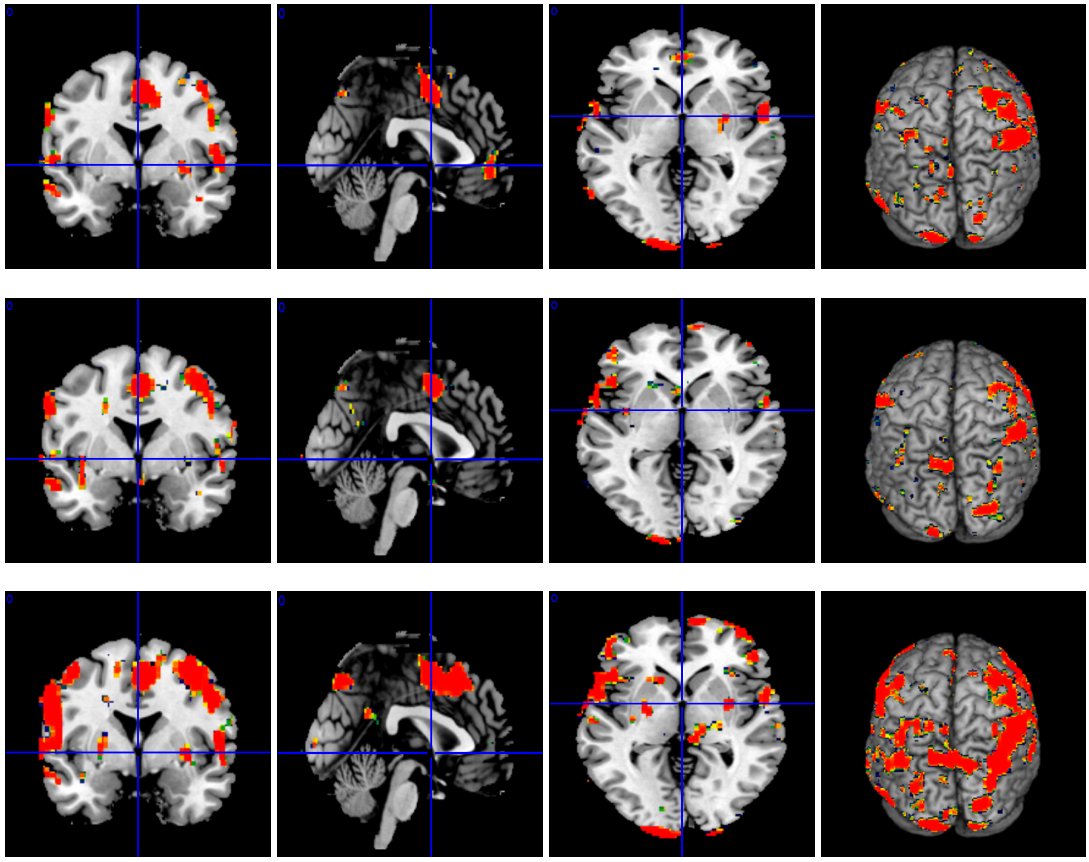


Figure 5.24: Predicting activation from the frequentist approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the second trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

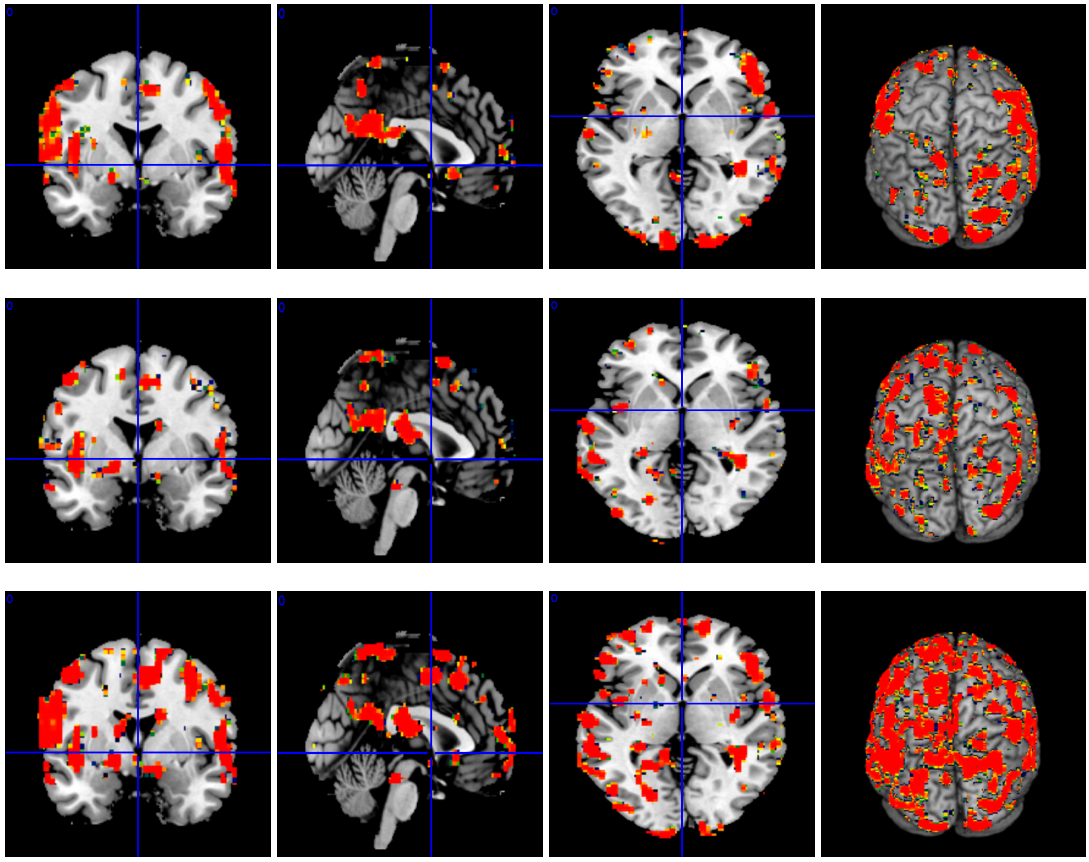


Figure 5.25: Predicting activation from the frequentist approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the third trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

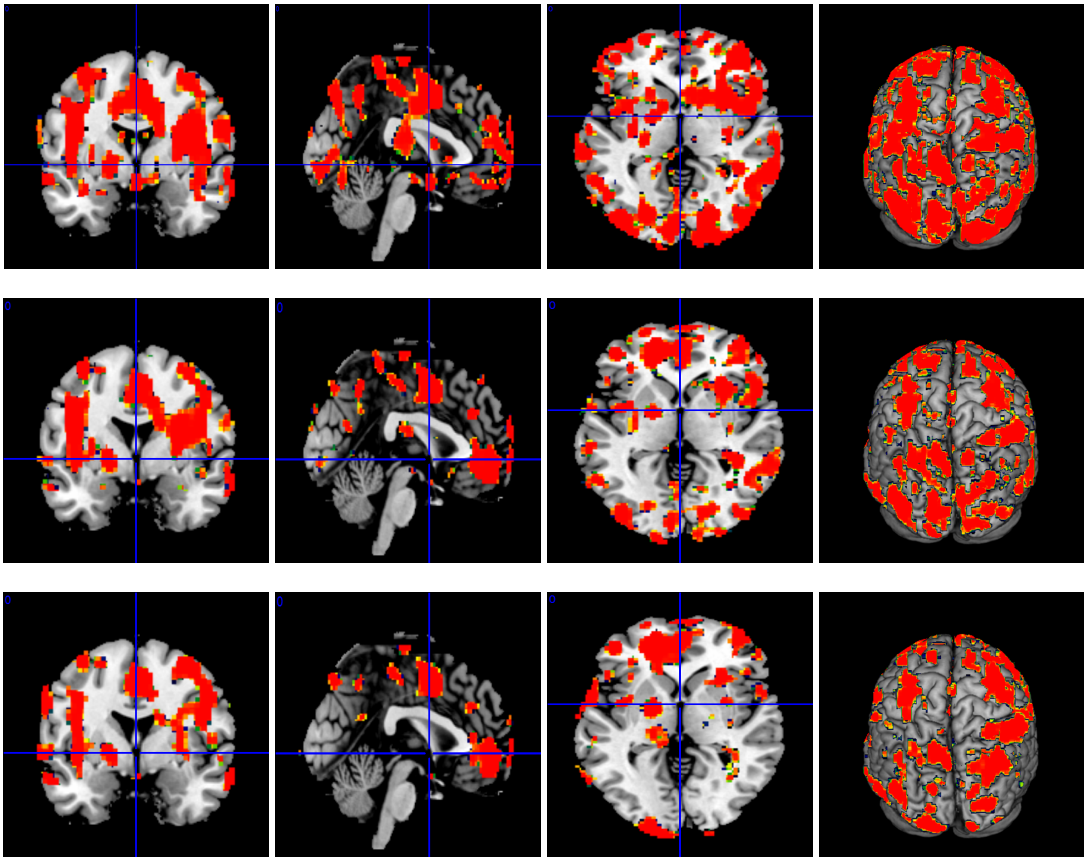


Figure 5.26: Predicting activation from the Bayesian approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the first trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

5.8 Gaussian Conditional Autoregressive Models

An alternative prior we consider for the binary variable $\boldsymbol{\gamma}=(\gamma_1, \ldots, \gamma_n)$ is Gaussian conditional autoregression (CAR) models differing in the degree to which spatial dependence is modeled by Ising distribution. All of them can model spatial structure; however, they accommodate it in different ways. Naturally, this begs the question about the respective

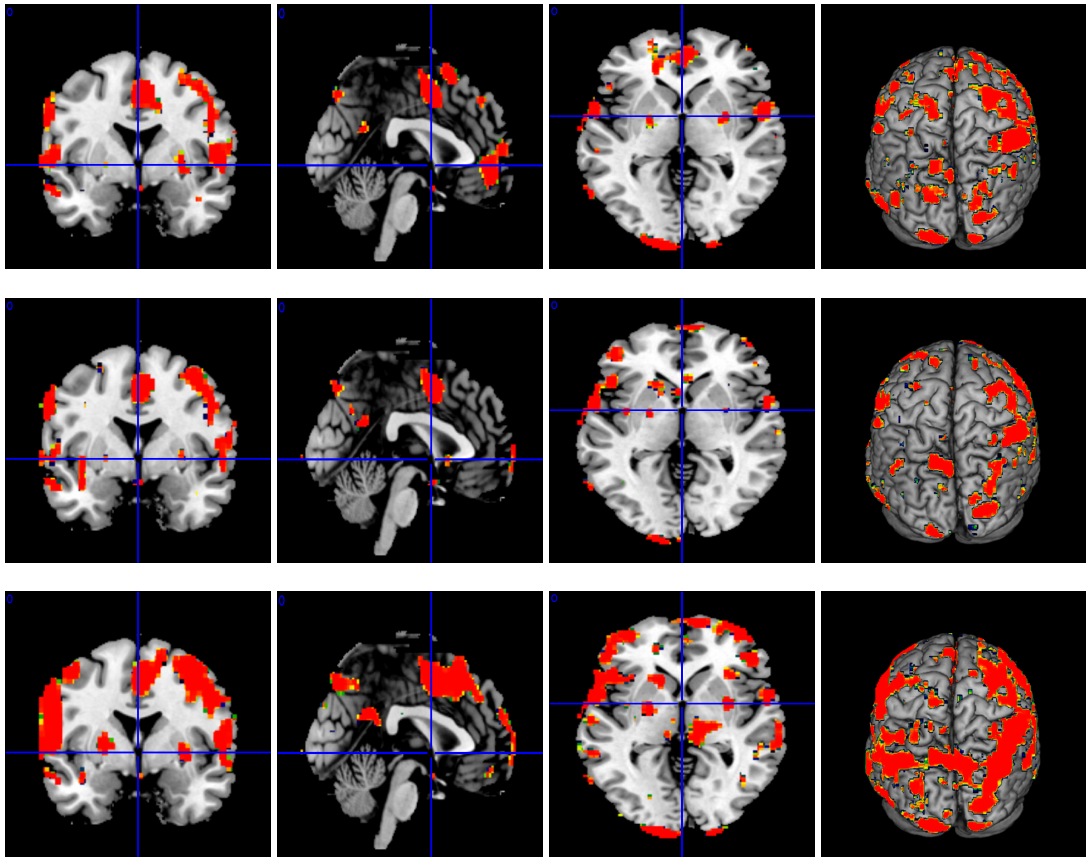


Figure 5.27: Predicting activation from the Bayesian approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the second trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

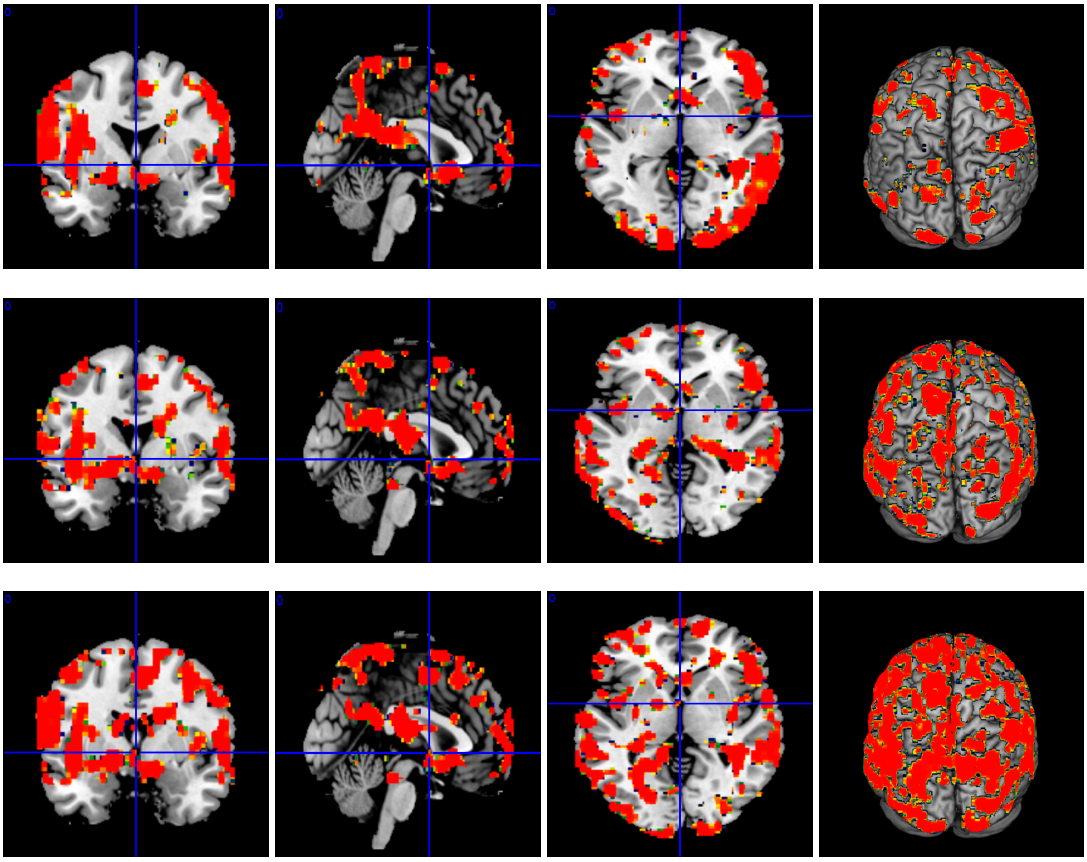


Figure 5.28: Predicting activation from the Bayesian approach with assumption of AR(1) dependence in error terms when performing "Ink Only", "Congruence," and "Interference" tasks in the third trial. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

merits and demerits of the various approaches. A short-sighted comparison of the two approaches that consider only the precision of parameter estimates is dangerous since the methods can represent completely different philosophies. We study different models and different algorithms

The prior distribution for $\boldsymbol{\beta}_v(\boldsymbol{\gamma}_v)$ for a particular voxel, v, is still assumed to be a Zellner's g-prior and the $\boldsymbol{\gamma}_v$ and ρ_v is confined within (-1, 1) and uniformly distributed. However, we take another prior, CAR, to model the spatial correlation between voxels.

The priors we consider for the binary variable $\boldsymbol{\gamma}_v$=$(\gamma_{v,1}, \ldots, \gamma_{v,p})$ are Gaussian conditional autoregression (CAR) models. CAR models are able to describe the interaction between neighbors and have been widely used in spatial statistics and Bayesian image analysis. The binary variable $\gamma_{v,j}$ equal to 1 or 0 in terms of the latent data $u_{v,j}$, that is,

$$\gamma_{v,j} = \begin{cases} 1, & \text{iff } u_{v,j} > 0; \\ 0, & \text{otherwise,} \end{cases}$$

for $v = 1, \ldots, N$. Let $\boldsymbol{u}_j$= $(u_{1,j}, \ldots, u_{N,j})$ be a vector of latent variables for regressor j. We assume that $\boldsymbol{u}_j$ is a Gaussian conditional autoregressive process given by

$$E(u_{v,j}|u_{(k,j)\neq(v,j)}) = \theta_j \sum_{k\in\delta_v} c_{v,k}u_{k,j}, \tag{5.17}$$

$$\text{Var}(u_{v,j}|u_{(k,j)\neq(v,j)}) = \eta^2_{v,j}, \quad v = 1, \ldots, N, \tag{5.18}$$

where the θ_j denote spatial dependence parameter, $c_{v,i}$ is the conditional autoregression coefficients with the diagonal elements $c_{v,v} = 0$, and δ_v represents the neighbors of site v. With the conditional means and variances given by (5.17) and (5.18), [79] show that these conditional distributions generate a valid joint multivariate normal distribution

$$\boldsymbol{u}_j \sim N(0, (I_j - \theta_j C_j)^{-1} H_j),$$

where H_j=diag$[\eta^2_{1,j}, \ldots, \eta^2_{N,j}]$ and $C_j = [c_{v,i}]$ and the covariance matrix $(I - \theta_j C_j)^{-1} H_j$ is symmetric and positive definite.

The two kinds o Gaussian conditional autogressions we consider are discussed here. The first one is used by [80] which is an intrinsic CAR model [81]. Let $\boldsymbol{u}_j = (u_{1,j}, \ldots, u_{N,j})$

be a conditional autoregressive Gaussian process and the conditional specification, labeled as CAR1, is given by

$$E(u_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)}) = \theta_j \sum_{k\in\delta_\upsilon} \frac{\omega_{k,\upsilon}}{\sqrt{\sum_{k\in\delta_\upsilon}\omega_{k,\upsilon}}} u_{k,j}, \tag{5.19}$$

$$\mathrm{Var}(u_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)}) = \frac{1}{\sum_{k\in\delta_\upsilon}\omega_{k,\upsilon}}, \quad \upsilon = 1,\ldots,n, j = 1,\ldots,p, \tag{5.20}$$

where

$$\omega_{k,\upsilon} = \begin{cases} 1, & \text{if } k \text{ and } \upsilon \text{ are neighbors;} \\ 0, & \text{otherwise.} \end{cases}$$

We have $\boldsymbol{u}_j \sim N\left(0, (I_j - \theta_j\Omega_j)^{-1}\Gamma_j\right)$, where the element of Ω_j in position (a,b) is $\omega_{a,b}$ and the diagonal elements are zero, and $\Gamma_j = \mathrm{diag}[\frac{1}{\sum_{k\in\delta_1}\omega_{k,1}}, \ldots, \frac{1}{\sum_{k\in\delta_N}\omega_{k,N}}]$. For simplicity, let $P_j^{-1} = (I_j - \theta_j\Omega_j)^{-1}\Gamma_j$, then

$$P_j(\upsilon,\upsilon) = \sum_{k\in\delta_\upsilon}\omega_{k,\upsilon},$$

$$P_j(k,\upsilon) = \begin{cases} -\theta_j\omega_{k,\upsilon}, & \text{if } k \text{ and } \upsilon \text{ are neighbors;} \\ 0, & \text{otherwise.} \end{cases}$$

To ensure that P_j is symmetric and positive definite, the parameter θ_j modelling spatial dependence is restricted to

$$\lambda^j_{\min} \leq \theta_j \leq \lambda^j_{\max},$$

where $\lambda^j_{\min}$ and $\lambda^j_{\max}$ are the minimum and maximum eigenvalues of matrix C_j, respectively.

Another type of CAR considered here is given as follows, labeled as CAR2,

$$E(u_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)}) = \theta_j \sum_{k\in\delta_\upsilon} \frac{\omega_{k,\upsilon}}{1+|\theta_j|\sum_{k\in\delta_\upsilon}\omega_{k,\upsilon}} u_{k,j}, \tag{5.21}$$

$$\mathrm{Var}(u_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)}) = \frac{1}{1+|\theta_j|\sum_{k\in\delta_\upsilon}\omega_{k,\upsilon}}, \quad i = 1,\ldots,n, j = 1,\ldots,p. \tag{5.22}$$

Then, the elements in matrix P_j different are

$$P_j(k,\upsilon) = \begin{cases} 1+|\theta|\sum_{k\in\delta_\upsilon}\omega_{k,\upsilon}, & \text{if } k = \upsilon; \\ -\theta_j\omega_{k,\upsilon}, & \text{if } k \neq \upsilon. \end{cases}$$

The advantage of the prior is that the P_j is positive definite for all real values of θ_j. Some properties of the model are explained in [82].

Then, the joint prior for $\boldsymbol{\beta}(\boldsymbol{\gamma})$, $\boldsymbol{\gamma}$, $\boldsymbol{u}$, $\boldsymbol{\rho}$, $\boldsymbol{\theta}$, and $\boldsymbol{\sigma}^2$ can be written as

$$\pi(\boldsymbol{\beta}(\boldsymbol{\gamma}), \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\theta}, \boldsymbol{\sigma^2}) \propto \pi(\boldsymbol{\beta}(\boldsymbol{\gamma})|\boldsymbol{\gamma})\pi(\boldsymbol{\gamma}|\boldsymbol{u})\pi(\boldsymbol{u}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\pi(\boldsymbol{\rho})\pi(\boldsymbol{\sigma^2}). \tag{5.23}$$

Therefore, the joint posterior density of $\boldsymbol{\beta}(\boldsymbol{\gamma})$, $\boldsymbol{\sigma}^2$, $\boldsymbol{\gamma}$, $\boldsymbol{u}$, $\boldsymbol{\rho}$, and $\boldsymbol{\theta}$, given the data, $\boldsymbol{y}$, can be obtained easily from combining the likelihood with the prior given by

$$\begin{aligned}
&\pi(\boldsymbol{\beta}(\boldsymbol{\gamma}), \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\theta}, \boldsymbol{\sigma^2}|\boldsymbol{y}) \\
&\propto p(\boldsymbol{y}|\boldsymbol{\beta}(\boldsymbol{\gamma}), \boldsymbol{\gamma})\pi(\boldsymbol{\beta}(\boldsymbol{\gamma})|\boldsymbol{\gamma})\pi(\boldsymbol{\gamma}|\boldsymbol{u})\pi(\boldsymbol{u}|\boldsymbol{\theta})\pi(\boldsymbol{\rho})\pi(\boldsymbol{\sigma^2}) \\
&\propto \prod_{\upsilon=1}^{N} \frac{1}{|\sigma_\upsilon^2 \Lambda_\upsilon|^{1/2}} \exp\left\{-\frac{1}{2\sigma_\upsilon^2}\left[\boldsymbol{y}_\upsilon - \mathbf{X}_\upsilon\boldsymbol{\beta}_\upsilon\right]' \Lambda_\upsilon^{-1}\left[\boldsymbol{y}_\upsilon - \mathbf{X}_\upsilon\boldsymbol{\beta}_\upsilon\right]\right\} \\
&\times \prod_{\upsilon=1}^{N} \frac{1}{|T_\upsilon\sigma_\upsilon^2(\mathbf{X}_\upsilon'\Lambda_\upsilon^{-1}\mathbf{X}_\upsilon)^{-1}|^{1/2}} \exp\left\{-\frac{1}{2T_\upsilon\sigma_\upsilon^2}\left[\boldsymbol{\beta}_\upsilon(\boldsymbol{\gamma}_\upsilon) - \hat{\boldsymbol{\beta}}_\upsilon(\boldsymbol{\gamma}_\upsilon)\right]' \mathbf{X}_\upsilon{}'\Lambda_\upsilon^{-1}\mathbf{X}_\upsilon\left[\boldsymbol{\beta}_\upsilon(\boldsymbol{\gamma}_\upsilon) - \hat{\boldsymbol{\beta}}_\upsilon(\boldsymbol{\gamma}_\upsilon))\right]\right\} \\
&\times \prod_{j=1}^{p} |P_j|^{1/2} \exp\left\{-\frac{\boldsymbol{u}_j' P \boldsymbol{u}_j}{2}\right\} \\
&\times \prod_{\upsilon=1}^{N} \frac{1}{\sigma_\upsilon^2}.
\end{aligned}$$

Consider a specific voxel, υ, after integrating out $\boldsymbol{\beta}_\upsilon(\boldsymbol{\gamma}_\upsilon)$ and σ_υ^2, we also have the same marginal posterior $p(\boldsymbol{\gamma}_\upsilon, \theta_\upsilon, \rho_\upsilon|\boldsymbol{y}_\upsilon)$ given in (5.8).

5.8.1 Bayesian Inference via MCMC Sampling

Here we present a Markov chain Monte Carlo sampling scheme for the spatial Bayesian variable selection models given above.

- Generate $(u_{\upsilon,j}, \gamma_{\upsilon,j})$ from $p(u_{\upsilon,j}, \gamma_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)}, \theta_\upsilon, \rho_\upsilon, \boldsymbol{y}_\upsilon)$, for $\upsilon = 1, 2, \ldots, N$ and $j = 1, \ldots, p$. In this step, the $(u_{\upsilon,j}, \gamma_{\upsilon,j})$ is generated as a pair in order to avoid reducible sampling. The joint density function of $(u_{\upsilon,j}, \gamma_{\upsilon,j})$ can be expressed as

  $$p(u_{\upsilon,j}, \gamma_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)}, \theta_\upsilon, \rho_\upsilon, \boldsymbol{y}_\upsilon) = p(\gamma_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)}, \theta_\upsilon, \rho_\upsilon, \boldsymbol{y}_\upsilon)p(u_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)}, \gamma_{\upsilon,j}, \theta_\upsilon)$$

Therefore, we first generate the $\gamma_{\upsilon,j}$ from $p(\gamma_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon,\rho_\upsilon,\boldsymbol{y}_\upsilon)$ and then $u_{\upsilon,j}$ from $p(u_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)},\gamma_{\upsilon,j},\theta_\upsilon)$. Since

$$p(\gamma_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon,\rho_\upsilon,\boldsymbol{y}_\upsilon)\propto p(\boldsymbol{y}_\upsilon|\gamma_{\upsilon,j},\rho_\upsilon)p(\gamma_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon), \tag{5.24}$$

to generate the $\gamma_{\upsilon,j}$ from $p(\gamma_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon,\rho_\upsilon,\boldsymbol{y}_\upsilon)$, we have to evaluate the density, $p(\gamma_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon)$, shown as following

$$\begin{aligned}
p(\gamma_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon) &= \int p(\gamma_{\upsilon,j},u_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon)du_{\upsilon,j}\\
&= \int p(\gamma_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon)p(u_{\upsilon,j}|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon)du_{\upsilon,j}\\
&\propto \int \phi(u_{\upsilon,j};\mu_{\upsilon,j},\tau^2_{\upsilon,j})du_{\upsilon,j},
\end{aligned}$$

where $\phi(\cdot;\mu,\varsigma^2)$ is the normal density function with mean μ and variance ς^2. On account of two priors considered, the means $\mu_{\upsilon,j}$ and variances $\tau^2_{\upsilon,j}$ corresponding to two priors are given, respectively,

$$\mu_{\upsilon,j}=\begin{cases}\dfrac{\theta_j\sum_{k\in\delta_\upsilon}u_{k,j}\omega_{k,j}}{\sum_{k\in\delta_\upsilon}u_{k,j}\omega_{k,j}}, & \text{for CAR1;}\\ \dfrac{\theta_j\sum_{k\in\delta_\upsilon}u_{k,j}\omega_{k,j}}{1+|\theta_j|\sum_{k\in\delta_\upsilon}u_{k,j}\omega_{k,j}}, & \text{for CAR2;}\end{cases}$$

and

$$\varsigma^2_{\upsilon,j}=\begin{cases}\dfrac{1}{\sum_{k\in\delta_\upsilon}u_{k,j}\omega_{k,j}}, & \text{for CAR1;}\\ \dfrac{1}{1+|\theta_j|\sum_{k\in\delta_\upsilon}u_{k,j}\omega_{k,j}}, & \text{for CAR2.}\end{cases}$$

Then, the probabilities when $\gamma_{\upsilon,j}=1$ or 0 are

$$p(\gamma_{\upsilon,j}=1|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon)\propto\int_0^\infty\phi(u_{\upsilon,j};\mu_{\upsilon,j},\tau^2_{\upsilon,j})du_{\upsilon,j}=1-\Phi(0,\mu_{\upsilon,j},\varsigma^2_{\upsilon,j}); \tag{5.25}$$

$$p(\gamma_{\upsilon,j}=0|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon)\propto\int_{-\infty}^0\phi(u_{\upsilon,j};\mu_{\upsilon,j},\tau^2_{\upsilon,j})du_{\upsilon,j}=\Phi(0;\mu_{\upsilon,j},\varsigma^2_{\upsilon,j}), \tag{5.26}$$

where $\Phi(\cdot;\mu,\varsigma^2)$ is the cumulative distribution function of a normal random variable with mean μ and variance ς^2.

Plug (5.25) and (5.26) back into (5.24), we have

$$\begin{aligned}
p(\gamma_{\upsilon,j}=1|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon,\rho_\upsilon,\boldsymbol{y}_\upsilon) &\propto p(\boldsymbol{y}_\upsilon|\gamma_{\upsilon,j}=1,\rho_\upsilon)\times p(\gamma_{\upsilon,j}=1|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon)\\
&\propto l_{\upsilon,j}(\gamma_{\upsilon,j}=1,\rho_\upsilon)\times\left(1-\Phi(0;\mu_{\upsilon,j},\varsigma^2_{\upsilon,j})\right);
\end{aligned}$$

and

$$p(\gamma_{\upsilon,j}=0|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon,\rho_\upsilon,\boldsymbol{y}_\upsilon) \propto p(\boldsymbol{y}_\upsilon|\gamma_{\upsilon,j}=0)\times p(\gamma_{\upsilon,j}=0|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon,\rho_\upsilon)$$
$$\propto l_{\upsilon,j}(\gamma_{\upsilon,j}=0,\rho_\upsilon)\times\Phi(0;\mu_{\upsilon,j},\varsigma^2_{\upsilon,j})$$

where $l_{\upsilon,j}(\gamma_{\upsilon,j},\rho_\upsilon)=p(\boldsymbol{y}_\upsilon|\gamma_{\upsilon,j},\rho_\upsilon)$ and

$$p(\boldsymbol{y}_\upsilon|\gamma_{\upsilon,j},\rho_\upsilon)\propto|\Lambda_\upsilon|^{-1/2}(1+T_\upsilon)^{-q_\upsilon/2}S(\boldsymbol{\gamma}_\upsilon,\rho_\upsilon)^{-T_\upsilon/2}.$$

After normalization, the $\gamma_{\upsilon,j}$ for $i=1,\ldots,N$ and $j=1,\ldots,p$ are generated from

$$p(\gamma_{\upsilon,j}=1|u_{(k,j)\neq(\upsilon,j)},\theta_\upsilon,\rho_\upsilon,\boldsymbol{y}_\upsilon)=\frac{1}{1+g}$$

where

$$g=\frac{l_{\upsilon,j}(\gamma_{\upsilon,j}=0,\rho_\upsilon)}{l_{\upsilon,j}(\gamma_{\upsilon,j}=1,\rho_\upsilon)}\left(\frac{\Phi(0)}{1-\Phi(0)}\right).$$

- Generate ρ_υ for $\upsilon=1,\ldots,N$, given $\boldsymbol{\gamma}_\upsilon$ and $\boldsymbol{y}_\upsilon$, where

$$p(\rho_\upsilon|\boldsymbol{\gamma}_\upsilon,\boldsymbol{y}_\upsilon)\propto|\Lambda_\upsilon|^{-1/2}[S(\boldsymbol{\gamma}_\upsilon,\rho_\upsilon)]^{T_\upsilon/2}.$$

- Generate θ_j from $p(\theta_j|\boldsymbol{u}_j)$ for $j=1,\ldots,p$ using Metropolis-Hastings algorithm. Suppose the current value is θ_j and new candidate is θ'_j generated from a $N(\theta_j,0.02)$, then the candidate is accepted with probability

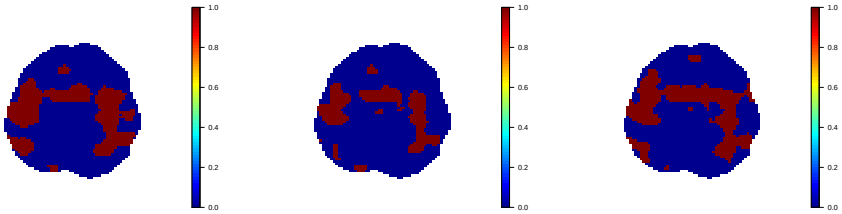
$$\min\left\{1,\frac{|P_j(\theta'_j)|^{1/2}\exp\{-\frac{\boldsymbol{u}'_jP_j(\theta'_j)\boldsymbol{u}_j}{2}\}}{|P_j(\theta_j)|^{1/2}\exp\{-\frac{\boldsymbol{u}'_jP_j(\theta_j)\boldsymbol{u}_j}{2}\}}\right\}.$$

5.8.2 Real Data Analysis

We apply the Bayesian selection models with three different spatial priors, Ising, CAR1, and CAR2 to the real dataset and compare the outputs to each other. Besides, we also provide the activation map obtained from the frequentist approach with $p=0.001$. Generating θ in CAR models is very computational effort because it needs to calculate the determinants of an $N\times N$ matrix P_j where N is the number of voxels. As a result, only Slice 57 and the region, parietal lobe, of interest are analyzed. In these datasets, we have $N=2225$ for Slice 57 and $N=8079$ in the parietal lobe. In Slice 57, the activation maps

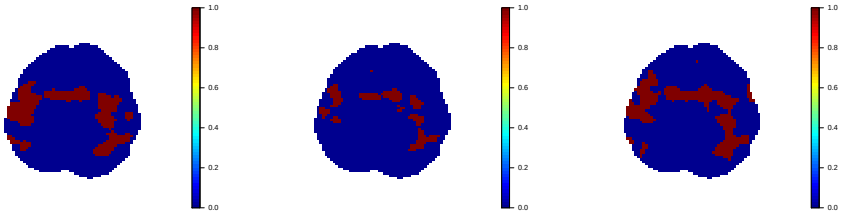
obtained from the frequentist approach are given in Figures 5.29 and 5.30. It can be found that the activation size in 5.30 is smaller than that in 5.29. Ignoring temporal correlation might lead to overestimating task effects on analyzing fMRI data, and there are some examples in the fMRI time-series [83] and [14] literature that address this issue in the regression context. Similarly, the activation maps for the parietal lobe are given in 5.38, 5.39, 5.34, 5.35, 5.36, and 5.37.

The activations maps obtained from the spatial Bayesian variable selection approaches show a little different activation patterns. The activation maps using Ising distribution to model spatial correlation display activation binding on each other. In contrast to Ising models, the activation maps using the autoregression models to model spatial correlation show separate activation areas.



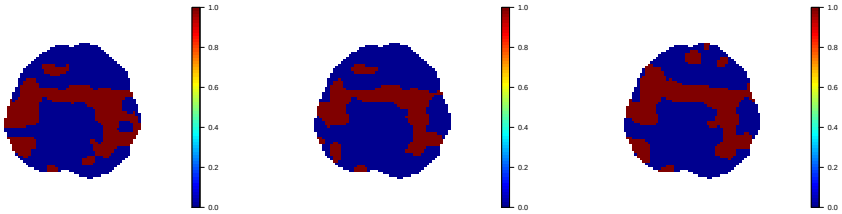
(a) Ink Only (b) Congruence (c) Interference

Figure 5.29: Predicting activations corresponding to different tasks obtained from the frequentist approach assuming independent error terms.



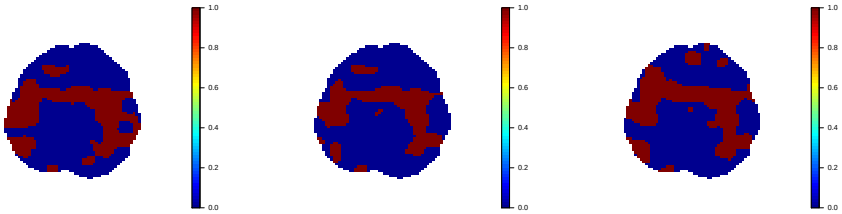
(a) Ink Only (b) Congruence (c) Interference

Figure 5.30: Predicting activations corresponding to different tasks obtained from the frequentist approach assuming AR(1) dependence in the error terms.



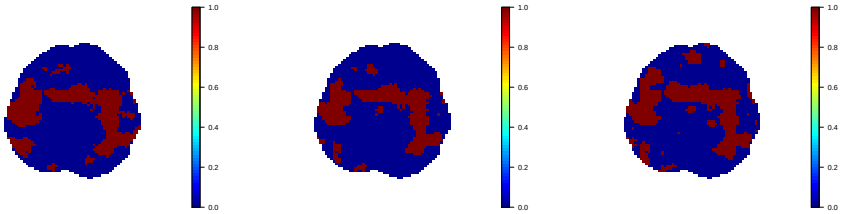
(a) Ink Only (b) Congruence (c) Interference

Figure 5.31: Predicting activation from the Bayesian variable selection approach with spatial relationship modeled by an Ising distribution.



(a) Ink Only (b) Congruence (c) Interference

Figure 5.32: Predicting activation from the Bayesian variable selection approach with spatial relationship modeled by the Gaussian autoregression model (CAR1).



(a) Ink Only (b) Congruence (c) Interference

Figure 5.33: Predicting activation from the Bayesian variable selection approach with spatial relationship modeled by the Gaussian autoregression model (CAR2).

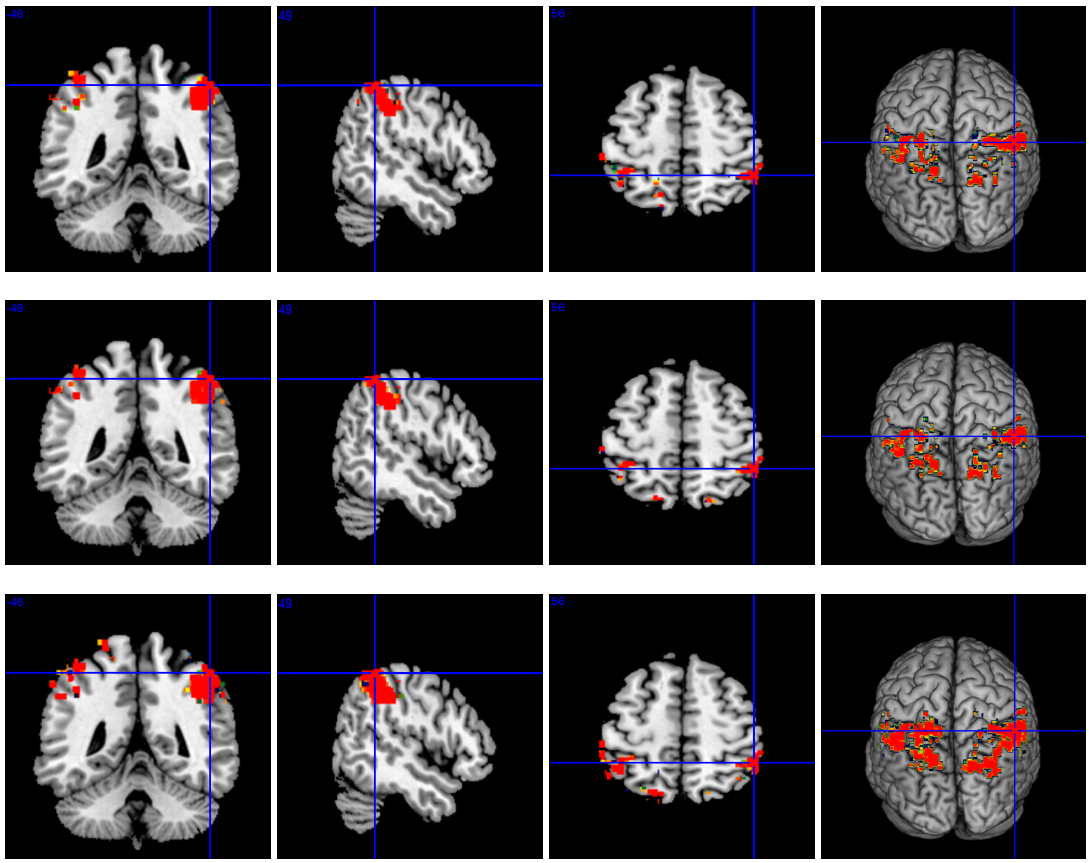


Figure 5.34: Predicting activation from the Bayesian approach when performing "Ink Only", "Congruence," and "Interference" by using Ising distribution to model spatial correlation and assuming independence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

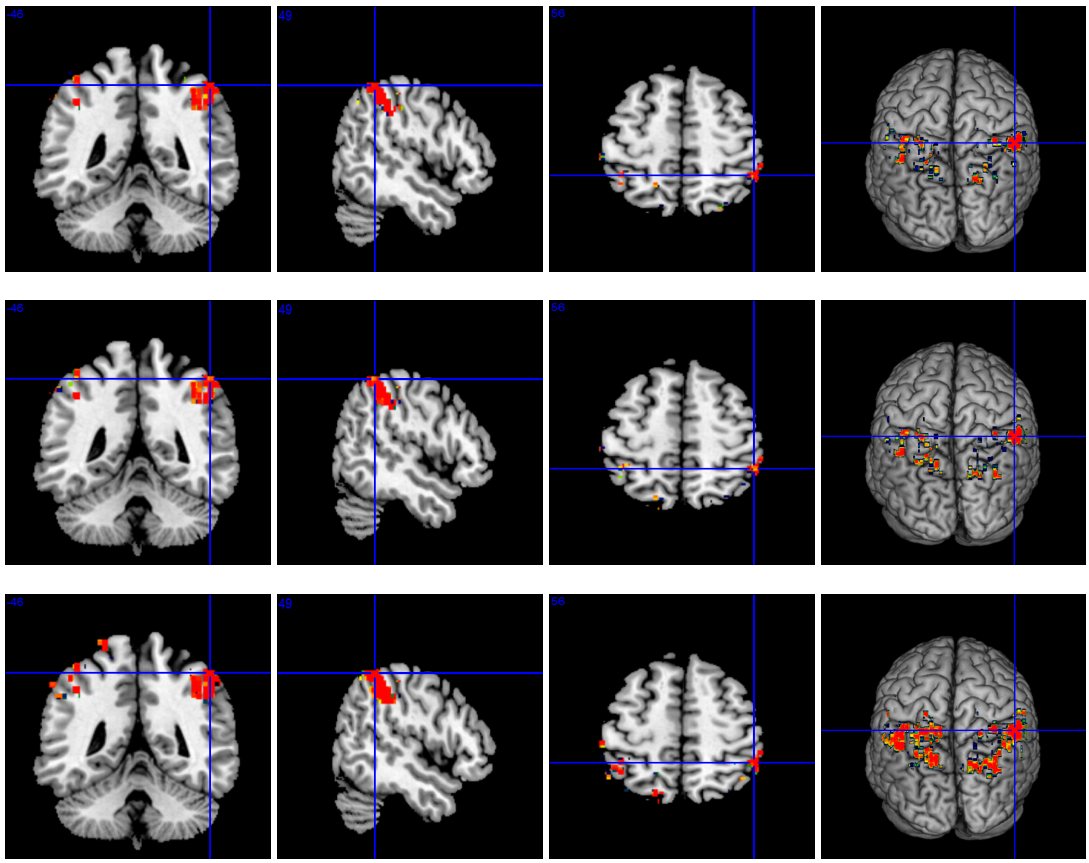


Figure 5.35: Predicting activation from the Bayesian approach when performing "Ink Only", "Congruence," and "Interference" by using Ising distribution to model spatial correlation and assuming AR(1) dependence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

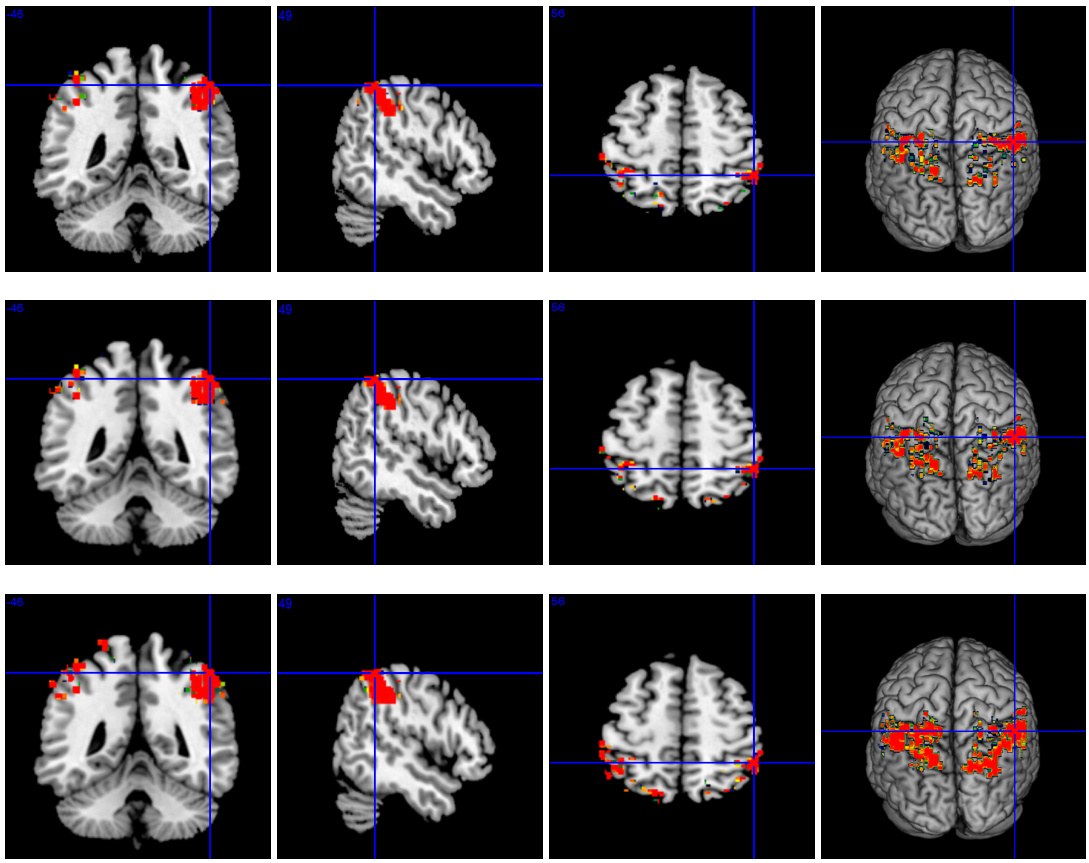


Figure 5.36: Predicting activation from the Bayesian approach when performing "Ink Only", "Congruence," and "Interference" by using CAR2 to model spatial correlation and assuming independence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

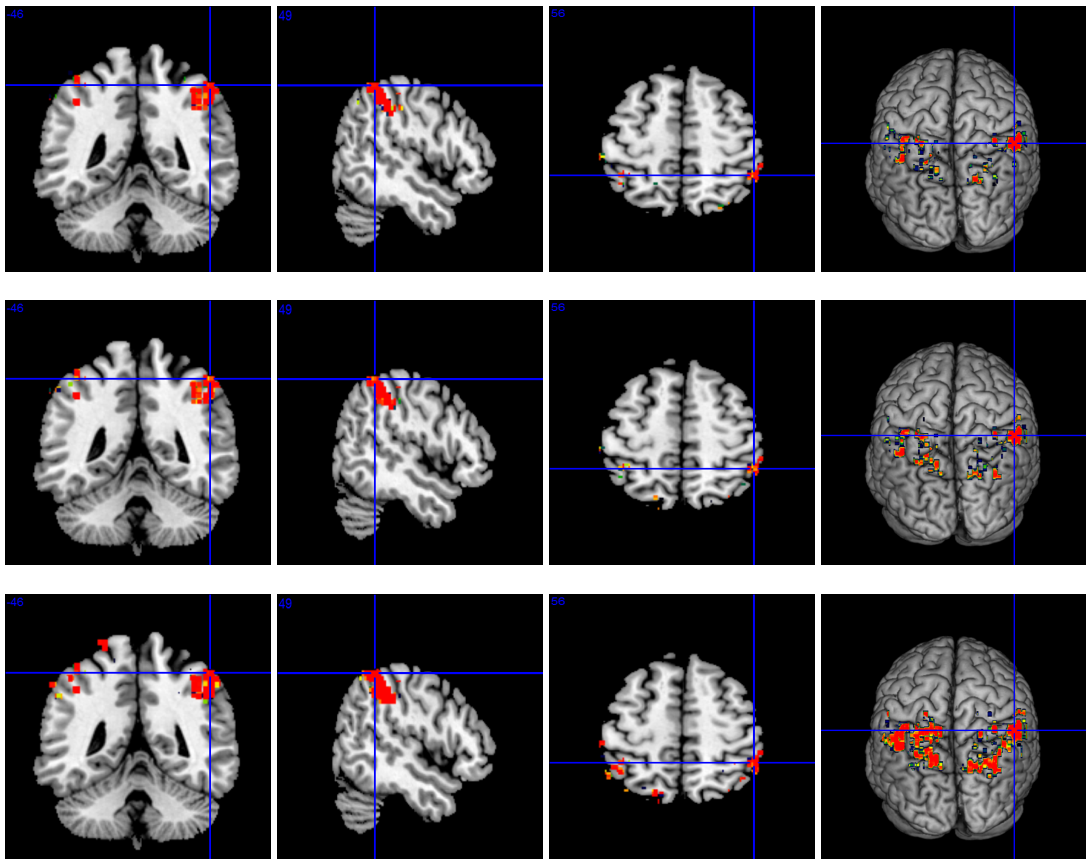


Figure 5.37: Predicting activation from the Bayesian approach when performing "Ink Only", "Congruence," and "Interference" by using CAR2 to model spatial correlation and assuming AR(1) dependence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

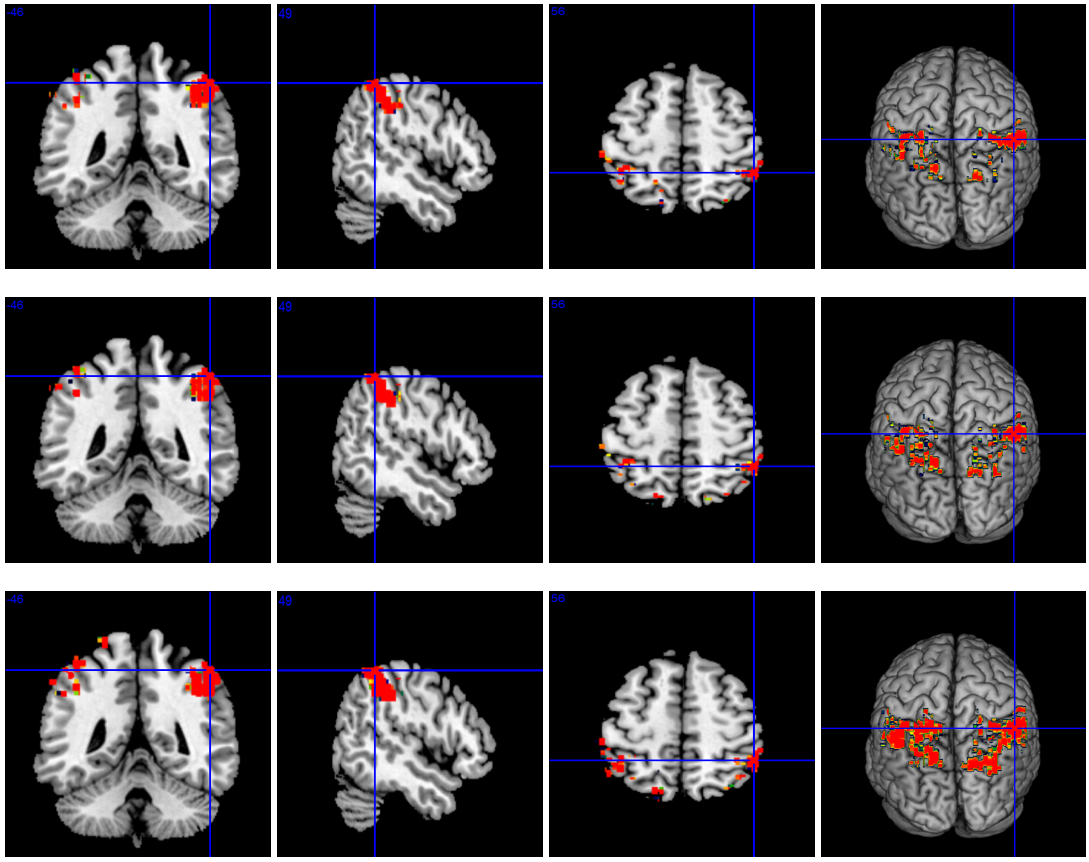


Figure 5.38: Predicting activation from the frequentist approach when performing "Ink Only", "Congruence," and "Interference" assuming independence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

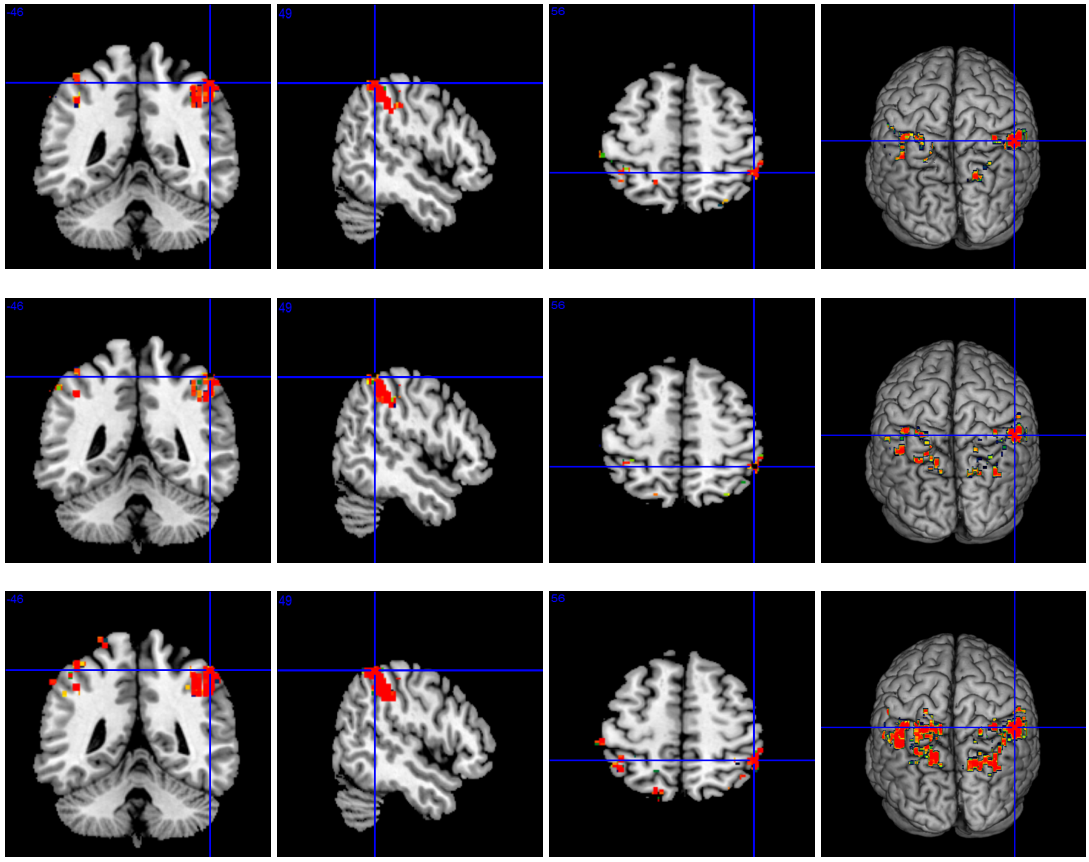


Figure 5.39: Predicting activation from the frequentist approach when performing "Ink Only", "Congruence," and "Interference" assuming AR(1) dependence in error terms. The top panel is activation maps when performing "Ink Only" task, the middle one for "Congruence" task, and bottom one for "Interference" task.

References

[1] Alzheimer's Association. 2010 Alzheimer's disease facts and figures. *Alzheimer's & Dementia: The Journal of the Alzheimer's Association*, 6:158–194, 2010.

[2] S.S Bassett1, D.M. Yousem, C. Cristinzio1, I. Kusevic1, M.A. Yassa1, B.S. Caffo, and S.L. Zeger. Familial risk for Alzheimer's disease alters fmri activation patterns. *Brain*, 129:1229–1239, 2006.

[3] M.A. Yassa, G. Verduzco, C. Cristinzio, and S.S. Bassett. Altered fMRI activation during mental rotation in those at genetic risk for Alzheimer disease. *Neurology*, 70:1898–1904, 2008.

[4] S.N. Thiyagesha, T.F.D. Farrowa, R.W. Parksa, H. Accosta-Mesad, C. Youngb, I.D. Wilkinsonc, M.D. Huntera, and P.W.R. Woodruffa. The neural basis of visuospatial perception in Alzheimer's disease and healthy elderly comparison subjects: An fMRI study. *Psychiatry Research: Neuroimaging*, 172:109–116, 2000.

[5] J.R. Stroop. Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, 7:643–661, 1935.

[6] C.J. Bench, C.D. Frith, P.M. Grasby, K.J. Friston, E. Paulesu, R.S.J. Frackowiak, , and R.J. Dolan. Investigations of the functional anatomy of attention using the Stroop test. *Neuropsychologia*, 31:907–922, 1993.

[7] C.S. Carter, M. Mintum, and J.D. Cohen. Interference and facilitation effects during selective attention: $H_2^{15}O$ study of Stroop task performance. *NeuroImage*, 280:747–749, 1995.

[8] L.M. Fisher, D.M. Freed, and S. Corkin. Stroop color-word test performance in patients with Alzheimer's disease. *Journal of Clinical and Experimental Neuropsychology*, 12:745–758, 1990.

[9] C. Li, J. Zheng, J. Wang, L. Gui, and C. Li. An fMRI Stroop task study of prefrontal cortical function in normal aging, mild cognitive impairment, and Alzheimer's disease. *Current Alzheimer Research*, 6:525–530, 2009.

[10] T.A. Polk, R.M. Drake, J.J. Jonides, M.R. Smith, and E.E. Smith. Attention enhances the neural processing of relevant features and suppresses the processing of irrelevant features in humans: A functional magnetic resonance imaging study of the Stroop task. *The Journal of Neuroscience*, 28:13786–13792, 2008.

[11] S.F. Taylora, S. Kornblumb, E.J. Lauberb, S. Minoshimac, and R.A. Koeppec. Isolation of specific interference processing in the Stroop task: PET activation studies. *NeuroImage*, 6:81–92, 1997.

[12] R.B. Buxton. The elusive initial dip. *Neuroimage*, 13:953–958, 2001.

[13] T.Q. Duong, D. Kim, and S. Kim. Spatiotemporal dynamics of the BOLD fMRI signals: Toward mapping submillimeter cortical columns using the early negative response. *Magnetic Resonance in Medicine*, 44:231–242, 2000.

[14] K.J. Friston, A. Holmes, K.J. Worsley, J.B. Polin, C. Frith, and R. Frackowik. Statistical parametric maps in functional image: A general linear approach. *Human Brain Mapping*, 2:189–210, 1995.

[15] K.J. Worsley. The geometry of random images. *CHANCE*, 9:27–40, 1996.

[16] K.J. Worsley, M. Andermann, T. Koulis, D. MacDonald, and A.C. Evans. Detecting changes in nonisotropic images. *Human Brain Mapping*, 8:98–101, 1992.

[17] M. Smith and L. Fahrmeir. Spatial Bayesian Variable Selection with Application to Functional Magnetic Resonance Imaging. *Journal of the American Statistical Association*, 102:417–431, 2007.

[18] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[19] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[20] S.H. Faro and F.B. Mohamed. *Functional MRI: Basic Principles and Clinical Applications*. Springer, 2006.

[21] R.A. Huettel, A.W. Song, and G. McCarthy. *Functional Magnetic Resonance Imaging, 2nd Edition*. Sinauer Associates, 2009.

[22] P. Jezzard, P.M. Matthews, and S.M. Smith. *Functional MRI: An Introduction to Methods*. Oxford Unversity Press., 2001.

[23] M. NessAiver. *All You really need to know About MRI Physics*. Simply Physics, 1996.

[24] S. Ogawa, T.M. Lee, A.R. Kay, and D.W. Tank. Brain magnetic resonance imaging with contrast dependent on blood osygenation. *Proceedings of the National Academy of Science*, 87:9869–9872, 1990.

[25] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. *In Y. Bizais, C. Barillot, and R. Di Paola, editors, Proc. Information Processing in Medical Imaging, Kluwer Academic Publishers*, pages 263–274, 1995.

[26] K.J. Worsley. Detecting activation in fMRI data. *Statistical Methods in Medical Research.*, 12:401–418, 2003.

[27] G. K. Aguirre, E. Zarahn, and M. D'Esposito. Empirical analyses of BOLD fMRI statistics. II. Spatially smoothed data collected under null-hypothesis and experimental conditions. *NeuroImage*, 5:199–212, 1997.

[28] S.M. Smith. *Preparing fMRI data for statistical analysis. In Jezzard P, Matthews P.M., Smith S.M. (Eds.), Functional MRI: an introduction to methods*. Oxford University Press, Oxford, UK., 2003.

[29] E. Zarahn, G. K. Aguirre, and M. D'Esposito. Empirical analyses of BOLD fMRI statistics. I. Spatially unsmoothed data collected under null-hypothesis conditions. *NeuroImage*, 5:179–197, 1997.

[30] C.A. Cocosco, A.P. Zijdenbos, and A.C. Evans. A fully automatic and robust brain MRI tissue classification method. *Medical Image Analysis*, 7:513ąV527, 2003.

[31] P.S. Mitra, V. Gopalakrishnan, and R.L. McNamee. Segmentation of fMRI data by maximization of region contrast. *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 88, 2006.

[32] C. Pluempitiwiriyawej, J. M. F. Moura, Y.-J. L. Wu, and C. Ho. A new active contour scheme for cardiac MR image segmentation. *IEEE Transactions on Medical Imaging*, 24:593ąV603, 2005.

[33] A.M. Dale, B. Fischl, and M.I. Sereno. Cortical surface-based analysis. I. segmentation and surface reconstruction. *Neuroimage*, 9:179–194, 1999.

[34] E. Salli, A. Visa, H.J. Aronen, A. Korvenoja, and T. Katila. Proc. of the 2nd international conference on medical image computing and computer assisted intervention - miccai'99. c. taylor, a. colchester (eds.). lecture notes in computer science. *Springer-Verlag*, 1679:481–488, 1999.

[35] J. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer., 2001.

[36] C.R. Robert and G. Casella. *Monte Carlo Staistical Methods*. Springer., 2004.

[37] J. M. Flegal, M. Haran, and G.L. Jones. Markov chain Monte Carlo: Can we trust the third significant figure. *Statistical Science*, 23:250–260, 2008.

[38] G.L. Jones, M. Haran, , B.S. Caffo, and R. Neath. Fixed-width output analysis for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 101:1537–1547, 2006.

[39] M.H. Chen, Q.M. Shao, and J.G. Ibrahim. *Monte Carlo Methods in Bayesian Computation*. Springer, 2000.

[40] G.L. Jones and J.P. Hobert. Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Statistical Science*, 16:312–334, 2001.

[41] A. Gelman. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13:163–185, 1998.

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

[43] J. M. Flegal, M. Haran, and G.L. Jones. Batch means and spectral variance estimators in Markov chain Monte Carlo. *Annals of Statistis*, 38:1034–1070, 2010.

[44] C.J. Geyer. Practical Markov chain Monte Carlo. *Statistical Science*, 7:473–511, 1992.

[45] M.S. Meketon and B. Schmeiser. Overlapping batch means: something for nothing? *Proceedings of the 16th conference on Winter simulation*, pages 227–230, 1984.

[46] I. Murray, Z. Ghahramani, and D.J.C. MacKay. MCMC for doubly-intractable distributions. In *UAI*. AUAI Press, 2006.

[47] G.M. Torrie and J.P. Valleau. Nonphysical sampling distribution in Monte Carlo free energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23:187–199, 1977.

[48] F. Wang and D.P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 81:2050 – 2053, 2001.

[49] V. Plagnol and S. Tavaré. Approximate Bayesian computation and MCMC. *Monte Carlo and quasi-Monte Carlo methods 2002*, pages 99–113, 2004.

[50] C. Zhang and J. Ma. Simulation via direct computation of partition functions. *Physical Review Letters E*, 76:036708–1–5, 2007.

[51] A. Gelman and X. Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13:163–185, 1998.

[52] J.A. Fill. An interruptible algorithm for perfect sampling via Markov chains. *Annals Applied Probabability*, 8:131–162, 1998.

[53] J.G. Propp and D.B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.

[54] F. Wang and D.P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 81:2050 – 2053, 2001.

[55] Faming Liang. A generalized wang-landau landau algorithm for monte carlo computation. *Journal of the American Statistical Association*, 100:1311–1327, 2005.

[56] B.A. Berg. Multicanonical simulations step by step. *Computer Physics Communications*, 153:397–406(10), 2003.

[57] W. Janke. Multicanonical monte carlo simulation. *Physica A*, 254:164–178, 1998.

[58] F. Liang. A theory on flat histogram monte carlo algorithms. *Journal of Statistical Physics*, 122:511–529, 2006.

[59] J.S. Wang. Flat histogram monte carlo method. *Physica A*, 281:147–150(4), 2000.

[60] D.P. Landau, S.H. Tsai, and M. Exler. A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling. *American Journal of Physics*, 72:1294–1302, 2004.

[61] C. Zhou and R.N. Bhatt. Understanding and improving the Wang-Landau algorithm. *Physical Review Letters E*, 72:025701–1–4, 2005.

[62] N. Lartillot and H. Philippe. Computing Bayes factor using thermodynamic integration. *Systematic Biology*, 55:195–207, 2006.

[63] J. Möller, A. N. Pettitt, R. Reeves, and K. K. Berthelsen. An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93:451–458, 2006.

[64] P. Marjoram, J. Molitor, V. Plagnol V, and S. Tavare. Markov chain monte carlo without likelihoods. *Proc. Natl. Acad. Sci. USA*, 100:(26):15324–8, 2003.

[65] S.A. Sisson and Y. Fan M.M. Tanaka. Sequential Monte Carlo without likelihoods. *Proc. Natl. Acad. Sci. USA*, 104:(6):1760–1765, 2007.

[66] G.O. Roberts and J. S. Rosenthal. Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18:349–367, 2009.

[67] K.J. Worsley, S. Marrett, P. Neelin, and A.C. Evans. A three-dimensional statistical analysis for CBF activation studies in human brain. *Journal of Cerebral Blood Flow and Metabolism*, 12:900–918, 1992.

[68] C.R. Genovese. A Bayesian Time-Course Model for Functional Magnetic Resonance Imaging Data. *Journal of the American Statistical Association*, 95:691–703, 2000.

[69] C. Gössl, L. Fahrmeir, and D.P. Auer. Bayesian Spatiotemporal Inference in Functional Magnetic Resonance Images. *Biometric*, 57:554–562, 2001.

[70] M.W. Woolrich, M. Jenkinson, J.M. Brady, and S.M. Smith. Fully Bayesian Spatio-Temporal Modeling for fMRI Data. *IEEE Transactions on Medical Imaging*, 23:213–231, 2004.

[71] W. D. Pennya, N. J. Trujillo-Barretob, and K. J. Friston. Bayesian fMRI Time Series Analysis with Spatial Priors. *NeuroImage*, 24:350–362, 2005.

[72] A. Quirós, R.M. Dieza, and D. Gamerman. Bayesian Spatiotemporal Model of fMRI Data. *NeuroImage*, 49:442–456, 2009.

[73] F.D. Bowman. Spatiotemporal Models for Region of Interest Analyses of Functional Neuroimaging Data. *Journal of the American Statistical Association*, 102:442–453, 2005.

[74] A. Zellner. On assessing prior distributions and Bayesian regression analysis with g-prior distributions. *In Bayesian Inference and Decision Techniques: Essays in Honor of Brunode Finetti North-Holland/Elsevier*, page 233ąV243, 1996.

[75] M.-H. Chen, J.G. Ibrahim, and C. Yiannoutsos. Prior Elicitation, Variable Selection and Bayesian Computation for Logistic Regression Models. *Journal of the Royal Statistical Society: Series B*, 61:223–242, 1999.

[76] E.I. George. The Variable Selection Problem. *Journal of the American Statistical Association*, 95:1304ąV1308, 2000.

[77] E.I. George and R.E. McCulloch. Approaches for Bayesian Variable Selection. *Statistica Sinica*, 7:339ąV374, 1997.

[78] R. Kohn, M. Smith, and D. Chan. Nonparametric regression using linear combinations of basis functions. *Statistics and Computing*, 11:313–322, 2001.

[79] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 36:192–225, 1974.

[80] I. Weir and A. Pettit. Binary probability maps and using a hidden conditional autoregression Guassian process with an application to Finnish common toad data. *Applied Statistics*, 49:473–484, 2000.

[81] J. Besag and C. Kooperberg. On conditional and intrinsic autoregressions. *Biometrika*, 82:733–746, 1995.

[82] A. Pettit. A conditional autoregressive Gaussian process for irregularly spaced multivariate data with application to modelling large sets of binary data. *Statistics and Computing*, 12:353–367, 2002.

[83] K.J. Friston, P.J. Jezzard, and R Turner. Analysis of functional mri time-series. *Human Brain Mapping*, 1:153–171, 1994.

[84] E. Ising. Beitrag zur theorie des ferromagnetismus. *Z Physik*, 31:253–258, 1925.

Appendix A

R Packages

Download two R-packages, NCIsing and SpBVS.fMRI, from the following links

1. http://www.stat.umn.edu/∼kjlee/R-Packages/NCIsing_1.0.tar.gz
2. http://www.stat.umn.edu/∼kjlee/R-Packages/SpBVS.fMRI_1.0.tar.gz

A.1 R-package 'NCIsing': Estimation of the Normalizing Constant in an Ising Model

In order to compare the accuracy and speed of the algorithms described above, we perform a Monte Carlo simulation for an Ising model [84] with free boundary condition on a 32×32 square-lattice E. Let $\mathbf{y} = \{y_1, \ldots, y_n\}$ be a data on E with y_i is represented by a binary indicator, $+1$ and -1 and assume a uniform prior $\varphi(\theta)$ on θ. The posterior distribution is characterized by

$$p(\theta|\mathbf{y}) = \frac{1}{Z(\theta)} \exp\left\{ -\theta \sum_{i \sim j} [w_{i,j} \times 1(y_i = y_j)] \right\} \varphi(\theta), \tag{A.1}$$

where $i \sim j$ denotes the nearest-neighbors between i and j and $w_{i,j}$ is prespecified weights. In this simulation study, we assumed $w_{i,j} = 1$ and θ is restricted over the interval $[0, 1]$. There are two different algorithms, coupling from the past (CFTP) [53] and Fill's interruptible algorithm [52], in perfect sampling method to exactly simulate sample from an

Ising model. Two 32×32 2-dimensional Ising models generated using CFTP Fill's interruptible algorithm are shown in Figure A.1, respectively. Figure A.2 gives Ising models corresponding to different spatial coefficient θ generated using CFTP. Next we introduce how to reproduce the outputs.

Download the NCIsing R-package from
http://www.stat.umn.edu/∼kjlee/R-Packages/NCIsing_1.0.tar.gz and then install the package. After including the library "NCIsing", for a given $\theta = 0.3$, the two algorithms are used to generate Ising images.

```
> library("NCIsing")
> n = 32


> par(mfrow = c(1, 2))
> theta = 0.3
> IsingImage = IsingImageGeneration(theta, n, "CFTP")
> image(IsingImage, main = "Coupling from the past", axes = FALSE,
+     frame = T)
> IsingImage = IsingImageGeneration(theta, n,
+     "Fill's interruptible algorithm")
> image(IsingImage, main = "Fill's interruptible algorithm",
+     axes = FALSE, frame = T)
```

Here we perform a simulation to see the difference of Ising models in different values of θ.

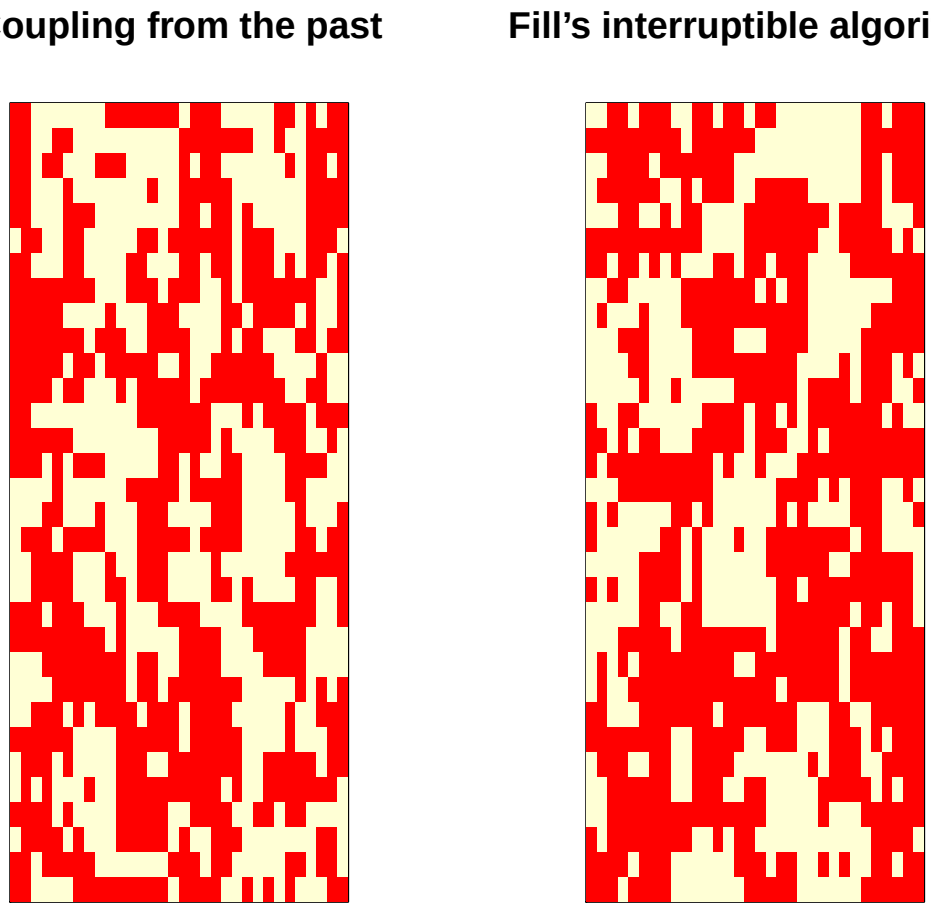


Figure A.1: Given a $\theta = 0.3$, 32×32 2D Ising images are generated using the perfect sampling with different algorithms.

```
> par(mfrow = c(1, 3))
> theta = 0.1
> IsingImage = IsingImageGeneration(theta, n, "CFTP")
> image(IsingImage, main = expression(paste(theta, "=0.1")),
+     axes = FALSE, frame = T)
> theta = 0.3
> IsingImage = IsingImageGeneration(theta, n, "CFTP")
> image(IsingImage, main = expression(paste(theta, "=0.3")),
+     axes = FALSE, frame = T)
> theta = 0.45
> IsingImage = IsingImageGeneration(theta, n, "CFTP")
> image(IsingImage, main = expression(paste(theta, "=0.45")),
+     axes = FALSE, frame = T)
```

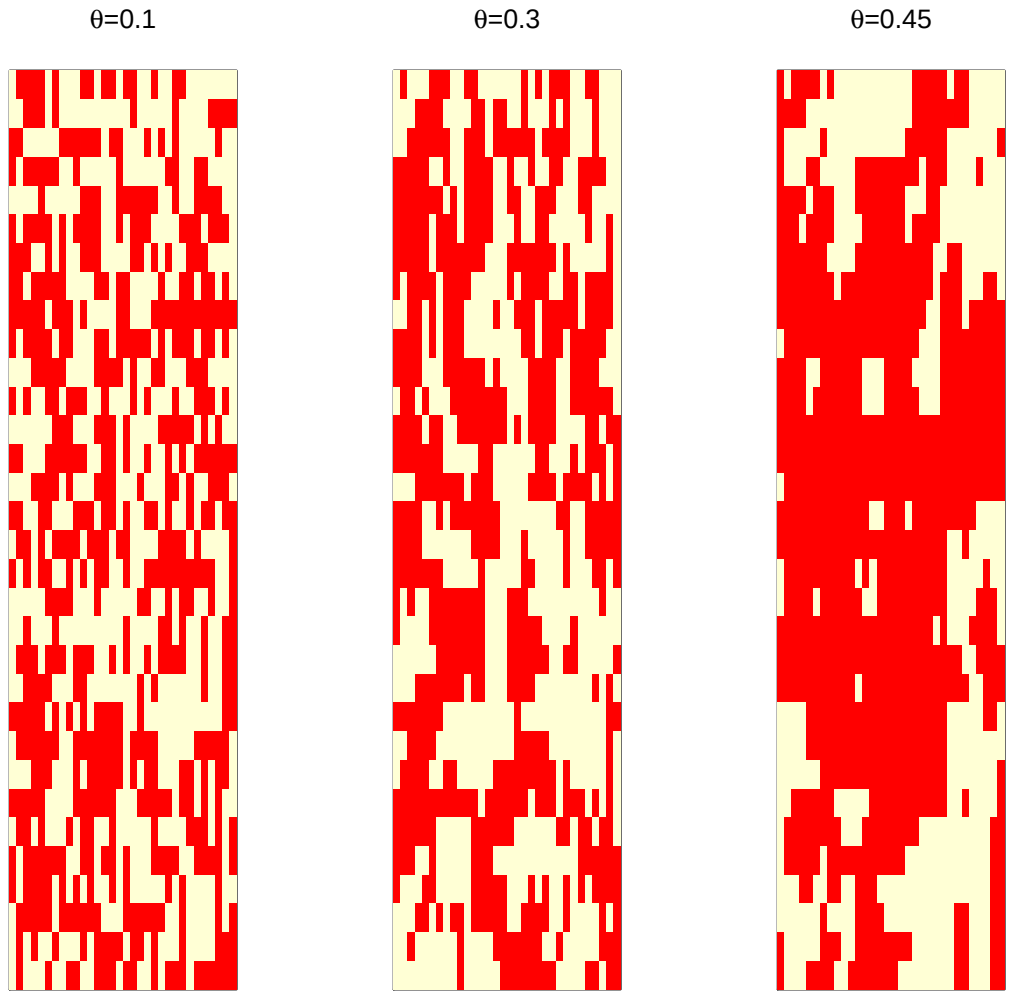


Figure A.2: 32×32 2D Ising images with different values of θ generated using the perfect sampling.

Let $D(\mathbf{y}) = \sum_{i \sim j} [w_{i,j} \times 1(y_i = y_j)]$. Then, the normalizing constant can be expressed as

$$Z(\theta) = \left(\sum_{\mathbf{y} \in \Upsilon} \exp\{-\theta \cdot D(\mathbf{y})\} \right)^{-1},$$

where Υ is the number of all possible outcomes in the configuration of $\mathbf{y}$. To simulate a sample from (A.1), the normalizing constant $Z(\theta)$ is necessarily to be evaluated in advance. It's, however, extremely difficult to compute the normalizing constant $Z(\theta)$ because it is required to make a summation over all possible values of $\mathbf{y}$, which involve 2^n terms where n is the number of data points. Therefore, we use the algorithms mentioned above to evaluate the normalizing constant before performing Gibbs sampler to generate the sample from A.1.

Given a $\theta = 0.3$, a data $\mathbf{y}$, i.e., an Ising image, is generated using perfect sampling [53]. We carry out different Monte Carlo algorithms to simulate the spatial coefficient θ, then use batch mean methodology [38] to calculate the estimates of θ and corresponding Monte carlo standard error (MCSE).

In NCIsing R-package, there are 6 approaches to estimate the parameter, θ, when given an Ising model. These 6 approaches are path sampling [41], Wang-Landau [48], modified Wang-Landau [50], umbrella sampling [47], single variable exchange (SVE) [46], and approximate Bayesian computation (ABC) [49] algorithms where SVE and ABC don't provide the estimate of the logarithm of the normalizing constant. The following are R codes to estimate the spatial coefficient θ for a given Ising image.

Path Sampling algorithm:

```
> library(NCIsing)
> theta = 0.3
> IsingImage = IsingImageGeneration(theta, n, "CFTP")
> iteration = 2000
> num.of.points.shown.trace.plot = 999
> start.time = proc.time()
> Output = SpatialCoefGenerator(IsingImage, "Path Sampling",
+       iteration)
> samples = Output$theta
```

```
> logZ = Output$logZ
> end.time = proc.time()
> time.elapse = end.time - start.time
> time.elapse
   user  system elapsed
 73.765   0.020  73.783
> id <- function(x) return(x)
> bm(samples)
$est
[1] 0.3086799

$se
[1] 0.0008207145

$bs
[1] "sqroot"


> par(mfrow = c(2, 2))
> plot(logZ, type = "l", col = 1, lwd = 2, lty = 1,
+     xlab = expression(theta), ylab = "Log Z", main = "")
> plot((iteration - num.of.points.shown.trace.plot):iteration,
+   samples[(iteration - num.of.points.shown.trace.plot):iteration],
+   main = "Trace plot", xlab = "Iteration",
+   ylab = expression(theta), type = "b")
> hist(samples, xlab = expression(theta), main = expression(paste(
+     "Histogram of ", theta)))
> acf(samples)
```

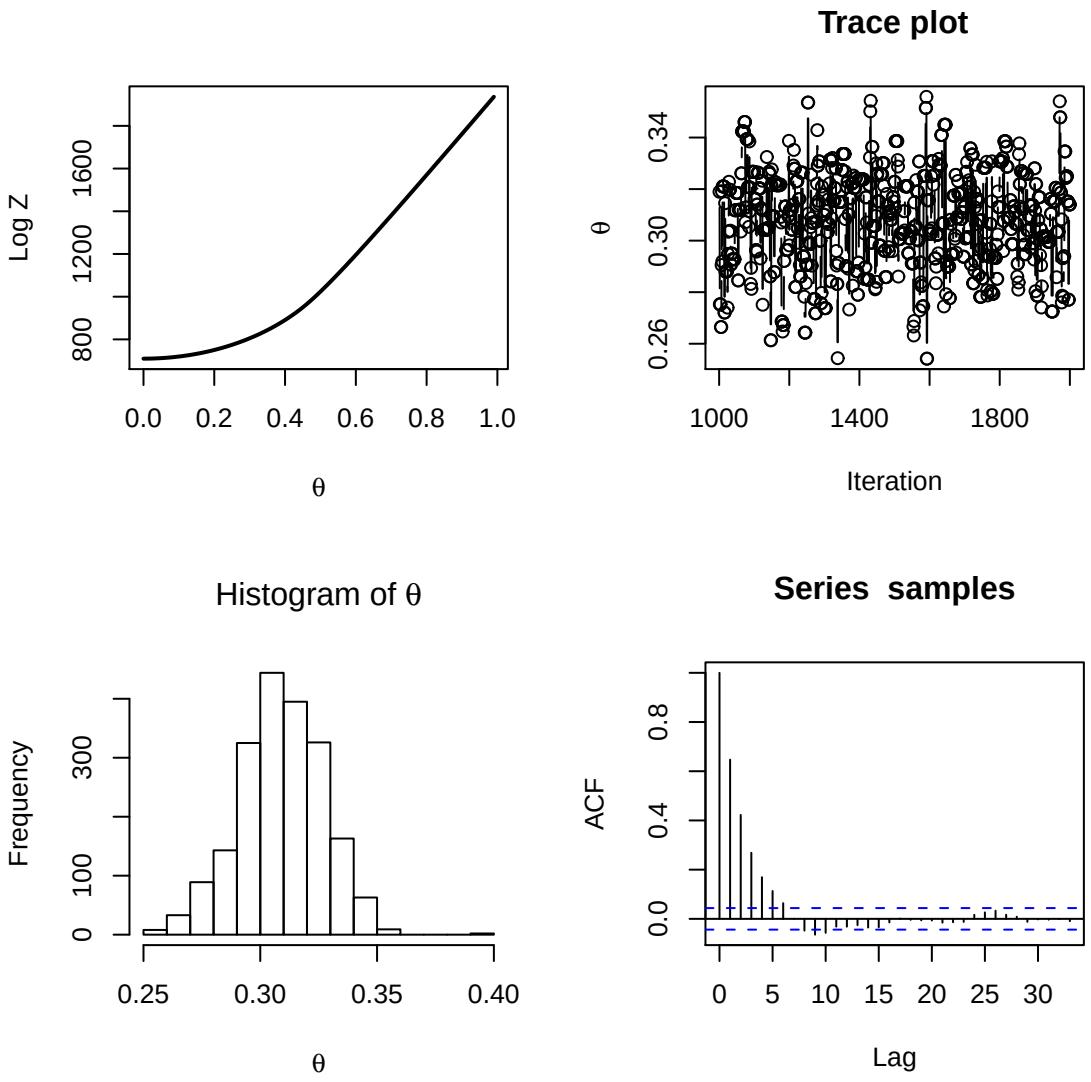


Figure A.3: Estimate of logarithm of the normalizing constant, trace plot, histogram and ACF plots for the sample generated using path sampling.

Wang Landau Algorithm:

```
> start.time = proc.time()
> Output = SpatialCoefGenerator(IsingImage, "Wang Landau Algorithm",
+     iteration)
> samples = Output$theta
> logZ = Output$logZ
> end.time = proc.time()
> time.elapse = end.time - start.time
> time.elapse
   user  system elapsed
 49.171   0.000  49.172
> id <- function(x) return(x)
> bm(samples)
$est
[1] 0.3052621

$se
[1] 0.0008132801

$bs
[1] "sqroot"


> par(mfrow = c(2, 2))
> plot(logZ, type = "l", col = 1, lwd = 2, lty = 1,
+     xlab = expression(theta), ylab = "Log Z", main = "")
> plot((iteration - num.of.points.shown.trace.plot):iteration,
+   samples[(iteration - num.of.points.shown.trace.plot):iteration],
+   main = "Trace plot", xlab = "Iteration", ylab = expression(theta),
+   type = "b")
> hist(samples, xlab = expression(theta), main = expression(paste(
```

```
+     "Histogram of ", theta)))
> acf(samples)
```

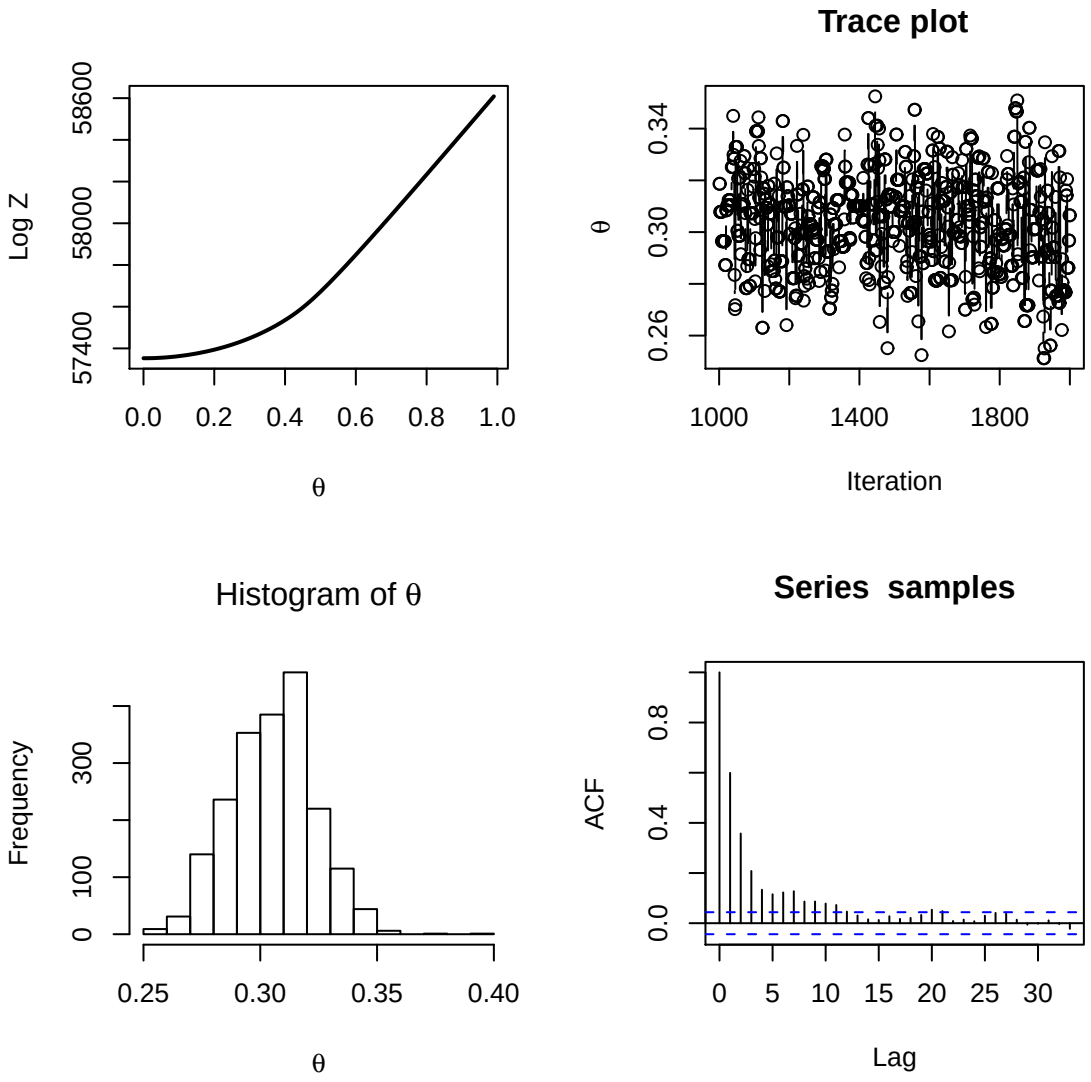


Figure A.4: Estimate of logarithm of the normalizing constant, trace plot, histogram and ACF plots for the sample generated using the modified Wang Landau algorithm. The plot of "Estimate of log Z" shows the corresponding estimate of logarithm of the normalizing constant at different values θ in [0, 1] up to a constant. Only the last 1000 sample points are shown in the trace plot.

Modified Wang Landau Algorithm:

```
> start.time = proc.time()
> Output = SpatialCoefGenerator(IsingImage,
+     "Modified Wang Landau Algorithm", iteration)
> samples = Output$theta
> logZ = Output$logZ
> end.time = proc.time()
> time.elapse = end.time - start.time
> time.elapse
   user  system elapsed
 20.513   0.000  20.517
> id <- function(x) return(x)
> bm(samples)
$est
[1] 0.3051080

$se
[1] 0.0008987626

$bs
[1] "sqroot"

> par(mfrow = c(2, 2))
> plot(logZ, type = "l", col = 1, lwd = 2, lty = 1,
+     xlab = expression(theta), ylab = "Log Z",
+     main = "Estimate of log Z")
> plot((iteration - num.of.points.shown.trace.plot):iteration,
+   samples[(iteration - num.of.points.shown.trace.plot):iteration],
+   main = "Trace plot", xlab = "Iteration", ylab = expression(theta),
+   type = "b")
> hist(samples, xlab = expression(theta), main = expression(paste(
```

```
+     "Histogram of ", theta)))
> acf(samples)
```

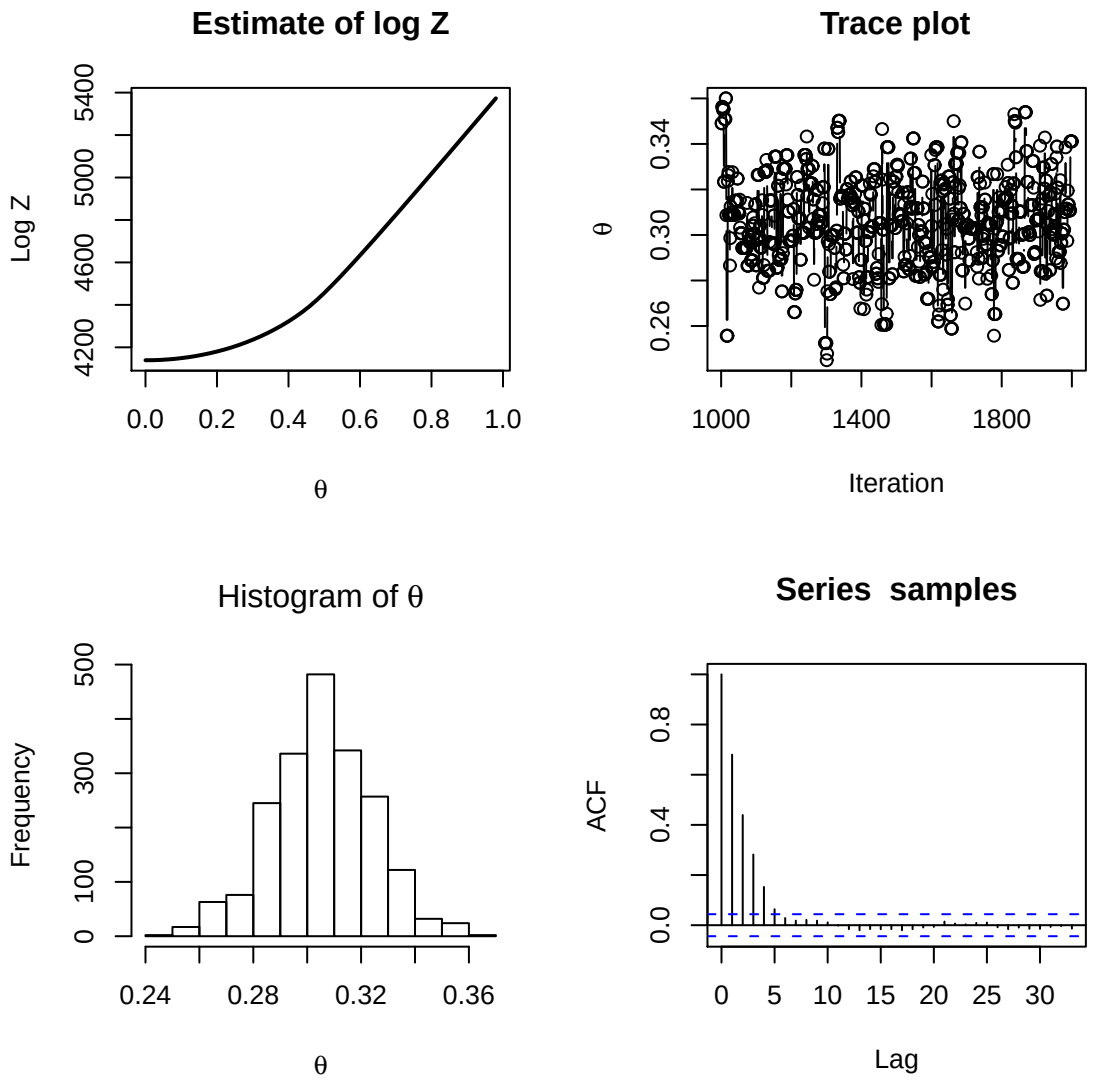


Figure A.5: Estimate of logarithm of the normalizing constant, trace plot, histogram and ACF plots for the sample generated using the modified Wang Landau algorithm. The plot of "Estimate of log Z" shows the corresponding estimate of logarithm of the normalizing constant at different values θ in [0, 1] up to a constant. Only the last 1000 sample points are shown in the trace plot.

Single Variable Exchange Algorithm:

```
> start.time = proc.time()
> samples = SpatialCoefGenerator(IsingImage,
+     "Single Variable Exchange Algorithm", iteration)
> end.time = proc.time()
> time.elapse = end.time - start.time
> time.elapse
   user  system elapsed
 77.981   0.020  78.015
> id <- function(x) return(x)
> bm(samples)
$est
[1] 0.3041708

$se
[1] 0.001097946

$bs
[1] "sqroot"


> par(mfrow = c(3, 1))
> plot((iteration - num.of.points.shown.trace.plot):iteration,
+   samples[(iteration - num.of.points.shown.trace.plot):iteration],
+   main = "Trace plot", xlab = "Iteration", ylab = expression(theta),
+   type = "b")
> hist(samples, xlab = expression(theta), main = expression(paste(
+     "Histogram of ",theta)))
> acf(samples)
```

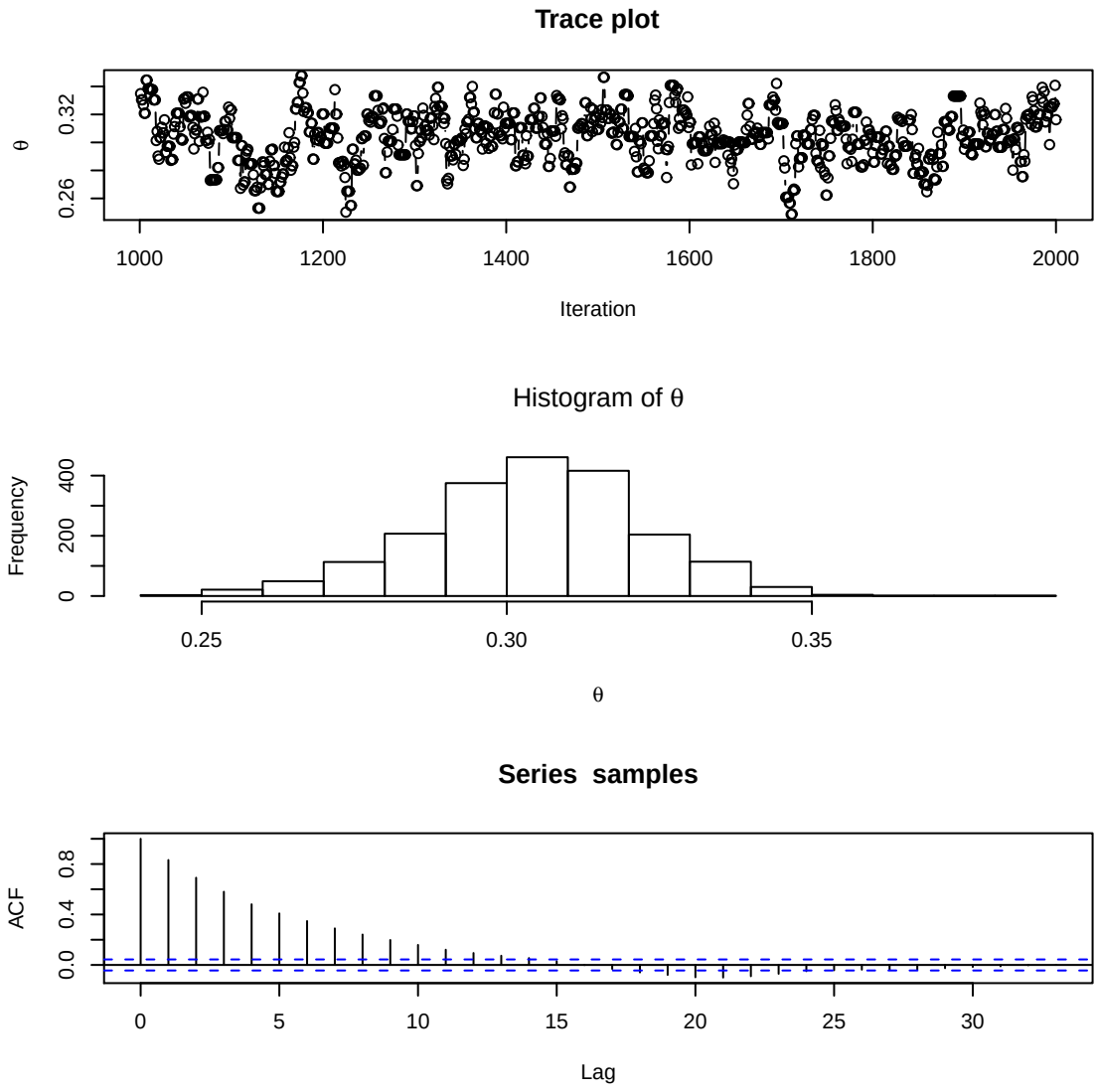


Figure A.6: Trace plot, histogram and ACF plots for the simulated sample using single variable exchange algorithm.

Approximate Bayesian Computation:

```
> start.time = proc.time()
> samples = SpatialCoefGenerator(IsingImage,
+  "Approximate Bayesian Computation", iteration)
> end.time = proc.time()
> time.elapse = end.time - start.time
> time.elapse
   user  system elapsed
  2.656   0.000   2.657
> id <- function(x) return(x)
> bm(samples)
$est
[1] 0.2913715

$se
[1] 0.004064412

$bs
[1] "sqroot"


> par(mfrow = c(3, 1))
> plot((iteration - num.of.points.shown.trace.plot):iteration,
+   samples[(iteration - num.of.points.shown.trace.plot):iteration],
+   main = "Trace plot", xlab = "Iteration", ylab = expression(theta),
+   type = "b")
> hist(samples, xlab = expression(theta), main = expression(paste(
+     "Histogram of ",theta)))
> acf(samples)
```

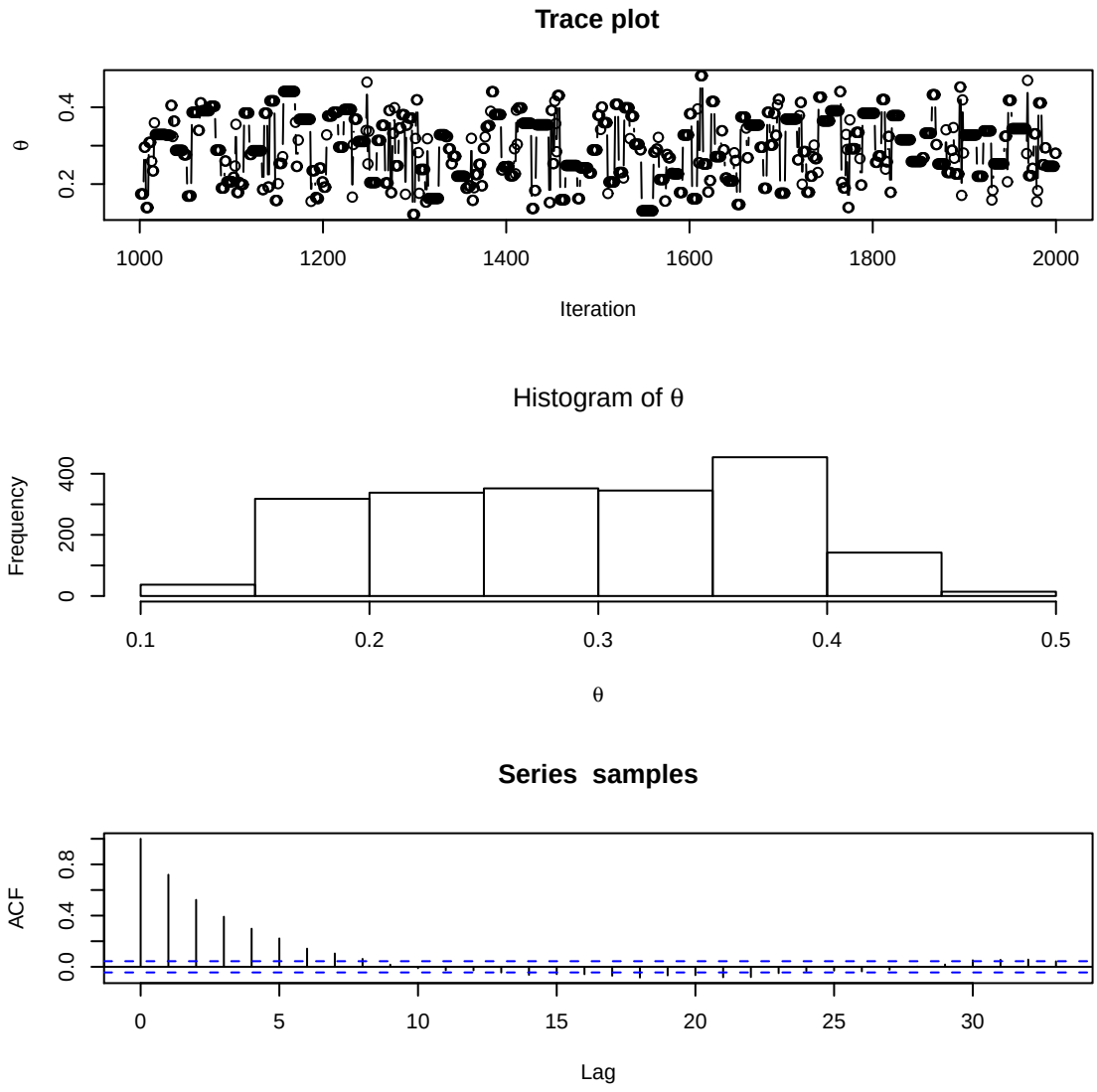


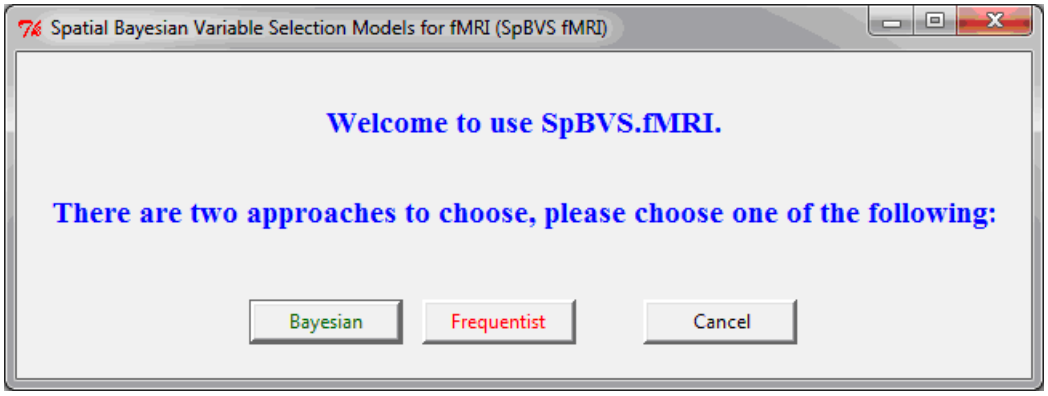
Figure A.7: Trace plot, histogram and ACF plots for the simulated sample using the approximate Bayesian computation algorithm.

A.2 R-package 'fMRI.SpBVS': Spatial Bayesian Variable Selection for fMRI Time Series Data.

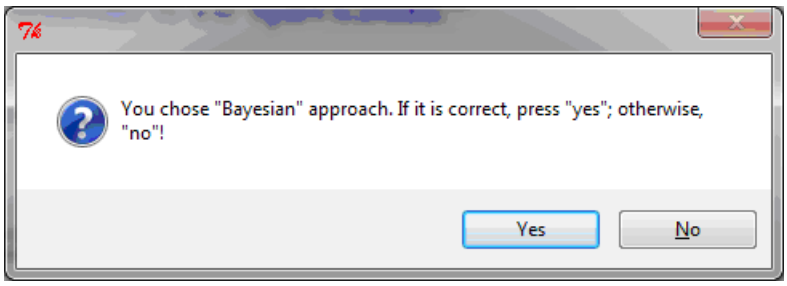
Available at http://www.stat.umn.edu/∼kjlee/

- Author: Kuo-Jung Lee kjlee@stat.umn.edu, Galin Jones galin@stat.umn.edu,
- Maintainer: Kuo-Jung Lee kjlee@stat.umn.edu

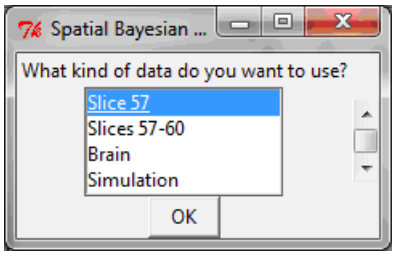
Two approaches are implemented to analyze the fMRI time-series data. Frequentist approaches are still under construction and coming soon.

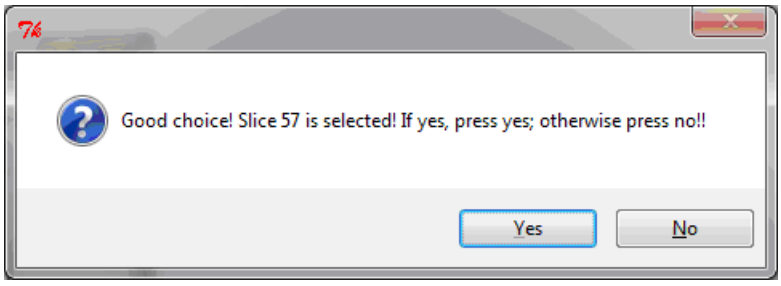


After pressing button "Bayesian" it will show the following image.

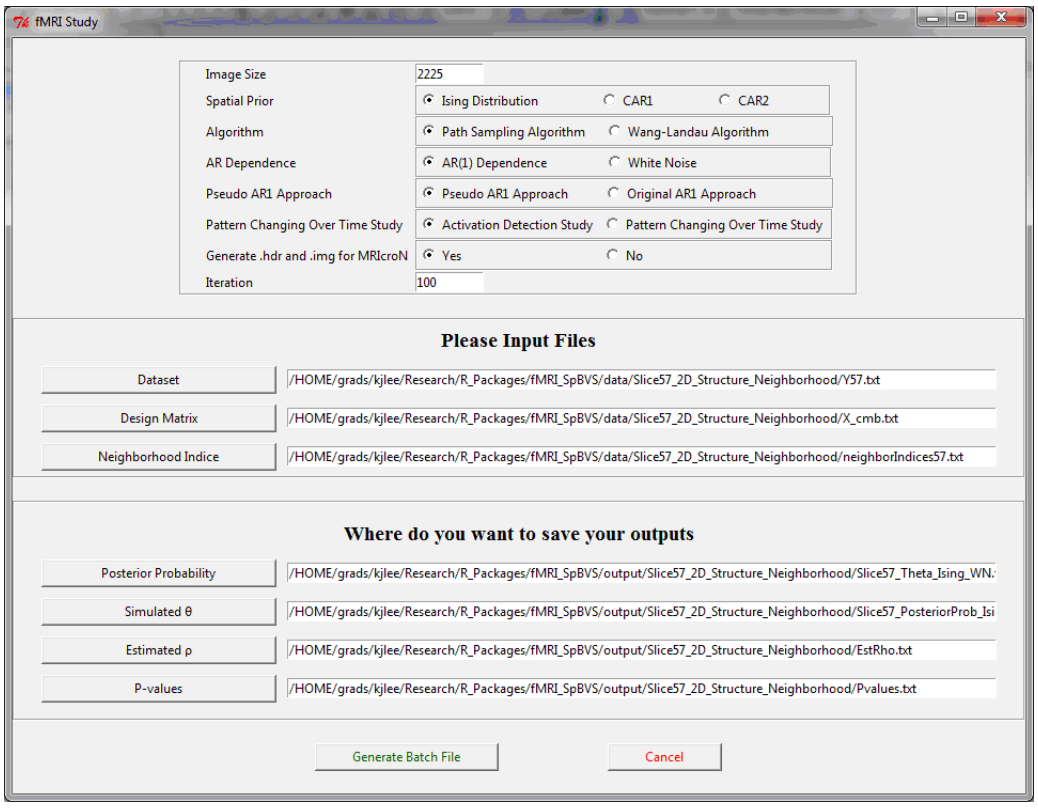


If it is correct, just press "Yes" and you'll be asked to choose one dataset for the following analysis. "Slice 57" means only voxels in Slice 57, "Brain" whole voxels in the brain, "Parietal_Inf_L" voxles in the left inferior parietal lobe and etc. are going to be analyzed. We have 7 different datasets so far, we'll add more dataset regarding to different regions of interest.

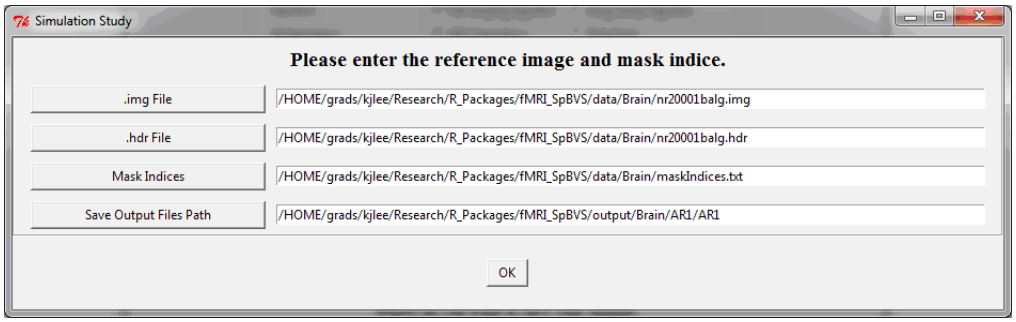




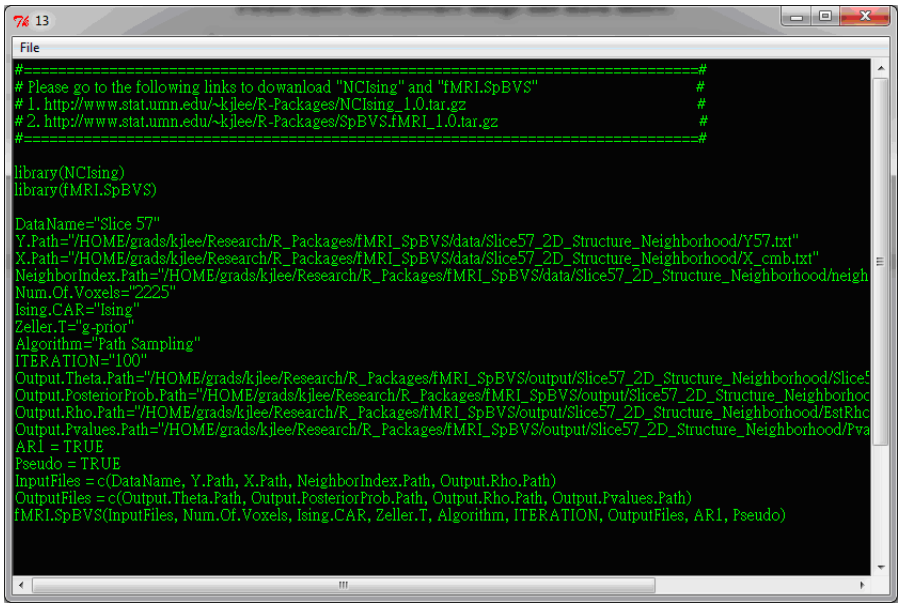
If you choose "Slice 57" and it is correct, please press yes and it'll show:



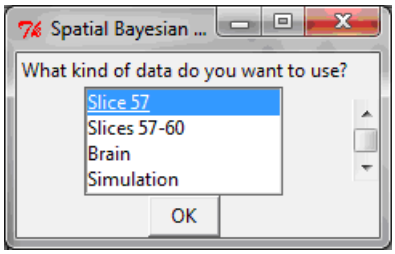
Enter all information needed for analysis, then press "Generate Batch File" then it'll show the following image if you choose to generate .img and .hdr data format for MRIcroN software to look at 3-dimensional images.



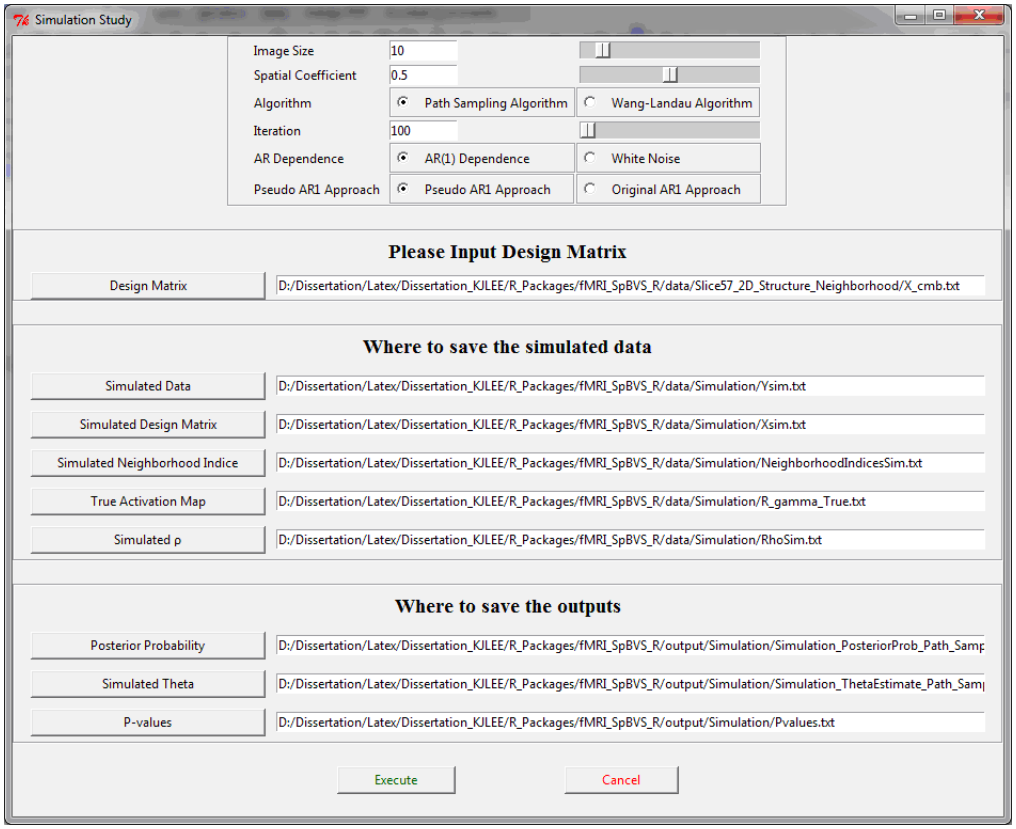
Then it'll generate a R batch file and just copy the following code to R and run.



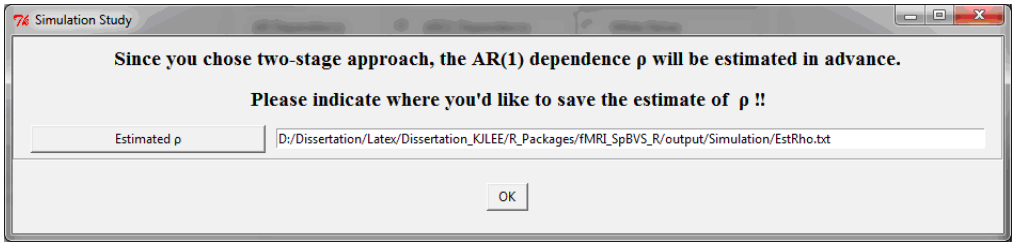
If it is correct, just press "Yes" and you'll be asked to choose one dataset for the following analysis. "Slice 57" means only voxels in Slice 57, "Brain" whole voxels in the brain, "Parietal_Inf_L" voxles in the left inferior parietal lobe and etc. are going to be analyzed. We have 7 different datasets so far, we'll add more dataset regarding to different regions of interest.

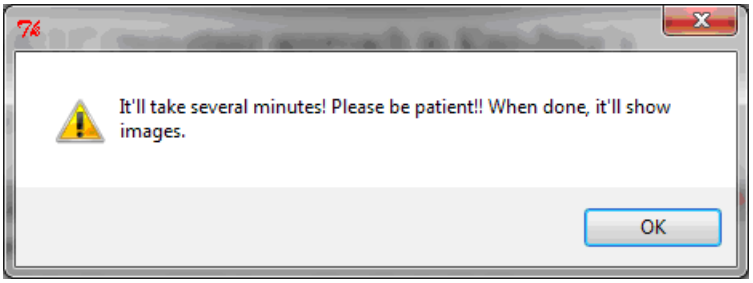


If you choose "Simulation," it'll show the following image. Then enter all necessary information and press "Execute."

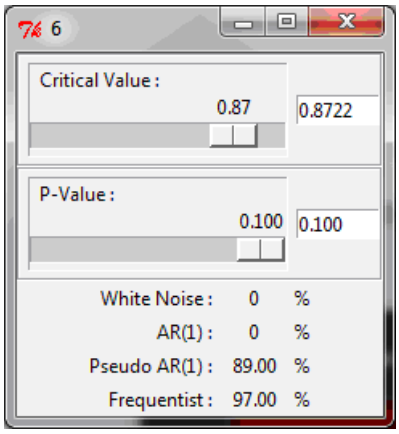


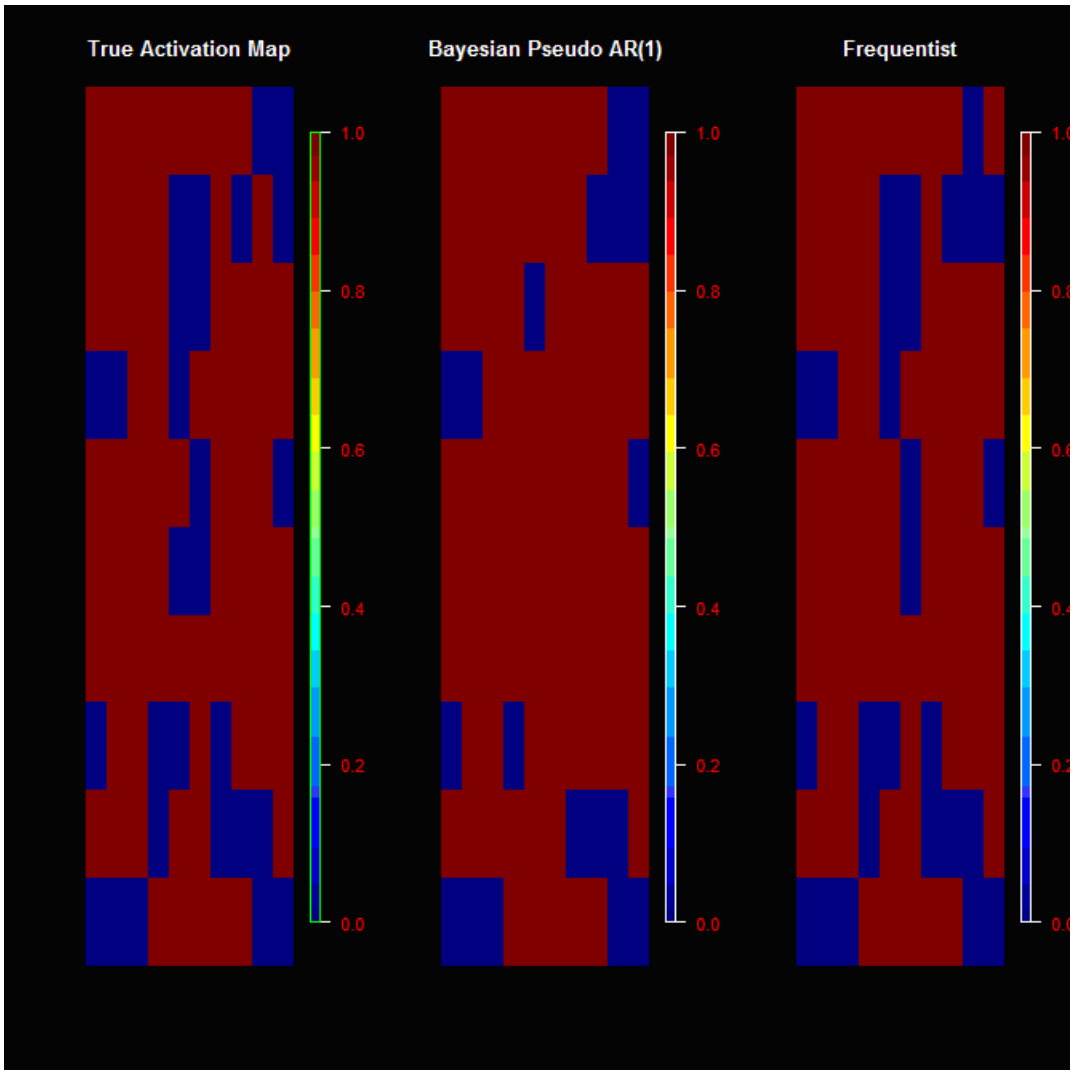
If you choose "Pseudo AR(1)" (two-stage estimation procedure in Kuo-Jung's dissertation), you have to save the ρ.

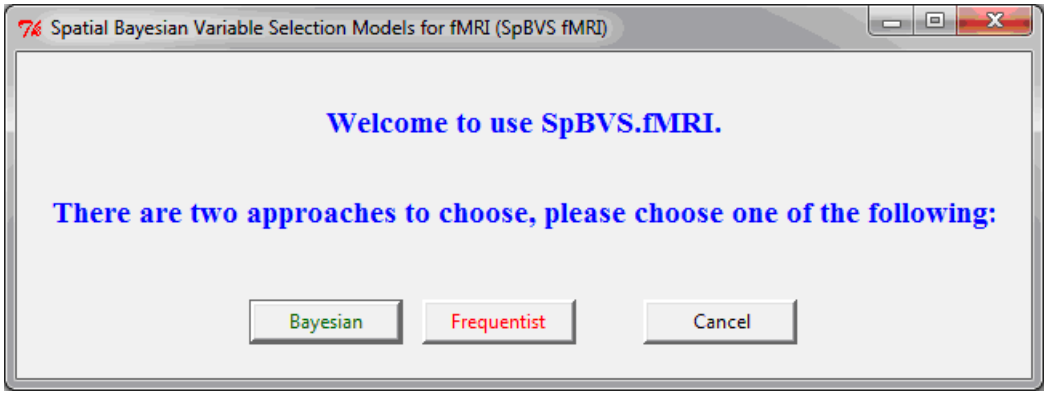


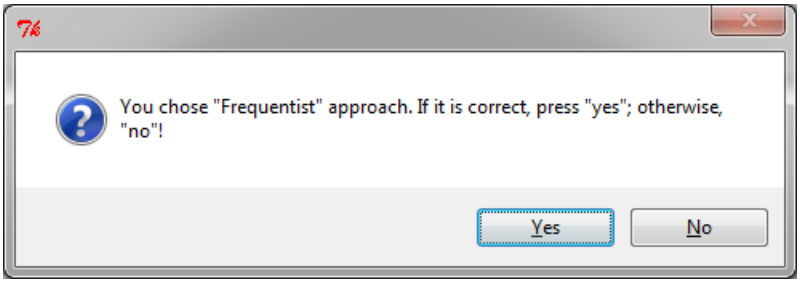


Then press "OK" and wait for output. When it's done, it'll show activation images and a corresponding box where you could choose critical values to see the difference of activation images.

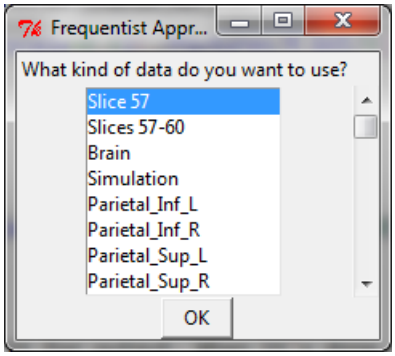


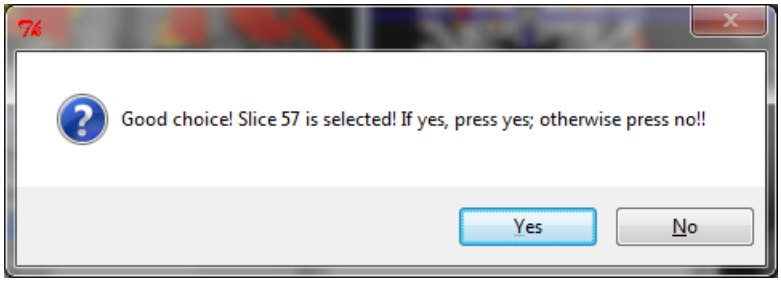

True Activation Map
Bayesian Pseudo AR(1)
Frequentist
1.0
0.8
0.6
0.4
0.2
0.0

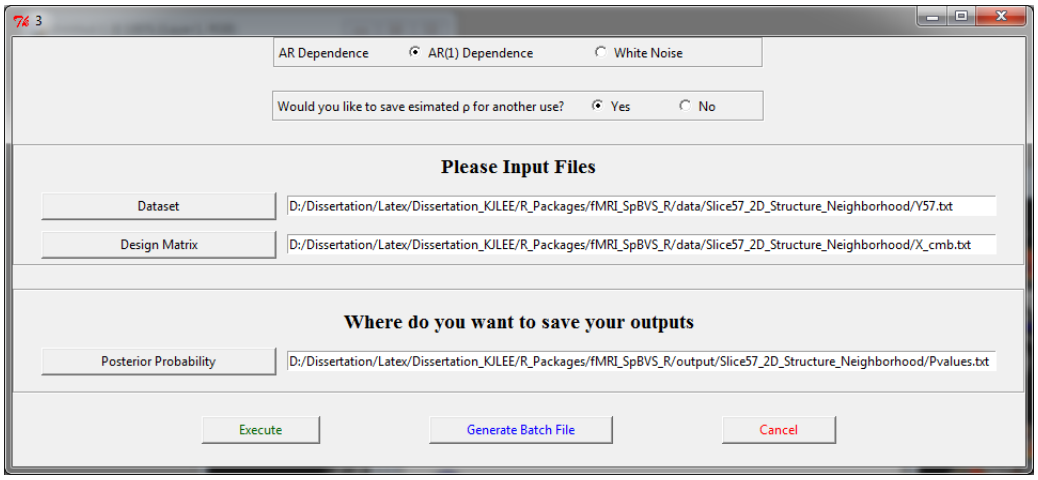




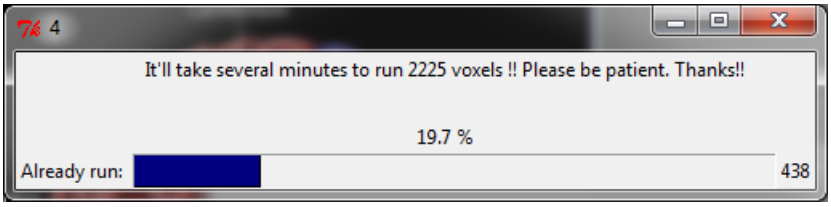
If you choose the frequentist approach and it is correct, press "yes" and the following image will show. Choose one of dataset for the following analysis and press "OK." After press "yes," then another dialog will show up and please input your data and indicate where you'd like to save the output.

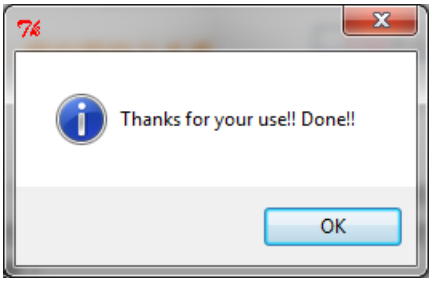

Frequentist Appr...
What kind of data do you want to use?
Slice 57
Slices 57-60
Brain
Simulation
Parietal_Inf_L
Parietal_Inf_R
Parietal_Sup_L
Parietal_Sup_R
OK


Good choice! Slice 57 is selected! If yes, press yes; otherwise press no!!
Yes
No

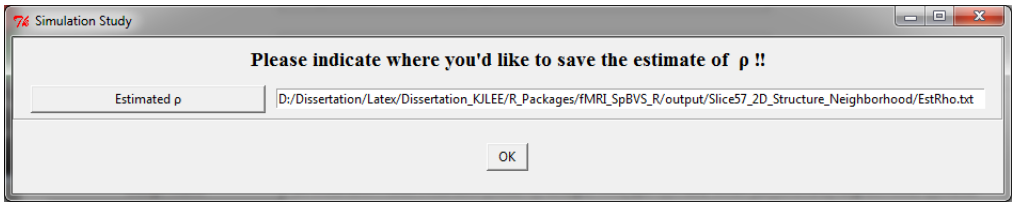

3
AR Dependence
AR(1) Dependence
White Noise
Would you like to save esimated ρ for another use?
Yes
No
Please Input Files
Dataset
D:/Dissertation/Latex/Dissertation_KJLEE/R_Packages/fMRI_SpBVS_R/data/Slice57_2D_Structure_Neighborhood/Y57.txt
Design Matrix
D:/Dissertation/Latex/Dissertation_KJLEE/R_Packages/fMRI_SpBVS_R/data/Slice57_2D_Structure_Neighborhood/X_cmb.txt
Where do you want to save your outputs
Posterior Probability
D:/Dissertation/Latex/Dissertation_KJLEE/R_Packages/fMRI_SpBVS_R/output/Slice57_2D_Structure_Neighborhood/Pvalues.txt
Execute
Generate Batch File
Cancel

If you choose "White Noise" in the "AR Dependence" dialog and then press "Execute" it will show you how much percentage is done.

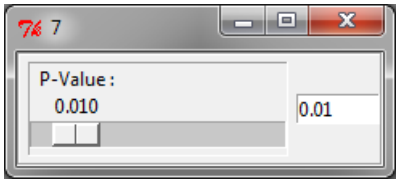


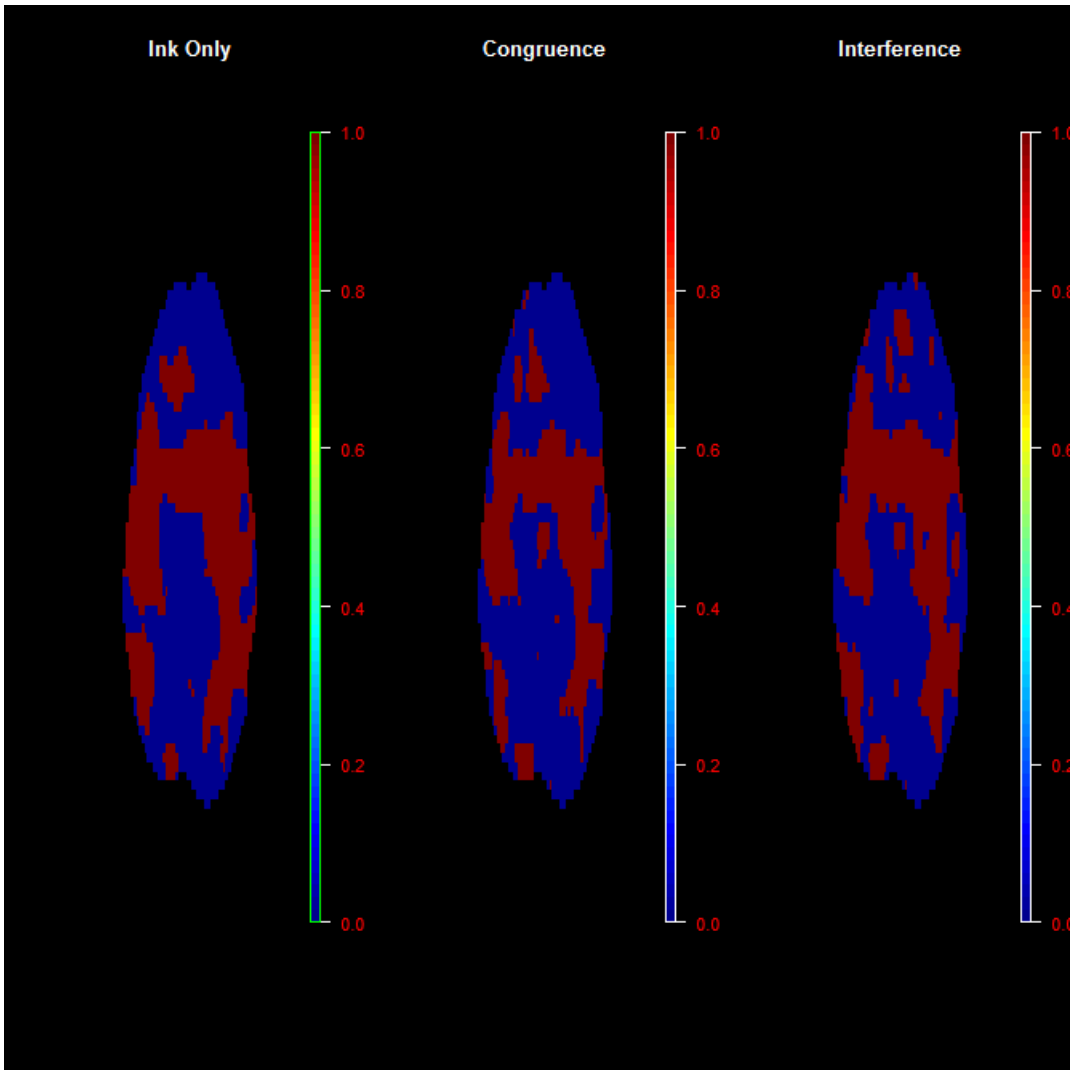


Otherwise, if you choose "AR1" in the "AR Dependence" dialog and "yes" in "Would you like to save the estimate ρ for another use?" another dialog will shown up to asked where you want to save the estimates.



After it's done, a box is shown up where you can choose the critical value and the corresponding activation images will be shown.




Ink Only
Congruence
Interference
1.0
0.8
0.6
0.4
0.2
0.0

Once you would deal with huge data, it is better to run in the background. To generate the **R** code, just press "Generate Batch File" then it gives you the code.