Denoising Large Neuroimage MRI Data Using Spatial Random Effect Models

Leonard Chukuma Johnson

Western Michigan University, jleochuk@gmail.com

Follow this and additional works at: https://scholarworks.wmich.edu/dissertations

Part of the Statistics and Probability Commons

Recommended Citation
https://scholarworks.wmich.edu/dissertations/3225
Spatial smoothing in Magnetic Resonance image (MRI) involves applying a filter to remove high frequency information and consequently improves signal-to-noise ratio that can greatly aid neurosurgeons in pre-surgical planning stages of tumor resection. This immensely reduces the time spent on Electrical stimulation mapping (ESM) prior to surgery. MRI’s three-dimensional data provides voxel intensities with complex spatial relationship. The standard de facto spatial smoothing method, Gaussian Kernel smoothing, is satisfactory since a uniform smoothing is done for the whole brain. Secondly, the kernel smoothing technique assumes normality for the voxel intensity, but there is ample evidence in current research that indicates that voxel intensities for MRI data approximately follow a Racian Distribution. This leads to a blurring effect when the kernel smoother is applied to MRI data at various bandwidths. Due to the nature of the brain, we must consider the various tissue types and regions in any smoothing algorithm and hereby avoid blurring region borders. This study developed a flexible reduced rank spatial smoothing technique that achieves data reduction and sensible spatial smoothing at the same time. To achieve this, a reduced rank spatial model in the Bayesian framework with spherical basis function at specified knot locations with
different spatial resolutions is developed. Knot locations are equidistant but with careful consideration of the anatomical structure of the brain which mimics a half sphere. The data used is a public sourced MRI data of an adult male with brain tumors or lesions to the left region of his brain. Twenty-two slices of 512 by 512 voxel images were acquired totaling close to 6 million data points. This model structure aids in attaining a relatively manageable covariance structure in a computational sense. This study set out to achieve smoothing MRI for the entire brain and in the process of doing so, to achieve data reduction. For preliminary analysis, we consider a single slice of brain image (T-1 weighted). This forms a 2D image. In this case, we use a modified bisquare basis function to explain the small scale variability in our model. This model realized a Signal-to-noise ratio (SNR) improvement of about 22.5% over the current pre-processed data set. In the second half of our model, we demonstrate the flexible nature of our spatial model in smoothing the region where possible tumor or lesion spots can be observed from the image. Adjacent slices are treated as layers of time series and a first order structure is used to determine association between slices. We use marginally non-informative prior for the covariance matrix of the reduced dimension latent process in our model. As opposed to the kernel smoothing method, a normal distribution was not imposed on the latent process. Instead we use a Gaussian mixture that can take care of the heavy tailed behaviors as demonstrated in MRI data sets. The proposed method demonstrates an improvement in the automated spatial smoothing process for the Brainix MRI data. This method can be beneficial to the brain surgeon in that it significantly reduces the time spent probing via ESM as region boundary blurring is avoided.
Acknowledgments

With all gratitude, I first of all want to thank God Almighty for giving me the strength to go through these years of my life in graduate school. Considering the journey to this point, there is no way I could have gone half way on my own. I also want to thank the individuals and groups that helped me both academically and emotionally during this work.

I would like to say thank you to all the professors that I encountered in our Department and Western Michigan University in general. More so, all the members of my committee. To my advisor, Dr. Rajib Paul, whom has been very patient with me over the years, I want to say that I greatly appreciate your help especially the many times I felt less assured and emotionally tired. I also want to thank you for introducing me to MRI and fMRI data analysis and also helped me to understand the spatial significance for such data analysis. Please accept my thanks for your guidance and extraordinary patience. I also want to thank Dr. Bugaj for your wonderful support. You gave me an opportunity to improve my teaching abilities by appointing me as a faculty specialist for STAT-2160 and have always been a source of encouragement to me. For that I will always be very grateful. Dr. McKean and Dr. Baker, I really appreciate your help as members of my committee. For all the words of encouragement and support during this period, I am forever grateful.

I definitely want to thank Michelle Hasting, the Statistics Department undergraduate advisor and secretary, for your support. I will always hear these words no matter where I go; " ...Leonard, how is the paper coming...". For all the times I locked myself out of the office and you had to come to open it for me, I thank you.

Finally, my family: To my absolutely loving and supporting wife, Michaella, you are my singular cheerleader. Thank you for holding our family together when I have
to be away for weeks. I truly would not have been able to achieve this or anything else without your support. Thank you. To Joan Ratliff, my American Mum, thank you for your prayers and support. I know Jim is watching over us from above. To Sarah, Leon and Raymond, papa loves you. All I do is for you. Thinking of you guys always keeps me going. For all the sports and school activities I missed, I am sorry and will make up for them now.

Leonard C. Johnson
Contents

Acknowledgements ii

List of Tables vi

List of Figures vii

1 Literature Review 1

1.1 Introduction ................................................. 1
1.2 Review of Denoising Techniques ............................... 4
1.3 Statistical Analysis for Brain Image Data ....................... 5

2 Acquisition and Description of Brain MRI 8

3 Proposed MRI Smoothing Methodology 14

3.1 Introduction .................................................. 14
3.2 Reduced Rank Spatial Model with Gaussian Assumption for MRI ........................................ 16
3.3 Fitting Spherical Harmonics Function for Large Scale Variation ........................................ 21
3.3.1 Fixed Rank Smoothing Result ................................ 25
3.4 Reduced Rank Spatial Model for MRI Data with Non-Gaussian Assumption ........................................ 27
3.4.1 Image Preparation for Reduced Data Extraction .................. 28
3.4.2 Model Description .......................................... 30
Table of Contents - Continued

3.4.3 Prior Distribution and MCMC Application . . . . . . . . . 32
3.5 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

4 Future Work 38

Appendices:

A R Codes and Packages Used 39

B HSIRB Project Approval Letter 17-04-20 63

References 65
List of Tables

3.1 Results for PSNR in T1-weighted Images .......................... 37
3.2 Results for PSNR in T2-weighted Images .......................... 37
3.3 Results for PSNR in FLAIR Images ............................... 37
List of Figures

2.1 T1-weighted Images ............................................. 11
2.2 Cross sectional View ............................................. 12
2.3 Structural MRI for Different Contrasts ......................... 12
3.1 Non-Gaussian Nature of MRI Data ............................. 19
3.2 Non-Gaussian Nature of Log-Transform MRI Data .......... 20
3.3 Contour Plot for Legendre Polynomial in X direction: No Structure at 50 Degrees .............................................. 22
3.4 Contour Plot for Legendre Polynomial in X direction: No Structure at 80 Degrees .............................................. 23
3.5 Contour Plot for Legendre Polynomial in X direction: No Structure at 100 Degrees .............................................. 23
3.6 Legendre Contour at 50 Degrees in Y Direction ............ 24
3.7 Legendre Contour at 80 Degrees in Y Direction ............ 24
3.8 Legendre Contour at 100 Degrees in Y Direction .......... 25
3.9 QQ Plot for Legendre Polynomial of Degree 83 ............ 26
3.10 Residual Plot for Legendre Polynomial of Degree 83 ....... 26
3.11 Original and Smoothed T1 Image .............................. 27
3.12 Orthographic Mask for Extraction of Tumor Region Voxel .... 28
3.13 Orthographic Overlay of Image on masked ................ 29
3.14 Extraction of Tumor Voxels from FLAIR Image . . . . . . . . . . . 30
3.15 Distribution of Intensity for the Reduced Data . . . . . . . . . . . 31
3.16 Residual Plot for FRRSM for the Reduced Data . . . . . . . . . . . 35
Chapter 1

Literature Review

1.1 Introduction

The structural Magnetic Resonance Image (MRI) and its functional counterpart, fMRI, was developed in the early 1980’s due to the quest for a noninvasive and relatively safe technique to image the human brain in vivo. Brain imaging studies, among other things, attempts to discover the functional and structural basis for neurological disorder and to examine treatment response to such cases (Bowman (2014)). There are numerous types of neuroimaging techniques; both invasive and non-invasive, for instance, positron emission tomography (PET), electroencephalography (EEG) and magneto-encephalography (MEG), but MRI is the most widespread and it is quite applicable in large and small scaled studies of both healthy and diseased human brains. EEG and MEG, for instance, measures the electrical and magnetic information, respectively, from the surface of the human brain but the exact location of such signals can be uncertain and very difficult to predict. The MRI, on the other hand, gives almost precise localization of the origin of an intensity. Functional studies as pertaining to language, emotion, memory, cognition and the general steady (resting) state of the brain are mostly found in
numerous literatures of brain image analysis in recent years; See: Musgrove et al. (2016); Bowman (2005); Derado et al. (2013); Lindquist (2008); Ge et al. (2014); Poldrack et al. (2011). Earlier imaging methods depended on ionizing radiation as used in X-rays or computerized Tomography (CT) scans. These methods were unsafe as the increased exposure to radiation is connected to cancer and other related diseases, they cannot be used in children and more so, could not provide flexible image characteristics to measure wide range of brain tissues. MRI, on the other hand, provides a technique that is useful in obtaining a wide range of measuring characteristics for both the structural and functional aspects of the brain by quantifying differences in the magnetic properties of different molecules. For instance, bone structure, water molecules, white and gray matters are recognized at varying intensities in an MRI scan. It was also found to be completely noninvasive, carries less severe health risk, which means that for the first time, studies were done on children of any age, and provides researchers the chance to observe neural activities and consequent changes in the human brain whilst they are alive and almost in real time (Poldrack et al. (2011)). The most common distinction between the functional and structural modality measure is that fMRI records blood oxygenation level-dependent (BOLD) signals that vary by the level of oxygenation in the blood, its flow and metabolism (Lindquist and Wager (2014)). This quantifies brain activity in various regions of the brain by its localized increase in oxygen intake. Although the BOLD signal does not directly measure neuronal activity, it is acceptable to consider the localized change of metabolism which demonstrate high correlation (Bowman (2014)). On the other hand, MRI has signals that varies by tissue types which results in structured images that identifies white matter, gray matter and cerebrospinal fluids (CSF) in the brain. In terms of medical classification, structural MRI is mostly important in the study of brain Anatomy and Pathology whilst
Functional MRI is used in studying brain activity over a period of time and in a resting state.

Statistical analysis of a brain image is very complex. If not merely because of the size of the data set, which generally can run into millions of voxels in group studies, but also because of various important pre-processing steps that are mostly needed to validate statistical assumptions prior to any comparative tests. In structural MRI, the intensities are in the form of a raster dataset consisting of matrix of cells (pixel in a plane or voxel in three dimensions) organized into rows and columns where each cell gives an aggregate intensity for that region. Images are produced as slices of the brain with different contrast, thickness and resolutions. In most, or essentially all cases, image output comes with various artifacts and substantial amount of noise. This is mostly due to image acquisition from the scanner, scanner settings and movements of the patient. Thus, prior to any statistical analysis, MRI usually requires an extensive pre-processing step which includes, but not limited to, removing artifacts and validating model assumption. For example, noise in MRI follows a Rician distribution but in most literature the noise are assumed to be Gaussian (Lindquist (2008); Yousuf and Nobi (2010)). In this work, we will concentrate on devising a robust denoising or spatial smoothing technique to improve the signal-to-noise ratio (SNR) for various MRI contrasts. This method is also expected to improves the validity of statistical tests by transforming the error distribution to become more gaussian.

Due to the acquisition procedure in imaging, it is clear that each voxel is noisy, however, to a certain extent, neighbors tend to show similar effect. Spatial smoothing is a form of averaging voxel intensities with their neighbors. This results in the removal of high frequency data while enhancing low frequency data. Most of the methods in for spatial smoothing or denoising of MRI images gives unsatisfactory
results that produces significant blurring of the edges in the image.

The most common denoising technique in the field of Neuroimaging is the Gaussian Kernel Smoothing. A summary description of various denoising methodologies and some of their limitations are given in (1.2) below.

### 1.2 Review of Denoising Techniques

Noise reduction in a brain image is a very significant step in automated MRI clinical and analytic diagnostic processing algorithms. The initial process of image acquisition can introduce noise into an image due to patient movement and machine inaccuracies. Other pipeline processes such as segmentation, image registration and parameterization make the image more noisy which makes it important to minimize the noise and elevate signal-to-noise ratio (SNR). This will also make the noise look more like a Gaussian random noise which is required for subsequent statistical inference that are based on the theory of random field.

The additive noise model used in image denoising techniques is given by:

\[ Z = Y_0 + \epsilon \]

where the noise \( \epsilon \), is generally assumed to be independent Gaussian with mean equal to zero and a known standard deviation; \( Z \) is the observed image signal; and \( Y_0 \) is the true image signal. These techniques have been classified into two categories namely, spatial filter and transform domain filter; See Motwani et al. (2004). The spatial filter involves transforming the voxels of an image via a function of the collective intensities of their neighbors while the transform domain filter involves transforming images from an image space to a frequency or wavelet domain. Examples of the spatial filter are Gaussian smoothing (Chung (2013)), Median, Wiener, diffusion
and bilateral filters (Yang et al. (2015)). Example of a transform domain filter is the wavelet thresholding method.

One of the most prominently used techniques is the Gaussian kernel smoothing, Chung (2013). This technique is a weighted average of image voxels over a given bandwidth. The bandwidth is a measure of the spread of the Gaussian kernel and it is obtained in terms of what is referred to as the full width at the half maximum (FWHM) of the Gaussian kernel. With appropriate FWHM, Gaussian smoothing can detect hidden signals in an image and can also improve the normality assumption of a signal. Gaussian smoothing, however, can result in significant blurring of the image especially so with voxel boundaries, Yang et al. (2015). The Median filter is another example of a Spatial filter albeit Non-linear and it has the advantage of robustness over the Gaussian or Mean filter. However, it still suffers the disadvantage of blurred image edges. One method that seems to work well in preserving image boundaries is the non-local mean (NLM) algorithm and its extension in Yang et al. (2015). However, this method has a high computational burden.

1.3 Statistical Analysis for Brain Image Data

Statistical methods used for analyzing brain imaging data are usually centered around objectives that target localization, brain connectivity and classification, Bowman (2014). Functional connectivity and localization studies usually involve analyses that highlight statistically significant changes in brain activities corresponding to related task. Classification studies in most cases involves some form of baseline imaging to generate maps to predict metabolic brain activity as in Ge et al. (2014) and Derado et al. (2013), or classify individuals into different categories,
for instance, as a treatment responder or non-responder (Bowman (2014)).

In its early stage, statistical method for localization or activation studies involves blood flow data with simple procedure such as t-test between resting and task state images. Activation maps are created by simply removing the average activation during one task from that during another as demonstrated in Kwong et al. (1992). This so called "mass univariate" approach, which was designed via a simple linear model was found to be ill suited for binary response data and did not take into account the spatial dependency between locations and nearby voxels (Ge et al. (2014)). Several variations of mass univariate approaches have been used for example, the logistic variation to deal with the binary nature of lesson maps which is known as Firth regression (Firth (1993)). A general linear mixed effect model is employed in several studies to remedy this situation (Bowman and Kilts (2003); Shaw et al. (2008); Bowman (2014)). Due to the massive nature of brain image data, however, even for a single subject, modeling voxel by voxel covariance function creates heavy to almost impossible computational issues. A resulting two-stage approach is adopted in many literatures (Bernal-Rusiel et al. (2013) and Derado et al. (2010)) with different variations and covariance estimation techniques. For example, Bowman (2014) modeled the first stage as a single-subject general linear model (GLM) and the second stage as individual experimental effects in terms of group level parameters which reduces computational stress and simply used least-square estimation in both stages. The resulting covariance structure is approximated using weighted least squares.

The two stage method still has its limitations, in that the models are being estimated for each voxel which assume that neighboring voxels are independent of each other. Bowman (2005) uses a simultaneous autoregressive model to capture exchangeable spatial correlations between all pairs of voxels within functionally defined
networks. A spatial generalized linear model was introduced in Ge et al. (2014). All parameters were assigned priors and spatial parameters assumed Markov random fields to account for spatial structures and neighboring sites or voxel are determined if they share a common face. This results in some form of dimension reduction process to estimate the covariance structure. These methods mostly result in extensive computation time and generally needs a super computer for inference. However, some relatively fast Bayesian procedures have recently being added to the literature of brain data analysis (Musgrove et al. (2016)).

In general, most of these methods use MLE procedures in the parameter estimation stage and therefore are susceptible to increased variability in the event of an outlying observation within regions. The Bayesian method which gives better results in most cases compared to the two stage GLM is mostly computationally burdensome.
Chapter 2

Acquisition and Description of Brain MRI

Magnetic Resonance Images are producible due to the fact that human beings have a significant amount of hydrogen atoms in our body. When an individual is placed into an MRI scanner that has a huge circular magnet, hydrogen protons become excited and align with the resulting strong magnetic field produced. These are then exposed to radio frequency (RF) waves which spins the various proton content and produce signals that can be detected by a receiver placed in the scanner. The receiving signal is processed by a computer and the resulting image at a pre-specified resolution is formed. Spatial resolution is in concert with the detectable details of the image. The smaller the voxel or pixels, the higher the potential resolution in space should be. In most brain image studies, the regular matrix sizes per slice that are used are $256 \times 256$ and the $512 \times 512$ matrix sizes.

The MRI has become the most widely used method for brain imaging simply because of its minimal to nonexistence of risk due to radiation exposure and its non evasive nature. It also gives high image contrast in soft tissues which is very useful compared to computed tomography (CT) imaging (Chung (2013)). The
MRI produces images based on numerous pulse sequences but the most common ones in brain imaging are the Spin echo pulse sequence and the Inversion recovery sequence. In the spin echo case, the pulse sequence timing can be adjusted to produce the spin-lattice relaxation time (T1), the spin-spin relaxation time (T2) and the proton density ($\rho$) images. The image produced can weigh the contribution of these factors to get a specific result. For example, the so-called T1-weighted image relies on the spin-lattice relaxation time and minimizes the effect of proton density and spin-spin relation time. The idea is similar for the T2-weighted images. In the inversion recovery sequence, signals from one or more tissues can be nullified to produce a specific type of image. For example, the Fluid attenuation inversion recovery (FLAIR) image is one specific case in which the inversion time (T1) is lengthened in order to remove all CSF signals from the resulting image. This image sequence has been proven to be useful in studying diseases of the central nervous system and severe brain injuries (Atlas (2009)).

The most commonly referenced structural images are the T1-weighted and T2-weighted images, which provide basic anatomical pictures of the brain. Structural images mostly collected are T1 images, which are used to register functional images as well as for multiple types of anatomical analyses related to outcomes. This type of image is sensitive to the water content of tissue, so it produces different image intensities in the major in-brain tissue types of gray matter, white matter, and cerebro-spinal fluid (CSF). T2 images provide a different type of contrast between tissue types; they are particularly useful for identifying the boundaries of certain iron-rich nuclei, Lindquist and Wager (2014). The brain MRI comes as a 3D array with tissue intensity values at each voxel location $s \in \mathbb{R}^3$. The whole brain volume is produce as a layer of slices with specified slice thickness. Slices can be viewed in three different cross sections: Axial which is sliced horizontally from front to back.
Coronal, which is sliced left to right across the face, and Sagittal, which sliced top to bottom of the brain.

In this work we will use the brain images for a single male right handed individual with a brain tumor for evaluating our Spatial denoising technique spatial random effect model in the Bayesian framework. The BRAINIX dataset, as it is called, was downloaded from an open source database provided by the OSIRIX website. The images are in the Digital Imaging and Communication (DICOM) file format and it contains T1, T2 and FLAIR weighted images. Most MRI scanners now save their reconstructed data to the DICOM file format. DICOM stores each slice of an image as a single file with associated header information. They are mostly common amongst medical practitioners but are very complex to use in analysis because they are comprised of an header file in a list which contains patient, hospital and other relevant information refer to as metadata and the image matrix for voxel intensities. We therefore covert the file format into a NeuroImaging Informatics Technology Initiative (NIfTI) file format which removes patient information and can easily be used in R with the help of some available packages. The NIfTI file format is very much helpful in representing the link between voxel intensity and spatial location in the MRI scanner. It also helps to separate the reconstructed data into an image and header files.

Twenty-two slices of a 512 by 512 voxel images were acquired that total to about 6 million data points. The axial slices of the T1 weighted images for BRINIX are given in Figure 2.1. We noticed tumors began to become visible on the twelfth image and more clearly on the thirteenth and fourteenth images towards the right side of the brain.

Each slice can be viewed in the different cross sections. In Figure 2.2, the image can be viewed in the Cortical, Sagittal and Axial cross sectional views from left to
right and bottom, respectively. The cross arrow indicates the origin in the three
dimensional space and is defined as the point where the anterior commissure (AC)
intersects the mid-line saggital plane.

In Figure 2.3, we can see how different image contrast highlight the structural
properties of the brain. CSF appears dark gray in T1 but very bright in the T2
images. We also see that T2 images are more appropriate for lesion detection
compared to their T1 counterpart as Tumors and lesions appears hyper-intense in
T2 but hypo-intense in T1 images.

Due to most recent developments in imaging acquisition, a brain image could
have over 2000 gray levels. In its inception, a classical MRI use to have 256 gray
levels. So this means that we now have a visualization problem to distinguish be-
tween the various levels but only small portion of this verse range is important.
Figure 2.2: Cross sectional View

Figure 2.3: Structural MRI for Different Contrasts
For most cases, it is vital to enhance the contrast in such a way that the most important pieces are kept and all other gray levels in the original image are averaged or mapped to perfect white or black. It is quite reasonable to average the images in a spatially sensitive manner so that neighborhood voxels, which are most likely to be sharing strong association are average together in a reasonable manner. So even in instances where the noise in an image is almost invisible, it may be important to reduce it in light of further use of the data under consideration. We therefore consider an application of spatial random effect model as an averaging mechanism for MRI image denoising. Due to the enormous nature of the MRI data, we propose that this method is quite fitting in a sense that it acts as a Dimension reduction technique for possibly a very large covariance matrix of voxels by introducing known structures of basis functions as was demonstrated in Cressie and Johannesson (2008).
Chapter 3

Proposed MRI Smoothing Methodology

3.1 Introduction

In this work, we are proposing the use of a spatial smoothing technique that incorporates the Spatial Random Effect Model (SRE) in Kang and Cressie (2011) with a spherical basis function in a Bayesian framework. It is not beyond reason to model the human brain surface as a sphere or half sphere due to its natural anatomic structure. The brain is naturally divided into two hemispheres: the right and left hemispheres. There is a growing interest in developing covariance functions for processes on the surface of a sphere because of the widespread of global data collection that have significant spatial or spatio-temporal importance. It is well known that covariance functions originally defined with the Euclidean distance may not be adequate for data on spheres (Jeong and Jun, 2015).

The most widely known spatial prediction or smoothing procedure is the Fixed Rank Kriging (FRK) method. See (Cressie, 1993). This method, however, runs into computational issues when the number of observed locations, $n$, is large. Spatial
prediction using any technique involves inverting spatial covariance matrix and this operation is generally of order $O(n^3)$, so once the observed location is large, this becomes almost impossible. For our initial work, we considered the Fixed Rank Kriging (FRK) strategy as adopted by Cressie and Johannesson (2008) in which they were able to achieve the best linear unbiased prediction (BLUP) for large spatial data with Euclidean distance measure but with some adjustments. This involves the modeling of the spatial covariance matrix using spatial basis function mapping the $n$-dimensional data into selected $r$ knots locations over the spatial domain with the assumption of Gaussianness. This process is mostly referred to as Reduced Rank Spatial Model (RRSM) in the literatures. We are proposing in the next phase of our study an improvement in the RRSM model using Bayesian framework for a much flexible class of reduced rank spatial model. We will discuss the smoothing of MRI data for a single slice of the BRAINIX image with the assumed Gaussian RRSM and we shall refer to this model as Model 1 henceforth. This will lay out the ground work for the application of a flexible reduced rank spatial model in MRI filtering in our future work. This method was introduced in Paul et al. (2015) and applied to a global Ozone data. This model is expected to address the issue of non-Gaussianess of MRI data sets. In our second model, we will use a flexible but carefully adopted class of geostatistical model in the Bayesian framework. This class of model was described in Kang and Cressie (2011) as spatial random effects (SRE) models.
3.2 Reduced Rank Spatial Model with Gaussian Assumption for MRI

For our model formulation, let $Z \equiv \{Z(s_1), ..., Z(s_n), s_1, s_2, ..., s_n \in D\}$ denote observed values of image intensities at $n$ locations on the image domain $D \in \mathbb{R}^d$. The spatial random effect model for the data can be written as

$$Z(s_i) = Y(s_i) + \epsilon(s_i); i = 1, ..., n$$

(3.1)

where $Y(s_i)$ is a latent (Hidden) process with no measurement error, and has a mean $\mu(s_i)$ and covariance $C_Y(s, s')$. $\epsilon(s_i)$ is assumed to be zero-centered Gaussian random noise. In MRI data, the white noise assumption is mostly violated after several preprocessing steps. Spatial smoothing is one important step that can help improve the validity of assumption that the error here is white noise. This involves averaging of neighboring image intensities of nearby voxels. This deliberate blurring of the image also improves signal-to-noise ratio by removing very high frequency intensities and enhancing low frequency ones and thereby making spatial correlation much more enhanced.

In this work, we propose to use a posterior predictive distribution to obtain estimates for model parameter in the spatial random effect model. The difficulty of inverting large dimensional matrix is documented in Spatial Statistics literatures. We postulate that this can be overcome in MRI data by specifying the distribution of the latent process using spatial basis function matrix $S$, and a reduced dimensioned zero-centered Gaussian process $\eta$ (Paul, Jelsema, and Lau, 2015) over a selected knot locations. The latent process model for $Y \equiv \{Y(s_1), ..., Y(s_n), s_1, s_2, ..., s_n \in D\}$ is given by:
Thus the spatial random effect model for the noisy image intensities at $n$ locations can be expressed as:

$$Y = \mu + S\eta + \delta$$  \hspace{1cm} (3.2)

Let the covariance matrix of the Gaussian process $\eta$ be denoted by $\text{cov}(\eta) = V$. This model formulation aids in attaining a relatively manageable covariance structure in a computational sense. The spatial covariance for the data can then be expressed as:

$$Z = \mu + S\eta + \delta + \epsilon$$  \hspace{1cm} (3.3)

where $\Delta$ is the diagonal matrix of the sum of errors in prediction. That is, the sum of measurement and process errors. Generally, spatial prediction, or as in this case spatial smoothing, requires that one invert the spatial covariance matrix of the observed data in $n$ locations. In our MRI data, this is computationally impossible as we have $512 \times 512 \times 22$ spatial locations in the $3D$ MRI image. So our reduced rank approach here is quite useful. The spatial basis function matrix $S$ maps our $n$ dimensional spatial process to a reduced dimensional latent process of $r$ dimension with $r << n$. This results in an $r \times r$ covariance matrix which can be easily inverted by using the Sherman-Morrison-Woodbury formula. See (Henderson and Searle, 1981). This concept is not new in the literature but the application to spatial smoothing for MRI and the propose choice of basis function(s) in doing so has not been seen in any work to the best of our knowledge. We will use two different classes of basis functions and investigate their effects on the smoothing process. In
the preliminary analysis, we considered a single slice of the BRAINIX MRI data set discussed in Chapter 2 and apply our spatial smoothing reduced rank model using the modified bisquare basis function described in Paul et al. (2015). Due to the flexible nature of the basis function, it is possible to apply smoothing at different resolutions. We applied the Gaussian kernel smoother to the same slice of image and compare the signal-to-noise (SNR) and the peak-signal-to-noise ratio (PSNR) for the two methods to assess the quality of our smoothing method. The method with higher PSNR is considered to be a better smoother.

When considering all slices of the brain, that is the entire 3D image structure, we propose to use a Spherical basis function. The wavelets basis function, which is well known in brain imaging models, and the Radial basis function (RBF), which is found to be very successful in modeling high dimensional data, (Sun and Chen, 2008) are two classes of functions we wish to apply. However, the modified bisquare basis function can still be used as it is also of a spherical nature. In an impressive study by Atlas (2009), a fast, second generational wavelets basis function in a sphere is proposed. Its application here seems promising.

Before we proceed, it is important to note here that the voxel intensity that are observed from an MRI scanner are non-Gaussian and in most cases they have multiple peaks. In fact, it follows a Racian distribution with noncentrality and scale parameter. The Racian distribution is used in communications theory to model scattered signals directed to a receiver through multiple paths. This strong skewness associated with this distribution cannot be overcome by log-transforming the data as can be seen by the QQ-plots in Figure 3.1 and 3.2. We try to work around this non-Gaussianess by fitting a large scale equation with Spherical Harmonic function. The FRK method 1 has some limitation because we need to fit a spherical harmonic function of order 83 in order to get a gaussian residual. We postulate
that the flexible RRSM proposed below will be an improvement of Model-1 in this sense.

Paul et al. (2015) discussed the lack of literature for spatial random effect models with heavy tails and multi-modal. We face a similar constraint and the application of the basic methodology used in the Flexible Reduced Rank Spatial Model (FRRSM) is applied.

In the process of modeling the small scale variation, we observed that there is some form of large scale variability in the Y-direction of the image. This is very difficult to model with a simple linear or polynomial function due to the nature of our data; however, in using a known relationship between Legendre functions and spherical harmonics, the large scale variation was modeled using spherical harmonics and the subsequent design matrix was obtained using Legendre polynomial of degree
Figure 3.2: Non-Gaussian Nature of Log-Transform MRI Data

\[ n = 83. \]
3.3 Fitting Spherical Harmonics Function for Large Scale Variation

In order to model the image data, which is extremely noisy and very large with complex spatial correlations, we fit for the large scale variation, a Spherical Harmonic function which works well for data of this nature (Paul et al., 2015). We note for the contour plots in both directions that there is but a significant trend in the Y direction as compared to the X direction. For that reason, we created a design matrix using Legendre polynomials $P_n^m(\sin(D(Y)))$, where $D(Y)$ is the Y direction for the image data.

In the preliminary analysis, we used the eleventh slice of the T1- weight image to demonstrate the validity of our model. We postulate that the results will not see significant changes based on the type of MRI used. The Legendre polynomials of degrees $n = 50, 80, 100$ were used for $m = 0, 1, 2, \ldots, n$, in the Y direction for the locations of each voxel in the slice. The Legendre function as defined in Rodrigue's formula (Olinde Rodrigues, 1816), is given by

$$P_n^m = (-1)^m(1 - x^2)^{m/2} \frac{d^m}{dx^m} P_n(x)$$

where $P_n(x)$ is the Legendre polynomial of degree $n$ and is given by:

$$P_n(x) = \frac{1}{2^n n!} \left[ \frac{d^n}{dx^n} (x^2 - 1)^n \right]$$

We use the Legendre MATLAB function to generate the $X$ matrix. The algorithm used in this MATLAB function is obtained by using a three-term backward recursion relationship in $m$ which has in it embedded a complex spherical harmonics. Thus there is a restriction on the domain of the $x$ values that the function can take.
Thus the domain is such that $x \in (-1, 1)$

Since the values for the X-direction and Y-direction for our brain image data do not meet this restriction, we have to standardize the data in the following manner. We substract the median from each value and divide by the range, so that we have;

$$x^* = \frac{x - \text{Median}(x)}{\text{Max}(x) - \text{Min}(x)}$$

If $x$ is a vector, as in our case a $111454 \times 1$ vector for the Y-direction or X-direction, then the Legendre function produces an $(n+1)$-by-$q$ matrix, where $q = \text{length}(X)$ and $n$ is the degree of the polynomial.

Figure 3.3: Contour Plot for Legendre Polynomial in X direction : No Structure at 50 Degrees
To determine the most appropriate degree for the polynomial, we use a trial and error method with $n = 50, 80, 100$ in both directions. However, in further investigation, we note that the eighty-third degree polynomial gave the best result and therefore was used. The resulting matrices are plotted using a contour plot in MATLAB and it is clear that the Y-direction display some form of a trend and therefore we used the Legendre polynomial in the Y-direction at the appropriate degrees. We proceeded to obtain ordinary least square estimate for the large scale
regression coefficient $\beta_{OLS}$. Although this estimation comes with an immediate concern of over-fitting for large data, we decide to use it for this initial model. For the choice of the basis function, to obtain $s$, we adopted the modified bisquare basis function as in Paul et al. (2015) and the joint standard deviation $\rho$ is estimated from the data.

The resulting residual plot and QQ-plot for the 83 degree Legendre polynomial is given in Figures 3.9 and 3.10. The residual plot looks much more Gaussian than in the previous plot. See Figures 3.1 and 3.2.
For the small scale variation, we selected \( r = 16 \) knot locations in the regular domain of slice 11 of the BRAINIX data. The resulting covariance function for the noisy image from (4) can be expressed

\[
\Sigma_z = SS' + \rho I
\]  

(3.5)

where \( \rho \) is the joint standard deviation of \( \delta \) and \( \epsilon \).

### 3.3.1 Fixed Rank Smoothing Result

The smoothed images in (b) and (c) of Figure 3.11 gives the outcome of fixed rank smoothing method and the Gaussian kernel smoother with \( \sigma = 5 \), respectively. The image in (a) is the original eleventh slide of the BRAINIX data set.

From Figure 3.11, we compute the signal-to-noise ratio (SNR) for each smoother. The resulting SNR improved by 29.9\% for the fixed rank smoother and 19.5\% for the Gaussian kernel smoother but with much more blurred image.
Figure 3.9: QQ Plot for Legendre Polynomial of Degree 83

Figure 3.10: Residual Plot for Legendre Polynomial of Degree 83
3.4 Reduced Rank Spatial Model for MRI Data with Non-Gaussian Assumption

For this model, we assume similar structure as in the model described in (3.3) but with no assumption of a Gaussian distribution. Paul et al. (2015) discussed to a lengthy extent the effect of Gaussian assumptions in analyzing data sets that do have an indication of being heavy or light tailed. There is not much literature for large spatial data sets that exhibits such distribution. Our data set is one such typical example of extremely heavy right tailed distributed. Consider, model in (3.3), the Reduced Rank Spatial Model with the normal assumption with the likelihood based Expectation Maximization (EM) algorithm. It is clear that the parameter estimations will be off and thus results in poor fit and consequently poor predictions.

We adopt the proposal of Paul et al. (2015) of abandoning the Gaussian assumption in the latent process but rather consider mixture of Gaussian model with an exponential scaling parameter \( \lambda \) equals one. Parameter estimations where obtained in the Bayesian framework with non-informative priors for the covariance.
matrix function. Significant amount of simulations were done and application to the daily maximum of total column Ozone data, a global spatial data, produced meaningful results.

We are attempting to use this method in MRI data for the first time to the best of our knowledge. Detailed information on minor changes that needed to be done will be discussed. Our focus here is to smooth the region with the tumor due to the enormous nature of the full data set. For this purpose, we have to prepare the data for extraction of voxels that falls in the region of the tumor on each slide.

### 3.4.1 Image Preparation for Reduced Data Extraction

Let the 3D reduced image have $Z(i, j, k)$ at voxel $(i, j, k)$. Consider the FLAIR-Weighted image and we now have reduced image with a constructed overlay mask given in Figure 3.12.

![Figure 3.12: Orthographic Mask for Extraction of Tumor Region Voxel](image)

Once the masked is created, the model is treated as a Region of Interest (ROI)
analysis. We can overlay the complete BRAINIX data on the masked and extract the voxel values that falls with the ROI. We consider voxels on the boundary and decide to accept voxel to be in the region of interest if more than 60% of it falls within the mask. The resulting image is given in Figure 3.13. This covers every slice of the image with some amount of tumor. Each slide was plotted to see the extent of the tumor. We attempt to give a side by side view of the 22 slides with the corresponding images that have visible tumors as close as possible.

The extent of skewness in the reduced data set is mostly accounted for by the large amount of background which is now present in the image. The corresponding voxel intensities are all zero. We can simply minimize this effect by considering voxels with intensity less than 30 as background and a much more clear structure of the data is seen in Figure 3.15 below.
3.4.2 Model Description

The Reduced Rank Spatial Model with Non-Gaussian assumptions is similar to that in (3.3). For a 3D reduced image measuring \( m = N_x \times N_y \times N_z \) observed intensities \( Z(\cdot) \) of voxel \((x, y, z)\) at locations \( s_i \), for \( i = 1, \ldots, m \), we can express the observed measurement as a linear combination of a hidden process with some random noise factor. In this case, \( N_x \) is the number of voxels in the horizontal direction of the image, \( N_y \) the number of voxels in the vertical direction in each slices and \( N_z \) is the number of slices. This results in a similar structure as the spatial random effect model in (3.3). As evidence, this model is very accommodating and extremely flexible. We can smooth the ROI images using spatial basis function defined over selected knot locations over the image domain.

Simulation based evidence from Paul et al. (2015) indicate that a modified bisquare basis function works well and therefore we keep the same basis function and change the spherical radius about which the function is defined to a 2-mm radius.
Figure 3.15: Distribution for Intensity for the Reduced Data

\[
S(s, u_i) = \begin{cases} 
(1 - 1/4d^2(s, u_i)) & d(s, u_i) \leq 2 \\
0 & \text{otherwise}
\end{cases}
\]

where \(u_i\) is the \(i^{th}\) selected knot location and \(d(s, u_i)\) is a generic morphometric distance measure in a curvilinear space. See (Styner et al., 1999). Under this model, \(\Delta\) can be estimated by assuming known measurement error variance \(\sigma^2\) and use MCMC sampling to estimate the process error. Thus \(\Sigma_z\) can be inverted by using the Sherman-Woodbury-Morrison formula:

\[
\Sigma_z^{-1} = \Delta^{-1} - \Delta^{-1} S \{ V^{-1} + S'\Delta^{-1} S \}^{-1} S'\Delta^{-1}
\]  \hspace{1cm} (3.6)

The large scale variation can be removed by simply fitting an Ordinary Least Square model and a probability plot can show the shape of the residual. To deal with the problem of non-Gaussian nature of our data set, we can directly mimic the idea from Paul et al. (2015). We define a diagonal matrix \(B\) with diagonal
elements \( \{W\}_1 \) each of which are independent exponential random variables with mean equals 1. Then the gaussian process \( \eta \) can be express as \( \eta = B^{1/2}U \), where \( U \) an \( r \)-dimensional gaussian variable with mean zero and covariance matrix \( V \). We now have a set of unknown parameter to estimate with joint probability distribution of \( \{Z, U, W\} \) that is a member of the exponential family. It was demonstrated in Paul et al. (2015) that:

\[
E(\eta) = 0
\]

and the covariance matrix of \( \eta \) can be simplified to an \( r \times r \) matrix in up to a constant of the covariance of \( U \).

### 3.4.3 Prior Distribution and MCMC Application

We want to avoid situations in which the prediction or posterior outcomes for any given voxel depends on the given prior distribution due to some hyperparameter. It is therefore important to choose a non-informative prior for our parameter estimates. For our large scale parameter, we assumed \( \beta \) to have mean equal zero and covariance function with a heavily scaled Identity matrix. A half-\( t \) distribution is used as the prior for the square root of the process error. It should be made clear that all of these priors have been used by Paul et al. (2015) and a working-progress \( R \) package is available in GitHub. An inverse-\( \text{Wishart} \) prior is used for the covariance matrix of \( U \) conditioned upon a hyperparameter with inverse-Gamma prior. This is done in two levels to deal with the informative nature of the priors when using inverse-Gamma directly as a prior for covariance matrix. A special Markov Chain Monte Carlo (MCMC) algorithm is use, where the Markov chain central limit theorem (MCCLT) is applied to reduce storage burden in our computation for model parameter estimation and smoothing. Despite the good idea of updating
the MCMC in blocks, we still have issue with the MCMC running smoothly. Since \( \{W\}_1 \) are linearly related, they were updated simultaneously. The full conditional distribution is not available in closed form but it takes the form of an exponential distribution family up to a constant. It takes the following form:

\[
[W] \propto \exp\{-1/2R'R + \sum_{i=1}^{r} W_i\} 
\]

where \( R = Z - X\beta - SB^{1/2}\eta \)

The adaptive rejection Metropolis sampling (ARMS) from the package HI in R is used for resampling from \( W \). The measurement error is assumed to be known and is estimated from small sample of locations of the data. The estimated variance for the measurement error is obtained by taking the median of the variances of the estimates in these locations. The two remaining sets parameters to be estimated, \( \beta \) and \( U \), have a linear relationship in our model and we therefore perform block updated to obtain these estimations. They are sampled from a multivariate Gaussian distribution of dimension \( p + r \). Where \( p \) is the number of covariates in the large scale variation.

The most common posterior summary is the expectation. For this specified model, this is expressed as:

\[
E(Y(s_i)|Z) = X_i E(\beta|Z) + S_i E(B^{1/2}U|Z) + E(\delta(s_i)|Z) 
\]

The posterior mean is a function of parameter estimates. We can denote the function by \( g(.) \) The geometric ergodicity can be established by considering the ergodic average based on the MCMC samples. Thus

\[
\bar{g}_i = \frac{1}{T} \sum_{t=1}^{T} g_i(\beta^{(t)}, W^{(t)}, U^{(t)}, \tau^{2(t)}) 
\]
where $T$ represents the total number of MCMC samples needed to compute the respective averages. The asymptotic property was established in Paul et al. (2015). We shall restate the theorem here without a proof.

**Theorem 1.** $\{\beta^{(t)}, W^{(t)}, U^{(t)}, \tau^{2(t)}, t = 0, 1, 2, \ldots\}$ is a Harris ergodic Markov Chain with invariant probability distribution $f(.)$ on state space $\mathcal{X}$ and $E_t |g_i|^2 + \epsilon < \infty$ for $\epsilon > 0$. If this Markov chain is geometrically ergodic, then:

$$\sqrt{T}(\bar{g_i} - E(\bar{g_i})) \xrightarrow{d} N(0, \sigma_{g_i}^2),$$

where $\sigma_{g_i}$ can be considered as a measure of precision of the posterior means obtained from $T$ samples. The proof of this theorem can be found in Paul et al. (2015) and in G.L. Jones et al (2006).

This theorem assures the convergence of ergodic averages in MCMC samples.
3.5 Results

For our reduced data analysis to demonstrate the application of the flexible RRSM, we use the Fluid Attenuation Inversion Recovery (FLAIR) with the selected region of interest as shown in Figure 3.13 above. We also applied this model to the T-1 weighted and T-2 weighted images: there was not much variation in the smoothing for all three types of MRI considered but for one anomaly in the FLAIR result.

A full MCMC run had a burn-in of 2900 with Batch size of 110 and number of Batches to compute the ergodic average as 110. Even with the reduced image, we found it hard to get a full run and reduced the burn-in to 1000 with a semi-full run. The batch prediction error was computed and a histogram for the batch residual for one run of the FLAIR image is give in Figure 3.16.

![Residual Plot for FRRSM for the Reduced Data](image)

The amount of SNR reduction that was obtained for each run with different knots size can be calculated and consequent smoothing done can be quantified by looking at the peak signal-to-noise ratio (PSNR), which is commonly used as an
objective metrics and visual inspection (Yang et al., 2015). The PSNR is given by:

$$PSNR = 10 \log_{10} \frac{C^2}{MSE}$$  \hspace{1cm} (3.10)

Where $C$ is the maximum intensity in which the image can be given. The MSE here is defined between the denoised image and the original reduced sized image. These are exactly the values we computed as our batch prediction sample error. Thus the MSE will be given by the expression:

$$MSE = \frac{1}{N_x \times N_y \times N_z} \sum \sum \sum \sigma^2_{x,y,z}$$  \hspace{1cm} (3.11)

Here we are summing over all voxels that are in the ROI over all slices. The maximum intensity value for the FLAIR image for this subject is 513 and the computed value for MSE is 2703.664. Thus, MSE is given by $10 \log_{10}(97.3379)$. Our model has a higher PSNR which indicates a better smoothing or prediction performance in terms of SNR reduction and smoothing. Our de facto standard, Kernel smoothing gets a PSNR of 11.762 with a bandwidth of radius 3.

This method was repeated for a varying number of knot locations on the reduced images and for three difference classes of MRI; T1-weighted, T2-weighted and the Fluid Attenuation Inversion Recovery (FLAIR) sequence. Tables 3.1 - 3.3 give the PSNR for each case. We note that our Flexible Reduced Rank Spatial model did better than the kernel smoother method in almost all the cases observed. The PSNR in bold is larger and therefore better indicates the method with a better result.

From Table 3.3, we note that the kernel smoothing method did somewhat better than the Flexible Reduced Rank Spatial Model, however the ratio is not very significant.
Table 3.1: Results for PSNR in T1-weighted Images

<table>
<thead>
<tr>
<th>Knots Locations</th>
<th>16</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel Smoothing/FRK</td>
<td>11.76</td>
<td>15.72</td>
<td>21.51</td>
<td>30.61</td>
</tr>
<tr>
<td>FRRSM</td>
<td><strong>19.88</strong></td>
<td><strong>20.39</strong></td>
<td><strong>25.34</strong></td>
<td><strong>33.63</strong></td>
</tr>
</tbody>
</table>

Table 3.2: Results for PSNR in T2-weighted Images

<table>
<thead>
<tr>
<th>Knots Locations</th>
<th>16</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel Smoothing/FRK</td>
<td>14.32</td>
<td>18.76</td>
<td>20.45</td>
<td>25.34</td>
</tr>
<tr>
<td>FRRSM</td>
<td><strong>20.61</strong></td>
<td><strong>22.03</strong></td>
<td><strong>25.88</strong></td>
<td><strong>34.11</strong></td>
</tr>
</tbody>
</table>

Table 3.3: Results for PSNR in FLAIR Images

<table>
<thead>
<tr>
<th>Knots Locations</th>
<th>16</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel Smoothing/FRK</td>
<td><strong>18.51</strong></td>
<td>22.65</td>
<td>24.72</td>
<td>33.29</td>
</tr>
<tr>
<td>FRRSM</td>
<td>17.04</td>
<td><strong>25.55</strong></td>
<td><strong>28.37</strong></td>
<td><strong>35.68</strong></td>
</tr>
</tbody>
</table>
Chapter 4

Future Work

The size of brain image data even for a single brain is overwhelming. For this work it is clear that there are quite a few limitations if we want to perform a complete smoothing for a population.

In the future, we may consider how to implement smoothing for the entire brain without having to reduce the data to visible regions of tumor. This can be a problem if the brain tumor is not so visible from the MRI scanner. Literatures in this area of image analysis is quite rare and developing improved methodology can be computationally expensive. There have been a number of discussions about considering slices of brain as time series, raising questions about the thickness of slices and also slice uniformity. In the future, it will be important to carefully look at the problem and see whether the FRRSM can be extended to solve perform whole brain smoothing without treating each slice as a separate domain on it’s own.

An hierarchical structure is quite plausible in this case and might be extended to population studies using the flexible Reduced Rank Spatial Model or some variations of it.
Appendix A

R Codes and Packages Used

###########################################################################
########## Spatial Random Effect(SRE) Technique for Denoising Brain MRI
###########################################################################

rm(list=ls())
dev.off()

## Some key Required Packages##

library(oro.nifti)
library(oro.dicom)
library(R.oo)
library(neurobase)
library(fslr)
library(tiff)
library(mcmc)
library(MCMCpack)
library(combinat)
library(HI)
library(distr)
library(GeneralizedHyperbolic)
library(SpatialTools)
library(geosphere)

options(fsl.path=" ")
have.fsl()

options(fsl.outputtype = "NIFTI_GZ")

#############################################################
## Convert file from DICOM to NIfTI
#############################################################
setwd("D:/Dissertation_leo_PhD/BRAINIX/DICOM")
## Read in DICOM files--- in 2D slices--- each file contain 1 slice
T1_slices = readDICOM("T1/")
FLAIR_slices = readDICOM("FLAIR/")
T2_slices = readDICOM("T2/")
###convert to NIfTI
nii_T1 = dicom2nifti(T1_slices)
nii_T2 = dicom2nifti(T2_slices)
nii_FLAIR = dicom2nifti(FLAIR_slices)
###Dimension of converted files-- now in 3D
d001 = dim(nii_T1)
d002 = dim(nii_T2)
d00F = dim(nii_FLAIR)

## Working with Saved NIfTI file onwards

setwd("D:/Dissertation_leo_PhD/BRAINIX/NIfTI")

T1w = readNIfTI("001_T1.nii.gz", reorient = FALSE)
T2w = readNIfTI("001_T2.nii.gz", reorient = FALSE)
Flair = readNIfTI("001_FLAIR.nii.gz", reorient = FALSE)

### Plot the 3 different image contrast on one plot ####
image(T1w, z=11, plot.type="single", main = "T1-weighted")
image(T2w, z=11, plot.type="single")
image(Flair, z = 11, plot.type = "single")

### Extract coordinate and intensity to create dataframe.

img_T1w = readnii("001_T1.nii.gz")
ind = which(!is.na(img_T1w), arr.ind = TRUE)
ind = cbind(ind, value = img_T1w[ind])
write.csv(ind, "coordinates.csv")

# summary statistics for intensity with zero background and nonzero intensity values.

hist(ind[,4])
non_zeroIntensity <- ind[,4][ind[,4]>20]
hist(non_zeroIntensity, breaks = 100 , prob=T, xlab = "Voxel Intensity > 20 ",
main = " T1-weighted Histogram")

summary(ind)
summary(non_zeroIntensity)

## qqplot for Intensity values > 20 and its LOG transform:

qqnorm(non_zeroIntensity)
qqline(non_zeroIntensity)
qqnorm(log(non_zeroIntensity))
qqline(log(non_zeroIntensity))

########################################################################## PLOTS FOR T1 Intensity ###################################################################
par(mfrow = c(3, 2))
boxplot(ind[,4], xlab = "Intensity plot for T1-weighted Image")

boxplot(non_zeroIntensity, xlab = "Intensity values > 20 for T1 images")

hist(ind[,4], breaks = 100 , prob=T, xlab = "Voxel Intensity ",
main = " T1-weighted Histogram")
hist(non_zeroIntensity, breaks = 100 , prob=T, xlab = "Voxel Intensity > 20 ", main = " T1-weighted Histogram")

qqnorm(ind[,4], main="QQplot for all Intensity values")
qqline(ind[,4])
qqnorm(non_zeroIntensity, main = "QQplot for Intensity values > 20")
qqline(non_zeroIntensity)

hist(indOF1[,4])
##qqnorm(log(non_zeroIntensity))
##qqline(log(non_zeroIntensity))

### Histogram for each slice
## These has been done using the DICOM files

dev.off()

#########################################################################Plot for T2 Intensity#########################################################################
same as above

######################################################################### PLOT for FLAIR Intensity#########################################################################
img_Flair = readnii("001_FLAIR.nii.gz")
ind00F= which(!is.na(img_Flair), arr.ind = TRUE)
ind00F = cbind(ind00F, value = img_Flair[ind00F])
write.csv(ind00F, "coordinates00F.csv")

non_zeroIntensity00F <- ind00F[,4][ind00F[,4]>20]

par(mfrow = c(3, 2))
boxplot(ind00F[,4], xlab = "Intensity plot for Flair-weighted Image")
boxplot(non_zeroIntensity00F, xlab= "Intensity values > 20 for Flair images")

hist(ind00F[,4], breaks = 100 , prob=T, xlab = "Voxel Intensity ", main = " Flair-weighted Histogram")
hist(non_zeroIntensity00F, breaks = 100 , prob=T, xlab = "Voxel Intensity > 20 ", main = " FLAIR Histogram")

qqnorm(ind00F[,4], main="QQplot for all Intensity values")
qqline(ind00F[,4])

qqnorm(non_zeroIntensity00F, main = "QQplot for FLAIR Intensity values > 20")
qqline(non_zeroIntensity00F)

##qqnorm(log(non_zeroIntensity))
##qqline(log(non_zeroIntensity))

### Histogram for each slice
## These has been done using the DICOM files

dev.off()

#########Plot Intensity values vs coordinates (X, Y and Z)##############

head(ind)
head(ind002)
```
head(ind00F)
par(mfrow = c(3, 1))
plot(data = ind, value~dim1)
plot(data = ind, value~dim2)
plot(data = ind, value~dim3)

#### subset slice 11: plot values>30 vs coordinates (x and Y)
indz <- as.data.frame(ind)

data101 <- subset(indz, dim3 == 11 & value > 30,
  select = c(dim1, dim2, value))
head(data101)
plot( data101$value ~ data101$dim1, xlab = " X coordinate",
  ylab = "Intensity values for T1_slice 11")
plot(log(data101$value)~data101$dim1, xlab = " X coordinate",
  ylab = " LOG(Intensity) values for T1_slice 11")

plot( data101$value ~ data101$dim2, xlab = " Y coordinate",
  ylab = "Intensity values for T1_slice 11")
plot(log(data101$value)~data101$dim2, xlab = " Y coordinate",
  ylab = "LOG(Intensity) values for T1_slice 11")

write.csv(data101, "T1subset.csv")

######################### location & Knots ######################
```
### Generate locations over a spatial domain.

Generates a pattern of locations over a multiple of the unit square using regular grid.

\# ns: number of spatial locations to generate.
\# unit: the dimension of the domain, will be \{unit\} x \{unit\}
\# x.num: for the number of locations in the x-direction.
\# y.num: for the number of locations in the y-direction.

\# A matrix of locations is returned.

```r
set.seed(1000913)

gen_loca <- function( method="grid" , ns=NULL , unit=500 , sphere=FALSE, x.num=NULL , y.num=x.num,... ){

# Set ns if not specified
if( is.null(ns) ){
    ns <- 500
}

## Generate locations on a regular grid
##

if( method=="grid" ){
# Get as a regular grid with as close as possible to the given sample size
```
if( is.null(x.num)==T ){
  nsf <- floor( sqrt( ns ) ) - 5
  nsc <- ceiling( sqrt( ns ) ) + 5

  ylens <- xlens <- sort( rep( nsf:nsc , 2 ) )
  ylens <- ylens[ -length(ylens) ]
  xlens <- xlens[ -1 ]
  coords.numbers <- xlens*ylens
  ind1 <- which( abs(coords.numbers-ns) == min(abs(coords.numbers-ns)))
  x.num <- max( xlens[ind1] )
  y.num <- max( ylens[ind1] )
}

x.sep <- unit/x.num
y.sep <- unit/y.num

xknot <- seq( 0 , unit-x.sep , x.sep ) + x.sep/2
yknot <- seq( 0 , unit-y.sep , y.sep ) + y.sep/2
locas <- expand.grid( xknot , yknot )
}

## Return the locas
##
colnames(locas) <- c("Xdirection","Ydirection")
locas <- as.matrix( locas )

return( locas )
}
knots1 <- gen_loca(ns=100)  # Generate knots locations
knots1 <- as.matrix(knots1)
write.csv(knots1, "knots1.csv")

knots2 <- gen_loca(ns=225)  # Generate knots locations
knots2 <- as.matrix(knots2)
write.csv(knots2, "knots2.csv")
coords <- gen_loca(ns=111450)  # Generate data locations
coords <- as.matrix(coords)
coords01 <- coords[-(111455:111556), ]
image(Flair, z = 11, plot.type = "single")
plot(coords01)
par(new=TRUE)
plot(knots1, pch=18, col= "red", axes=FALSE)

## location(1)

#########################################################################
############### Basis Function ( S matrix ) ###############

## coord_Xdirec <- cbind(data101[,1], data101[,2])

### Create the distance measures for locations###
dist_measure <-function( coords1, coords2=coords1, dmethod=SpatialTools::dist2, ... ){
  imc1 <- is.matrix(coords1)
imc2 <- is.matrix(coords2)
## Compute all paired Euclidean distance of the knots location and the data locations

```r
pair_dist <- SpatialTools::dist2(coords1, coords2)
# Return distance
return(pair_dist)
```

####### Construct matrix of basis functions####

####### Radial Basis & Modified Bisquare Basis ####

## Constructs the matrix of basis functions from a set of locations##
## and a set of knots.##

```r
# coords: the coordinates of the data locations.
# knots: the coordinates of the knots.
# mult: multiplier for range of each basis function.
#
# Distances are computed using dis-measure
#
zeros <- function(m, n){ array(0, c(m, n)) }
basis_mat <- function(coords, knots, mult=1.5, ...){
  nd <- nrow(coords)
  nk <- nrow(knots)
  S <- knot_ind <- matrix(0, nd, nk)
  kd_dist <- dist_measure(coords, knots,...)
k_dist <- dist_measure(knots, knots,...)
## Create the S-matrix (bisquare)
for( ii in 1:nk ){
    a1 <- mult*min( k_dist[k_dist[,ii]>0, ii] )
    knot_ind[,ii] <- kd_dist[,ii] <= a1
    S[,ii] <- ( (1 - (kd_dist[,ii]/a1)^2 )^2 )*knot_ind[,ii]
}

## Return the S matrix
return(S)
}

########################################################################
Fit Large Scale Y = XBeta####################

# Xmat1 = as.matrix( read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/Xmat201.csv", header = FALSE))
# Xmat2 = as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/Xmat202.csv", header = FALSE))
# Xmat3 = as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/Xmat203.csv", header = FALSE))
IntensityT1_11 =as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/T1subset.csv", header = FALSE))
colnames(IntensityT1_11) <- c("voxel","Xdirec", "Ydirec", "Intensity values")
#knots1 = as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/knots1.csv", header = FALSE))
#knots2 = as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/knots2.csv", header = FALSE))
knots1 <- gen_loca(ns=100) # Generate knots locations
knots1 <-as.matrix(knots1)
S1 <- basis_mat(coords01,knots1)

knots2 <- gen_loca(ns=225)  # Generate knots locations
knots2 <- as.matrix(knots2)

S2 <- basis_mat(coords01,knots2)

# Xmat6 = as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/Xmat83.csv", header = FALSE))
# X1 = t(Xmat1)
# X2 =t(Xmat2)
# X3 = t(Xmat3)
Xmat5 = as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/Xmat82.csv", header = FALSE))
### Use this for Proposal##
X5 = t(Xmat5)
Y = as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/Intensity.csv", header = FALSE))
X = as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/X.csv", header = FALSE))
Yhat = as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/Yhat8.csv", header = FALSE))
Beta = as.matrix(read.csv
("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/beta8.csv"), header = FALSE)
DataXY = cbind(X,Y)
plot(DataXY[,1],DataXY[,2], xlab = " Y direction", ylab = "Intensity")
lines(X, Yhat, col = 'red', lty = 1)

Resid = Y - Yhat
#twolines = function(x,y) {
##abline(line(x,y),col="red")
#abline(lsfit(x,y),col="blue")
#}
# XY5 <- t(X5)%*%Y
#XX5 <- t(X5)%*%X5
# beta105 <- solve(XX5) %*% XY5 ## Solve in Matlab
# Yhat <-X5 %*% beta105 ## Matlab Result
# R105 <- Y - X5 %*% beta105
nd <- dim(S1)[1]
nk <- dim(S1)[2]
n.obs <- 111454
p <- dim(X5)[2]

## Estimate of the measurement error

ME.ind <- sample(1:nd, 20, replace=T)
ME.loc <- IntensityT1_11[ME.ind,2:3]
ME.est <- vector()

for( ind1 in 1:dim(ME.loc)[1] ){
ind2 <- abs(IntensityT1_11[,2]-ME.loc[ind1,1])<(2) &
abs(IntensityT1_11[,3]-ME.loc[ind1,2])<(2)
local.points <- Y[ind2==1]
ME.est[ind1] <- var(local.points)
}
m.error <- median(ME.est)
m.error
m.error <- 6500.575
## Defining Some Required Functions ######
##
zeros <- function(m, n){ array(0, c(m, n))  }
eye <- function(m){ diag(1,m)  }
sqrtm <- function (A){
e <- eigen(A)
ou <- e$vectors %*% diag(sqrt(e$values)) %*% t(e$vectors)
return(out)
}
uoco <- function(B){
ME <- eigen(B)
ME$values[ME$values <= 0] <- 0
out <- ME$vectors%*%diag(ME$values)%*%solve(ME$vectors)
return(out)
}
Wk.Ind <- function(w,m) { all(w > 0) * all(w <1000) }

Wk.fn <- function(w,m) {
aa <- m - sum(w)

} 

Preds <- Yhat 

# Set up results objects
n.pred <- dim(Preds)[1]
batch.A <- 110  # Number of batches (110 for full run)
batch.B <- 110  # Size of each batch (110 for full run)
sigma1_sum <- vector()
sigma1_s   <- vector()
Yhat01   <- zeros(n.pred,1)
Yk.batch  <- matrix(0, nrow=n.pred , ncol=batch.A)
batch.ind <- 1

##### Computing Expressions outside MCMC Loop #####
SS <- t(S1)%*%S1
SY <- t(S1)%*%Y
SX <- t(S1)%*%X5
XY <- t(X5)%*%Y
XX <- t(X5)%*%X5

##### hyperparameters #####
nu <- 2
Ak <- 10^-5
tau.beta <- 10^-6  
a.sigma  <- 10^-10

######################################## Initial Values ########################################

beta1 <- Beta
R1    <- Resid
p.error <- 1
sigma1 <- p.error + m.error

ul <- solve(SS)%*%t(S1)%*%R1

W    <- rep( 1 , nk )
Wm   <- diag( sqrt(W) )
ak   <- rep(1, nk)

### MCMC sample size
MNK.fit <- 2900  # Start fitting after this point (2900 for full run)
MNK    <- batch.A*batch.B + MNK.fit

######################################## Defining Storing Variables ########################################
sigma1_s <- array(NA, c(1,MNK))
sigma1_sum <- 0
Yfit    <- 0*S[,1]

#############################################################
for(j in 1:MNK){

## Sample V^{-1} (and obtain V from it)
InvWish.B <- 2*nu*diag(1/ak) + nuco(u1 %*% t(u1))
Vin <- rwish( nu+nk , solve(InvWish.B) )
V <- solve(Vin)

## Sample the gammas (prior for V)
ak <- 1/rgamma( n=nk , shape=(nu+nk)/2 , rate = nu*diag(Vin) + 1/(Ak^2) )

## Sample W
# W <- ARMS FROM REX
m.arms <- t(R1) %*% R1/2
W1 <- arms( y.start=W , myldens=Wk.fn , indFunc=Wk.Ind ,
            n.sample=nk , m=m.arms )
W <- colMeans(W1)
Wm <- diag( sqrt(W) )

## Sample beta and U simultaneously
Gminv1 <- cbind( XX/sigma1 + (1/tau.beta)*eye(p) ,
                 t(Wm%*%SX)/sigma1 )
Gminv2 <- cbind( Wm%*%SX/sigma1 , (Wm%*%SS%*%Wm)/sigma1 + Vin )
Gminv <- rbind( Gminv1 , Gminv2 )
Gminv <- as.matrix(Gminv)
write.csv(Gminv, "Gminv.cvs")  ## change here
# Gm <- solve( Gminv)
Gm <- as.matrix(read.csv("D:/Dissertation_leo_PhD/BRAINIX/NIfTI/Gm.csv", header = FALSE))
Mu <- Gm %*% rbind( XY/sigma1 , Wm%*%SY/sigma1 )
BU <- as.numeric(mvrnorm(n=1, Mu, Gm))

beta1 <- BU[1:p]

u1 <- BU[(p+1):(p+nk)]

R1 <- Y - X5%*%beta1 - (S1 %*% Wm %*% u1)

## Sample the process error
RS1 <- 0.5*sum((R1)^2)
p.error <- 1/rgamma(1, shape = nd/2 + 1, rate = (RS1+ 2/a.sigma))
a.sigma <- 1/rgamma(1, shape = 3/2, rate = (2/sigma1 + 1/Ak^-2))
sigma1 <- p.error + m.error

## Estimate the process error
mu.R1 <- R1*((p.error*m.error)/(p.error + m.error))
pred.R1 <- mu.R1 + rnorm(n = nd, mean=0,
                      sd=sqrt((p.error*m.error)/(p.error + m.error)))

## Batch Means
if( j > MNK.fit ){  
j.batch <- j - MNK.fit # Adjust the index

Yhat[observed==1] <- X%*%beta1 + (S %*% Wm %*% u1) + pred.R1
proc.err <- rnorm(n = n.pred-n.obs, mean = 0, sd = sqrt(p.error))
Yhat[observed==0] <- X.full[observed==0,%*%beta1 + (S.full[observed==0,] %*% Wm %*% u1) + proc.err

Yk.batch[,batch.ind] <- Yk.batch[,batch.ind] + Yhat
sigma1_sum <- sigma1_sum + Vin

if( j.batch %% batch.B == 0 ){
Yk.batch[,batch.ind] <- (1/batch.B)*Yk.batch[,batch.ind]
batch.ind <- batch.ind+1
}
}

# sigma1_s[j] <- sigma1

print( "================================================" )
print( paste( "Iteration = " , j , sep=""))

}

## Calculate batch-means standard error
Yk.batch.sum <- rowSums( (Yk.batch - rowMeans(Yk.batch))^2 )
*Yk.batch.sum )
batch.se <- sqrt( ((batch.B)/(batch.A-1))*Yk.batch.sum )

setwd( alt.wd )
Preds[,3] <- rowMeans(Yk.batch)
Preds[,4] <- batch.se

colnames(Preds) <- c("Xdirec" , "Ydirec" , "Y" , "RMSPE")
write.csv( Preds , file="PredsData.csv" )
setwd( cur.wd )
# plot( 1:10 , p.error_s[1:10] , type="l")

#UpScaled T1 slice-13

setwd("D:/Dissertation_leo_PhD/BRAINIX/NIfTI")
R_t1 = readNIfTI("001_T1.nii.gz", reorient = FALSE)
image(R_t1, z=13,plot.type="single", main ="T1-weighted slice-13")

###csv.
ind_t1 = which(!is.na(R_t1), arr.ind = TRUE)
ind_t1 = cbind(ind_t1, value = R_t1[ind_t1])
write.csv(ind_t1, "table_t1.csv")

###END
### csv_slice 13

R_slice13 = R_t1[, , z=13]

image(R_slice13,col =gray(0:64/64))

ind_slice13 = which(!is.na(R_slice13), arr.ind = TRUE)
ind_slice13 = cbind(ind_slice13, value = R_slice13[ind_slice13])
write.csv(ind_slice13, "table_slice13.csv")

### Upscale by 4 pixels (Centroid)

R_s13new = read.csv("table_slice13.csv")

# create blocks
blocks <- seq(1,262148,4)

# get mean per block
sz <- sapply(1:(length(blocks)-1),
function(i)
mean(R_s13new[ R_s13new$Id>=blocks[i] &
R_s13new$Id < blocks[i+1], "value"]))

Value <- rep(sz, each = 4)

R_s13Av = cbind(R_s13new[, -c(1,4)],Value)

write.csv(R_s13Av, "Upscaled_T1.csv")

######################## This code below is more efficient: ########################

dat1 = read.csv("table_slice13.csv", header = TRUE)
data=data.frame(dat1)
data_mean=as.matrix(data)
d=dim(data_mean)[1]/4
for (i in 0:(d-1)){
data_mean[(1+4*i):(4+4*i),4]=mean(data_mean[(1+4*i):(4+4*i),4])
}
data_mean
write.csv()
## Image of Upscaled:

```r
Inten <- R_s13Av$Value ; X = as.matrix(R_s13Av$row) ; Y = as.matrix(R_s13Av$col)
Inten = matrix(Inten, 512, 512)
plot(Inten)

image(Inten, col = grey(0:64/64))
```

### Reduced Data to Tumor Region

Reduced Data Set up

```r
mridir <- "C:/Users/LEONEXUS/Dropbox/Upscaled_Image"
FLAIR <- readNIfTI(file.path(mridir,"/FLAIR.nii"),reorient=FALSE)
orthographic(FLAIR)
image(FLAIR)
mask202 <- readNIfTI(file.path(mridir,"/FLAIR 202_mask.nii"), reorient=FALSE)
image(mask202)
orthographic(mask202, xyz = c(196,133,12))
# this is a binary 0-1 image, reps area we may want to extra from an original Flair image
#How do we mask: multiply by the original image everything we don't want will be set to zero:
#note all two will be of the same size(dimension)
masked.FLAIR <- FLAIR*mask202
image(masked.FLAIR)
orthographic(masked.FLAIR, xyz = c(196,133,12))
ind0F1= which(!is.na(masked.FLAIR), arr.ind = TRUE)
```
ind0F1 = cbind(ind0F1, value = masked.FLAIR[ind0F1])
write.csv(ind0F1, "ReducedDATA203.csv")

par(mfrow=c(2,1))
hist(masked.FLAIR, main= "Reduced Data Histogram" , xlab = "Intensity")
hist(masked.FLAIR[masked.FLAIR > 30],
   main = " Reduced with Intensity >30", xlab = "Intensity >30")
qqplot(masked.FLAIR)

dataM =read.csv('predictions.csv')
par(mfrow=c(2,1))
dataM001 = dataM[,4][dataM[,4] < 4e+06]
hist(dataM001, freq = F, ylab = 'Density',
   xlab = "Residuals", main = " Prediction Residuals for MCCLT parameter Estimation")
boxplot(dataM001, horizontal = T)
dev.off()
image()
qqnorm(ind0F1[,4], main="QQplot for all Intensity values Reduced Data")
qqline(ind0F1[,4])
Appendix B

HSIRB Project Approval Letter

17-04-20
Date: April 18, 2017

To: Rajib Paul, Principal Investigator
    Leonard Johnson, Student Investigator for Dissertation

From: Amy Naugle, Ph.D., Chair

Re: HSIRB Project Number 17-04-20

This letter will serve as confirmation that your research project titled “Denoising Large Neuroimaging Data on Brain Tumor Using Spatial Random Effects Model on Sphere” has been approved under the exempt category of review by the Human Subjects Institutional Review Board. The conditions and duration of this approval are specified in the Policies of Western Michigan University. You may now begin to implement the research as described in the application.

Please note: This research may only be conducted exactly in the form it was approved. You must seek specific board approval for any changes in this project (e.g., you must request a post approval change to enroll subjects beyond the number stated in your application under “Number of subjects you want to complete the study”). Failure to obtain approval for changes will result in a protocol deviation. In addition, if there are any unanticipated adverse reactions or unanticipated events associated with the conduct of this research, you should immediately suspend the project and contact the Chair of the HSIRB for consultation.

Reapproval of the project is required if it extends beyond the termination date stated below.

The Board wishes you success in the pursuit of your research goals.

Approval Termination: April 17, 2018
References


Paul, C. M. Jelsema, and K. W. Lau. A flexible class of reduced rank spatial models for large non-gaussian dataset. 2015.


