



Faculty of Electrical and Computer Engineering
Institute of Circuits and Systems (IEE)
Chair of Fundamentals of Electrical Engineering

Enhancing intraoperative neuroimaging by incorporating spatial regularization into semiparametric regression models

Nishant Kumar

Master Thesis

Supervisors:

Prof. Dr. phil. nat. habil. Ronald Tetzlaff
Dr. rer. nat. Nico Hoffmann

Date and Place:

11-05-2018 Dresden

Declaration of Authorship

I, Nishant Kumar, declare that:

- This work was done completely by me while in candidature for a master degree at the Technische Universität Dresden.
- The thoughts and ideas taken from the published work of others have been indicated.
- In the selection and evaluation of the work as well as in the production of the document, I have received support from Dr. Nico Hoffmann.
- I have explicitly acknowledged all the other sources of help.
- This thesis was not previously presented to another examination board and has not been published.
- I accept the current Master degree regulations of the Faculty of Electrical and Computer Engineering, Technische Universität Dresden.

Place & Date

Signature

“Imagination is more important than knowledge”.

Albert Einstein

Abstract

The development of intraoperative imaging-guided neurosurgery represents a substantial improvement in the microsurgical treatment of malignant tissues in the human brain. The changes in regional cerebral blood flow dramatically alter the emitted heat radiation of the cortex leading to non-linear random behavior. In this work, semiparametric regression model has been developed to combat this non-linearity. This model adds the deterministic or parametric components of the state of the art Generalized Linear Models (GLM) with non-parametric components such as P-splines. To model spatial-temporal interactions in the thermographic brain imaging data, the semiparametric regression framework has been extended by Markov random field (MRF) components. The MRF requires fast inference schemes such as Tree-reweighted message passing (TRWS) to fulfill intra-operative performance requirements. The advancements proposed in this work should aid the neurosurgeons to achieve accurate detection of intraoperative neuronal activity in the somatosensory cortex with the removal of background noise from the surroundings.

Acknowledgement

First and foremost, I must thank my adviser and supervisor, Dr. Nico Hoffmann, for his support and guidance throughout this project. It has been a privilege to be able to draw on Nico's vast experience and breadth of knowledge. My research has greatly benefited from his keen insights and critical feedback over the months. I would also like to thank Prof. Dr. Ronald Tetzlaff for his support as my primary supervisor and for making this thesis possible. I was fortunate to have him supervise my work. A special thanks to Prof. Dr. Stefan Gumhold for the brief but fruitful discussions.

I would like to thank Dr. Andre Lange for providing me the excellent opportunity to work as a part-time researcher in a fast-paced environment of Fraunhofer and quickly adjust to the German culture and ethics. My stints with Samsung India Electronics Ltd and HCL Technologies Ltd turned out to be the building blocks of my career. Therefore, I would like to especially thank Mr. Vaibhav Verma and Mr. Purnananda Kumar Divakaruni for their guidance during my early professional life.

On a personal note, I would also like to thank my parents and Prachi for their unwavering confidence in my abilities and their encouragement in all of my endeavors while giving me the spiritual and emotional strength to cross one of the most challenging times of my life. They always motivated me to strive for perfection. I am grateful for their love and support. I like to thank many amazing folks outside my research area that I have had the opportunity to meet along the way during my Master studies. Thank you, everyone.

Contents

List of Figures	viii
List of Tables	xii
1 Introduction	1
1.1 Motivation	1
1.2 Intraoperative Thermal Neuroimaging	2
1.3 Applications and Summary	3
2 Related Work	5
2.1 Summary	15
3 Background theory	16
3.1 B-splines	16
3.2 Discretization of spline coefficients	17
3.2.1 Gaussian Mixture Model (GMM)	17
3.2.1.1 Expectation Maximization Algorithm	18
3.3 Spherically Invariant Random process (SIRP)	19
3.3.1 Meijer G-function	20
4 Semiparametric regression	21
4.1 The Univariate spline regression Model	21
4.1.1 Parametric component	21
4.1.2 Non-parametric components as B-splines	22
4.1.3 Penalized B-splines	22
4.2 Optimal Lagrange Multiplier λ	23
4.3 Computing Z-statistics	25
4.4 Summary	25
5 Spatial Regularization using Markov Random fields	26
5.1 MRF based on Lagrange Multiplier λ	26
5.1.1 Unary potentials	26
5.1.2 Pairwise potentials	28
5.1.3 Analysis of pairwise potential weight γ	29
5.1.4 Computing new Z-statistics	29
5.1.5 Drawbacks	29
5.2 MRF based on Z-statistics	29
5.2.1 Unary potentials	30
5.2.2 Pairwise potentials	30
5.2.3 Advantages	30
5.3 MRF on spline coefficients	31
5.3.1 Single non-parametric component	31
5.3.1.1 Unary potentials	31

5.3.1.2	Pairwise potentials	32
5.3.1.3	Refit of the parametric component	33
5.3.1.4	Drawbacks	34
5.3.2	Two non-parametric components	35
5.3.2.1	Unary potentials	36
5.3.2.2	Pairwise potentials	37
5.3.2.3	Refit of the parametric component	37
6	Performance Evaluation	39
6.1	Spectral Analysis	40
6.1.1	Analysis of Gaussian activity pattern	40
6.1.2	Analysis of raw vs. superimposed data	41
6.1.3	Sum of Square Error (SSE) analysis	42
6.2	Number of Gaussian components for Discretization	43
6.3	Accuracy Analysis	44
6.3.1	Univariate spline regression model	44
6.3.2	MRF on Lagrange Multiplier λ	44
6.3.3	MRF on spline coefficients	45
6.3.4	Effect of pairwise potential weight vs Accuracy	45
6.4	TRWS runtime measurement	51
6.5	General runtime measurements	52
7	Discussions and Conclusions	54
8	Future Work	55
8.1	Superpixel based Markov random fields	55
8.2	GPU implementation of Gaussian Mixture Model	55
9	Bibliography	56
Appendix		62
A	OpenGM implementation of Graphical Model	63
B	Principal component analysis (PCA)	65
C	K-means clustering	66

List of Figures

1.1	(a) White light image of the exposed human cortex (b) Thermographic image of the cortex	2
1.2	The time series of the 25 th pixel in the raw intraoperative data with $n = 1024$ time points.	3
1.3	The Gaussian activation pattern with $\mu = 30$ and $\sigma = 10$. An alternate cycles of rest and stimuli is visible. Higher levels of response indicate stimulus while the lower levels indicate rest.	3
1.4	A typical CBF response to brief neuronal activation	4
2.1	The left figure is an undirected graph ($(\mathbf{X}, \mathbf{G}) = (\mathbf{V}, \mathbf{E})$) where circles resembles a set of nodes \mathbf{V} and the lines connecting the nodes are the set of edges \mathbf{E} . The random variables \mathbf{X} are indexed on the nodes. The right figure shows a first order neighborhood systems of the left graph where red node e has green neighboring nodes b, d, f and h	8
2.2	(a) shows a first order neighborhood system while (b)-(d) shows different types of cliques possible with different combination of nodes.	9
2.3	(a) shows a typical thermographic image of the cortex , (b) is the representative factor graph of a small region of the image with 5 pixels and (c) shows the resultant label diagram of the image after finding the maximum a posteriori probability.	12
2.4	A descriptive diagram of flow of messages between two nodes during an inward pass towards root node.	14
2.5	A typical TRWS Algorithm	14
2.6	An example TRWS and its monotonic chains	15
3.1	a) shows an example of a quadratic curve b) and (c) are its two polynomial segments. The blue dots represent the control points.	17
3.2	An example of a Gaussian mixture model with 4 multivariate Gaussian distributions	18
3.3	Expectation Maximization Algorithm	19
3.4	The left figure is the image of our raw data at 0 th time point. The right figure shows the discretized pixels with a label using Gaussian Mixture Model with $k = 12$ Gaussians components and $p = 431$ spline control points.	19
4.1	A typical sequence of brain activation starting from stimuli till the temperature changes leading to heat radiation from the active sites.	21
4.2	An overview of the univariate semiparametric regression model for the modelling of functional intraoperative spatial temporal data	23
4.3	The plot between different values of Lagrange Multiplier λ and corresponding values of Akaike Information criterion, $AIC(\lambda)$ for the pixel (say 306869 th).	24
4.4	Time series of the pixel 306869 th . The blue dots resembles the original signal \mathbf{Y} for $n = 1024$ time points. The red line shows the semiparametric regression fit $\hat{\mathbf{Y}}$ using the optimum AIC score.	24

5.1	An overview of the proposed MRF model based on Lagrange Multiplier λ for modelling of intraoperative thermal imaging data	27
5.2	Unary potentials assigned to each labels of pixel p and q for MRF based on λ	28
5.3	The interaction between labels of adjacent pixels. In this figure, k labels are defined on each pixel.	28
5.4	The univariate spline regression model extended by an MRF model on Z-statistics.	30
5.5	Unary potentials assigned to each labels of pixel p and q respectively	30
5.6	Figure showing the interaction between labels of adjacent pixels for MRF on Z values.	30
5.7	μ_{k-1} represents mean value for the Gaussian component $k - 1$	31
5.8	An overview of the proposed MRF model based on a single non-parametric component.	32
5.9	The unary potentials defined for k Gaussian components at pixels p and q	33
5.10	The left figure shows the labels assigned to each pixel of our intraoperative data by the Gaussian Mixture Model. The right figure shows the inferred labels after MAP estimate using TRWS algorithm. The results are for $p = 431$ control points, $m = 307200$ pixels, $k = 3$ components, $\gamma = 1$ and $n = 1024$ time points. A relative smoothness has been observed in the second figure.	33
5.11	The figure shows the time series values of \mathbf{Y} , $\hat{\mathbf{Y}}_{mrf}$, and $\hat{\mathbf{Y}}_{fit}$ for a pixel (say 306869 th). It is visualized from the figure that $\hat{\mathbf{Y}}_{mrf}$ results in a rough estimate of \mathbf{Y} . After compensating $\hat{\mathbf{Y}}_{mrf}$ from \mathbf{Y} , $\hat{\mathbf{Y}}_{fit}$ is obtained which is centered around zero. The figure only shows the first 200 time points out of total $n = 1024$ time points for better visualization.	35
5.12	An overview diagram of the proposed MRF model based on spline coefficients of a single non-parametric component in a two non-parametric component spline regression framework.	37
6.1	Ground truth showing four circular areas of neuronal activity	39
6.2	Energy contribution vs Number of spectral components	41
6.3	Figure shows the plot between the number of components and their respective AIC scores	43
6.4	Analysis of the accuracy on four different models.	47
6.5	The left figure shows the MRF on Z values at $\gamma = 0.1$ with $p = 81$ control points. The right figure shows the result with $\gamma = 10$	48
6.6	The effect of pairwise potential weight on overall accuracy for three models after incorporation of spatial regularization on Z values. Blue bar plots are for spatial regularization of only Z values, Green bar plots are for spatial regularization of both spline coefficients and Z values in a one non-parametric component framework, Red bar plots are for spatial regularization of both spline coefficients and Z values in a two non-parametric component framework.	49
6.7	Spatial Z value map of univariate spline regression framework without any spatial regularization. The results are for $p = 431$ control points.	50
6.8	Spatial Z value map after spatial regularization on spline coefficients as well as on Z values. The results are for two non-parametric components framework. $p = 431$ control points, $\gamma = 1.4$	50
6.9	TRWS Energy and bound curve with respect to number of steps for MRF on spline coefficients. The result are for the semiparametric regression framework containing two non-parametric components but only one non-parametric component is spatially regularized. $p = 431$ control points and $k = 12$ Gaussian components and $\gamma = 1$	51
6.10	TRWS Energy and bound curve with respect to runtime for MRF on spline coefficients. The result are for the semiparametric regression framework containing two non-parametric components but only one non-parametric component is spatially regularized. $p = 431$ control points and $k = 12$ Gaussian components and $\gamma = 1$	51
6.11	TRWS Energy and bound curve with respect to number of steps for MRF on Z values. $k = 2$ components, and $\gamma = 1.4$	52

LIST OF FIGURES

6.12 TRWS Energy and bound curve with respect to runtime for MRF on Z values. $k = 2$ components, and $\gamma = 1.4$	52
A.1 Defining graphical model and running inference in OpenGM	64
C.1 K-Means Algorithm	66
C.2 a) shows the image of our raw data at 0 th time point. b) shows the discretization map of $m = 307200$ pixels using K-means algorithm with $k = 12$ clusters. Each color represents one of the 12 cluster labels assigned to each pixel. The result is obtained using $p = 431$ control points.	67

List of Tables

2.1	Average Performance Metrics for 50 Replicates of the fMRI dataset simulation based on the Ising and Gaussian priors. The metrics corresponding to the best performing prior are in bold. The table shows the % of voxels which were misclassified at different noise levels.	11
6.1	The table shows total energy at target sites for all 512 components as well as relevant [50, 100] components. The values are for both raw and smoothed raw data. γ are the pairwise potential weight. The results are for $p = 431$ control points and $k = 12$ Gaussian components for discretization.	42
6.2	The table shows total energy at target sites for all 512 components as well as relevant [50, 100] components. The values are for both superimposed data and spatially smoothed superimposed data. γ are the pairwise potential weight. The results are for $p = 431$ control points and $k = 12$ Gaussian components for discretization.	42
6.3	The table showing Sum of squares Error (SSE) values. The second column calculates the SSE between the raw and the superimposed data. The third column calculates the SSE between the raw and the superimposed data after 3D spline smoothing. The fourth column calculates the SSE between the raw and the superimposed data after spatial regularization based on our proposed model.	42
6.4	Quantitative analysis of existing univariate spline regression model for different values of control points p	44
6.5	Quantitative analysis of our proposed spatial regularization on Lagrange Multiplier λ . Pairwise potential weight γ is 1	45
6.6	Quantitative analysis of our proposed spatially regularized semiparametric regression model with single non-parametric component. GMM discretization and subsequent spatial regularization is done on the spline coefficients of the single non-parametric component. Pairwise potential weight γ is 1 and $k = 12$ components.	46
6.7	Quantitative analysis of our proposed spatially regularized semiparametric regression model with two non-parametric components. GMM discretization and subsequent spatial regularization is done on spline coefficients of only single non-parametric component. Pairwise potential weight γ is 1 and $k = 12$ components.	46
6.8	Quantitative analysis of spatial regularization of Z values in an existing univariate spline regression framework. The results are for $p = 431$ control points.	47
6.9	Quantitative analysis of spatial regularization on Z values in a univariate spline regression framework with discretization and spatial regularization of spline coefficients of one non-parametric component already incorporated. The results are for $p = 431$ control points, $k = 12$ Gaussian components and pairwise potential weight for MRF on one non-parametric component is 1.	48

LIST OF TABLES

6.10 Quantitative analysis of spatial regularization of Z values in a univariate spline regression framework containing two non-parametric components with discretization and spatial regularization of spline coefficients of one non-parametric component already incorporated. The results are for $p = 431$ control points, $k = 12$ Gaussian components and pairwise potential weight for MRF on one non-parametric component is 1.	49
6.11 The runtime results are for $p = 431$ control points, $k = 12$ Gaussian components and $\gamma = 1.4$ for MRF on Z values	53

Chapter 1

Introduction

1.1 Motivation

The human brain is one of the most intricate and vital organs in the human body consisting of nerve cells known as neurons. Neurons receive a stimulus and carry electrochemical pulses called action potentials to various regions of the brain or body. It is this mechanism that controls most of the activities in the human body. Action potentials get generated by voltage-gated ion channels embedded in a cell's plasma membrane. These channels are closed when the action potential is below the threshold voltage, but they are opened if the potential increases to a precisely defined threshold voltage [BL07]. Hence, neuronal activity occurs when the action potential in neurons is above this threshold voltage [BL07]. However, it is quite challenging to image the neuronal activity as the action potentials cause only small electrical signals.

There is an extensive variety of brain disorders that dramatically affects the memory, sensation, and even the personality of the concerned person. The brain tumor is one such disorder which is caused by the unrestricted growth of cells in the brain. Glioblastoma is one of the categories of brain tumor which originates from the uncontrolled growth of Glial cells [PH06]. This growth generates pathophysiological symptoms like edema [PH06]. World Health Organisation (WHO) [LPR⁺16] classifies malignancy grade of tumors based on mean survival times [PH06]. Malignant tumors are one of the most dangerous types of tumors with fast growth rate and survival time between 6 to 15 months only. Ionizing radiation is one of the major causes of brain tumors. In Germany alone, approximately 6,920 residents develop brain tumor every year [KSH⁺13].

Typically, the medical treatment of such tumor involves neurosurgery supported by neuroimaging and brain mapping. Neuroimaging consists of two different type of procedures namely Structural and Functional Imaging. Structural imaging techniques such as Computed Tomography (CT) and Ultrasonography helps in visualization of disorders such as tumors. Functional imaging techniques such as Optical imaging, Functional magnetic resonance imaging (fMRI) and Electroencephalography (EEG) aids in visualization of neuronal activity. However, each neuroimaging technique cannot achieve both the functions together. Intraoperative thermal neuroimaging is one such method which provides means to analyze both the structural and functional information. It assesses the emitted heat radiation from the exposed human brain and transforms them into temperature values [GHK⁺03] [STG⁺93]. Figure 1.1(b) shows a thermographic image of the exposed human cortex of a 33-year-old patient [JH14].

The intraoperative thermal neuroimaging experiment involves exposing the human brain to a temporal sequence of stimulus conditions while repeated scans of the brain's region of interest are taken [Glo11]. This process alternates between equal length periods of stimulus and rest for many cycles. In our case, the patient's central nerve receives electrical stimulations based on a fixed protocol. This procedure gets repeated for a set time period. The local neurovascular coupling results in detectable changes in blood flow [HKP⁺17]. The recently activated sites are replenished with the blood flow allowing the identification of the neuronal activity in specific regions of human

brain[Glo11].

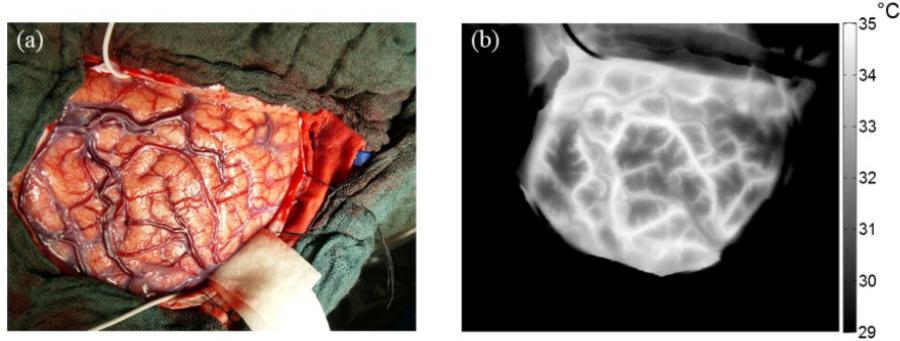


Figure 1.1: (a) White light image of the exposed human cortex (b) Thermographic image of the cortex [JH14]

1.2 Intraoperative Thermal Neuroimaging

Thermal imaging is readily influenced by environmental noise which makes it challenging to detect neuronal activity in the exposed human cortex. Hysteresis effects in the detector hardware elevate the noise effects [HKP⁺¹⁷]. The infrared thermographic camera consists of a focal-plane array of microbolometer detectors that convert the emitted electromagnetic radiation of infrared range into small changes in the electrical resistance [HKP⁺¹⁷]. These changes result in temperature values. It leads to heating up of the camera and its body until it reaches a convergence temperature. The resultant behavior induces non-linear temperature drifts into the data recordings that are compensated by a periodic gain and offset correction also known as non-uniformity correction (NUC) [doi10]. The data from the microbolometer is normalized by the non-uniformity correction and gives temperature and drift normalized image just after the NUC [HKP⁺¹⁷].

The data we are considering in this thesis consists of sequences or time series of the intraoperative thermal imaging. Let $\mathbf{n} \in [0, 1, \dots, n - 1]$ indicates the number of time points in the sequence, $\mathbf{m} \in [0, 1, \dots, m - 1]$ conveys the number of pixels at each time point, then $\mathbf{Y} \in \mathbb{R}^{n*m}$ denotes the response variable consisting of temperature values. In this thesis, each image has $\mathbf{m} = 480 \times 640$ pixels. The rows of the response variable \mathbf{Y} represents pixels at a particular time point, and each column represents the time series of responses at individual pixels.

Figure 1.2 shows a time series of a single pixel (say 25th) with $\mathbf{n} = 1024$ time points. The response variable, i.e., the temperature recordings in the vertical axis of the figure has been normalized to zero mean. We discussed that the experimental subject is scanned while being exposed to alternating periods of rest and stimuli. So, we use Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ as our synthetic activation pattern to evoke neuronal activity with predefined mean $\mu \in \mathbb{R}$ and variance $\sigma^2 \in \mathbb{R}$. Figure 1.3 shows the Gaussian activation pattern which is superimposed with the raw intraoperative thermal imaging data acting as our baseline raw dataset.

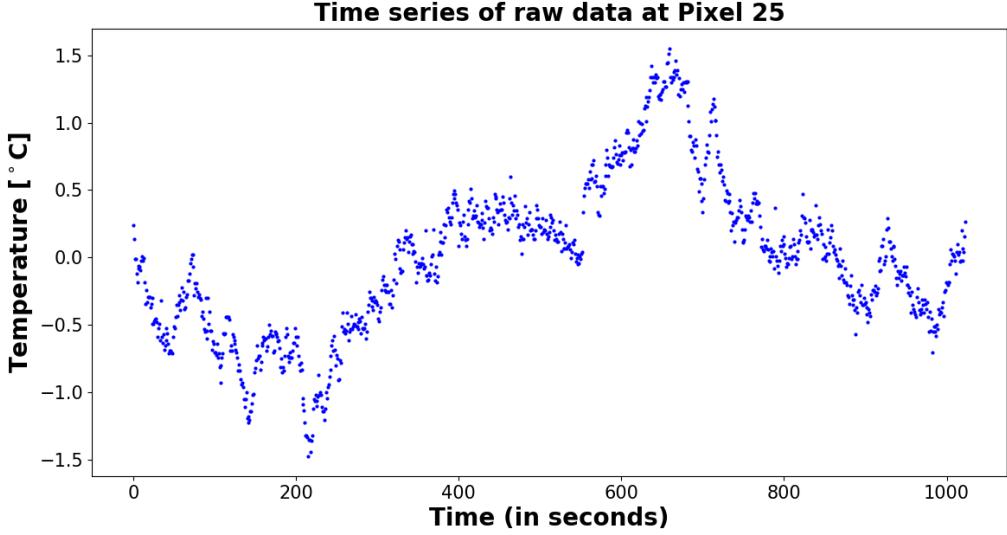


Figure 1.2: The time series of the 25th pixel in the raw intraoperative data with $n = 1024$ time points.

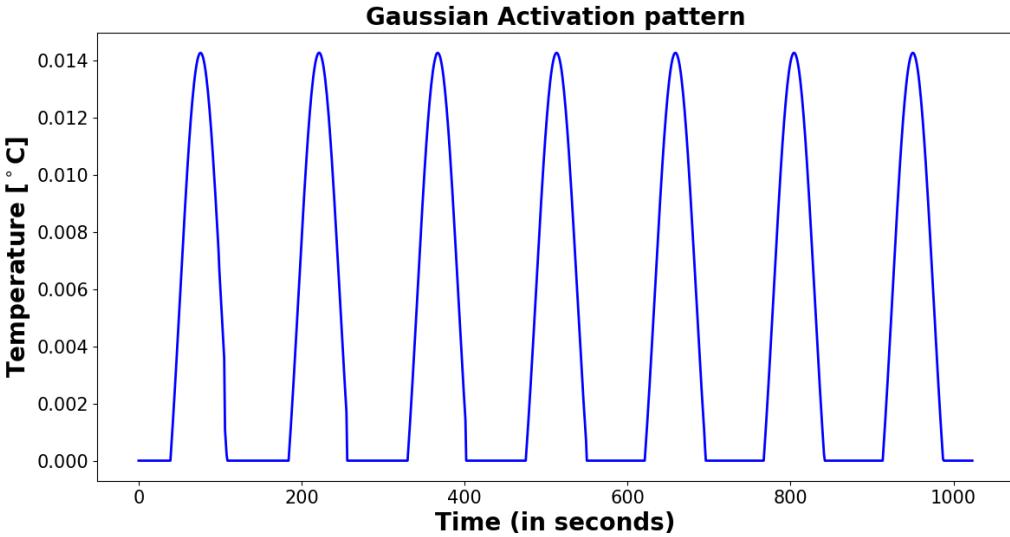


Figure 1.3: The Gaussian activation pattern with $\mu = 30$ and $\sigma = 10$. An alternate cycles of rest and stimulus is visible. Higher levels of response indicate stimulus while the lower levels indicate rest.

1.3 Applications and Summary

Intraoperative thermal neuroimaging identifies neuronal activity as neurovascular coupling helps to correlate local changes in cortical perfusion with neuronal activity. Figure 1.4 shows a typical neuronal activation followed by cerebral blood flow (CBF) response. The imaging technique also provides information about the tissue composition and cell metabolism [HKP⁺17]. However, there are other applications of thermal imaging as well which are equally engrossing. The study of temperature is widely used in medical and non-medical fields. We focus on its medical applications. Work by [CC70] showed that the surface temperature of an arthritic joint was related to the intra-articular joint, and to other biochemical markers of inflammation obtained from the exudate. This experiment ultimately helped in diagnosing inflammatory arthritis. [VPRK04] compared

thermograms with radiographs from patients with hand osteoarthritis. They reported increased temperatures associated with even slight degenerative changes and low temperatures in more severe disease. [STG⁺93] successfully measured neuronal activity in rat brains with the help of thermal imaging. [GHK⁺03] showed the visualization of neuronal activity in thermal imaging with a statistical approach ranging from speech mapping to sensory activations. Thermal imaging is also widely used in dentistry[GGRS⁺96] as well as tumor segmentation [GHKO04]. More detailed applications of thermal imaging are provided in [RA12].

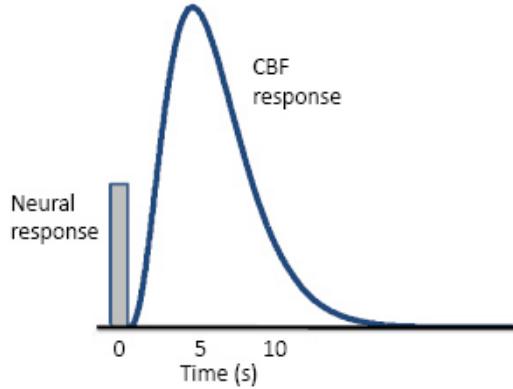


Figure 1.4: Schematic of a typical CBF response to brief neuronal activation [PF08]

There are broad categories of statistical methodologies that have been proposed in the past for the analysis of such data. The thesis includes an extensive literature review of techniques applied in the study of imaging experiments. The literature review in the next chapter considers statistical methodologies for analysis, estimation, and modeling of spatial-temporal data. The chapter also describes the literature on state of the art statistical inference approaches. After the literature review, the thesis aims to present a profound semiparametric regression model to accurately detect somatosensory neuronal activity by increasing the signal to noise ratio (SNR) in the target regions. As already discussed, the data is multidimensional containing thousands of pixels with time series of a thousand or more scans. Also, the data is correlated both in space and time. However, it is noisy with high environmental interference, and the magnitude of the signal increase due to activation is very low [Glo11]. Additionally, the non-uniformity correction induces a jump discontinuity into the time-series of each pixel and alters the low-frequency time-behavior [HHS⁺14]. All these effects contribute to the background noise in the signal and make it challenging to detect neuronal activity accurately. Therefore, it becomes vital to compensate these non-linear background noise. Furthermore, the data exhibits stochastic or random process [Pit79]. In its purest form, a stochastic process is a description of the movement of a random process over time. At every new unit of time, the random process assumes one of many possible values, and each value has a probability associated with it [Pit79]. While we do not know the exact path that the random process takes, we make inferences about the path it might take based on those probabilities. The thesis proposes the extension of the semiparametric regression model with spatial regularization using Markov random fields (MRF). The proposed framework is described in the subsequent chapters, and an extensive performance evaluation has been done. The proposed MRF model should help in the efficient modeling of pixel time courses by minimizing the background noise and catching the linear as well as non-linear behavior present in the data. Finally, the results have been visualized to decipher the neuronal activity in the human brain accurately.

Chapter 2

Related Work

This chapter discusses popular works conducted in the past to handle noisy multidimensional data. [NW72] proposed Generalized Linear model by stating linear relationships between response and independent variables. [FHW⁺94] used this model for the analysis of the functional imaging data. They stated that if $\mathbf{Y} \in \mathbb{R}^{n*m}$ is the response variable and $\mathbf{X} \in \mathbb{R}^{n*p}$ is the independent variable with n, m, p being positive integers, then the relationship between these variables according to GLM is given by $\mathbf{Y} = \mathbf{X}\beta$ where $\beta \in \mathbb{R}^{p*m}$ are the coefficients of the independent variable \mathbf{X} . The GLM model for such sequences is of the form:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \quad (2.1)$$

For our use case, the coefficient matrix β represent regression coefficient values at each pixel location. $\epsilon \in \mathbb{R}^{n*m}$ is the error term which are modeled using probability distributions such as normal, binomial, Poisson, Bernoulli, gamma distributions, among others. Extending the GLM model with non-parametric components is necessary to catch non-linear and random effects. [RWC03] proposed a semiparametric regression framework that combines parametric components with non-parametric components such as a spline polynomial functions. [DB01] defines spline function of a particular order as a piecewise polynomial function that passes through many points called knots and generated fitted curves. Splines enable the creation and control of complex shapes using an ideal number of knots. Spline are categorized into open and closed curves. When the first knot coincides with the last knot, then a closed spline curve is formed. These type of spline curves, therefore, has no endpoints. However, when the first and the last knots are distinct from each other, an open spline curve is formed. [DB01] uses cubic B-splines which are third order polynomials and they possess a high level of smoothness. However, choosing an optimal number and position of the knots is a tricky task. Too many knots lead to overfitting of the data while too few leads to underfitting [EM96].

[OYJ86] presented penalized B-splines or P-splines as a solution to the challenge of ideal fit of splines. They used the integral of a squared higher derivative of the fitted curve as the penalty. [EM96] generalized this approach by proposing a relatively large number of knots and a higher order finite difference penalty on coefficients of adjacent B-Splines. This method reduced the dimensionality of the problem to the number of knots instead of the number of time points. [WTH09] presented a penalized matrix decomposition (PMD), for computing a rank-K approximation for a matrix using l_1 -penalties, which yielded a decomposition of \mathbf{X} using sparse vectors. They showed that when the PMD is applied using a l_1 -penalty on coefficients, it points to a method for sparse principal components. [Ng04] studied two different regularization methods l_1 and l_2 for preventing overfitting. Focusing on logistic regression, they showed that the sample complexity grows only logarithmically after performing l_1 regularization of the parameters. However, there is no closed form solution to l_1 regularization which make inference computationally demanding.

[HT90] developed a generalized additive model (GAM) approach to enhance the properties of generalized linear model with a linear predictor depending linearly on unknown smooth functions

of some predictor variables. They stated:

$$g(E(\mathbf{Y})) = \mathbf{X}_0\boldsymbol{\beta} + f_1(\mathbf{X}_1) + f_2(\mathbf{X}_2) + \cdots + f_{n-1}(\mathbf{X}_{n-1}) \quad (2.2)$$

where $\mathbf{X}_0\boldsymbol{\beta}$ is strictly parametric, f_i are the functions which may be specified parametrically, or non-parametrically, index $i \in [1, 2, \dots, n - 1]$, $E(\mathbf{Y})$ is the expectation of the response variable \mathbf{Y} , \mathbf{X} are the independent variables for each i . An exponential family distribution is specified for \mathbf{Y} along with a link function g . The P-spline functions combined with the parametric component gives a good approximation of $g(E(\mathbf{Y}))$ in the GAM approach. However, this method suffers with numerical rank deficiency. [Woo04] proposed GAM with a ridge penalty to prevent numerical rank deficiency. It is based on the pivoted QR decomposition and the singular value decomposition. [Woo04] examined singular values and removed all the values less than the most substantial singular value multiplied by the square root of the machine precision. This deletion had the effect of recasting the problem into a reduced space in which the model parameters were identifiable. Therefore, this approach dealt adequately with the severe problem of rank deficiency that occurs only over a part of the smoothing parameter space and gives an approximate solution to the initial approaches.

The thermal imaging data concerned in this thesis is spatial-temporal in behavior. Therefore, multivariate or spatial interpolation becomes essential on functions of more than one variable. [JH13] proposed discrete wavelet transform on preprocessed thermal brain imaging data. This technique was able to detect tumors in the brain, and the method showcased the use of time-resolved thermography. [HHS⁺14] developed a wavelet shrinkage scheme based on subspace analysis in 1D wavelet domain to remove motion related patterns from the intraoperative thermographic brain image sequence. The results of the simulation study and the intraoperative measurements revealed an efficient method for improving perfusion and analyzing neuronal activity. [ME05] built a two-dimensional coefficient surface that allows for interaction across the indexing plane of the regressor array. They presented a penalized signal regression using penalized B-spline tensor products, where difference penalties are placed on rows and columns of the tensor product coefficients. The size of this model was not a problem, but the intermediate step with flattened basis led to computational issues. Imagine an image of 1000 * 1000 pixels and 1000 tensor products; the basis matrix then have 10^9 elements which consumes an enormous amount of memory. [ECD06] proposed a fast algorithm that takes advantage of the unique structure of array data and the model matrix as a tensor product. The algorithm avoids computation of full basis matrix and computes the normal equations directly. The algorithm was designed to handle large basis functions.

[Boo89] discussed thin plate splines as an approach to smooth two dimensional surface. The thin-plate spline was visualized as a technique which provided an optimal solution to the problem of deforming a flat piece of sheet metal at a finite number of locations [WM04]. In principle, thin-plate splines (TPS) is used as the model for multidimensional surfaces, but they have too many parameters to estimate with a vast system of equations. [Woo03] showed fitting a thin plate spline to n data points involves estimation of n parameters and a smoothing parameter λ . This results in $O(n^3)$ operations causing computational issues if it is implemented in a semiparametric framework for example. Normally, spatial smoothers such as Thin plate splines uses Euclidean distances between observations even though this distance may not be a measure of spatial proximity. Euclidean distance is extremely sensitive to the scales of the variables involved since all variables are measured in the same units of length. Secondly, the Euclidean distance is blind to correlated variables. [Mah36] presented the Mahalanobis distance as an alternative measure of the distance. The Mahalanobis distance of an independent variable $\mathbf{X} \in \mathbb{R}^{n*p}$ with mean $\boldsymbol{\mu} \in \mathbb{R}^{n*p}$ and covariance matrix $\mathbf{C} \in \mathbb{R}^{n*n}$ is defined as:

$$D(\mathbf{X}) = \sqrt{(\mathbf{X} - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{X} - \boldsymbol{\mu})}. \quad (2.3)$$

The Mahalanobis distance uses the covariance among the variables for calculating the distances. With this measure, the problems of scale and correlation inherent in the Euclidean distance are no longer an issue. A point to note is that the Mahalanobis distance is the generalized form of

Euclidean distance. If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance.

[WR07] replaced straight-line distances with geodesic distances in a smoother that is a sort of approximate thin plate spline known as Geodesic Low rank Thin Plate Splines (GLTPS). However, this algorithm is costly for large datasets. The principle difficulty in interpreting the results of this method is that it is unclear what their penalty term penalizes. To overcome these challenges, [MW14] used a method of spline smoothing with respect to generalized distances proposed originally by [Duc77]. They modeled a response variable $\mathbf{Y} = f(\mathbf{e})$, where f is a smooth function, dependent on $\mathbf{e} \in \mathbb{R}^n$, a vector of generalized distances between n observations. Finally they approximate the model as $f(\mathbf{e}) = f_n(\mathbf{X}(\mathbf{e}))$, where $\mathbf{X}(\mathbf{e})$ is the location of the point with distance vector \mathbf{e} in the n dimensional Euclidean space. The method smooths over a Euclidean space in which the Euclidean inter-observation distances are approximately equal to the original generalized distances which is given by $\|\mathbf{X}(\mathbf{e}_i) - \mathbf{X}(\mathbf{e}_j)\| \approx e_{ij}$. e_{ij} is the generalized distance between points i and j ($\|$ is the Euclidean norm) where $i, j \in [0, 1, \dots, n - 1]$.

[Ram02] suggested a method named FELSPLINE (Finite element L Spline) in which a connection between smoothing with differential operators based penalties and partial differential equations were used to produce a smoother which solved partial differential equation problem defined only over a finite area. However, an extreme boundary condition was required to ensure a unique solution to the L-spline smoothing which might not work in all kind of problems. [WBH⁺08] proposed an alternative model by experimenting with the physical analogy of a soap film which are represented as a basis penalty smoother, and has better boundary behavior. Mathematically, the soap film consists of two sets of basis functions, one that is based entirely on the domain and the other that is induced by the known or estimated boundary values. One problem with soap film smoothing is that the basis function setup is computationally expensive. Additionally, no distinction exists between open boundaries (a boundary that is simply the limit of the region) and hard boundaries (real physical barriers).

[CDE06] introduced generalized linear array model (GLAM), where the data is arranged in an array structure or regular grid. GLAM model is based on generalized linear model (GLM) with the design matrix denoted as a Kronecker product \otimes . The algorithm takes advantages of the structure of the multidimensional data since the fitted values and standard errors are in the correct space. Suppose the response variable $\mathbf{Y} \in \mathbb{R}^{n*m}$ is arranged in a n -dimensional array with each array containing m values, the design matrix of the independent variable $\mathbf{X} \in \mathbb{R}^{n*p}$ is the Kronecker product $\mathbf{X} = \mathbf{X}_{n-1} \otimes \mathbf{X}_{n-2} \otimes \dots \otimes \mathbf{X}_0$ where $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_{n-1} \in \mathbb{R}^p$. The analysis of GLM with response variable \mathbf{Y} and design matrix \mathbf{X} proceeds by the evaluation of the algorithm using the equation:

$$\mathbf{X}^T \mathbf{W} \mathbf{X} \hat{\boldsymbol{\beta}} = \mathbf{X}^T \mathbf{W} \mathbf{Z}, \quad (2.4)$$

where $\hat{\boldsymbol{\beta}}$ is the estimate of $\boldsymbol{\beta} \in \mathbb{R}^{p*m}$, $\mathbf{W} \in \mathbb{R}^{n*n}$ is the diagonal weight matrix, and $\mathbf{Z} = \boldsymbol{\eta} + \mathbf{W}^{-1}(\mathbf{Y} - \boldsymbol{\mu})$ is the working variable with $\boldsymbol{\mu}$ being the mean values. Computationally, GLAM provides array algorithms to calculate the linear predictor, $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$ and the weighted inner product $\mathbf{X}^T \mathbf{W} \mathbf{X}$ without evaluation of the model matrix \mathbf{X} thereby avoiding computational issues in storage and managing huge amount of data with high speed and efficient computations during model estimation.

A thorough research has been done in the previous works to analyze spatial-temporal interactions in image datasets. Smoothing spatial-temporal model is suitable to estimate the spatial and temporal trends simultaneously. [KW03] proposed geoadditive models with Gaussian random fields where they imply that response variable is modeled as the sum of spatial and temporal effects given as $f(\text{space}) + f(\text{time})$. P-Spline ANOVA type interaction model for spatial-temporal smoothing proposed by [LD11] allows spatial-temporal interactions. They used penalized splines in mixed model (semiparametric regression) framework for smoothing spatial-temporal data and is given by:

$$\mathbf{Y} = \kappa + f_s(\mathbf{X}_1, \mathbf{X}_2) + f_t(\mathbf{X}_t) + f_{st}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_t) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2). \quad (2.5)$$

where κ is linear predictor, ϵ is the Gaussian error term with covariance σ^2 , \mathbf{X}_1 and \mathbf{X}_2 are

spatial covariates, \mathbf{X}_t is the temporal covariate, f_s is the function for two-dimensional spatial interaction, f_t is the function to capture temporal trends and f_{st} is the function for the spatial-temporal interaction. The temporal trends were modeled using unidimensional P-splines whereas the spatial and spatial-temporal interactions used two-dimensional and three-dimensional tensor products of P-splines respectively. However, the issue with such implementation is that with large datasets, the tensor products are computationally expensive. For instance, when the data has a strong seasonal trend, the size of basis matrices explodes, and the tensor products are easily of the order of thousands [LD11].

Lately, the literature of image analysis has seen an upsurge in the use of Graphical models to spatially smooth multidimensional data and to express prior and generic knowledge. Graphical models [Lau96] helps in dividing a complex model into a combination of simpler parts. Probability theory provides a conceptual idea for combining these parts into a consistent model. A probabilistic graphical model is a pair $((\mathbf{X}, \mathbf{G}) = (\mathbf{V}, \mathbf{E}))$ of random variables \mathbf{X} and a graph \mathbf{G} where \mathbf{V} is a set of nodes and \mathbf{E} is a set of edges [Li09]. The random variables \mathbf{X} is indexed by nodes of \mathbf{G} such that each node $v \in \mathbf{V}$ is associated with a random variable X_v with $x_v \in X_v$. x_v is the value assigned to the random variable X_v . The edge set \mathbf{E} reveals the conditional independence on \mathbf{X} . A special kind of graphs called undirected graphs contains only undirected edges, which are represented by a set of unordered edge pairs [Li09]. Figure 2.1 shows the example of an undirected graph. Since the pixels in our intraoperative thermal image data are non-causal in nature, undirected graphs particularly become useful for use in our problem set.

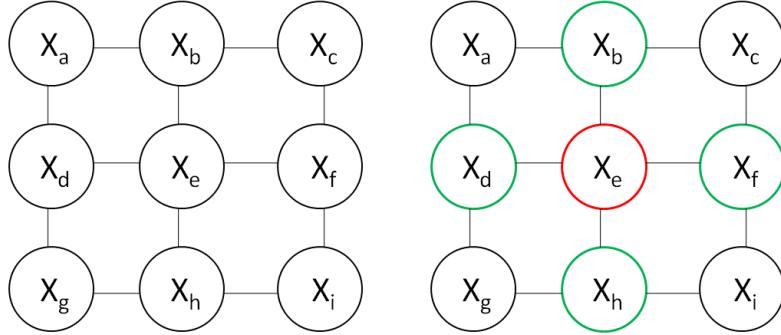


Figure 2.1: The left figure is an undirected graph $((\mathbf{X}, \mathbf{G}) = (\mathbf{V}, \mathbf{E}))$ where circles resemble a set of nodes \mathbf{V} and the lines connecting the nodes are the set of edges \mathbf{E} . The random variables \mathbf{X} are indexed on the nodes. The right figure shows a first order neighborhood systems of the left graph where red node e has green neighboring nodes b, d, f and h

The nodes in the set \mathbf{V} are related to one another via a neighborhood system. A neighborhood system is defined as the set of nodes neighboring a node v excluding itself [BK73]. In the first order neighborhood system, also called the 4-neighborhood system, every interior node has four neighbors. The nodes at the boundaries have fewer neighbors. Figure 2.2 (a) shows an example of a first order neighborhood system. A clique \mathbf{S} for (\mathbf{V}, \mathbf{E}) is defined as a subset of nodes in \mathbf{V} i.e $\mathbf{S} \subset \mathbf{V}$ [Li09]. The single-node, horizontal and vertical pair-nodes cliques, constitute the first order neighborhood system as shown in Figure 2.2 (b-d). It consists of either the single node $S_1 = (v)$, or a pair of neighboring nodes $S_2 = (v, v')$, or a triple neighboring nodes $S_3 = (v, v', v'')$ [Li09]. We intend to limit our discussion with only single and pair node cliques. The collection of a single node and pair node cliques for the first order neighborhood system is given as:

$$\mathbf{S} = S_1 \cup S_2 \quad (2.6)$$

A lot of research has been conducted in the past to build random fields as an extension to the undirected graphical model. A random field model ([Bes74][Bes86b]) assigns a potential to each site, which represents intensity, depth or a category label. As previously discussed, the random variable set $\mathbf{X} = (X_a, \dots, X_v)$ is defined on the set of nodes \mathbf{V} in which each random variable

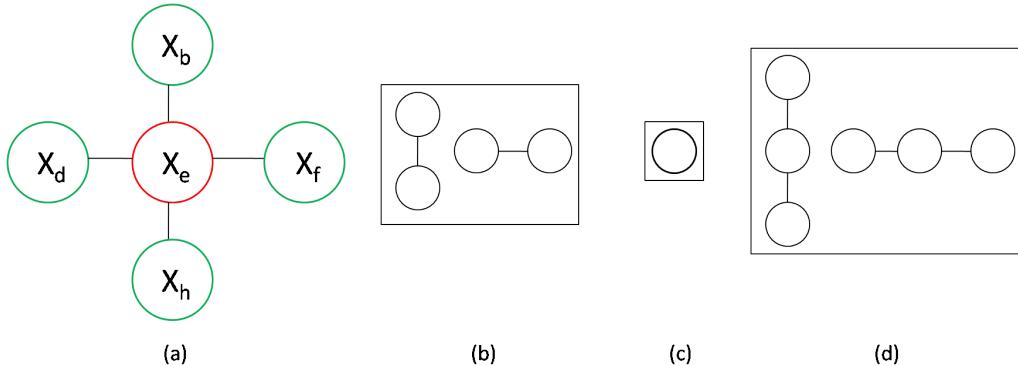


Figure 2.2: (a) shows a first order neighborhood system while (b)-(d) shows different types of cliques possible with different combination of nodes.

takes a value from a discrete or a continuous label set. We use the notation $X_v = x_v$ to denote the event that X_v takes the value x_v and the notation $(X_a = x_a, \dots, X_v = x_v)$ to denote the joint event [Li09]. Hence, the probability that random variable X_v takes the value x_v is denoted by $P(X_v = x_v)$, abbreviated $P(X_v)$ and the joint probability is denoted by $P(\mathbf{X} = \mathbf{x}) = P(X_a = x_a, \dots, X_v = x_v)$ and abbreviated $P(\mathbf{X})$ [Li09]. A Markov random field is the random variable set \mathbf{X} to system $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ which satisfies Markov property and is connected to each other in an undirected graph [KS80]. The properties that it needs to satisfy are:

$$P(\mathbf{X}) > 0, \forall x \in \mathbf{X} \quad (Positivity) \quad (2.7)$$

$$P(X_v|X_{V-v}) = P(X_v|X_{S_v}), \quad (Markovianity) \quad (2.8)$$

The Markov property (Markovianity) can be termed as the property of a random variable where the conditional probability distribution of the random variable depends only upon its cliques S_v instead of all the random variables present in the system $\mathbf{V} - v$. Gibbs Random Field (GRF) is another type of random field where a set of random variables \mathbf{X} is said to obey a Gibbs distribution on \mathbf{V} with respect to \mathbf{S} [Gib]. [KS80] showed that a discrete Gibbs random field (GRF) provides a global model by stipulating a probability mass function. It describes the global properties of an image regarding the joint distribution of potentials for all pixels. [DE87] presented a new approach to the use of Gibbs distributions. They proposed dynamic programming based segmentation algorithms for noisy and textured images, considering a statistical maximum a posteriori probability (MAP) criterion. [AHK65] used Markov assumption to model spatial-temporal data. An MRF is characterized by its local property (the Markovianity) whereas a GRF is characterized by its global property (the Gibbs distribution). The Hammersley-Clifford theorem [HC71] established the equivalence of these two types of random field models. It states that a unique Gibbs Random Field exists for every Markov Random field as long as Gibbs random field is defined in terms of a neighborhood system. The joint probability function of a Markov Random field can be postulated in terms of Gibbs Random field as [Li09]:

$$P(\mathbf{X}) = \frac{1}{J} e^{-\frac{1}{T} U(\mathbf{X})} \quad (2.9)$$

where $J \in \mathbb{R}$ is a normalizing constant called the partition function, $T \in \mathbb{R}$ is a constant called the temperature which is assumed to be 1 unless otherwise stated, and $U(\mathbf{X}) \in \mathbb{R}$ is the energy function given as:

$$U(\mathbf{X}) = \sum_{s \in \mathbf{S}} \theta_s(\mathbf{X}) \quad (2.10)$$

$U(\mathbf{X})$ is the sum of clique potentials $\theta_s(\mathbf{X})$ over all possible cliques \mathbf{S} in a neighborhood. The value of $\theta_s(\mathbf{X})$ depends on the local configuration of the clique s . We discuss a special case where

cliques of only first order neighborhood with single and pair nodes are considered. In this case, the energy function is written as

$$U(\mathbf{X}) = \sum_{v \in \mathbf{V}} \theta_1(X_v) + \gamma \sum_{v \in \mathbf{V}} \sum_{v' \in \mathbf{S}} \theta_2(X_v, X_{v'}) \quad (2.11)$$

where θ_1 and θ_2 are the unary and pairwise clique potentials for n random variables and $\gamma \in \mathbb{R}$ is the pairwise potential weight. It is intuitive from the Markov-Gibbs equivalence that the probability of assigning a label x to a random variable \mathbf{X} is increased if we decrease the overall energy function $U(\mathbf{X})$. Hence, if the response variable is \mathbf{Y} and the random variable assignments are \mathbf{X} , then according to the Bayes rule [Bay63], the maximum a posteriori probability (MAP) is written as:

$$\text{argmax} P(\mathbf{X}|\mathbf{Y}) = \text{argmax} P(\mathbf{Y}|\mathbf{X})P(\mathbf{X}) \quad (2.12)$$

where $P(\mathbf{X}|\mathbf{Y})$ is called the a posteriori probability. It is to be noted that $P(\mathbf{X})$ is termed as $P(\mathbf{X}|\mathbf{Y})$.

The ubiquity of Gaussian random variables in statistical applications had led to the use of continuous random fields, called Gaussian Markov Random Fields (GMRF) as image models. [Che89] proposed the algorithm for sampling GMRFs. Pixel or sites were assigned continuous labels which had joint Gaussian distributions with means μ , standard deviations σ , and correlations controlled by some specific parameters. GMRF is a pairwise interaction model with the random variables at each pixel permitted to take on any real value.

[FKL04b][FL01a][FL01b][FL][Mar95][LF01][BKL05] proposed extensions of penalized spline semiparametric space-time regression model using a Bayesian perspective. The non-linear effects of continuous covariates and time trends were modeled using Bayesian versions of penalized splines, while correlated spatial effects were solved using Gaussian Markov random field prior in a continuous schema. The inference has been performed using Empirical Bayes approach on the generalized linear mixed model representation. This approach of inference has been termed a posteriori mode estimation and resembles penalized likelihood estimation. The advantage of the Bayesian approach is that all unknown functions and parameters are in a unified framework. The empirical approach is based on generalized linear mixed model (GLMM) developed by [LZ99] for longitudinal data analysis using smoothing splines, or for geoadditive models using stationary Gaussian random fields [KW03]. The time trends of the continuous independent variables have been modeled using P-Splines, and the spatial effect has been captured using Gaussian Markov random field which also works in the continuous schema. [FKL04a] compared the performance of Empirical Bayes (EB) using Gaussian Markov Random field, and Full Bayes (FB) approaches with Markov Chain Monte Carlo (MCMC) technique. The spatial-temporal model using EB approach showed better accuracy and runtime as compared to FB approach, confirming that the inclusion of the spatial information was substantial [FKL04a]. The author concluded that Empirical Bayes inference is a promising alternative to full Bayes inference even for relatively large data sets since Monte Carlo simulations are computationally expensive as compared to Empirical Bayes approach which uses Markov random field priors.

In popular literature, a different version of Markov random fields known as discrete Markov Random Fields has become very common. In these models, the random variable defined on each node or pixel takes only discrete values. Discrete MRF is subdivided into Ising prior and Potts prior. The random variables defined using Ising prior have only binary values whereas Potts prior allows them to choose from a range of discrete values. These two priors are different from Gaussian MRF priors as these two involve discrete labels as compared to continuous labels of Gaussian MRF. Various attempts have been made in the past to perform a comparison of their properties, complexities as well as performances. [SS06] compared Discrete MRF based on Ising prior to Gaussian MRF in the context of two different empirical examples. Both the examples featured a two-dimensional regular lattice, and for both types of priors, the neighborhood structure involved eight immediate neighboring pixels. In the first example, the author found that Ising prior estimated higher posterior probabilities of the pixels as compared to Gaussian MRF. The tendency

of the Ising prior to fill voids and smooth over small gaps were noticeable in the first example. Overall, the Ising prior appeared to promote strong clustering and greater segmentation accuracy than the Gaussian prior. However, the author observed that Gaussian prior consumed lesser time than Ising prior to achieve the optimal solution. The high runtime of Discrete Markov random fields is explained from the fact that they are NP-hard problem [Sim93]. The second example in [SS06] was a spatial-temporal functional magnetic resonance imaging (fMRI) data in human brain imaging environment. The data had high levels of noise, and spatial smoothing was crucial to obtain quality results. The data was fitted using the Ising and the Gaussian priors followed by the simulation study based on the fitted data. The objective in fMRI studies of the human brain was to identify areas where an increase in the blood oxygenation levels occurs in response to the presence of an external stimulus, as measured by an observed fMRI signal time series at each site (labeled as 'voxel') on a large regular three-dimensional lattice. The dataset represented voxels of the brain as a dependent variable in a $72 * 86$ lattice with time series regressions, each with 63 observations. The Discrete MRF, i.e., Ising prior and Gaussian MRFs were then used to smooth the data. [SS06] found that Ising prior had less 'starring' (isolated voxels being classified as active) than the Gaussian prior, and accuracy for clustering active voxels were higher with Ising prior than Gaussian prior as shown in Table 2.1.

Noise level	Ising prior	Gaussian prior
0.5	0.095	0.109
1	0.642	0.698
2	1.748	1.781

Table 2.1: Average Performance Metrics for 50 Replicates of the fMRI dataset simulation based on the Ising and Gaussian priors. The metrics corresponding to the best performing prior are in bold. The table shows the % of voxels which were misclassified at different noise levels [SS06].

To investigate further, the author undertook a simulation study on fitted regressions and found that the Ising prior provided superior performance based on real misclassification rates of voxels. Overall, the simulations revealed Gaussian prior as the weaker of the two prior. [NTC13] performed similar comparisons on Potts prior and Gaussian MRFs and found that the Gaussian prior had better runtime results but the final segmentation accuracy was worse than Potts prior.

Discrete Markov random fields as probabilistic models have been widely used in previous works. These models are modeled based on factor graphs [KFL06] in which the graph expresses relations on a set of random variables [Pea88][KF09]. The conditional independence of random variables in an undirected discrete graphical model implies that factorization of the variables is imminent to carry out graph-based algorithms. A very important notion of factor graph is the message, which can be understood as a random variable X_a telling something about the random variable X_b , when the message is passed from X_a to X_b . It is therefore common to use factor graphs to visualize this factorization of the graphical model [KFL06][Loe04]. Factors enable the recursive structure of calculating messages, making the message passing [Kol06] or belief propagation [FH06] algorithms easier to understand and implement. From the equations (2.9-2.11), it is written that:

$$P(\mathbf{X}|\mathbf{Y}) \propto \exp\left(-\left(\sum_{v \in V} \theta_1(X_v) + \gamma \sum_{v \in V} \sum_{v' \in S} \theta_2(X_v, X_{v'})\right)\right) \quad (2.13)$$

Factor graphs alleviates the challenge of solving the above equation by defining factor or clique potentials not in log space by:

$$P(\mathbf{X}|\mathbf{Y}) \propto \prod_{v \in V} \varphi_1(X_v) \prod_{v, v' \in S} \varphi_2(X_v, X_{v'}) \quad (2.14)$$

where φ_1 and φ_2 are the unary and the pairwise factors defined on n random variables respectively. The maximum a posteriori probability (MAP) is defined by the simple product of these small factor

functions. Due to the addition of the factors in the graphical model, the notation now becomes $(\mathbf{X}, \mathbf{G}) = (\mathbf{V}, \mathbf{F}, \mathbf{E})$ where \mathbf{F} are the factors that are assigned to the set of nodes \mathbf{V} and the set of edges \mathbf{E} .

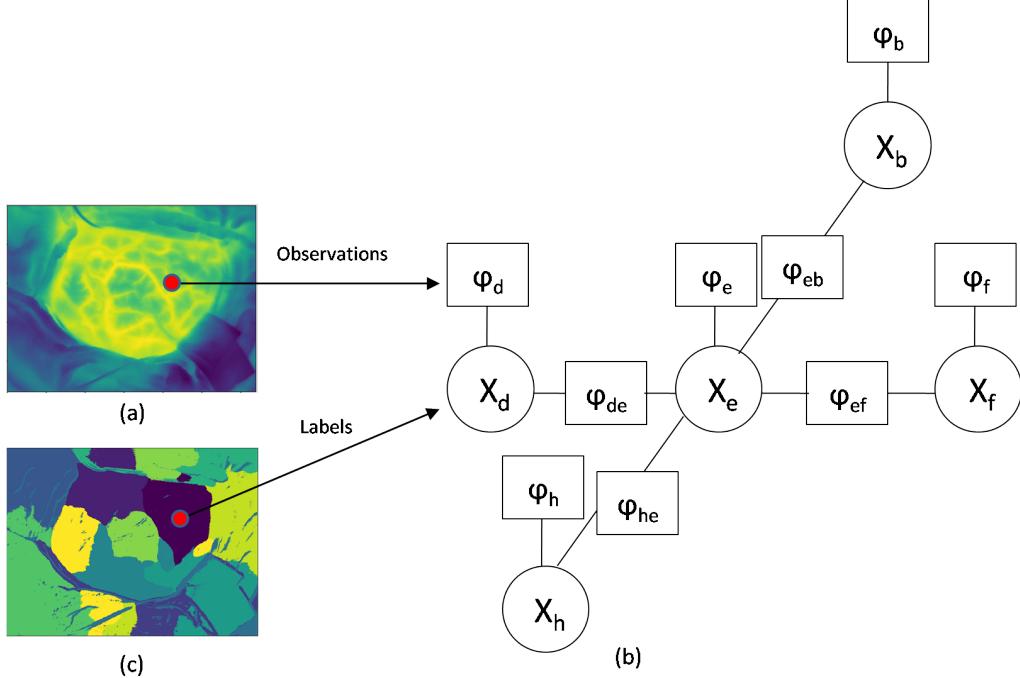


Figure 2.3: (a) shows a typical thermographic image of the cortex , (b) is the representative factor graph of a small region of the image with 5 pixels and (c) shows the resultant label diagram of the image after finding the maximum a posteriori probability.

The problem of finding the maximum a posteriori probability (MAP) seeks the most probable label assignments that describe a set of random variables. There exists several optimal solutions which make the distribution $P(\mathbf{X}|\mathbf{Y})$ multi-modal. Also, solving the Markov random field formulation of the MAP problem is known to be NP-hard [Coo90][Shi94]. This means that we can not expect, that the MAP-problem is exactly solvable. However, even if the optimal solution cannot be calculated in suitable time, one can find approximate solutions. In such cases, it is favorable to use methods which give upper and lower bounds on the optimal posterior probabilities and energies, respectively.

Search based algorithms such as ICM [Bes86a] keeps all variables except one fixed in each step and adjusts the free variable such that the objective function is minimized. The variables are repeatedly visited in a particular order until no alteration of a single variable can further reduce the value of the objective function. These algorithms have an exponentially high runtime if they are applied to a large number of variables [AKK⁺10]. [Pea88] [MWJ13][WF06] uses message passing algorithm called Loopy Belief Propagation (LBP) whose primary operation is to send messages between the nodes of the graph. It acts as an approach related to dynamic programming if the messages are sent in serial order. However, even if the messages are sent in the parallel order, it guarantees to converge to the optimal solution, but with some overhead of calculations. However, the prime issue with LBP is that there is an uncertainty on really what it optimizes [AKK⁺10]. [WF06] showed that fixpoints of the sum-product-LBP coincide with stationary points of the Bethe variational problem [Bet35].

[WJW05][WJ08] proposed algorithms which are similar to LBP but based on a convex relaxation of the MAP problem by considering a convex upper bound. A decomposition of the original problem into several tree structured subproblems acts as a promising solution [WJW05]. By re-parametrization of the decomposition, which re-weights the trees, a bound obtained by

this decomposition is optimized. This method is known as Tree reweighted Belief Propagation or TRBP [WJW05]. While TRBP process fix-point updates, the algorithm gets stuck in local optima [Kol06]. Furthermore, TRBP has no guarantee for convergence if all messages are calculated in a parallel manner. [WJ08] proposed a consistent tree representation with some ordering of nodes. This ordering restricts the choice of the decomposition defining trees for fully connected graphs. Dual decomposition proposed by [GK87][Gui03] is a conventional technique in convex optimization to decompose the problem in a set of simpler interdependent subproblems. [Kol06] proposed a sequential version of tree-reweighted message passing (TRW-S) which is guaranteed to converge to minima satisfying the so-called weak tree agreement. [CR12] implemented TRWS algorithm in FPGA hardware to reap the benefits of its sequential execution and utilize significant parallelism and memory bandwidth for acceleration.

[KAH⁺14] performed a comparative study of these inference techniques for Discrete Markov Random fields (DMRF). The author discussed pixel-based-models, superpixel based models, etc. with varying orders. In pixel-based models, each pixel in a 2D lattice is a variable whereas, in superpixel based models, a cluster of pixels acts as a single variable. The models were evaluated based on different neighborhood structures. The study attempted to compare the inference techniques based on runtime, energy, bound, amount of memory consumed and so on. The inference algorithms were broadly classified into linear programming methods, move making methods and message passing methods. In all these inference methods, algorithms based on Monte Carlo simulations were not considered. The findings were summarized for problems such as Stereo Matching, Inpainting, and Photomontage. The author found that with good stopping conditions, Tree-reweighted message passing (TRWS) performed well for all models to which it can be applied. Compared to BP, TRW-S has better convergence properties guaranteed by its tree-sequential style of message passing on the tree-decomposed graph [BBHS07] [BGPC10], and in many cases performed better in its quality of labeling results than BP [Ibm11].

Belief propagation [Pea88] finishes message passing in two passes: message passing from leaf nodes to the root node of the tree (upward pass) and from the root to the leaves (downward pass). Figure 2.4 shows an inward pass with each node having four possible labels. The message that node q receives from node p is given as [Kol06]:

$$M_{pq}(j) = \min_i \{\theta_p(i) + \gamma \theta_{pq}(i, j)\} \quad (2.15)$$

where $i, j \in [0, 1, 2, 3]$ are the possible labels in node p and q respectively, $\theta_p(i) \in \mathbb{R}$ is the unary potential for node p at label i and $\theta_{pq}(i, j) \in \mathbb{Z}_{\geq 0}$ is the pairwise potential for node p and q at labels i and j respectively. The message that node r receive from node q therefore becomes [Kol06]:

$$M_{qr}(k) = \min_j \{\theta_q(j) + M_{pq}(j) + \gamma \theta_{qr}(j, k)\} \quad (2.16)$$

where k is the number of possible labels for node r . After two-pass message passing, the beliefs for all the nodes and edges are obtained. If all the node beliefs have the unique minimum assignment, the global optimum label for each node is found based on its belief individually. In the case of graphs without loops, the belief propagation finds the optimum assignment with minimum energy.

With belief propagation, we expect that the belief of one node is propagated to the other nodes far apart in the graph. With loopy graphs, however, the message can also go along the loop back to the node itself and impact its original beliefs [WJW05]. It is often the case that when running belief propagation on a loopy graph, messages do not converge but oscillate. Even if messages are converged, and we find the local best label assignment, the minimum energy of this assignment is usually higher than the true global minimum [WJW05]. Tree-reweighted message passing (TRWS) [WJW05] is designed to avoid this weakness by transforming the original energy minimization problem on a loopy graph to a set of minimization problems on trees that cover the graph. The original energy function on an MRF with parameters can be decomposed into a sum of energy functions on trees as follows:

$$U(\mathbf{X}|\boldsymbol{\theta}) = \sum_T \omega_T U(X_T|\theta_T) \quad (2.17)$$

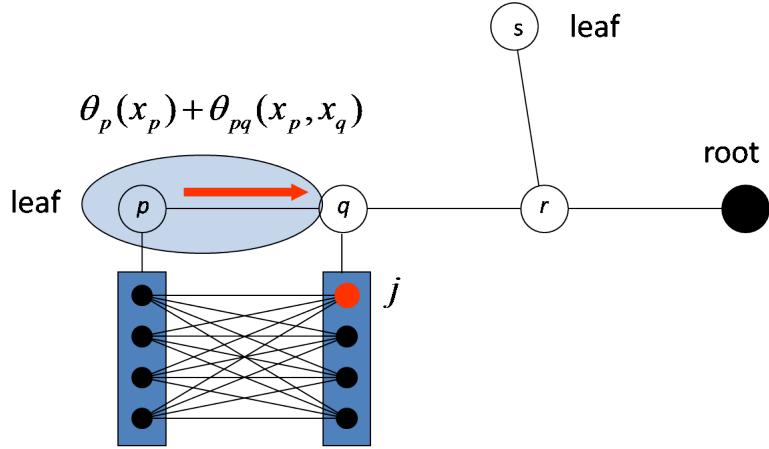


Figure 2.4: A descriptive diagram of flow of messages between two nodes during an inward pass towards root node [Kol06].

where ω_T is a set of parameters defined on the tree T , θ are the unary and pairwise potentials defined on the graph. We obtain the lower bound of the original energy minimization problems by applying Jensens inequality as follows [WJW05]:

$$\min_X U(\mathbf{X}|\theta) \geq \sum_T \omega_T \min_X U(X_T|\theta_T) \quad (2.18)$$

One advantage of this lower bound is that it efficiently finds the best label assignment for each tree using belief propagation [WJW05]. Any subset of the union of the best label assignments for all the trees results in locally optimum energy. The equality is satisfied when the label assignments for all the trees agree with each other. The union of these assignments is proven to be the exact global optimum assignment that we want to achieve [WJW05]. After decomposition, the goal is to find the proper set of parameters on trees that maximizes the lower bound. A modified version of Tree reweighted message passing finds the best parameters by iteratively updating beliefs [Kol06]. For example, to update a belief of one node, we first take all the trees that contain this node and perform BP on each tree. The average among the beliefs of the node in the corresponding trees is then calculated. The averaged belief is used as a new unary potential for the update of the other nodes [Kol06].

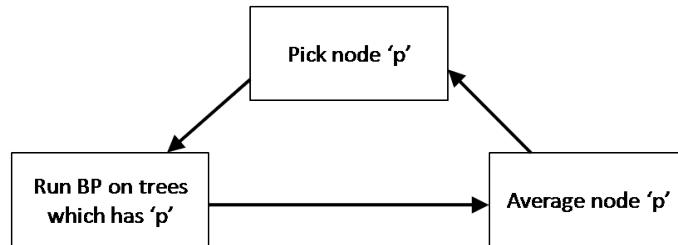


Figure 2.5: A typical TRWS Algorithm

An important property of the TRW-S method is that if we repeat the belief update one node at a time, the lower bound is guaranteed not to decrease [CR12]. In other words, we avoid the energy value oscillating by updating belief sequentially [CR12]. The first diagram in Figure. 2.6 shows a 3x3 factor graph, where the dark and white rectangles represent unary and pairwise factors, respectively. The second diagram shows the tree-decomposition of the first figure. Each row or column corresponds to a tree; the tree T1 consists of node 1, 2 and 3. Note that the update order (the numbers in circles) is monotonically increasing for every tree. Thus, a message once

computed is reused in the later update; the message from 1 to 2 computed in the update of node 2 is reused in the update of node 3. If we follow this monotonic order to pass messages from top-left to bottom right (forward pass) and in the opposite direction (backward pass), we obtain the equivalent result of sequential belief update while computing each message only once [CR12].

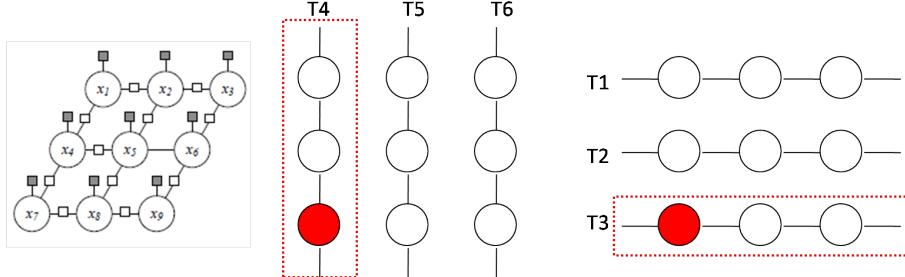


Figure 2.6: An example 3x3 grid factor graph with its monotonic chains [CR12]

In this thesis, a discrete factor graph is constructed based on the implementation of the OpenGM framework [ATK12] (see Appendix A). It has built-in functions which help in defining unary and pairwise factors, construct a graphical model and finally run inference algorithms like TRWS to minimize the energy function.

2.1 Summary

Various statistical methods were presented in this chapter which gave insights into the analysis of spatial-temporal datasets. The study focused primarily on robust estimation of such data and showcased the advantages as well as the issues involved in the practical implementation of the methods. The basic concepts related to semiparametric regression framework, graphical model, Markov Random fields and factor graphs were also discussed. It is clear that the optimization of factor graphs by minimizing the overall energy function of a probabilistic model is vital. Therefore, the crucial trade-offs between various inference schemes were explored. After taking into consideration various factors involved, subsequent chapters put forward the proposed model developed in this thesis.

Chapter 3

Background theory

3.1 B-splines

B-splines are attractive for univariate non-parametric modelling of time series since they offer more control and flexibility than Bezier curves [DB01]. B-spline curves are composed from many pieces of polynomials. Consider $w + 1$ real equidistant values t_i called knots with $i \in [0, 1, \dots, w]$. The knots has two endpoints, t_0 and t_w with:

$$t_0 \leq t_1 \leq \dots \leq t_w$$

For order $j \in [1, \dots, d]$, d being the order of the spline, a set of real valued B-spline basis functions B_{ij} is defined for each augmented knot set t_i as [DB01]:

$$B_{i0}(t) = \begin{cases} 1, & \text{if } t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$B_{i,j}(t) = \psi_{i,j}(t)B_{i,j-1}(t) + [1 - \psi_{i+1,j}(t)]B_{i+1,j-1}(t)$$

where

$$\psi_{i,j}(t) = \begin{cases} \frac{t - t_i}{t_{i+j} - t_i}, & \text{if } t_{i+j} \neq t_i \\ 0, & \text{otherwise} \end{cases}$$

The above equations assumes that $0/0$ is 0. Therefore, a spline function of order d is composed of a linear combination of basis B-splines given by [DB01]:

$$\sum_{i=0}^{p-1} \beta_i B_{i,d}(t)$$

The $\beta \in \mathbb{R}^p$ are control points or de-boor points [DB01] or simply spline coefficients. It is to be noted that $p = w - d - 1$. As discussed in the previous chapter, spline curves are categorized into open and closed curves. Open spline curves are beneficial to use in our case because the first and last knots do not coincide with each other. Defining the position of knots t_i is crucial in a spline regression framework. For a discontinuous data, a non-uniform position of knots seems to be the best solution. However, the intraoperative thermal imaging data is uniform with equal distance between each time points. [UAE93] showed that defining the knots and their respective locations in a non-uniform space are computationally costly as well. Therefore, equally spaced knots are considered for modeling the time series of each pixel in the intraoperative data. Another factor which determines the amount of the spline fit to the data is the number of knots. A higher number of knots leads to overfitting while a lower number of knots leads to under-fitting of the data.

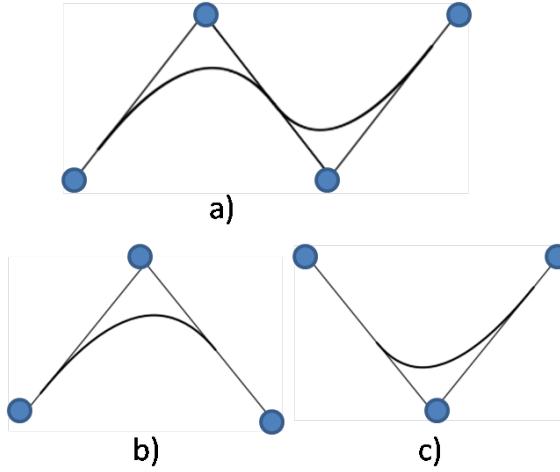


Figure 3.1: a) shows an example of a quadratic curve b) and (c) are its two polynomial segments. The blue dots represent the control points.

3.2 Discretization of spline coefficients

The thesis aims to discretize the spline coefficients to model them using the discrete Markov Random field priors. There are broad categories of discretization methods that are used to transform these continuous features. Some unsupervised clustering algorithms like binning [CGB96], outputs a mesh over the multi-dimensional continuous feature space, where each feature is partitioned into regions independent of the other attributes. Equal width binning method merely divides the range of observed values for a variable into k equal sized bins, where k is the user supplied parameter. [Cat91] discussed that this type of discretization is susceptible to outliers that may profoundly skew the range. The method also does not use instance labels for partition boundaries hence it is likely that the crucial information is lost by binning. It also leads to high discretization error. K-means clustering [HW79] (see Appendix C) is an unsupervised clustering algorithm which aims to classify the data through a certain number of components (k). However, K-means algorithm is prone to converge at a local minimum. The algorithm also requires a prior specification of the number of components. The algorithm is also very inefficient with the noisy data containing many outliers. It also heavily relies on the proper initialization of mean values to provide accurate results.

3.2.1 Gaussian Mixture Model (GMM)

Gaussian Mixture Model is a probabilistic model with a mixture of k normally distributed components [Rey09]. Instead of hard assigning a data point to a particular Gaussian component, it assigns probability of a Gaussian component belonging to a data point [Rey09]. The algorithm helps in solving the problem of local minima. Gaussian Mixture Model maintain many of the theoretical and computational benefits of Gaussian models, making them practical for efficiently modeling huge datasets such as intraoperative functional thermal imaging data.

Let $\beta \in \mathbb{R}^{m*p}$ are the spline coefficients with m number of pixels and p number of control points. Assuming total of k Gaussian components then a Gaussian mixture model is parameterized by three types of values, the mixture component weights $\omega \in \mathbb{R}^k$, the k^{th} component means $\mu_k \in \mathbb{R}^{m*p}$ and the k^{th} component covariances $C_k \in \mathbb{R}^{m*m}$. The mixture component weights are defined with the constraint that $\sum_{i=1}^k \omega_i = 1$ so that the total probability distribution normalizes to 1. The probability is therefore the summation of k normal distributions given as:

$$P(\boldsymbol{\beta}) = \sum_{i=1}^k \omega_i \mathcal{N}(\boldsymbol{\beta} | \boldsymbol{\mu}_i, \mathbf{C}_i) \quad (3.1)$$

where \mathcal{N} represents multivariate Gaussian distribution given by:

$$\mathcal{N}(\boldsymbol{\beta} | \boldsymbol{\mu}_i, \mathbf{C}_i) = \frac{1}{\sqrt{(2\pi)^k |\mathbf{C}_i|}} \exp \left(-\frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\mu}_i)^T \mathbf{C}_i^{-1} (\boldsymbol{\beta} - \boldsymbol{\mu}_i) \right) \quad (3.2)$$

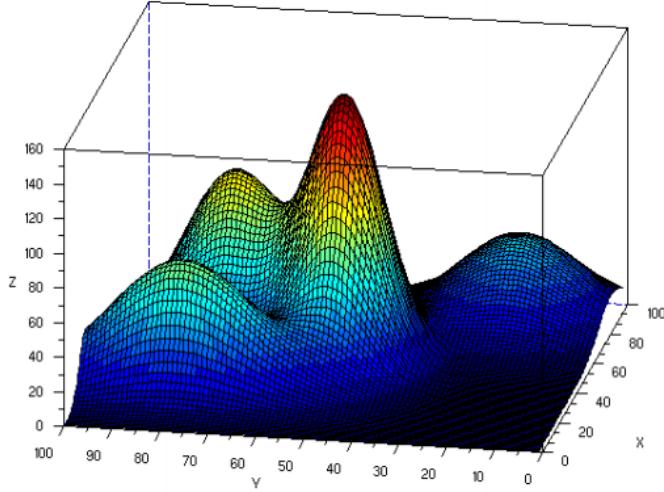


Figure 3.2: An example Gaussian mixture model with $k = 4$ Gaussian components [TP09]

3.2.1.1 Expectation Maximization Algorithm

To learn the parameters $\boldsymbol{\omega}$, $\boldsymbol{\mu}$ and \mathbf{C} , Expectation Maximization algorithm [DLR77] is used. Expectation maximization (EM) is an iterative technique for maximum likelihood estimation where the maximum likelihood of the data increases with each subsequent iteration, meaning it is guaranteed to converge and therefore, it is not NP-hard [DLR77]. Figure 3.3 shows the flow of expectation maximization algorithm. The first step is the initialization step where the algorithm assigns initial values to $\hat{\boldsymbol{\omega}}_k$, $\hat{\boldsymbol{\mu}}_k$ and $\hat{\mathbf{C}}_k$. In the second step, i.e., the expectation step, the probability that a data point is generated by each of the k Gaussian components are computed. It computes this probability by the following equation:

$$\hat{P}_{ik} = \frac{\hat{\omega}_k \mathcal{N}(\boldsymbol{\beta}_i | \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{C}}_k)}{\sum_{j=1}^k \hat{\omega}_j \mathcal{N}(\boldsymbol{\beta}_i | \hat{\boldsymbol{\mu}}_j, \hat{\mathbf{C}}_j)} \quad (3.3)$$

The third step is the maximization step where the algorithm updates weights $\hat{\boldsymbol{\omega}}_k$, means $\hat{\boldsymbol{\mu}}_k$, and covariances $\hat{\mathbf{C}}_k$ by means of the following equations:

$$\hat{\omega}_k = \sum_{i=1}^m \frac{\hat{P}_{ik}}{m} \quad , \quad \hat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^m \hat{P}_{ik} \boldsymbol{\beta}_i}{\sum_{i=1}^m \hat{P}_{ik}} \quad \text{and} \quad \hat{\mathbf{C}}_k = \frac{\sum_{i=1}^m \hat{P}_{ik} (\boldsymbol{\beta}_i - \hat{\boldsymbol{\mu}}_k)^2}{\sum_{i=1}^m \hat{P}_{ik}} \quad (3.4)$$

The steps of expectation and maximization are performed iteratively until the values of $\hat{\boldsymbol{\omega}}_k$, $\hat{\boldsymbol{\mu}}_k$ and $\hat{\mathbf{C}}_k$ converges, giving the maximum likelihood estimate. Figure 3.4 shows the discretization map after applying Gaussian Mixture Model on our intra-operative thermal imaging data using $k = 12$ Gaussian components and $p = 431$ control points.

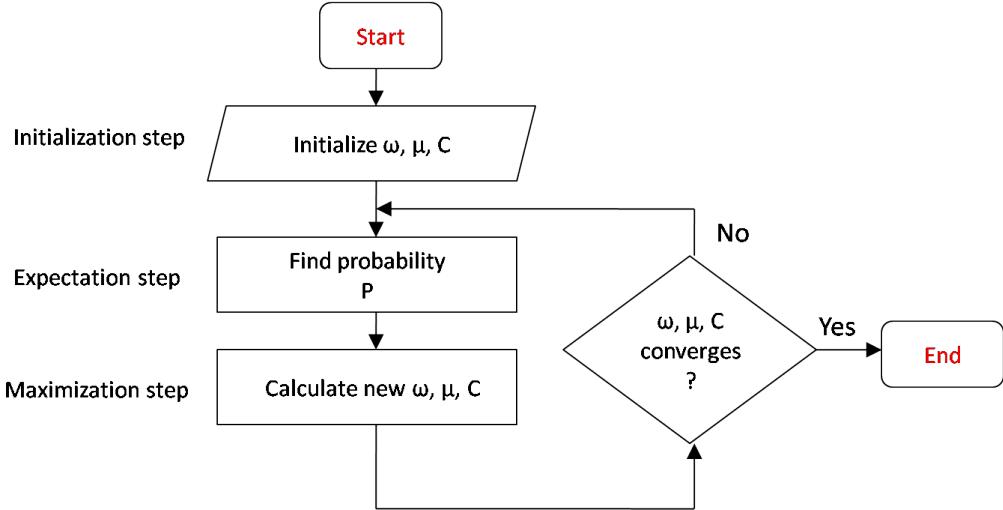


Figure 3.3: Expectation Maximization Algorithm

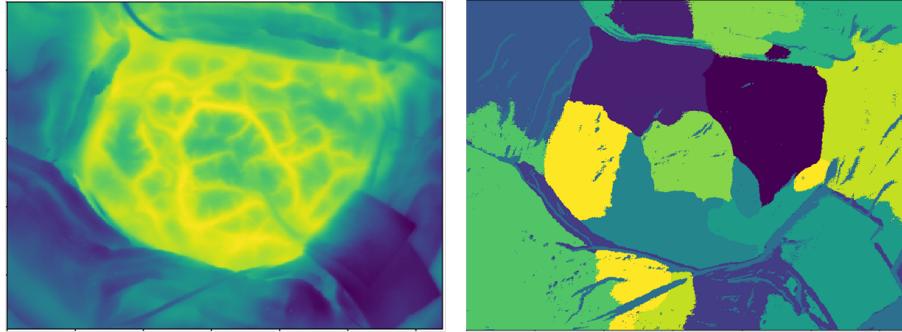


Figure 3.4: The left figure is the image of our raw data at 0th time point. The right figure shows the discretized pixels with a label using Gaussian Mixture Model with $k = 12$ Gaussians components and $p = 431$ spline control points.

3.3 Spherically Invariant Random process (SIRP)

[Ver64] introduced spherically invariant random process and showed that a random process is spherically invariant if the process has zero mean and unit variance Gaussian distribution. Hence, the multivariate Gaussian or normal distribution is generalized in the form of spherically invariant random process (SIRP) [Bre82]. The major advantage of using SIRP to represent a multivariate Gaussian distribution is that their probability density functions (PDF) is denoted as a quadratic function instead of an exponential function. This decreases the overall time complexity of running the algorithm. Assuming n be the size of random processes and l being a constant then SIRP implementation with a quadratic function has a runtime of $O(n^2)$ whereas a non-SIRP implementation with an exponential function has $O(l^n)$. A generalized SIRP notation is given by:

$$P(\mathbf{X}) = \pi^{-n/2} f(\mathbf{X}^T \mathbf{X}; n) \quad (3.5)$$

where \mathbf{X} is the n^{th} order random process and $P(\mathbf{X})$ is the joint probability density function (PDF) of n^{th} order. It is understood from the above equation that SIRP is a generalization of multivariate Gaussian distribution with $\mathcal{N}(0, \sigma^2)$ and $f(\mathbf{X}^T \mathbf{X}; n)$ being exponential. [Bre82] shows that using SIRP, the multivariate probability density function of an odd order is obtained from the first-order PDF only by means of differentiation, however multivariate PDF of even order is explicitly given by means of G-functions. Hence, [Bre82] proposed to model a SIRP implementation of multivariate

PDF, in terms of G-functions.

3.3.1 Meijer G-function

G functions are generalized form of hypergeometric functions which are used to describe any special function such as differentiation, integration, Laplace transform, probability density function among others. [Bre82] shows the implementation of SIRP with multivariate PDF using Meijer-G functions. Meijer-G function is given by the following formula in a complex plane [Bre82]:

$$G_{p,q}^{m,n}\left(\begin{matrix} e_1, \dots, e_p \\ h_1, \dots, h_q \end{matrix} \middle| z\right) = \frac{1}{2\pi i} \int_L \frac{\prod_{j=1}^m \Gamma(h_j - s) \prod_{j=1}^n \Gamma(1 - e_j + s)}{\prod_{j=m+1}^q \Gamma(1 - h_j + s) \prod_{j=n+1}^p \Gamma(e_j - s)} z^s ds \quad (3.6)$$

where

- $\mathbf{e}_p = (e_1, e_2, \dots, e_p)$ and $\mathbf{h}_q = (h_1, h_2, \dots, h_q)$ are two sets of complex parameters.
- $\Gamma(s)$ denotes the gamma function in s plane. $z \neq 0$
- m, n, p and q are integer numbers with $0 \leq m \leq q$ and $0 \leq n \leq p$
- $e_k - h_j \neq 1, 2, 3, \dots$ for $k \in [1, 2, \dots, n]$ and $j \in [1, 2, \dots, m]$

The above equation for G-function is important to finally formulate a spherically invariant Random Process for multivariate Gaussian distribution. The first order joint probability density function is written in terms of G functions as [Bre82]:

$$P(\mathbf{X}) = A G_{p,q}^{m,n}\left(\begin{matrix} \mathbf{e}_p \\ \mathbf{h}_q \end{matrix} \middle| \lambda \mathbf{X}^T \mathbf{X}\right) \quad (3.7)$$

where A is a normalizing factor and λ is a constant which yields unit variance. The above equation provides a simpler way of representing multivariate Gaussian distribution and convert an exponential time complexity to a quadratic time complexity by using explicit notation of G functions.

Chapter 4

Semiparametric regression

Somatosensory stimulations such as touch, pressure, or heat create neuronal activity along a sensory nerve to the spinal cord and finally in the concerned regions of the brain. The neuronal activity lead to an increase in regional cerebral blood flow due to neurovascular coupling. The blood flow at the active sites cause minute temperature changes in the range of 40-80 mK which is finally captured by thermal imaging [GHK⁺03]. Successful detection of this neuronal activity and negating any background noise is a challenging task in the field of intraoperative functional thermal imaging.

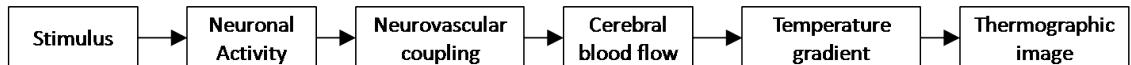


Figure 4.1: A typical sequence of brain activation starting from stimuli till the temperature changes leading to heat radiation from the active sites.

4.1 The Univariate spline regression Model

Figure 4.2 shows a detailed flow of the univariate spline regression that has been developed to model the intraoperative thermal imaging data. The parametric component is combined with a non-parametric component such as P-splines to form a semiparametric regression framework. The penalized normal equation has been used to estimate the spline coefficients. Depending on the varying values of the Lagrange multiplier λ (to be discussed later), the estimated spline coefficients are derived for its model statistics such as AIC to determine the optimal fit to the intraoperative data.

4.1.1 Parametric component

Neurovascular coupling induces temperature changes into thermal time series. This behavior is propagated through several tissue layers due to which it is expected that the thermal signal resembles a bell like curve [JH14] [NH18]. Given $t_i \in [t_0, t_1, \dots, t_{n-1}]$ time points with $0 \leq i \leq n-1$, the parametric component, $\mathbf{X} \in \mathbb{R}^n$ of the semiparametric regression framework is modeled with a Gaussian activation function which is given as [NH18]:

$$\kappa(t_i | \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(t_i - \mu)^2}{2\sigma^2}} \quad (4.1)$$

where $\mu \in \mathbb{R}$ being the mean or time to peak value and $\sigma^2 \in \mathbb{R}$ being the variance or steepness of temperature change. It is expected that μ correlate with the depth of the focal activation, meaning that the neuronal activity in deeper tissue layers lead to a weaker amplitude of the measured signal

[HKP⁺17] [NH18]. In this case, the shape of the signal is also expected to have less curvature and therefore higher σ^2 . Figure 1.3 shows an example of Gaussian activation pattern with $\mu = 30$, $\sigma = 10$. The experimental conditions are modeled as the dilated and shifted variants of Eq. 4.1 and stacked into \mathbf{X} [NH18]. The signal $\kappa(t_i|\mu, \sigma)$ is vectorized with time points to form a vector $\Gamma(\mu, \sigma) = [\kappa(t_0|\mu, \sigma), \kappa(t_1|\mu, \sigma), \dots, \kappa(t_{n-1}|\mu, \sigma)]$. In case of $s + 1$ electrical stimulations with a period of Δ seconds, the number of parametric components hence becomes $s + 1$ [NH18]. So, the design matrix is given by [NH18]:

$$\mathbf{X}(\mu, \sigma) = [\Gamma(\mu, \sigma) \Gamma(\mu + \Delta, \sigma) \dots \Gamma(\mu + s\Delta, \sigma)] \quad (4.2)$$

4.1.2 Non-parametric components as B-splines

The non-parametric component is introduced in the parametric model to capture the random and non-linear effects. The addition of this component helps in modeling the low and high frequent non-linear behavior in the intraoperative thermal imaging data [NH18], and therefore the model is termed as semiparametric regression framework. The univariate regression model proposed in this thesis contains a non-parametric component [NH18], using B-spline basis function. Both parametric and non-parametric components are added to a semi-parametric regression framework as shown below:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\alpha} + \mathbf{B}_1\boldsymbol{\beta}_1 + \mathbf{e} \quad (4.3)$$

where $\mathbf{Y} \in \mathbb{R}^{n*m}$ are the response variable with n being the number of time points and m being the number of pixels, $\mathbf{X} \in \mathbb{R}^{n*1}$ is the parametric component, $\boldsymbol{\alpha} \in \mathbb{R}^{1*m}$ are the coefficients of the non-parametric component, $\mathbf{B}_1 \in \mathbb{R}^{n*p}$ are the spline basis functions with p being the number of spline control points, $\boldsymbol{\beta}_1 \in \mathbb{R}^{p*m}$ are the spline coefficients and $\mathbf{e} \in \mathbb{R}^{n*m}$ are the error terms modeled as the Gaussian distributed noise. Single matrices $\mathbf{G} \in \mathbb{R}^{n*(p+1)}$ stacks the two components as $\mathbf{G} = [\mathbf{X} \quad \mathbf{B}_1]$. Therefore, the semiparametric regression becomes:

$$\mathbf{Y} = \mathbf{G}\boldsymbol{\beta} \quad (4.4)$$

4.1.3 Penalized B-splines

A special care needs to be taken with the non-parametric component. To prevent issues of underfitting and overfitting due to inappropriate selection of control points, a specific penalty matrix $\mathbf{P} \in \mathbb{R}^{p*p}$ is added to the \mathbf{G} matrix [HKP⁺17]. By constructing a block diagonal matrix $\mathbf{S} = blkdiag(0_k, \mathbf{P}_1)$ where 0 are added for parametric components and \mathbf{P}_1 is the penalty matrix of non-parametric component, we make sure that the penalty is added to the estimate of non-parametric component only. The minimization problem then becomes [NH18]:

$$\min_{\boldsymbol{\beta}} \|\mathbf{G}\boldsymbol{\beta} - \mathbf{Y}\|_2^2 + \lambda \|\mathbf{S}\boldsymbol{\beta}\|_2^2 \quad (4.5)$$

where λ is the Lagrange Multiplier. Finally, the coefficients are estimated by the penalized normal equation given by [NH18]:

$$\begin{vmatrix} \hat{\boldsymbol{\alpha}} \\ \hat{\boldsymbol{\beta}}_1 \end{vmatrix} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{S}^T \mathbf{S})^{-1} \mathbf{G}^T (\mathbf{Y}) \quad (4.6)$$

The estimates $[\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}_1]$ are used to obtain the semiparametric fit of \mathbf{Y} i.e. $\hat{\mathbf{Y}} \in \mathbb{R}^{n*m}$ and is given as:

$$\hat{\mathbf{Y}} = \mathbf{G} \begin{vmatrix} \hat{\boldsymbol{\alpha}} \\ \hat{\boldsymbol{\beta}}_1 \end{vmatrix} \quad (4.7)$$

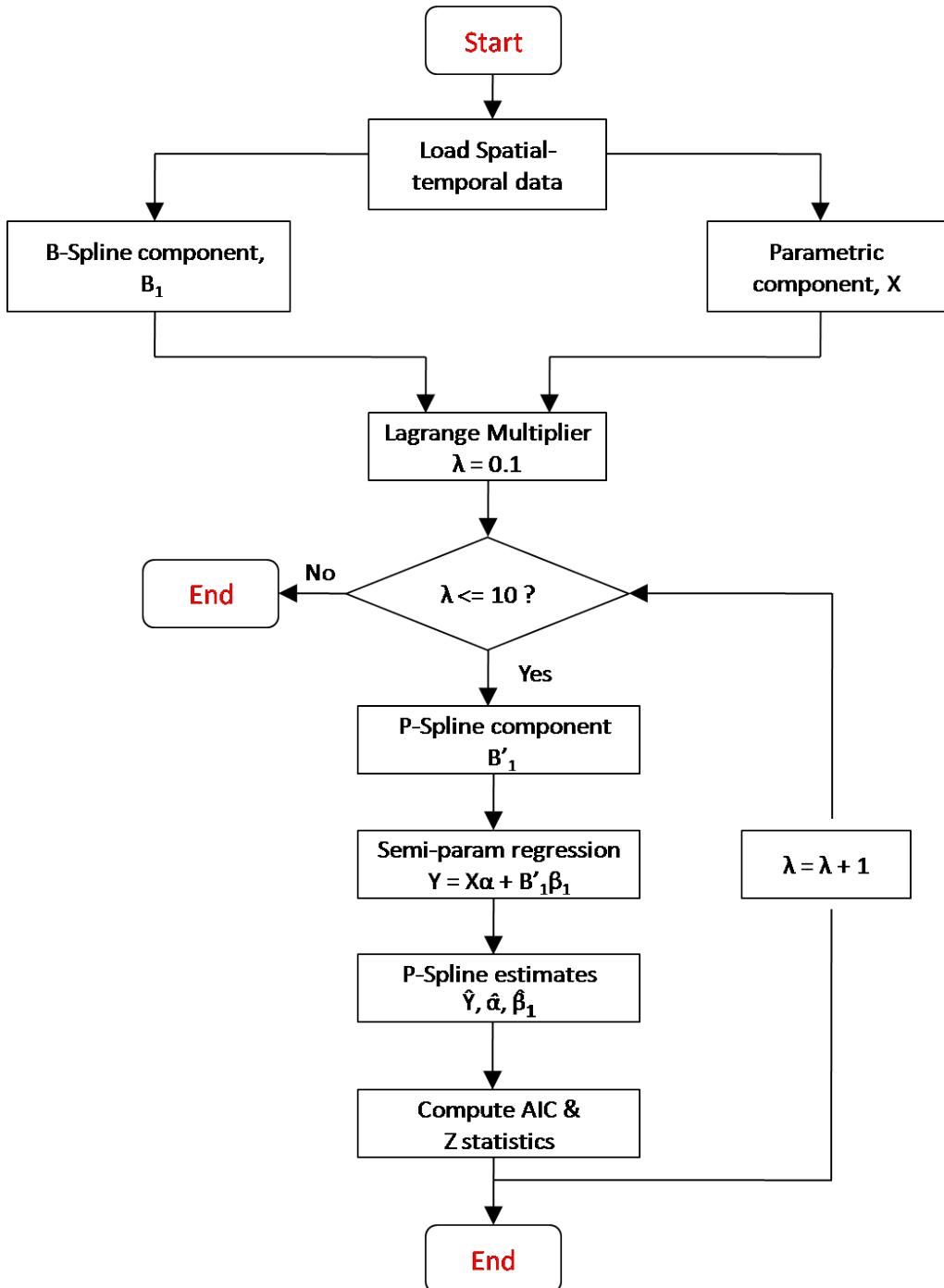


Figure 4.2: An overview of the univariate semiparametric regression model for the modelling of functional intraoperative spatial temporal data

4.2 Optimal Lagrange Multiplier λ

As the Lagrange Multiplier λ influences the extent of P-spline fit, Akaike Information criterion [Aka11] is used to choose the optimal value of λ . AIC optimizes the log likelihood of the fitted model for the effective number of parameters i.e. spline coefficients. The definition of AIC is equivalent to:

$$AIC(\lambda) = \frac{\log(||\mathbf{Y} - \hat{\mathbf{Y}}||^2) + 2\text{tr}(\mathbf{H}) + 2}{m - \text{tr}(\mathbf{H}) - 2} \quad (4.8)$$

where m are time points, the hat matrix $\mathbf{H} = \mathbf{G}(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{S}^T \mathbf{S})^{-1} \mathbf{G}^T$ and $\text{tr}(\mathbf{H})$ is the trace of the hat matrix. $AIC(\lambda)$ is evaluated for different values of λ . The best fitted model is the one which minimizes AIC. Figure 4.3 shows the plot between AIC scores for an arbitrary pixel (say 306869th) calculated with different λ values ranging from 0.1 till 10 for $p = 431$ spline control points. Each iteration of semiparametric regression framework with a different λ leads to a new AIC score for each pixel. Hence, for ten different λ , ten AIC scores are generated for each pixel.

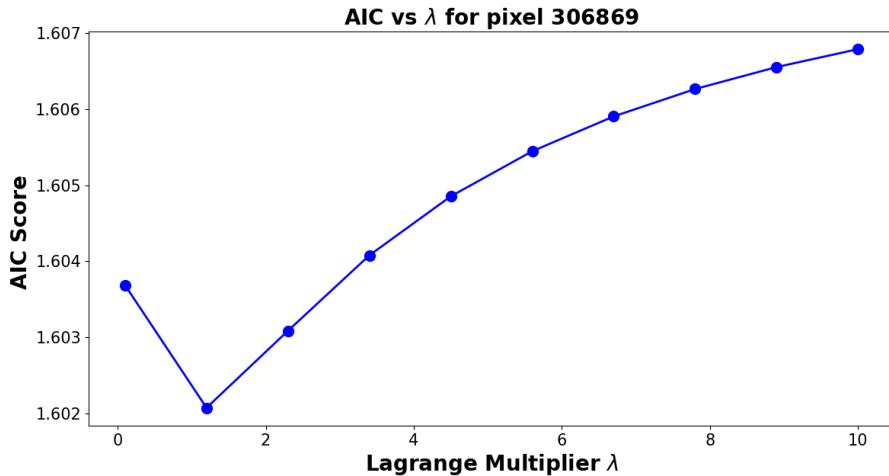


Figure 4.3: The plot between different values of Lagrange Multiplier λ and corresponding values of Akaike Information criterion, $AIC(\lambda)$ for the pixel (say 306869th).

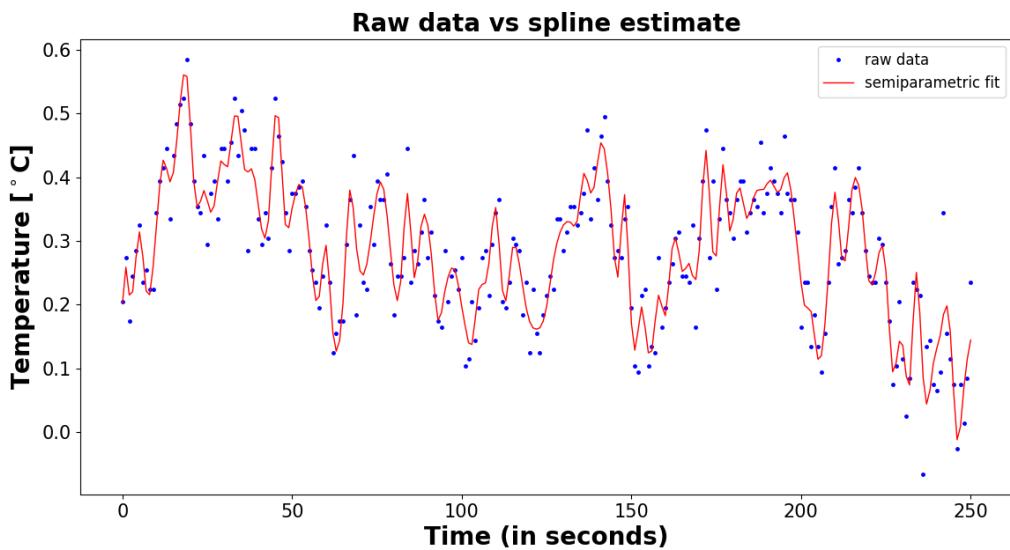


Figure 4.4: Time series of the pixel 306869th. The blue dots resemble the original signal \mathbf{Y} for $n = 1024$ time points. The red line shows the semiparametric regression fit $\hat{\mathbf{Y}}$ using the optimum AIC score.

From Figure 4.3, it is observed that $AIC(\lambda)$ is minimized at $\lambda \approx 1$. Therefore, the estimated spline coefficients at that λ value are the best estimates for the pixel 306869th. Similarly, we found

optimum $AIC(\lambda)$ for all pixels. We visualized the optimum fit of time series of the pixel 306869th in the Figure 4.4 using the semiparametric regression framework. It can be reasoned from Figure 4.4 that the framework does gives us a decent fit by taking care of high bias (under-fitting) and high variance (overfitting).

4.3 Computing Z-statistics

To evaluate the performance of the univariate spline regression model, we calculated Z-statistics which helps in estimating the robustness of the model to detect neuronal activity. Z-statistics gives us an idea about the number of standard deviations by which the parametric component is above the mean value. We first calculate the residual sum of squares (RSS) using the normalized \mathbf{Y} and its P-spline estimate $\hat{\mathbf{Y}} \in \mathbb{R}^{n*m}$. RSS is given by:

$$RSS = \sum_{i=0}^{n-1} (\mathbf{Y} - \hat{\mathbf{Y}})^2 \quad (4.9)$$

Therefore, the Z-statistics are given by [NH18]:

$$\mathbf{Z} = \frac{\hat{\boldsymbol{\alpha}}}{\sqrt{RSS' * \sigma_{\alpha}^2}} \quad (4.10)$$

where $RSS' = RSS/(n - tr(\mathbf{H}) - 1)$ and σ_{α}^2 is the covariance matrix for the parametric component. For fixing the threshold of the Z-statistics to decide about the significant activations, we implemented Bonferroni corrected thresholding levels ($Z \geq 5.2$) as in [RWC03]. Since we evaluate semiparametric regression framework based on ten different λ , there are ten Z-statistics generated for each pixel. Taking an example of pixel 306869th, the AIC gets minimized at $\lambda \approx 1$, therefore, optimum Z-statistics for pixel 306869 is obtained at $\lambda \approx 1$.

4.4 Summary

The P-splines modeled in the semiparametric regression framework provided a decent fit to our intraoperative thermal imaging data. It also fits well in the generalized linear model approach, and its properties are more natural to verify and apprehend. The crucial advantage of using semiparametric regression framework is that its regression property helps in extending univariate spline regression with the parametric components. However, the downside of such a model is that no spatial information is known a priori about the spline coefficients $\hat{\boldsymbol{\beta}}$. Therefore, it provides an opportunity to spatially regularize the spline coefficients with discrete Markov random fields to improve the accuracy of detecting neuronal activity.

Chapter 5

Spatial Regularization using Markov Random fields

This chapter illustrates the extension of univariate spline regression framework by spatial regularization using Markov random fields to compensate the background noise in the intraoperative thermal imaging data. The extended model is expected to improve the detection of specific patterns correlating with neuronal activity. The subsequent sections in this chapter form the core contribution of this thesis. We implemented four different model setups presenting different ways to approach spatial regularization using Markov Random Fields.

5.1 MRF based on Lagrange Multiplier λ

We designed an MRF model based on Lagrange multiplier λ . As discussed in the previous chapter, different values of λ provide a varying degree of fit for a time series. Each values of λ are evaluated based on the Akaike Information Criterion (AIC). However, to maximize the final value of our final Z-statistics, we replaced the AIC evaluation for the optimal λ by a Markov random field approach on all possible λ values. The model does not require any discretization step since the number of possible λ are the number of possible discrete labels k for each pixel. The intuition behind building an MRF model based on λ is that the time series of the adjacent pixels shows similar smoothness behavior. Therefore, after spatial regularization, adjacent pixels should attain similar optimum λ . Figure 5.1 showcases the extension of univariate spline regression framework with a Markov Random field model based on Lagrange multiplier λ . The preceding steps mentioned in the figure are related to the loading of spatial-temporal data and defining parametric as well as non-parametric components whereas the intermediate steps are related to adding a penalty on the non-parametric component and formulating a semiparametric regression framework. The figure is highlighted with two blue rectangular boxes. The right rectangular box represents the steps already discussed in Chapter 4 while the left rectangular box contains the proposed extension of the univariate model with MRF on Lagrange Multiplier λ . Similar representations have been followed in other models as well which are proposed later in this chapter.

5.1.1 Unary potentials

Each iteration of the univariate regression model with a different Lagrange multiplier λ gives the corresponding P-spline estimate \hat{Y} and its Z statistics. Suppose, the univariate spline regression model is evaluated based on k different λ values, then k discrete labels in the range of $[0, 1, \dots, k-1]$ are possible on each of the pixels in the factor graph. Therefore, each pixel now consists of k different Z values for each corresponding λ with $Z \in \mathbb{R}^{k*m}$ and m being the number of pixels. The unary potentials are modeled using the reciprocal of the Z-statistics at each pixel of the intraoperative data. The reciprocal of Z-statistics helps in minimizing the overall energy function

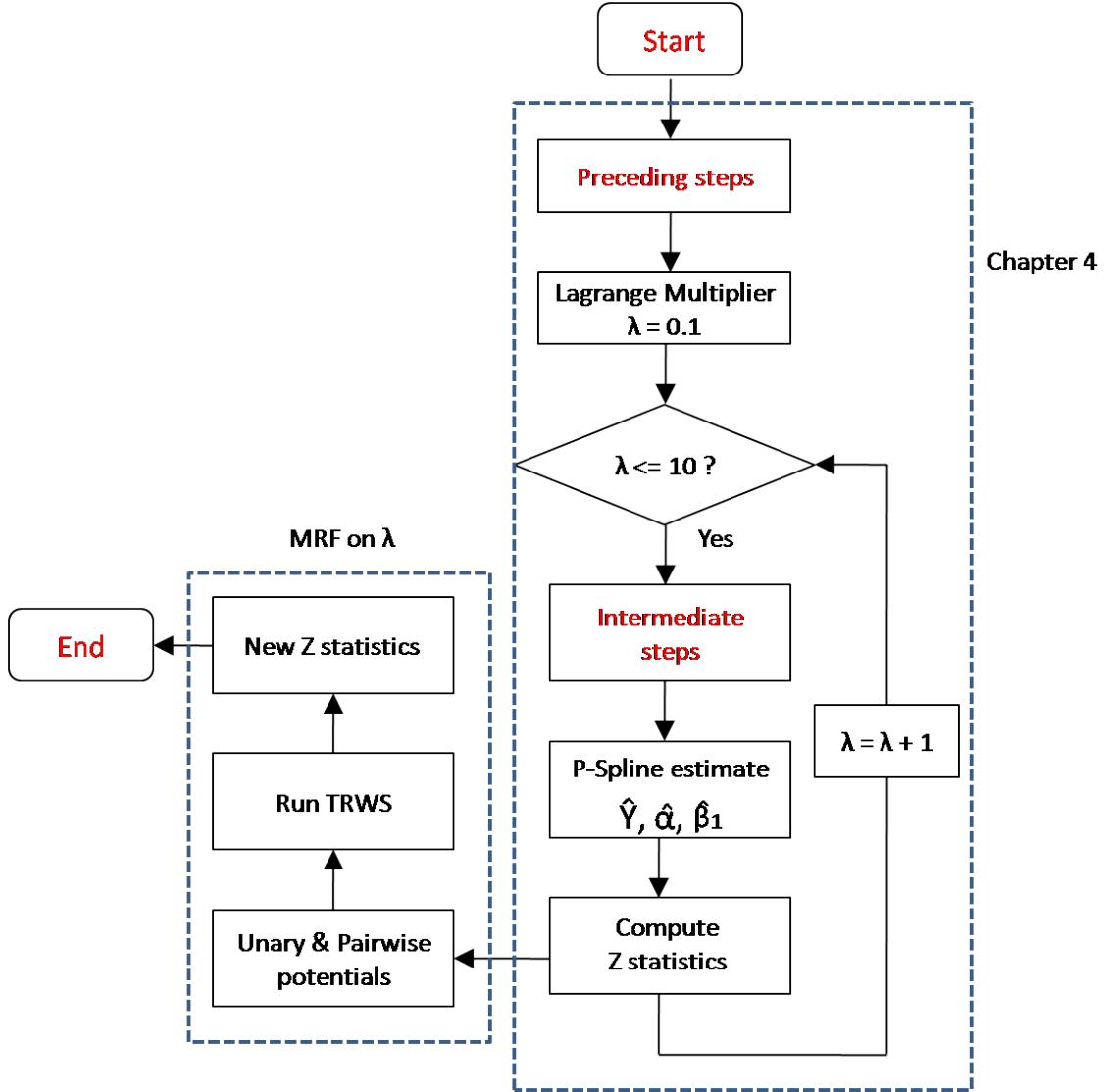


Figure 5.1: An overview of the proposed MRF model based on Lagrange Multiplier λ for modelling of intraoperative thermal imaging data

in the Markov random field framework by penalizing the discrepancy between the Z-statistics for the adjacent pixels and ultimately finding the maximum a posteriori probability (MAP). It is especially helpful for performance since there is no discretization error involved and the solution is more likely to be unique. Hence, the unary potential for a pixel q at a specific λ_k is calculated as:

$$\theta_q(\lambda_k) = \frac{1}{Z_q(\lambda_k)} \quad (5.1)$$

where $Z_q \in \mathbb{R}$ is the Z-statistics for the pixel q at λ_k where $k \in [0, 1, \dots, k - 1]$ are the possible discrete labels. Therefore, pixel q has the unary potential vector $\theta_q \in \mathbb{R}^k$ given as:

$$\theta_q = \begin{bmatrix} \theta_q(\lambda_0) \\ \theta_q(\lambda_1) \\ \vdots \\ \theta_q(\lambda_{k-1}) \end{bmatrix},$$

Figure 5.2 shows the assignation of unary potentials to the pixels p and q respectively. The pixels have been assigned k different discrete labels. The label which minimizes the overall energy function is assigned to the pixel as the maximum a posteriori estimate.

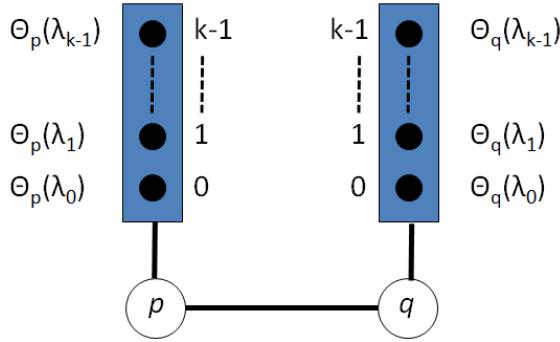


Figure 5.2: Unary potentials assigned to each labels of pixel p and q for MRF based on λ

5.1.2 Pairwise potentials

Generally, in real images including our intraoperative thermal imaging data, neighboring pixels are homogeneous which means that they usually have similar characteristics such as temperature intensity. Hence, pairwise potential is defined as the cost of assigning different labels to adjacent pixels whereas there is no cost for two adjacent pixels with the same label. Higher the difference between the value of labels between adjacent pixels, more significant is the cost. We apply l_1 norm to define pairwise potential between the adjacent pixels which is given as:

$$\theta_{pq}(i, j) = \gamma ||i - j||_1, \quad (5.2)$$

where $\theta_{pq} \in \mathbb{Z}_{\geq 0}^{k*k}$ is the pairwise potential for all combinations of discrete label values, and $i, j \in [0, 1, \dots, k - 1]$ are the discrete labels assigned to pixel p and q respectively, $\gamma \in \text{IR}$ is the pairwise potential weight. It is to be noted that pairwise potential doesn't depend on the input data and can be fixed a priori for the fixed number of labels.

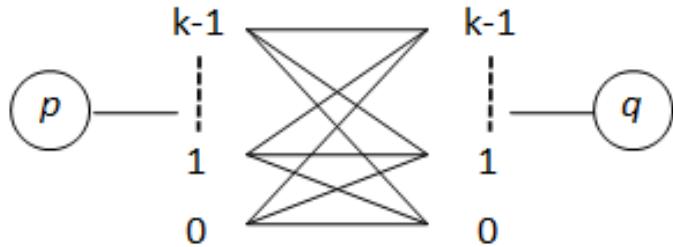


Figure 5.3: The interaction between labels of adjacent pixels. In this figure, k labels are defined on each pixel.

5.1.3 Analysis of pairwise potential weight γ

γ plays a crucial role in determining the extent of spatial regularization achieved. It is intuitive from the energy function $U(\mathbf{X})$ discussed in Chapter 2 that the value of γ determines the energy $U(\mathbf{X})$. When the adjacent pixels (say p and q) have different labels assigned to them, the joint probability $P(\mathbf{X})$ is affected by a factor e^γ . A higher value of γ results in a higher value of the energy $U(\mathbf{X})$. Therefore, there is a higher cost for assigning different labels to the adjacent pixels. Eventually, a higher level of smoothness is obtained after finding the Maximum a posteriori estimate (MAP) estimate since the adjacent pixels have the tendency of attaining similar discrete labels. In short, γ is the fundamental property which determines the amount of spatial smoothness. If the value of γ is very high, the pairwise potential is high resulting in over smoothing while a very low value of γ leads to under smoothing. A value of zero doesn't effect smoothing at all due to which we see no spatial regularization.

5.1.4 Computing new Z-statistics

After defining unary and pairwise potentials, a graphical model based on the factor graph is created. The inference procedure Sequential Tree reweighted message passing (TRWS) as in [Kol06] is applied to the factor graph. The stopping condition for the TRWS algorithm is defined as the state when the gap between the energy and bound values is less than 10^{-6} . When the maximum a posteriori probability or the minimum energy is attained by maximizing the lower bound, the inference algorithm converges, and the inferred optimum labels are assigned to each pixel. Let the optimum labels inferred for each pixel be given as $k_{inf} \in \mathbb{Z}_{\geq 0}^m$, therefore, the optimum Z values for every pixel after spatial regularization based on Lagrange Multiplier λ is given as:

$$\mathbf{Z} = \begin{bmatrix} Z_{k_{inf}}^0 \\ Z_{k_{inf}}^1 \\ \vdots \\ \vdots \\ \vdots \\ Z_{k_{inf}}^{m-1} \end{bmatrix} \quad (5.3)$$

where $\mathbf{Z} \in \mathbb{R}^m$ are the new spatially regularized values of Z for all m pixels, $Z_{k_{inf}}^{m-1} \in \mathbb{R}$ is the Z value at the optimal label k_{inf} for the $(m - 1)^{th}$ pixel.

5.1.5 Drawbacks

The drawback of the spatial regularization based on Lagrange Multiplier λ is that it uses an arbitrary range of λ values and the number of λ acting as the number of labels k are also uncertain. Currently, we defined the range of λ between 0 and 10 with $k = 10$. However, a value of λ out of this range may be an optimum λ value for a particular time series. It is observable that the limitation is inherited from the univariate spline regression framework.

5.2 MRF based on Z-statistics

The second model has been designed to spatially regularize the final Z-statistics obtained from univariate spline regression model in Chapter 4. In this model, we define two possible discrete labels on each pixel. The selection of two labels i.e. $k = 2$ has been made based on our eventual target of differentiating the pixels only by the criterion of existence or non-existence of neuronal activity. Figure 5.4 shows the workflow of the model. The workflow is identical to the semiparametric regression model discussed in the Chapter 4, however with an extension of Markov random field component based on Z-statistics.

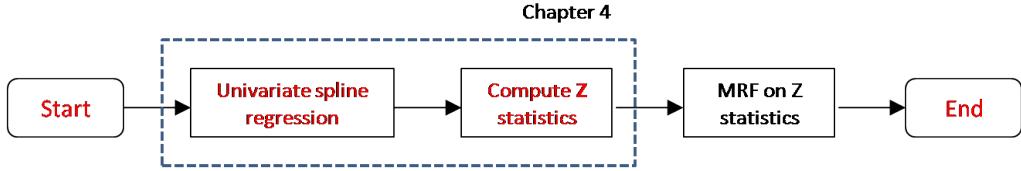


Figure 5.4: The univariate spline regression model extended by an MRF model on Z-statistics.

5.2.1 Unary potentials

As discussed in Chapter 4, the threshold value of the Z-statistics which determines the presence/non-presence of neuronal activity has been set as 5.2. We, therefore, aim to evaluate the l_2 norm between the Z values of the pixel and the threshold value 5.2. The following equations model the unary potentials of a pixel q as:

$$\begin{aligned}\theta_q(0) &= \|Z_q - 5.2\|_2^2 \\ \theta_q(1) &= \|5.2 - Z_q\|_2^2\end{aligned}\tag{5.4}$$

where $\theta_q \in \mathbb{R}^2$, $\theta_q(0)$ defines unary potential at 0th label, $\theta_q(1)$ defines unary potential at 1st label and $Z_q \in \mathbb{R}$ is the Z value at pixel q . Figure 5.5 shows the assignation of unary potentials to the pixels p and q . The pixels have been assigned 2 discrete labels. The label which minimizes the overall energy function is assigned to each of the pixels after inference.

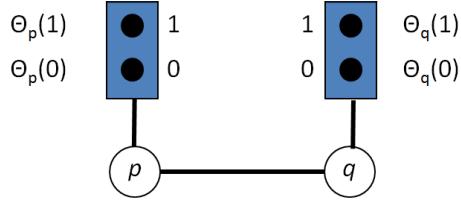


Figure 5.5: Unary potentials assigned to each labels of pixel p and q respectively

5.2.2 Pairwise potentials

We use l_1 norm to model the pairwise potentials as shown in section 5.1.2. However, the label set i and j only take the values $[0, 1]$ and therefore the pairwise interactions with the adjacent pixels look like the Figure 5.6 shown below. The value of pairwise potential weight γ plays a crucial role in determining the degree of smoothness in this model as well.

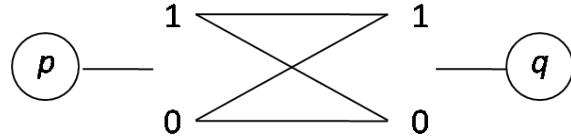


Figure 5.6: Figure showing the interaction between labels of adjacent pixels for MRF on Z values.

5.2.3 Advantages

One of the main advantages of creating a Markov random field on Z statistics is that it is computationally inexpensive. Since a maximum of two labels are assigned to each pixel in the factor graph, TRWS has a short runtime till the energy function is minimized.

5.3 MRF on spline coefficients

Although, the univariate spline regression model presented in chapter 4 and the two MRF frameworks so far discussed provided robust models to estimate our spatial-temporal data, the background noise due to high-frequency physiological effects and periodic non-uniformity correction still propagates to the spline coefficients β and the signal estimates $\hat{\mathbf{Y}}$. So far, the MRF models that have been discussed doesn't compensate these effects. Therefore, the motivation of this section is to develop a model which spatially regularizes the spline coefficients $\hat{\beta}$. We discretize the spline coefficients using Gaussian Mixture model with k Gaussian components. We then construct a graphical model based on the derived mean values of each component and finally run the inference procedure TRWS on the graphical model. The resultant MRF estimate $\hat{\mathbf{Y}}_{\text{mrf}}$ is negated from our input data \mathbf{Y} in the penalized normal equation to get new estimates of spline coefficients $\hat{\beta}_{\text{fit}}$ thereby compensating the background noise. We intend to develop two different sub-models based on spatial regularization of spline coefficients.

5.3.1 Single non-parametric component

The first sub-model is based on the semiparametric regression framework with a single non-parametric component. Figure 5.8 shows the overall flow of the proposed MRF model. The univariate spline regression model provides P-spline estimate $\hat{\mathbf{Y}}$, the coefficients of parametric component $\hat{\alpha}$ and non-parametric component $\hat{\beta}_1$.

The discretization strategy employs Gaussian Mixture Model (GMM) on the spline coefficients $\hat{\beta}_1 \in \mathbb{R}^{P*m}$ of the single non-parametric component. The GMM algorithm considers the spline coefficients, i.e., $\hat{\beta}_1$ as its data input on which discretization is performed with k number of Gaussian components. For obvious reasons, we term the number of labels as the number of Gaussian components. Each component k has its Gaussian distribution and a diagonal covariance matrix \mathbf{C} . The algorithm calculates optimum mean values $\mu_k \in \mathbb{R}^P$ for each k . It also predicts optimum labels for each data point m in $\hat{\beta}_1$. The Expectation Maximization algorithm calculates the optimal means $\hat{\mu}_k$. The steps of expectation and maximization are performed iteratively until the values of $\hat{\mu}_k$ converge, providing the maximum likelihood estimate. An optimum mean value $\hat{\mu}_k$ for component k is given by:

$$\hat{\mu}_k = \frac{\sum_{i=0}^{m-1} \hat{P}_{ik} \beta_{1i}}{\sum_{i=0}^{m-1} \hat{P}_{ik}} \quad (5.5)$$

where m are the number of data points or pixels, \hat{P}_{ik} is the probability that a data point $\hat{\beta}_{1i}$ are the spline coefficients at pixel i . Figure 5.7 shows that a discrete label corresponds to a mean value obtained from the discretization.

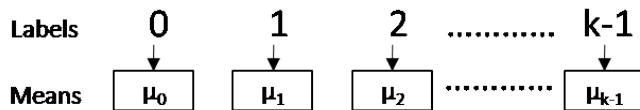


Figure 5.7: μ_{k-1} represents mean value for the Gaussian component $k - 1$.

5.3.1.1 Unary potentials

During the discretization of the spline coefficients, we assign a discrete label to each pixel and calculate mean values $\hat{\mu}_k$. The unary potential introduces a cost for assigning a discrete label to a particular pixel. We attempt to use l_2 Norm to model the deviation of the signal \mathbf{Y} for a particular pixel from its estimate $\hat{\mathbf{Y}}_k$. The estimate $\hat{\mathbf{Y}}_k$ for a Gaussian component k is given as:

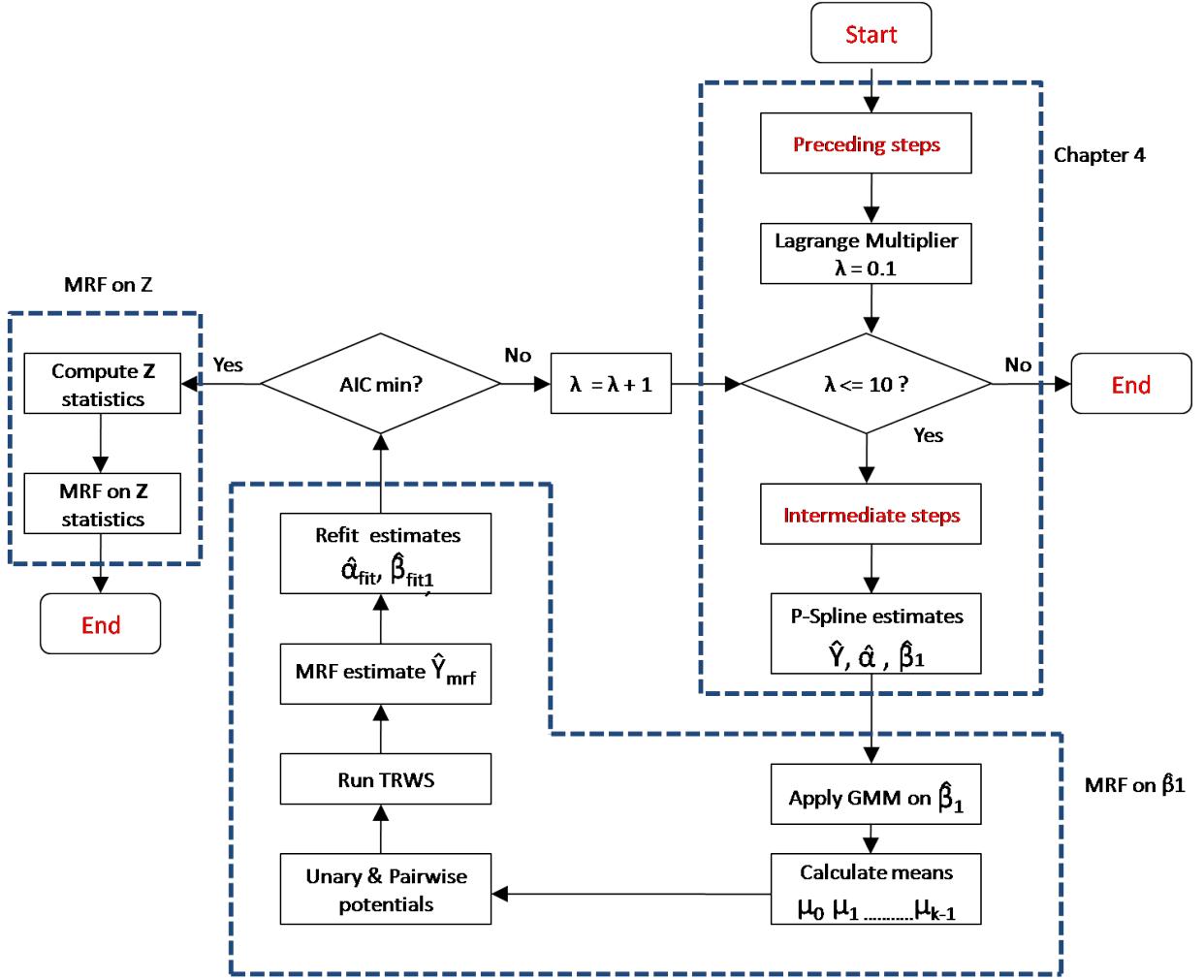


Figure 5.8: An overview of the proposed MRF model based on a single non-parametric component.

$$\hat{\mathbf{Y}}_k = \mathbf{G} \begin{vmatrix} \hat{\boldsymbol{\alpha}} \\ \hat{\mu}_k \end{vmatrix} \quad (5.6)$$

where $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{1*m}$ are the estimated parametric coefficients from our univariate spline regression model, $\hat{\mu}_k \in \mathbb{R}^{p*m}$ are the means calculated using Gaussian Mixture model for the k^{th} component with p being the number of control points, $\mathbf{G} \in \mathbb{R}^{n*(1+p)}$ is the matrix $[\mathbf{X} \ \mathbf{B}_1]$ containing parametric as well as the single non-parametric basis component with n being the number of time points. The unary potential for pixel q are calculated by iterating the below equation for each of the Gaussian components k :

$$\theta_q(\mu_k) = \|\mathbf{Y} - \hat{\mathbf{Y}}_k\|_2^2 \quad (5.7)$$

where $\theta_q \in \mathbb{R}^k$ are the unary potentials for pixel q at all k Gaussian components, $\mathbf{Y} \in \mathbb{R}^{n*m}$ is the input signal and $\hat{\mathbf{Y}}_k \in \mathbb{R}^{n*m}$ is the signal estimate based on discretization of spline coefficients.

5.3.1.2 Pairwise potentials

We use l_1 norm for defining pairwise potentials as stated in the previous models. The pairwise potential $\theta_{pq}(i, j)$ for pixel p and q is given as:

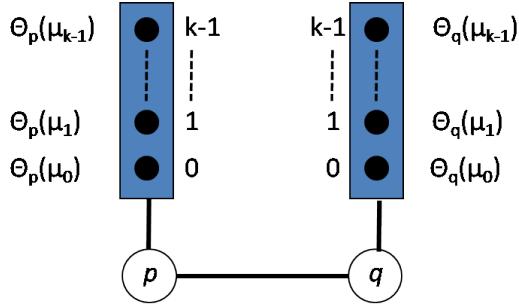


Figure 5.9: The unary potentials defined for k Gaussian components at pixels p and q .

$$\theta_{pq}(i, j) = \gamma \|i - j\|_1, \quad (5.8)$$

where $i, j \in [0, 1, \dots, k - 1]$ are the possible labels on each pixel, γ is the pairwise potential weight. The only difference between the pairwise potentials applied in this model with the previous models is the way we define the labels. The MRF on Lagrange Multiplier takes the number of different values of λ as the possible labels on each pixel, whereas MRF on Z-statistics have pixels with only two labels defined on them. The effect of pairwise potential weight γ on overall degree of smoothness is same as explained before. Figure 5.3 shows the pixel-pixel interactions to calculate the pairwise potential.

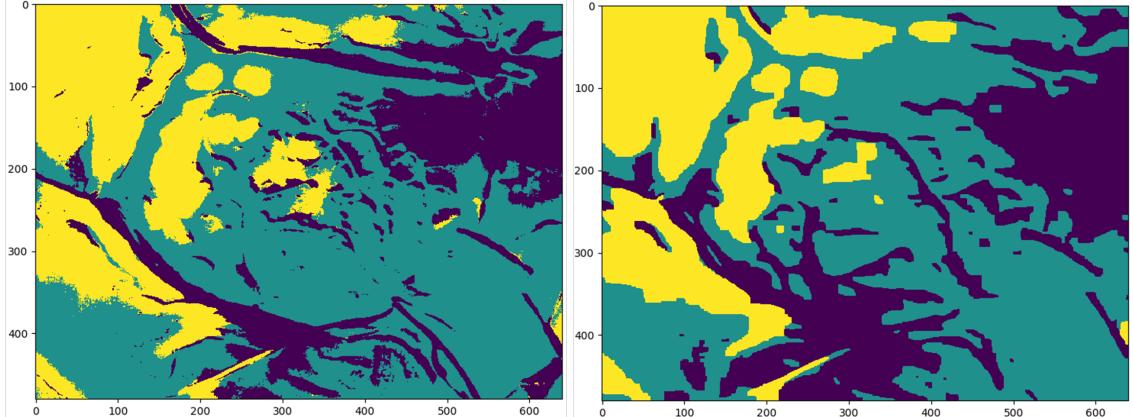


Figure 5.10: The left figure shows the labels assigned to each pixel of our intraoperative data by the Gaussian Mixture Model. The right figure shows the inferred labels after MAP estimate using TRWS algorithm. The results are for $p = 431$ control points, $m = 307200$ pixels, $k = 3$ components, $\gamma = 1$ and $n = 1024$ time points. A relative smoothness has been observed in the second figure.

5.3.1.3 Refit of the parametric component

The pixels are assigned their optimum mean values corresponding to there optimum labels after running TRWS inference. Let $k_{inf} \in \mathbb{Z}_{\geq 0}^m$ is the inferred label on all pixels and $\mu_{inf}^{m-1} \in \mathbb{R}^p$ corresponds to the mean value for the inferred label on $(m - 1)^{th}$ pixel. We therefore calculate the MRF spline coefficients estimate $\hat{\beta}_{mrf} \in \mathbb{R}^{p*m}$ by stacking all the inferred mean values of each pixels using the following equation:

$$\hat{\beta}_{mrf} = \begin{vmatrix} \mu_{inf}^0 \\ \mu_{inf}^1 \\ \vdots \\ \mu_{inf}^{m-1} \end{vmatrix} \quad (5.9)$$

The MRF signal estimate $\hat{\mathbf{Y}}_{mrf}$ are now calculated by the following operation:

$$\hat{\mathbf{Y}}_{mrf} = \mathbf{B}_1 \hat{\beta}_{mrf} \quad (5.10)$$

where $\mathbf{B}_1 \in \mathbb{R}^{n*p}$ is the spline basis matrix or the single non-parametric component. From Figure 5.11 it is clear that $\hat{\mathbf{Y}}_{mrf}$ leads to a more rough fit on the signal data \mathbf{Y} and shifts some local effects into the target pixels. Therefore, the parametric component has to be refitted to validate the target signal with the hypothesis that they still possess Gaussian distribution activation pattern. This is done by finding new parametric and spline coefficients $\hat{\alpha}_{fit}$ and $\hat{\beta}_{fit1}$ respectively when the signal \mathbf{Y} has been deducted by the MRF estimate $\hat{\mathbf{Y}}_{mrf}$ using the penalized normal equation. The raw data \mathbf{Y} shows increasing or decreasing linear trends called step artifacts due to periodic non-uniformity correction. This deduction of $\hat{\mathbf{Y}}_{mrf}$ from the raw data \mathbf{Y} compensates these effects. It can be observed from the figure that $\hat{\mathbf{Y}}_{fit}$ doesn't show any trends and is centered around zero. The penalized normal equation to calculate the refitted coefficients is given as:

$$\begin{vmatrix} \hat{\alpha}_{fit} \\ \hat{\beta}_{fit1} \end{vmatrix} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{S}^T \mathbf{S})^{-1} \mathbf{G}^T (\mathbf{Y} - \hat{\mathbf{Y}}_{mrf}) \quad (5.11)$$

The refitted model estimate $\hat{\mathbf{Y}}_{fit} \in \mathbb{R}^{n*m}$ is now given by:

$$\hat{\mathbf{Y}}_{fit} = \mathbf{G} \begin{vmatrix} \hat{\alpha}_{fit} \\ \hat{\beta}_{fit1} \end{vmatrix} \quad (5.12)$$

The residuals $\hat{\mathbf{Y}}_{res} \in \mathbb{R}^{n*m}$ becomes:

$$\hat{\mathbf{Y}}_{res} = (\mathbf{Y} - \hat{\mathbf{Y}}_{mrf}) - \hat{\mathbf{Y}}_{fit} \quad (5.13)$$

The model statistics such as Residual Sum of Squares (RSS) are calculated using the following equation:

$$RSS = \sum_{i=0}^{n-1} (\hat{\mathbf{Y}}_{res})^2 \quad (5.14)$$

where n are the number of time points. Using the RSS values, the final Z-statistics are calculated as:

$$\mathbf{Z} = \frac{\hat{\alpha}}{\sqrt{RSS' * \sigma_\alpha^2}} \quad (5.15)$$

where $RSS' = RSS/(n - tr(\mathbf{H}) - 1)$, the hat matrix $\mathbf{H} = \mathbf{G}(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{S}^T \mathbf{S})^{-1} \mathbf{G}^T$ and $tr(\mathbf{H})$ is the trace of the hat matrix, σ_α^2 is the covariance matrix for the parametric component. We finally apply spatial regularization on the resultant Z values as explained in the section 5.2. By doing so, we integrate the MRF on spline coefficients of a single non-parametric component with MRF on Z-statistics to attain an improved accuracy of detecting neuronal activity.

5.3.1.4 Drawbacks

The discretization of spline coefficients using Gaussian Mixture Model is computationally expensive. It is due to the high dimensionality of input data (i.e., spline coefficients). If k is the number of Gaussian components used, m is the number of data points or pixels and i being the number of iterations of Expectation Maximization (EM) algorithm then to get the optimum values

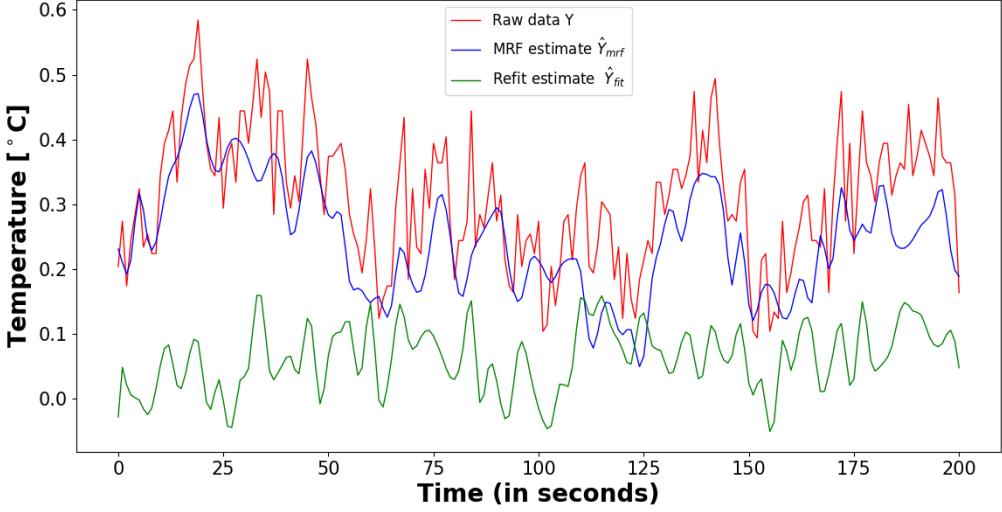


Figure 5.11: The figure shows the time series values of \mathbf{Y} , $\hat{\mathbf{Y}}_{mrf}$, and $\hat{\mathbf{Y}}_{fit}$ for a pixel (say 306869th). It is visualized from the figure that $\hat{\mathbf{Y}}_{mrf}$ results in a rough estimate of \mathbf{Y} . After compensating $\hat{\mathbf{Y}}_{mrf}$ from \mathbf{Y} , $\hat{\mathbf{Y}}_{fit}$ is obtained which is centered around zero. The figure only shows the first 200 time points out of total $n = 1024$ time points for better visualization.

after convergence of EM algorithm, the time complexity becomes $O(i * m^k)$. One alternative is to use spherically invariant random process (SIRP) representation of the multivariate Gaussian distribution. However, SIRP still needs to use k Gaussian components as well as all the pixels to calculate the means using EM algorithm. This method, however, should decrease the time complexity since it does not use exponential functions but we do not expect a drastic reduction since the dimensionality of the input data remains the same.

Another possible solution is the dimensionality reduction using Principal component analysis (PCA) (see Appendix B). PCA converts the spline coefficients in a lower dimensional space. This instance of spline coefficients in smaller dimension is used as the input data for the Gaussian Mixture model. However, one of the fundamental requirements of PCA algorithm is to choose a discrete number of principal components based on the explained variance. However, by doing so, some crucial feature information from the data is lost since the principal components with the least explained variance also contribute to the detection of neuronal activity.

5.3.2 Two non-parametric components

We designed a second sub-model based on spatial regularization of spline coefficients. In this sub-model, we use two non-parametric components in the univariate spline regression framework and spatially regularize the spline coefficients of one of the component. The idea behind adding a new non-parametric component is to address the interference of high-frequency effects in our data such as heart rate or respiration. Two non-parametric components with a high number of control points should compensate these effects better than using a single non-parametric component since two non-parametric components acts as an enhanced low pass filter. The semiparametric regression model with two non-parametric components therefore becomes:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\alpha} + \mathbf{B}_1\boldsymbol{\beta}_1 + \mathbf{B}_2\boldsymbol{\beta}_2 + \mathbf{e} \quad (5.16)$$

where $\mathbf{B}_1 \in \mathbb{R}^{n*p}$ is the B-spline basis matrix of the first non-parametric component with p being the number of control points and $\mathbf{B}_2 \in \mathbb{R}^{n*p}$ is the second B-spline basis matrix of the second non-parametric component, $\boldsymbol{\beta}_1 \in \mathbb{R}^{p*m}$ and $\boldsymbol{\beta}_2 \in \mathbb{R}^{p*m}$ are there corresponding spline coefficients.

The above configuration therefore changes the $\mathbf{G} \in \mathbb{R}^{n*(2p+1)}$ matrix in the following structure :

$$\mathbf{G} = [\mathbf{X} \quad \mathbf{B}_1 \quad \mathbf{B}_2] \quad (5.17)$$

The penalty matrix $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^{p*p}$ is added to the block diagonal \mathbf{S} matrix:

$$\mathbf{S} = blkdiag(0_k, \mathbf{P}_1, \mathbf{P}_2) \quad (5.18)$$

The univariate spline regression estimate of the coefficients, $\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2$ are calculated by using the following penalized normal equation:

$$\begin{vmatrix} \hat{\alpha} \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{vmatrix} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{S}^T \mathbf{S})^{-1} \mathbf{G}^T (\mathbf{Y}) \quad (5.19)$$

The discretization strategy employs Gaussian Mixture Model(GMM) on the spline coefficients of the second non-parametric component i.e $\hat{\beta}_2$. The procedure for discretization is as per the discussion in the previous sub-model. We again assume k as the number of Gaussian components each having its Gaussian distribution. The Expectation Maximization algorithm is applied to calculate optimum mean values $\hat{\mu}_k$ for each of the Gaussian components. It also predicts the optimum labels for each pixel, i.e., data points. We obtain an optimum mean value $\hat{\mu}_k$ for the component k which is given by:

$$\hat{\mu}_k = \frac{\sum_{i=0}^{m-1} \hat{P}_{ik} \beta_{2i}}{\sum_{i=0}^{m-1} \hat{P}_{ik}} \quad (5.20)$$

where m are the number of data points or pixels, \hat{P}_{ik} is the probability that a data point β_{2i} is generated by the Gaussian component k .

5.3.2.1 Unary potentials

Although we still use the l_2 norm as we did in calculating the unary potentials for the previous models, there are slight modifications in its definition for the current model. The \mathbf{G} matrix now has \mathbf{X} , \mathbf{B}_1 and \mathbf{B}_2 as its components. Also, the mean values are calculated only for spline coefficients $\hat{\beta}_2$. Therefore, the estimate $\hat{\mathbf{Y}}_k$ for Gaussian component k is given as:

$$\hat{\mathbf{Y}}_k = \mathbf{G} \begin{vmatrix} \hat{\alpha} \\ \hat{\beta}_1 \\ \hat{\mu}_k \end{vmatrix} \quad (5.21)$$

where $\hat{\alpha} \in \mathbb{R}^{1*m}$ are the estimated parametric coefficients, $\hat{\beta}_1 \in \mathbb{R}^{p*m}$ are the spline coefficients for the first non-parametric component and $\hat{\mu}_k \in \mathbb{R}^{p*m}$ are the means calculated using Gaussian Mixture model on spline coefficients $\hat{\beta}_2$. $\mathbf{G} \in \mathbb{R}^{n*(1+2p)}$ is the matrix containing parametric as well as both non-parametric components with n being the number of time points. The unary potentials for pixel q are calculated by iterating the below equation for each Gaussian components k :

$$\theta_q(\mu_k) = \|\mathbf{Y} - \hat{\mathbf{Y}}_k\|_2^2 \quad (5.22)$$

where $\theta_q \in \mathbb{R}^k$ is the unary potential for pixel q at all Gaussian components, $\mathbf{Y} \in \mathbb{R}^{n*m}$ is the input signal, $\hat{\mathbf{Y}}_k \in \mathbb{R}^{n*m}$ is the estimate based on discretization of spline coefficients. In order to visualize the assignation of unary potentials, we refer the Figure 5.9.

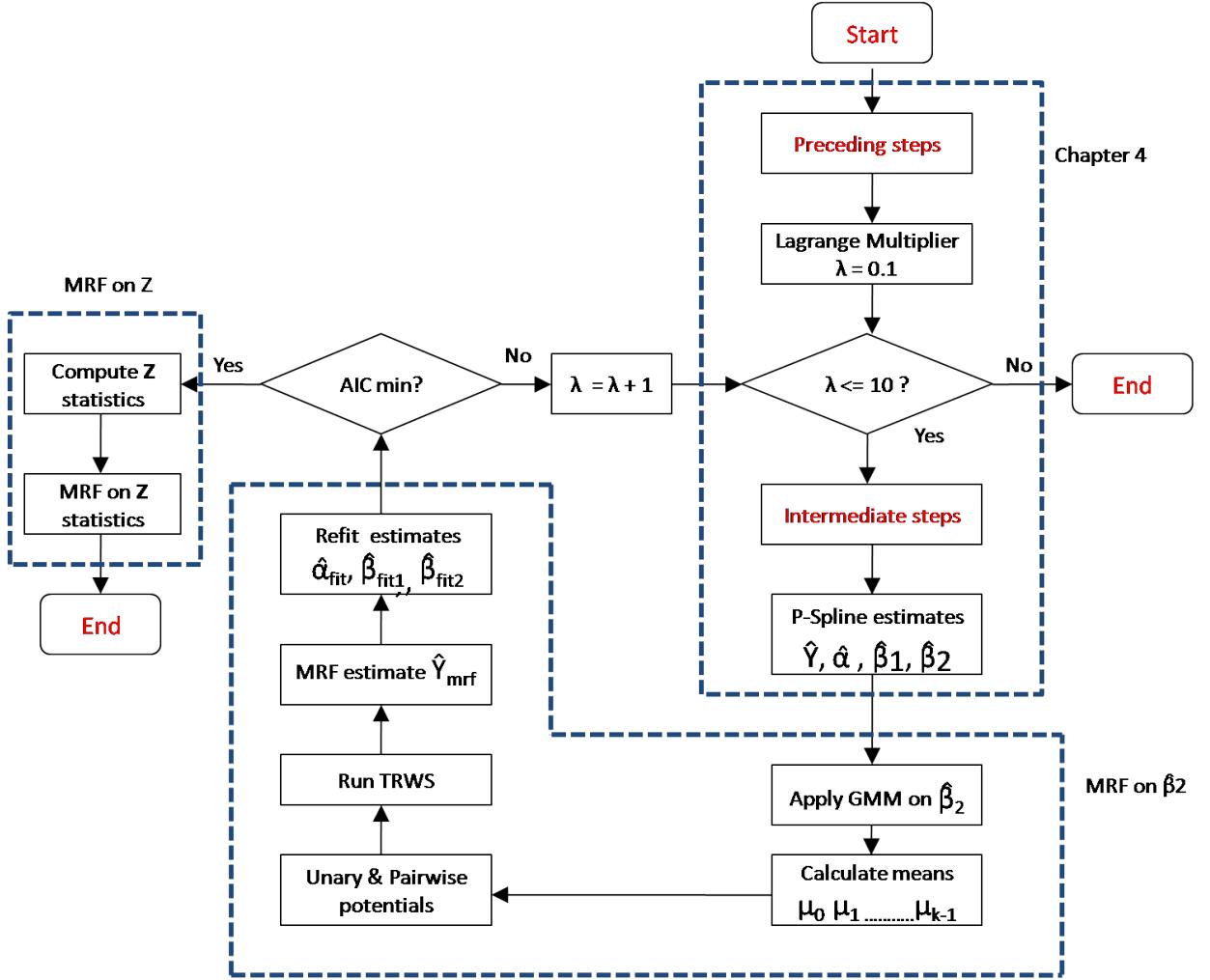


Figure 5.12: An overview diagram of the proposed MRF model based on spline coefficients of a single non-parametric component in a two non-parametric component spline regression framework.

5.3.2.2 Pairwise potentials

We use l_1 norm for modeling the pairwise potentials and all the properties of the pairwise potential are as defined with the MRF on single non-parametric component. Figure 5.3 shows the interactions between adjacent pixels. The pairwise potential is therefore given as:

$$\theta_{pq}(i, j) = \gamma ||i - j||_1, \quad (5.23)$$

where $i, j \in [0, 1, \dots, k - 1]$ are the possible labels on each pixel, γ is the pairwise potential weight.

5.3.2.3 Refit of the parametric component

Let $k_{inf} \in \mathbb{Z}_{\geq 0}^m$ are the inferred labels on all the pixels after TRWS inference convergence. Let $\mu_{inf}^{m-1} \in \mathbb{R}^p$ corresponds to the mean values for the inferred label on $(m - 1)^{th}$ pixel. We calculate the spatially regularized spline coefficients $\hat{\beta}_{2mrf} \in \mathbb{R}^{p*m}$ by stacking all the inferred mean values of each pixels using the following notation:

$$\hat{\beta}_{2mrf} = \begin{vmatrix} \mu_{inf}^0 \\ \mu_{inf}^1 \\ \vdots \\ \vdots \\ \vdots \\ \mu_{inf}^{m-1} \end{vmatrix} \quad (5.24)$$

The MRF signal estimate $\hat{\mathbf{Y}}_{mrf}$ is finally calculated as:

$$\hat{\mathbf{Y}}_{mrf} = \mathbf{B}_2 \hat{\beta}_{2mrf} \quad (5.25)$$

where $\mathbf{B}_2 \in \mathbb{R}^{n*p}$ is the basis matrix for the second non-parametric component. We now deduct $\hat{\mathbf{Y}}_{mrf}$ from \mathbf{Y} and refit the parametric component after performing the following penalized normal equation:

$$\begin{vmatrix} \hat{\alpha}_{fit} \\ \hat{\beta}_{fit1} \\ \hat{\beta}_{fit2} \end{vmatrix} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{S}^T \mathbf{S})^{-1} \mathbf{G}^T (\mathbf{Y} - \hat{\mathbf{Y}}_{mrf}) \quad (5.26)$$

The refitted model estimate $\hat{\mathbf{Y}}_{fit} \in \mathbb{R}^{n*m}$ is therefore given by:

$$\hat{\mathbf{Y}}_{fit} = \mathbf{G} \begin{vmatrix} \hat{\alpha}_{fit} \\ \hat{\beta}_{fit1} \\ \hat{\beta}_{fit2} \end{vmatrix} \quad (5.27)$$

We find the residual estimates, RSS and Z-statistics using Eq. (5.13 - 5.15) of the previous section. We finally spatially smooth the resultant Z-statistics based on our second model mentioned in section 5.2 to improve the accuracy further.

Chapter 6

Performance Evaluation

Some of the basic definitions that needs to be discussed before we reflect on the performance results are summarized below:

Resting state data (Raw data): It contains the temperature recordings of the cortex when the subject is not involved with an explicit stimulus. The resting state is observed through changes in blood flow which creates blood-oxygen-level-dependent (BOLD) signal. The signal is influenced by many physiological factors other than neuronal activity such as respiration and heartbeat. Therefore, the data is noisy and contains low and high-frequency temporal interactions. These intraoperative recordings act as our baseline dataset.

Augmented data (Signal data): In order to generate a synthetic activity in the human cortex in addition to its resting state behavior, we augment the baseline raw dataset by a synthetic activity function in order to accurately quantify the performance of our proposed frameworks. We model the activation pattern with a Gaussian distribution with mean $\mu = 30$, standard deviation $\sigma = 10$ and a phase duration of 30 seconds.

Ground truth : The activation pattern is added to four circular areas of the baseline dataset with a varying radius. Out of $m = 307200$ pixels in our intraoperative data, 303707 pixels are not affected by the activation whereas rest of the pixels are affected by this augmentation.

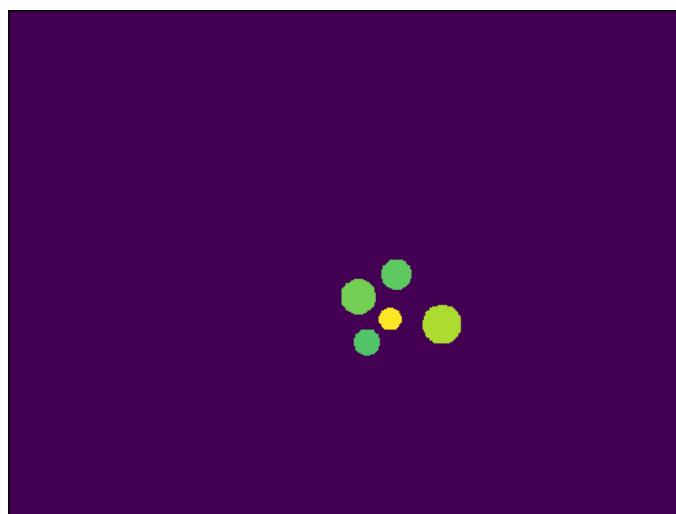


Figure 6.1: Ground truth showing four circular areas of neuronal activity

True positive rate (TPR): True positive rate measures the proportion of the number of

pixels which are correctly identified as containing neuronal activity among all the activated pixels.

$$TPR = \frac{TP}{TP + FN} \quad (6.1)$$

where TP are the number of pixels which are correctly identified as containing neuronal activity, FN are the number of pixels which are incorrectly identified as not containing neuronal activity.

False positive rate (FPR): False positive rate measures the proportion of the number of pixels which are incorrectly identified as containing neuronal activity among all the non-activated pixels.

$$FPR = \frac{FP}{FP + TN} \quad (6.2)$$

where FP are the number of pixels which are incorrectly identified as containing neuronal activity whereas TN are the number of pixels which are correctly identified as not containing neuronal activity.

Accuracy: Accuracy measures the number of pixels which are correctly assigned its ground truth value among all the pixels in the data.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (6.3)$$

F1 Score: Since we have a very high number of negative samples than positive samples, we implement F1 score as the tool for evaluating the accuracy of our model. F1 is the harmonic average between precision and true positive rate and is given by:

$$F1 = 2 * \frac{(precision * TPR)}{(precision + TPR)} \quad (6.4)$$

where $\text{precision} = \frac{TP}{TP+FP}$

6.1 Spectral Analysis

We analyze the intraoperative thermal imaging raw data by superimposing it with a Gaussian target activation pattern signal. Hence, the superimposition causes the data to become semi-synthetic. Since the data is spatial-temporal in behavior with stochastic properties, there is much cross-talk that overlaps in the time domain. Therefore, it becomes imperative to analyze the signals in frequency domain. The idea is to analyze by what amount our activation pattern and its frequencies are compressed after smoothing with Markov random fields. Fast Fourier transform [CT65] is a widely used algorithm which converts the time signal into the frequency domain using the following equation:

$$X(k) = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \quad (6.5)$$

where $N = 1024$ time points in our case, $k = 0, 1, \dots, N - 1$, x_n is the original time series signal while $X(k)$ is the Fourier transformed frequency domain signal.

6.1.1 Analysis of Gaussian activity pattern

The Gaussian activation function is generated with specific mean and variance which results in four circular areas containing synthetic neuronal activity. It means that the rest of the pixels are unharmed with the addition of Gaussian activation pattern and should retain its resting state behavior. This pattern now acts as a ground-truth dataset. The raw data, the Gaussian activation pattern, and the superimposed signal data are Fourier transformed separately so that the signal spreads in frequency domain thereby making it easier to analyze the energy contribution of its spectral components. The values of the Fourier transform, $|X(k)|$ instantly tells us how much

energy, the signal x_n has at a particular spectral component. The energy contribution, $E(p)$ by some components p is given as:

$$E(p) = \sum_{n=1}^p \frac{|X(n)|}{|X(0)| + |X(1)| + \dots + |X(N/2)|} \quad (6.6)$$

It is to be noted that this transformation also results in a Gaussian function but in the frequency domain. Figure 6.2 shows the positive 512 spectral components contribution to the energies after Gaussian activation pattern was Fourier transformed. We try to take the first $k = 128$ spectral components of our Gaussian target activation pattern for further analysis with our intraoperative thermal imaging data since they contribute 80% to the energy. However, it is better to not include all coefficients from 1 to 128 since some of the initial components contain low frequency drifts. Therefore an arbitrary k_1 till k_2 components are chosen for further analysis. It is intuitive that $k_1, k_2 < k$.

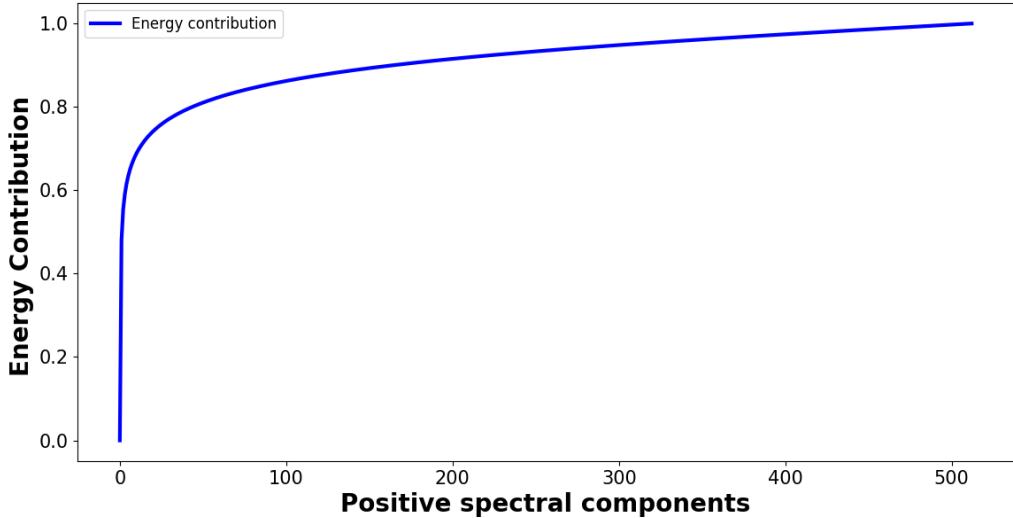


Figure 6.2: Energy contribution vs Number of spectral components

6.1.2 Analysis of raw vs. superimposed data

We first intend to use the raw intraoperative data (without superimposition of activation pattern) and perform spatial regularization using our proposed MRF model on spline coefficients in a single non-parametric component regression framework. We then perform Fast Fourier transform on the raw as well as spatially regularized raw data to record energies at the target circular regions with the relevant components. The energies are summarized in Table 6.1. E_{tot} represents the total energy at target pixels for all $k = 512$ spectral components whereas E_{rel} represents the total energy at target pixels for the relevant $[k_1, k_2]$ spectral components with $k_1 = 50$ and $k_2 = 100$.

Now, we superimpose our raw data with Gaussian Activation pattern and then perform spatial regularization using our proposed MRF model on spline coefficients in a single non-parametric component regression framework. We then perform fast Fourier transform on the superimposed as well as spatially regularized superimposed data to record energies at the target circular regions with the relevant components. The energies are summarized in Table 6.2. The values of k_1 and k_2 remains the same.

If we analyze the E_{tot} and E_{rel} of raw and superimposed data, we see a slight increase in energies for the superimposed data. This is intuitive from the fact that the superimposed data contains the energy from raw data as well as energies from the Gaussian activation pattern. This validates the augmentation of Gaussian activation pattern. Now, analyzing the E_{tot} and E_{rel} of

Weight γ	Raw Data		Smoothed Raw Data	
	E_{tot}	E_{rel}	E_{tot}	E_{rel}
0	20934.118	2694.555	72847.471	9088.964
0.5	20934.118	2694.555	68259.297	8491.747
1	20934.118	2694.555	71216.19	8895.446
1.5	20934.118	2694.555	72447.587	9057.304
2	20934.118	2694.555	74072.141	9316.75

Table 6.1: The table shows total energy at target sites for all 512 components as well as relevant [50, 100] components. The values are for both raw and smoothed raw data. γ are the pairwise potential weight. The results are for $p = 431$ control points and $k = 12$ Gaussian components for discretization.

Weight γ	Superimposed Data		Smoothed Superimposed Data	
	E_{tot}	E_{rel}	E_{tot}	E_{rel}
0	20935.828	2694.844	72697.028	9063.58
0.5	20935.828	2694.844	68125.034	8233.761
1	20935.828	2694.844	70766.546	8811.024
1.5	20935.828	2694.844	69587.281	8662.67
2	20935.828	2694.844	72841.526	9071.214

Table 6.2: The table shows total energy at target sites for all 512 components as well as relevant [50, 100] components. The values are for both superimposed data and spatially smoothed superimposed data. γ are the pairwise potential weight. The results are for $p = 431$ control points and $k = 12$ Gaussian components for discretization.

smoothed raw and superimposed data, we observe an opposite pattern. The energies of smoothed superimposed data has decreased as compared to the smoothed raw data. This reveals that our spatial regularization framework compressed some energies from the Gaussian activation pattern when it was superimposed with the raw data. However, the extent of compression is less which conveys that the spatial regularization of spline coefficients using Markov random fields still gives a decent fit to our raw data.

6.1.3 Sum of Square Error (SSE) analysis

Weight γ	Non-smoothed	3D Spline Smoothing	Proposed Model
0	0.108	1.91	2.91
0.5	0.108	1.91	56.41
1	0.108	1.91	170.04
1.5	0.108	1.91	197.34
2	0.108	1.91	267.43

Table 6.3: The table showing Sum of squares Error (SSE) values. The second column calculates the SSE between the raw and the superimposed data. The third column calculates the SSE between the raw and the superimposed data after 3D spline smoothing. The fourth column calculates the SSE between the raw and the superimposed data after spatial regularization based on our proposed model.

We compare the SSE values obtained from an existing 3D-Spline smoothing method with our proposed model. The results are summarized in Table 6.3. Typically, the values should approach zero if the estimates match with the original signal, otherwise, it can be inferred that the model absorbed some energy of the activation pattern. The results shows that as we increase the value of pairwise potential weight γ , the SSE values increases for our proposed model. The reasoning is that higher γ leads to over smoothing of spline coefficients which results in a rougher estimate of the signal.

6.2 Number of Gaussian components for Discretization

Determining the number of Gaussian components k is crucial for improving the accuracy of our model. The number of components are not known a priori, and there are many common methods in modern research to estimate it. Information criterion such as the Akaike information criterion (AIC) [Aka11] has been used to calculate the optimal k . A model which minimizes the AIC values provides a good fit. Figure 6.3 shows the graph between the number of Gaussian components and the corresponding AIC values after GMM applied on the spline coefficients of our intra-operative thermal imaging data in a single non-parametric component framework. The results are for $p = 431$ control points. From the figure, it is visualized that even if we increase the number of components beyond $k = 100$, the AIC values do not converge to a minimum. This is as per the expectation because theoretically, it should converge to a minimum when the number of components k is equal to the number of control points p . In that case, the discretization error is zero and AIC should reach its minimum value. However, selecting $k = p$ leads to unexpectedly large computations. Hence our goal is to choose a value of k that is not approaching the number of control points but still has a low AIC. In Figure 6.3, we observe that the point of diminishing returns is at $k = 12$ since the marginal loss in AIC drops at this point. This means that the first 12 components explain most of the variance in the data which also aligns with the elbow criterion [Tho53] for selection of the optimal k components. In the performance evaluation, we, therefore, intend to use k as 12 Gaussian components.

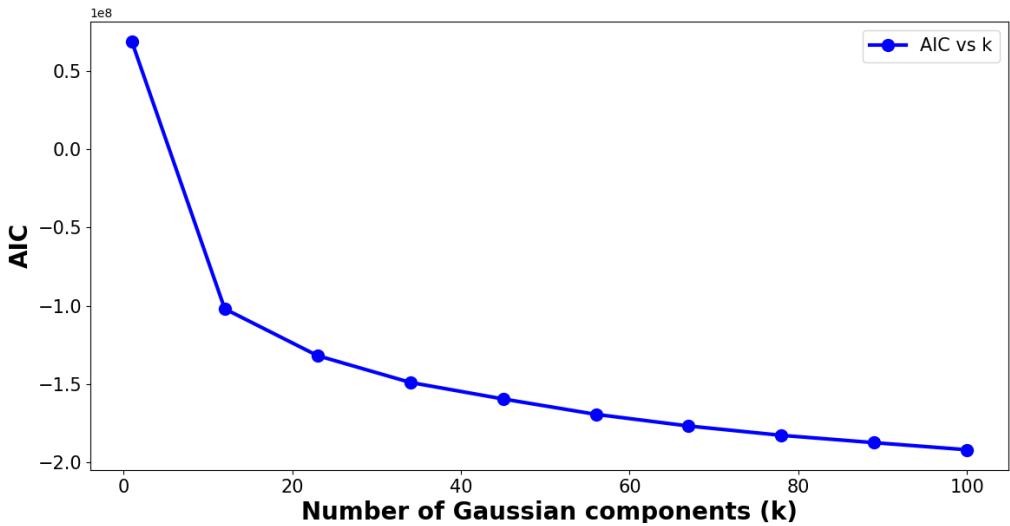


Figure 6.3: Figure shows the plot between the number of components and their respective AIC scores

6.3 Accuracy Analysis

We have calculated Z-statistics to test the accuracy of our novel semiparametric regression framework. The distribution of the test statistic under the null hypothesis is approximated by a normal distribution. For each significance level, the Z-test has a single critical value, Z_{crit} (in our case 5.2) which makes it more convenient than the t-statistics [RWC03] which has separate critical values for each sample size. Since the number of time points in our data is more than 30 and the standard deviation has already been calculated, a Z-test is more appropriate than the t-test. Throughout our evaluation, we assume Z_{crit} greater than or equal to 5.2 to be a positive response by the pixel for detecting neuronal activity while values less than 5.2 is assumed to be the negative response for detecting neuronal activity.

6.3.1 Univariate spline regression model

The Table 6.4 shows the performance of the existing univariate spline regression model. The best accuracy in terms of F1 score is achieved at $p = 431$ control points. There is an increase in overall accuracy of detecting neuronal activity as we increase the number of control points. However, selecting a very high number of control points leads to overfitting and decreases the accuracy.

Control points	TP	FP	TN	FN	TPR	FPR	F1
43	1537	1956	271452	32255	0.4400	0.0064	0.0862
100	2338	1155	301710	1997	0.6693	0.0038	0.6159
158	2577	916	302941	766	0.7377	0.0030	0.7618
215	2558	935	303332	375	0.7323	0.0030	0.7993
272	2684	809	303322	385	0.7683	0.0026	0.8205
329	2378	1115	303469	238	0.6807	0.0036	0.7794
387	2352	1141	303503	204	0.6733	0.0037	0.7784
431	2613	880	303464	243	0.7480	0.0029	0.8239
482	2519	974	303491	216	0.7211	0.0032	0.8100
539	2547	946	303515	192	0.7291	0.0031	0.8184
597	2452	1041	303554	153	0.7019	0.0034	0.8048
635	2327	1166	303592	115	0.6661	0.0038	0.7848

Table 6.4: Quantitative analysis of existing univariate spline regression model for different values of control points p

6.3.2 MRF on Lagrange Multiplier λ

Table 6.5 summarizes the performance of our proposed spatial regularization model based on Lagrange Multiplier λ . By comparing the accuracy values of this model with the existing univariate spline regression framework, it is observed that overall there is a slight decrease in the accuracy of detecting neuronal activity. This shows that finding optimum Z values based on AIC criterion as done in univariate spline regression framework is better than building an MRF model on λ and finding optimum λ for each pixel as a MAP estimate. Another possible reason for the decrease in accuracy is the uncertainty of the optimum range of λ values.

Control points	TP	FP	TN	FN	TPR	FPR	F1
43	2654	839	283490	20217	0.7598	0.0027	0.2021
100	2536	957	301479	2228	0.726	0.0031	0.6191
158	2532	961	302898	809	0.7248	0.0032	0.746
215	2555	938	303310	397	0.7314	0.0031	0.7952
272	2706	787	303300	407	0.7746	0.0026	0.8217
329	2418	1075	303455	252	0.6922	0.0035	0.7854
387	2340	1153	303505	202	0.6699	0.0038	0.7761
431	2572	921	303480	227	0.7363	0.003	0.8183
482	2417	1076	303525	182	0.6919	0.0035	0.7942
539	2474	1019	303533	174	0.7082	0.0034	0.8063
597	2319	1174	303569	138	0.6639	0.0038	0.78
635	2185	1308	303598	109	0.6255	0.0043	0.7554

Table 6.5: Quantitative analysis of our proposed spatial regularization on Lagrange Multiplier λ . Pairwise potential weight γ is 1

6.3.3 MRF on spline coefficients

Table 6.6 represents the performance of modeling MRF on spline coefficients of a single non-parametric component without an MRF on Z values. Table 6.7 shows the performance of modeling MRF on spline coefficients of a single non-parametric component in a two non-parametric component framework without an MRF on Z values. The results shows that by adding an extra non-parametric component in the spline regression framework, the accuracy increases. This conveys that the model with two non-parametric component is better in compensating high frequency effects and its interference with the intraoperative data is minimized. Comparing the performance of univariate spline regression framework in Table 6.4 and performance of our MRF model in Table 6.7, it is observed that the overall accuracy for $p = 431$ control points increased from 0.8239 to 0.8245. Figure 6.4 plots the results of the proposed models in comparison with the existing univariate spline regression framework at different value of control points.

6.3.4 Effect of pairwise potential weight vs Accuracy

We saw that incorporating two non-parametric components in our model and spatially regularizing one of them led to a better accuracy results as compared to existing univariate spline regression framework. To improve the overall accuracy further, we applied spatial regularization on Z values. We expected that it should further smooth the Z values by minimizing the overall energy. We applied MRF on Z values in three different frameworks and evaluated the effect of pairwise potential weight γ on the accuracy.

Table 6.8 shows the spatial regularization on Z values in an existing semiparametric regression framework. For $p = 431$ control points, the accuracy increased from 0.8239 in Table 6.4 to 0.8581 in Table 6.8 at $\gamma = 1.4$. It is observable that γ has a crucial effect in overall accuracy. It is inferred from the Table 6.8 that higher values of γ leads to over smoothness and therefore the circular areas of synthetic neuronal activity is also compensated decreasing the true positives.

Figure 6.5 showcases the effect of pairwise potential weight γ on the overall smoothness. A $\gamma = 0.1$ shows under smoothness while $\gamma = 10$ shows over smoothness. The results are for $p = 81$ control points. Table 6.9 and Table 6.10 shows the performance of spatial regularization on spline

Control points	TP	FP	TN	FN	TPR	FPR	F1
43	1609	1884	243078	60629	0.4606	0.0062	0.0543
100	2165	1328	301325	2382	0.6198	0.0044	0.5983
158	2578	915	302902	805	0.738	0.003	0.761
215	2563	930	303326	381	0.7338	0.003	0.7997
272	2692	801	303295	412	0.7707	0.0027	0.8186
329	2380	1113	303464	243	0.6814	0.0037	0.7792
387	2337	1156	303498	209	0.669	0.0038	0.7746
431	2611	882	303455	252	0.7475	0.0029	0.822
482	2518	975	303481	226	0.7208	0.00321	0.8085
539	2552	941	303504	203	0.7306	0.0031	0.8178
597	2463	1030	303549	158	0.7951	0.0034	0.8065
635	2328	1165	303590	117	0.6666	0.0038	0.7848

Table 6.6: Quantitative analysis of our proposed spatially regularized semiparametric regression model with single non-parametric component. GMM discretization and subsequent spatial regularization is done on the spline coefficients of the single non-parametric component. Pairwise potential weight γ is 1 and $k = 12$ components.

Control points	TP	FP	TN	FN	TPR	FPR	F1
43	1464	2029	262535	41172	0.4191	0.0066	0.0677
100	1894	1599	300434	3273	0.5422	0.0052	0.5058
158	2579	914	302855	852	0.7383	0.003	0.76
215	2569	924	303323	384	0.7355	0.003	0.7999
272	2693	800	303291	416	0.7709	0.0026	0.8187
329	2397	1096	303458	249	0.6862	0.0036	0.7819
387	2346	1147	303495	212	0.6716	0.0037	0.7763
431	2624	869	303456	251	0.7503	0.0028	0.8245
482	2522	971	303472	235	0.722	0.0032	0.8082
539	2560	933	303494	213	0.7328	0.003	0.8179
597	2468	1025	303543	164	0.7066	0.0034	0.8065
635	2334	1159	303584	123	0.6681	0.0038	0.785

Table 6.7: Quantitative analysis of our proposed spatially regularized semiparametric regression model with two non-parametric components. GMM discretization and subsequent spatial regularization is done on spline coefficients of only single non-parametric component. Pairwise potential weight γ is 1 and $k = 12$ components.

coefficients and Z values. The integration of MRF on spline coefficients and Z values results in

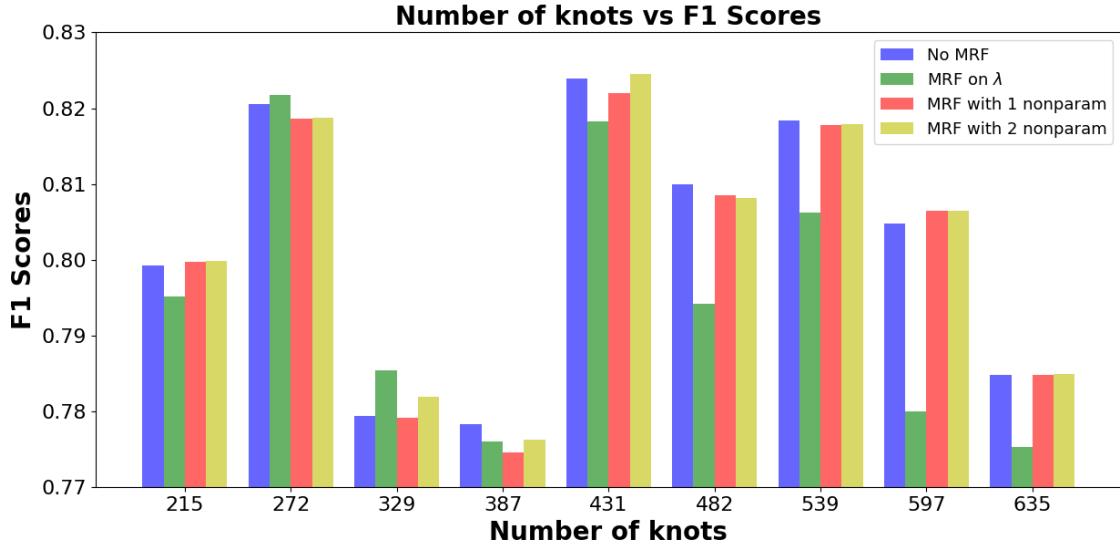


Figure 6.4: Analysis of the accuracy on four different models.

γ	TP	FP	TN	FN	TPR	FPR	F1
0	2613	880	303464	243	0.748	0.0029	0.8239
0.2	2641	852	303567	140	0.756	0.0028	0.8418
0.4	2667	826	303625	82	0.7635	0.0027	0.8545
0.6	2644	849	303647	60	0.7569	0.0027	0.8533
0.8	2637	856	303682	25	0.7549	0.0028	0.8568
1	2632	861	303688	19	0.7535	0.0028	0.8567
1.2	2629	864	303694	13	0.7526	0.0028	0.857
1.4	2625	868	303707	0	0.7515	0.0028	0.8581
1.6	2623	870	303707	0	0.7509	0.0028	0.8578
1.8	2623	870	303707	0	0.7509	0.0028	0.8577
2	2619	874	303707	0	0.7497	0.0028	0.857

Table 6.8: Quantitative analysis of spatial regularization of Z values in an existing univariate spline regression framework. The results are for $p = 431$ control points.

better accuracy as compared to the results achieved with spatial regularization on only Z values. This is true especially in the case when two non-parametric components are added in the spline regression framework and the MRF is applied on the spline coefficients of one of them. It is observed that for $p = 431$ control points, we achieved an improvement in accuracy from 0.8239 in Table 6.4 to 0.8589 in Table 6.10 by extending the univariate spline regression framework with MRF on spline coefficients and MRF on final Z values. Figure 6.6 shows the accuracy comparison of three different models based on pairwise potential weight γ with $p = 431$ control points. Figure 6.7 shows the final Z value map for the existing univariate spline regression at $p = 431$ controls points. Observe few noisy pixels in the background region. Figure 6.8 shows the final Z value map for our proposed MRF model on spline coefficients as well as Z values. There are no noisy pixels in the background and some smoothing achieved in the circular regions of neuronal activity.

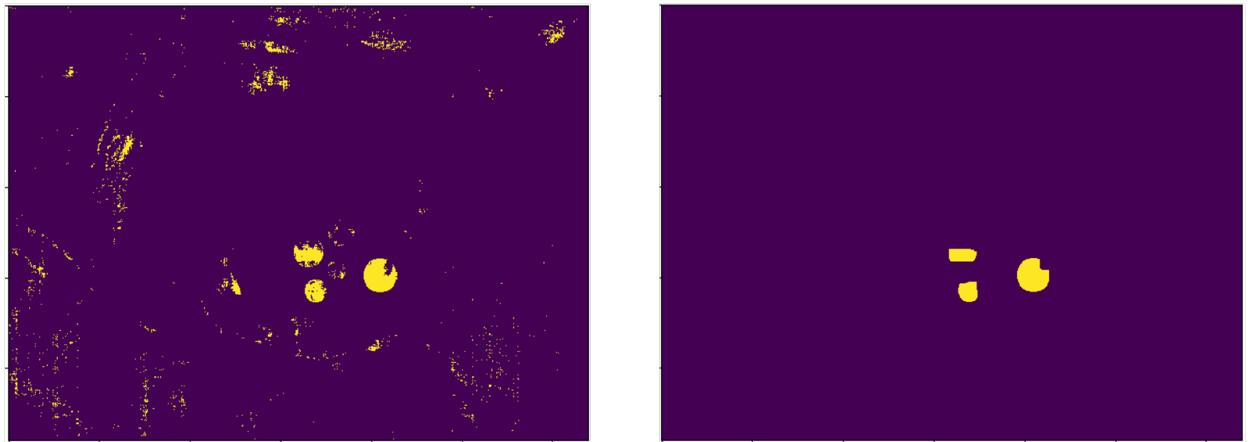


Figure 6.5: The left figure shows the MRF on Z values at $\gamma = 0.1$ with $p = 81$ control points. The right figure shows the result with $\gamma = 10$.

γ	TP	FP	TN	FN	TPR	FPR	F1
0	2611	882	303455	252	0.7475	0.0029	0.822
0.2	2637	856	303565	142	0.7549	0.0028	0.8409
0.4	2673	820	303624	83	0.7652	0.0027	0.8555
0.6	2652	841	303646	61	0.7592	0.0028	0.8547
0.8	2633	860	303682	25	0.7538	0.0028	0.8561
1	2623	870	303688	19	0.7509	0.0029	0.855
1.2	2625	868	303696	11	0.7515	0.0028	0.8566
1.4	2620	873	303707	0	0.75	0.0029	0.8571
1.6	2615	878	303707	0	0.7486	0.0028	0.8563
1.8	2613	880	303707	0	0.7481	0.0029	0.8559
2	2613	880	303707	0	0.7481	0.0029	0.8559

Table 6.9: Quantitative analysis of spatial regularization on Z values in a univariate spline regression framework with discretization and spatial regularization of spline coefficients of one non-parametric component already incorporated. The results are for $p = 431$ control points, $k = 12$ Gaussian components and pairwise potential weight for MRF on one non-parametric component is 1.

γ	TP	FP	TN	FN	TPR	FPR	F1
0	2624	869	303456	251	0.7503	0.0028	0.8245
0.2	2659	834	303565	142	0.7612	0.00275	0.8449
0.4	2674	819	303624	83	0.7655	0.0027	0.8557
0.6	2663	830	303651	56	0.7624	0.0027	0.8574
0.8	2653	840	303680	27	0.7595	0.0027	0.8595
1	2636	857	303688	19	0.7547	0.0028	0.8575
1.2	1617	876	303698	9	0.7492	0.0029	0.8554
1.4	2629	864	303707	0	0.7526	0.0028	0.8589
1.6	2629	864	303707	0	0.7526	0.0028	0.8589
1.8	2627	866	303707	0	0.752	0.0028	0.8585
2	2627	866	303707	0	0.752	0.0028	0.8585

Table 6.10: Quantitative analysis of spatial regularization of Z values in a univariate spline regression framework containing two non-parametric components with discretization and spatial regularization of spline coefficients of one non-parametric component already incorporated. The results are for $p = 431$ control points, $k = 12$ Gaussian components and pairwise potential weight for MRF on one non-parametric component is 1.

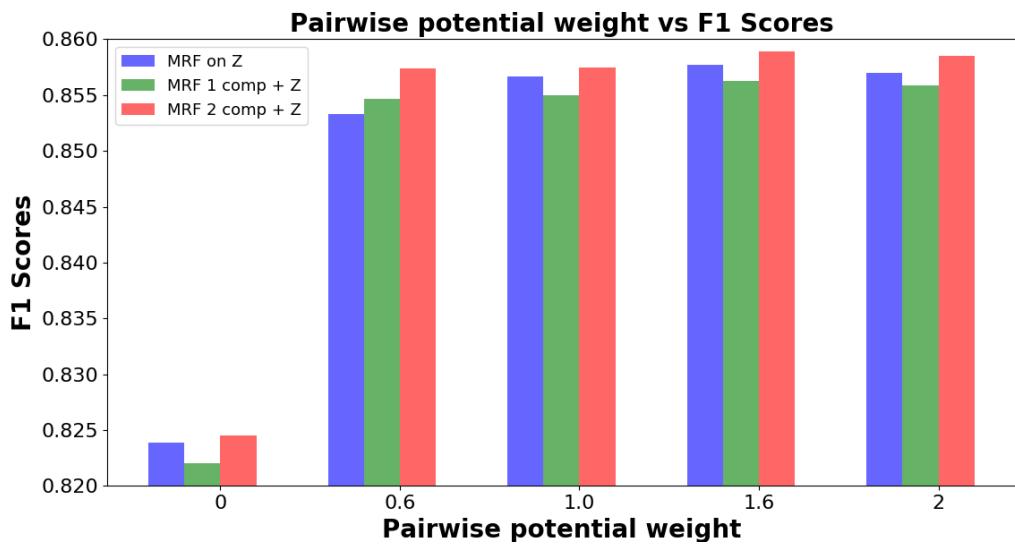


Figure 6.6: The effect of pairwise potential weight on overall accuracy for three models after incorporation of spatial regularization on Z values. Blue bar plots are for spatial regularization of only Z values, Green bar plots are for spatial regularization of both spline coefficients and Z values in a one non-parametric component framework, Red bar plots are for spatial regularization of both spline coefficients and Z values in a two non-parametric component framework.

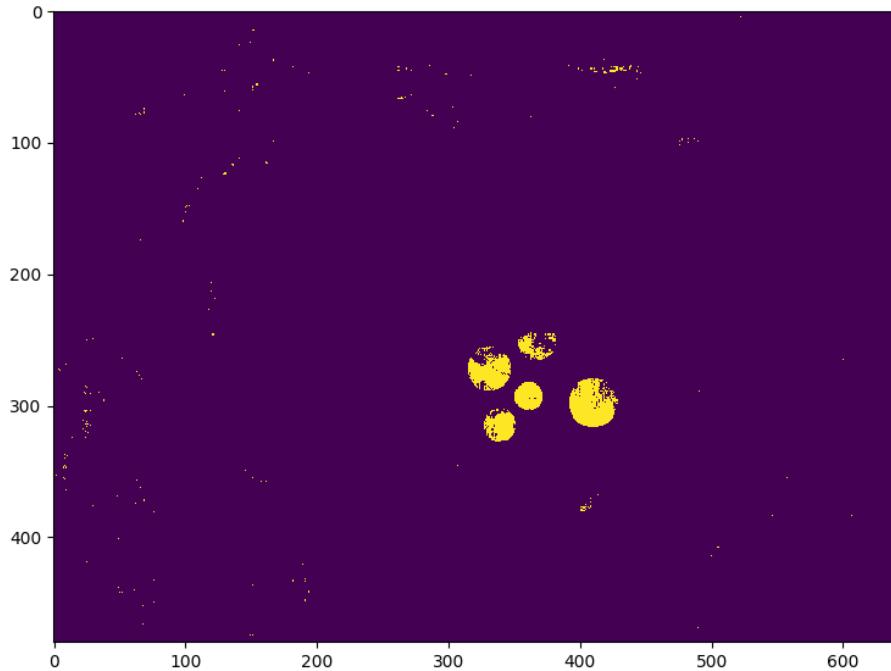


Figure 6.7: Spatial Z value map of univariate spline regression framework without any spatial regularization. The results are for $p = 431$ control points.

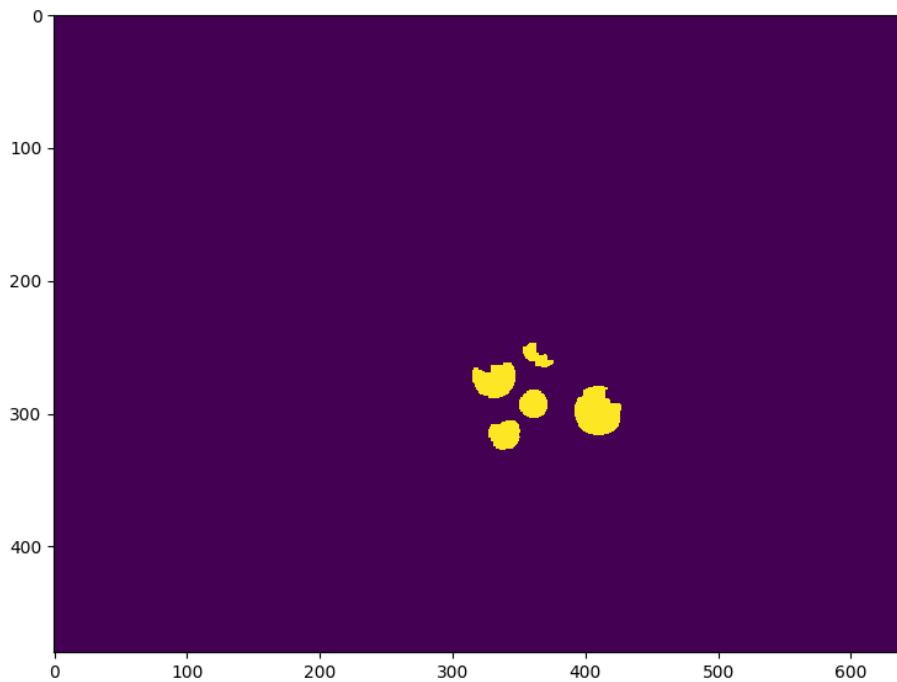


Figure 6.8: Spatial Z value map after spatial regularization on spline coefficients as well as on Z values. The results are for two non-parametric components framework. $p = 431$ control points, $\gamma = 1.4$.

6.4 TRWS runtime measurement

Figure (6.8 - 6.11) shows TRWS Energy and bound curve with respect to runtime and the number of steps. The inference stops when the bound and energy gap is less than 10^{-6} .

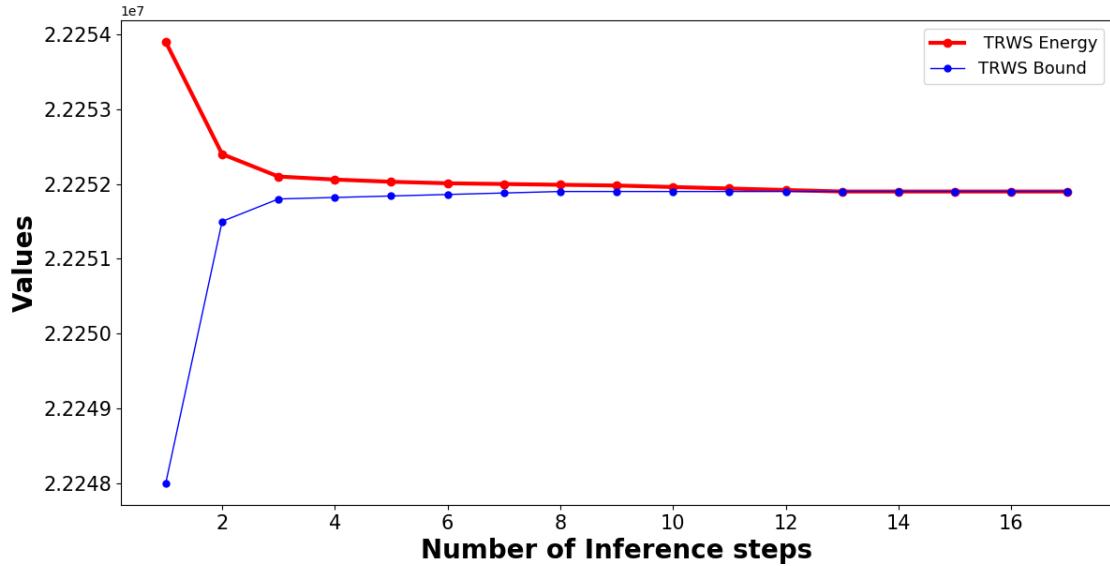


Figure 6.9: TRWS Energy and bound curve with respect to number of steps for MRF on spline coefficients. The result are for the semiparametric regression framework containing two non-parametric components but only one non-parametric component is spatially regularized. $p = 431$ control points and $k = 12$ Gaussian components and $\gamma = 1$.

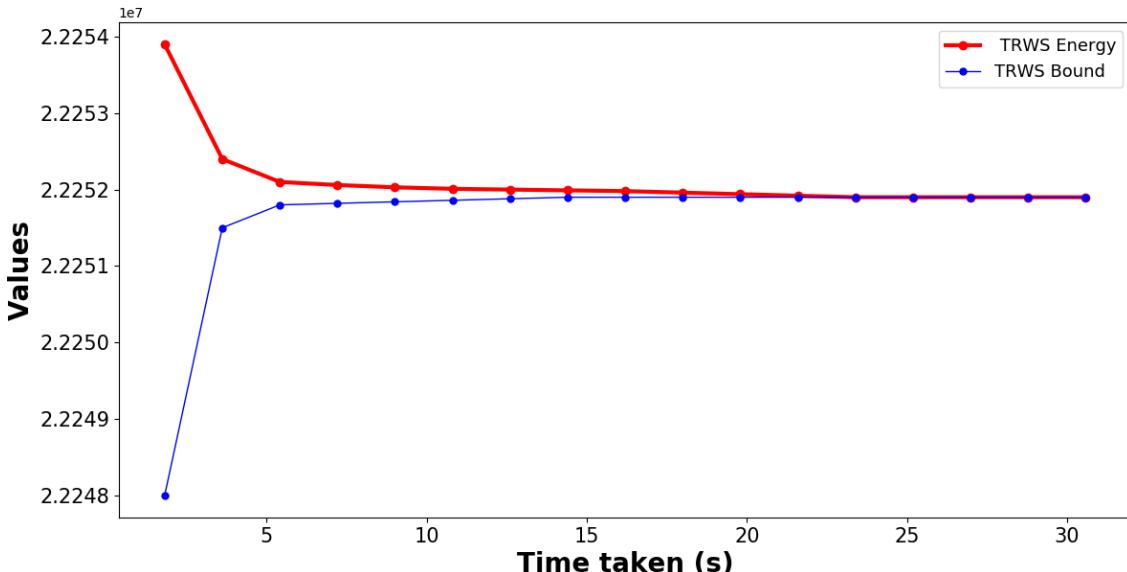


Figure 6.10: TRWS Energy and bound curve with respect to runtime for MRF on spline coefficients. The result are for the semiparametric regression framework containing two non-parametric components but only one non-parametric component is spatially regularized. $p = 431$ control points and $k = 12$ Gaussian components and $\gamma = 1$.

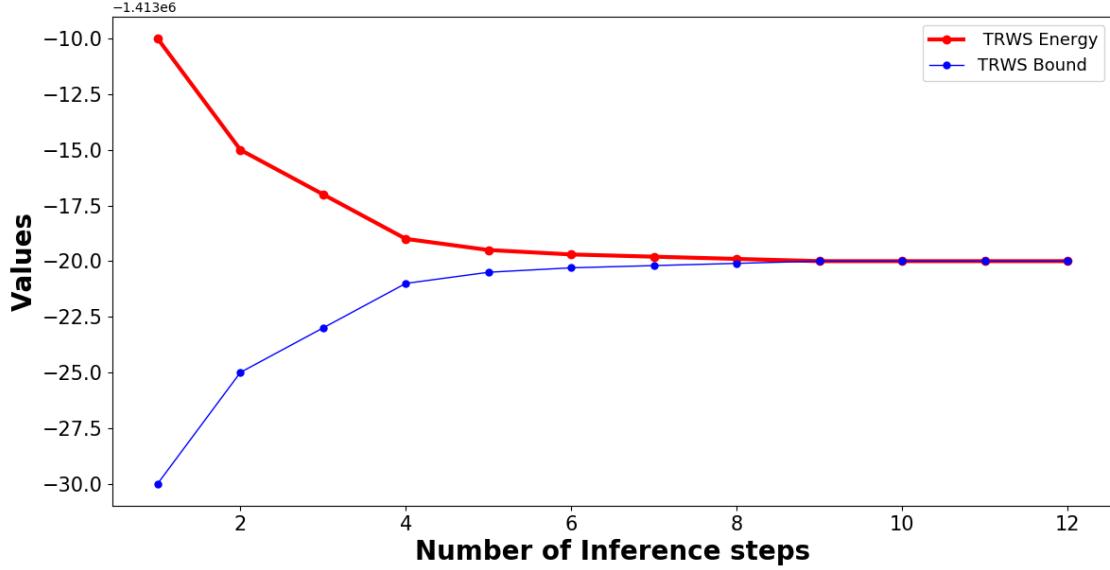


Figure 6.11: TRWS Energy and bound curve with respect to number of steps for MRF on Z values. $k = 2$ components, and $\gamma = 1.4$.

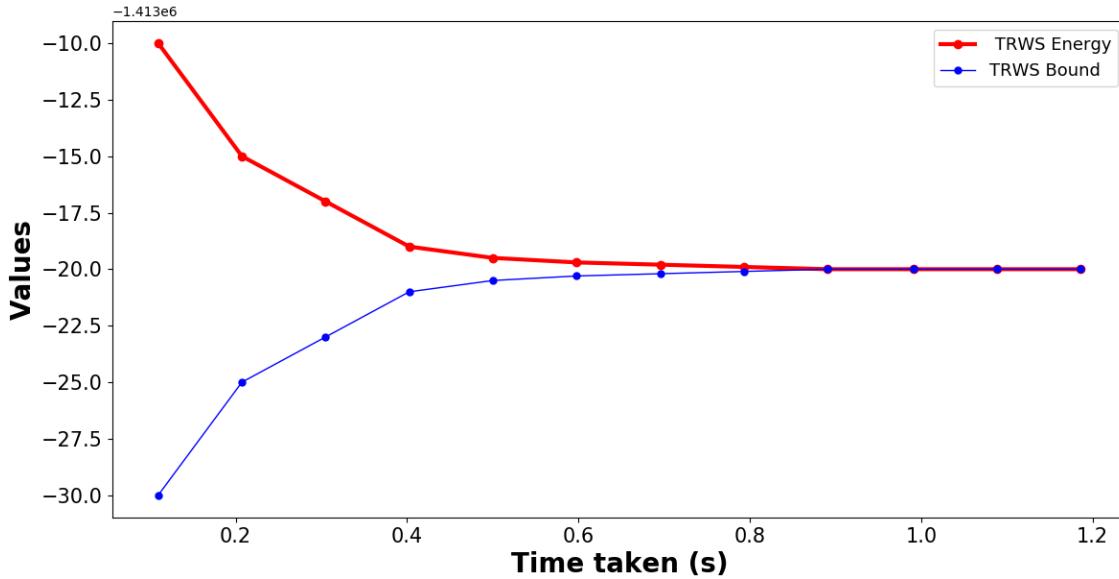


Figure 6.12: TRWS Energy and bound curve with respect to runtime for MRF on Z values. $k = 2$ components, and $\gamma = 1.4$.

6.5 General runtime measurements

The overall timing measurements of our proposed model has been summarized in Table 6.11. Clearly, the runtime of Gaussian Mixture Model and assignation of unary potentials are two of the most computationally expensive steps in our proposed model. Even though the spatial regularization using Markov Random fields provides an improved accuracy in detecting neuronal activity, the existing univariate spline regression framework comparably has lesser runtime. The univariate spline regression model has a runtime of around 30 seconds [NH18] as compared to the total runtime of 366.6 seconds for our proposed model. However, we feel these two bottlenecks

could be sidelined with the help of some optimization procedures discussed in future work.

Type	Time elapsed (s)
Gaussian Mixture Model	212.2
Assigning Unary potential	116.1
MRF on spline coefficients	32
MRF on Z values	1.2
Others	5.1
Total	366.6

Table 6.11: The runtime results are for $p = 431$ control points, $k = 12$ Gaussian components and $\gamma = 1.4$ for MRF on Z values

Chapter 7

Discussions and Conclusions

The visualization of neuronal activity is vital during neurosurgical tumor resections which conclusively helps in guiding medical personnel to determine the extent of tumor mass removal. The postoperative functional defects dramatically affect the patients healthy life due to which tissue resections require robust intraoperative schemes to unveil the regions of neuronal activity. However, various environmental factors such as physiological effects among others contribute to the background noise in the intraoperative thermal imaging data. This thesis extended the semiparametric regression framework by incorporating spatial regularization using Markov Random fields and improved the detection of neuronal activity by compensating the background noise. A synthetic activation pattern modeled using Gaussian distribution was superimposed on the resting state intraoperative thermal imaging data in alternating time periods. This superimposition helped in developing a model that includes the expected thermal behavior of neuronal activity as well as the characteristic thermal behavior of intraoperative thermal imaging. We demonstrated the overall applicability of our proposed framework to visualize statistically significant eloquent areas of the exposed human cortex.

The thesis showed that the spatial regularization on spline coefficients as well as on Z values achieved a substantial improvement in the accuracy of detecting neuronal activity. The existing semiparametric regression framework had an accuracy of 0.8239 at $p = 431$ control points. The expansion of the semiparametric regression framework with the Markov Random field components improved the accuracy to 0.8589. This improvement showed the robustness of the proposed model for the intraoperative thermal imaging of human cortex. We expect that the model could now be used for non-synthetic ways of evoking neuronal activity. In this work, we proposed three different Markov random field models which were built as an extension to the semiparametric regression framework. We also extended two of these models by spatially regularizing their final Z values to improve the accuracy even further. The results showed that the model which used two non-parametric components and spatially regularized one of them achieved higher accuracy compared to other models. The discretization of the spline coefficients was done using Gaussian Mixture Model (GMM); however, the runtime of the GMM was high which opens an area for improvement. The contribution of the pairwise potential weight was discussed which plays a vital role in determining the extent of smoothness.

Overall, the improvements achieved in detecting neuronal activity should help neurosurgeons determine the region for tumor resections during neurosurgical operations by decreasing false detections of areas with tumor tissues and improved removal of the regions of the tumor without affecting nearby healthy tissues. This should help in combating the postoperative neurological disorders in the patient. To the best of our knowledge, we are the first to apply spatial regularization of spline coefficients using Markov random fields in an intraoperative thermal brain imaging data.

Chapter 8

Future Work

8.1 Superpixel based Markov random fields

While the thesis exhibits a promising model to remove the background noise from the intraoperative thermal imaging data, a critical area that could be further investigated is a way to increase the True Positive Rate. As it is clear from the results that the background noise has been dramatically reduced, however, the detection of the areas containing neuronal activity could be improved further. This improvement is possible by incorporating superpixel based Markov random fields [SBS12] on the final Z values of our model. The MRF presented in this work defines pixels acting as nodes with factors assigned to them in a discrete graphical model framework. This construction gives the graphical model a uniform look. However, such a model only allows the local interaction between pixels and doesn't infer the relevant features in larger regions of the data. A superpixel Markov random fields define a cluster of pixels as a node based on their similar characteristics [SBS12]. On this node, unary and pairwise potentials could be defined, and therefore an irregular factor graph is formed. The interaction and message passing happen between these regions instead of each pixel. For superpixel based MRF models, there are fewer nodes in the graph which reduces the computation as well. We expect that this implementation should further increase the overall accuracy by increasing the True Positive Rate.

8.2 GPU implementation of Gaussian Mixture Model

As previously discussed, the runtime for discretization of spline coefficients using Gaussian Mixture Model is very high and the methods such as PCA and SIRP have their own disadvantages. As a future work, we propose to implement an efficient EM algorithm of the Gaussian Mixture Model using Graphical processing units (GPU) based on NVIDIA's Compute Unified Device Architecture (CUDA). The runtime of this implementation should dramatically reduce compared to the currently implemented GMM since the computations of the EM algorithm are divided into multiple GPU's. Thus a considerable speedup could be achieved without any approximations made in the estimation formulas. The significant advantage of a GPU is its parallel and computationally powerful processing. [TKD⁺16] shows a robust GPU implementation of Gaussian Mixture Model which is created on Tensorflow.

Chapter 9

Bibliography

- [AHK65] K. Abend, T. Harley, and L. Kanal. Classification of binary random patterns. *IEEE Transactions on Information Theory*, 11(4):538–544, October 1965. 9
- [Aka11] Hirotugu Akaike. *Akaike’s Information Criterion*, pages 25–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 23, 43
- [AKK⁺10] Björn Andres, Jörg H. Kappes, Ullrich Köthe, Christoph Schnörr, and Fred A. Hamprecht. An empirical comparison of inference algorithms for graphical models with higher order factors using opengm. In Michael Goesele, Stefan Roth, Arjan Kuijper, Bernt Schiele, and Konrad Schindler, editors, *Pattern Recognition*, pages 353–362, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 12
- [ATK12] B. Andres, Beier T., and J. H. Kappes. OpenGM: A C++ library for discrete graphical models. *ArXiv e-prints*, 2012. 15, 63
- [Bay63] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Phil. Trans. of the Royal Soc. of London*, 53:370–418, 1763. 10
- [BBHS07] A. O. Balan, M. J. Black, H. Haussecker, and L. Sigal. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, Oct 2007. 13
- [Bes74] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974. 8
- [Bes86a] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986. 12
- [Bes86b] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986. 8
- [Bet35] H. A. Bethe. Statistical theory of superlattices. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 150(871):552–575, 1935. 12
- [BGPC10] D. Batra, A. C. Gallagher, D. Parikh, and T. Chen. Beyond trees: Mrf inference via outer-planar decomposition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2496–2503, June 2010. 13
- [BK73] Coen Bron and Joep Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, September 1973. 8

- [BKL05] Andreas Brezger, Thomas Kneib, and Stefan Lang. Bayesx: Analyzing bayesian structural additive regression models. *Journal of Statistical Software, Articles*, 14(11):1–22, 2005. 10
- [BL07] Mark W Barnett and Philip M Larkman. The action potential. *Practical Neurology*, 7(3):192–197, 2007. 1
- [Boo89] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989. 6
- [Bre82] H. Brehm. Description of spherically invariant random processes by means of g-functions. In Dieter Jungnickel and Klaus Vedder, editors, *Combinatorial Theory*, pages 39–73, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg. 19, 20
- [Cat91] J. Catlett. *On changing continuous attributes into ordered discrete attributes*, pages 164–178. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991. 17
- [CC70] A J Collins and J A Cosh. Temperature and biochemical studies of joint inflammation. a preliminary investigation. *Annals of the Rheumatic Diseases*, 29(4):386–392, 1970. 3
- [CDE06] I. D. Currie, M. Durban, and P. H. C. Eilers. Generalized linear array models with applications to multidimensional smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(2):259–280, 2006. 7
- [CGB96] Michal R. Chmielewski and Jerzy W. Grzymala-Busse. Global discretization of continuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning*, 15(4):319 – 331, 1996. Rough Sets. 17
- [Che89] R Chellappa. Two-dimensional discrete gaussian markov random field models for image processing. *IETE Journal of Research*, 35(2):114–120, 1989. 10
- [Coo90] Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2):393 – 405, 1990. 12
- [CR12] J. Choi and R. A. Rutenbar. Hardware implementation of mrf map inference on an fpga platform. In *22nd International Conference on Field Programmable Logic and Applications (FPL)*, pages 209–216, Aug 2012. 13, 14, 15
- [CT65] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. 40
- [DB01] Carl De Boor. *A practical guide to splines; rev. ed.* Applied mathematical sciences. Springer, Berlin, 2001. 5, 16
- [DE87] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using gibbs random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(1):39–55, January 1987. 9
- [DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977. 18
- [doi10] *Fundamentals of Infrared Thermal Imaging*, chapter 1, pages 1–72. Wiley-Blackwell, 2010. 2
- [Duc77] Jean Duchon. *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, pages 85–100. Springer Berlin Heidelberg, Berlin, Heidelberg, 1977. 7

- [ECD06] Paul H. C. Eilers, Iain D. Currie, and Maria Durbán. Fast and compact smoothing on large multidimensional grids. *Comput. Stat. Data Anal.*, 50(1):61–76, January 2006. 6
- [EM96] Paul H. C. Eilers and Brian D. Marx. Flexible smoothing with b-splines and penalties. *Statist. Sci.*, 11(2):89–121, 05 1996. 5
- [FH06] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, Oct 2006. 11
- [FHW⁺94] K. J. Friston, A. P. Holmes, K. J. Worsley, J.-P. Poline, C. D. Frith, and R. S. J. Frackowiak. Statistical parametric maps in functional imaging: A general linear approach. *Human Brain Mapping*, 2(4):189–210, 1994. 5
- [FKL04a] Ludwig Fahrmeir, Thomas Kneib, and Stefan Lang. Penalized structured additive regression for space-time data: A bayesian perspective. *Statistica Sinica*, 14(3):731–761, 2004. 10
- [FKL04b] Ludwig Fahrmeir, Thomas Kneib, and Stefan Lang. Penalized structured additive regression for spacetime data: a bayesian perspective. *Statistica Sinica*, pages 731–761, 2004. 10
- [FL] Ludwig Fahrmeir and Stefan Lang. Semiparametric bayesian time-space analysis of unemployment duration. 10
- [FL01a] Ludwig Fahrmeir and Stefan Lang. Bayesian inference for generalized additive mixed models based on markov random field priors. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 50(2):201–220, 2001. 10
- [FL01b] Ludwig Fahrmeir and Stefan Lang. Bayesian semiparametric regression analysis of mult categorical time-space data. *Annals of the Institute of Statistical Mathematics*, 53(1):11–30, Mar 2001. 10
- [F.R01] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 65
- [GGRS⁺96] B M Gratt, S B Graff-Radford, V Shetty, W K Solberg, and E A Sickles. A 6-year clinical assessment of electronic facial thermography. *Dentomaxillofacial Radiology*, 25(5):247–255, 1996. PMID: 9161178. 4
- [GHK⁺03] Alexander M. Gorbach, John Heiss, Conrad Kufta, Susumo Sato, Paul Fedio, William A. Kammerer, Jeffrey Solomon, and Edward H. Oldfield. Intraoperative infrared functional imaging of human brain. Master’s thesis, 2003. 1, 4, 21
- [GHKO04] Alexander M. Gorbach, John D. Heiss, Leonid Kopylev, and Edward H. Oldfield. Intraoperative infrared imaging of brain tumors. *Journal of Neurosurgery*, 101(6):960–969, 2004. PMID: 15599965. 4
- [Gib] J. Willard (Josiah Willard) Gibbs. *Elementary principles in statistical mechanics developed with especial reference to the rational foundation of thermodynamics* /. New York :C. Scribner,, <https://www.biodiversitylibrary.org/bibliography/32624>. 9
- [GK87] Monique Guignard and Siwhan Kim. Lagrangean decomposition: A model yielding stronger lagrangean bounds. *Mathematical Programming*, 39(2):215–228, Jun 1987. 13
- [Glo11] Gary H. Glover. Overview of functional magnetic resonance imaging. *Neurosurgery Clinics of North America*, 22(2):133 – 139, 2011. Functional Imaging. 1, 2, 4

- [Gui03] Monique Guignard. Lagrangean relaxation. *Top*, 11(2):151–200, Dec 2003. 13
- [HC71] J. M. Hammersley and P. E. Clifford. Markov random fields on finite graphs and lattices. Unpublished manuscript, 1971. 9
- [HHS⁺14] Nico Hoffmann, Julia Hollmach, Christian Schnabel, Yordan Radev, Matthias Kirsch, Uwe Petersohn, Edmund Koch, and Gerald Steiner. Wavelet subspace analysis of intraoperative thermal imaging for motion filtering. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, pages 411–420, Cham, 2014. Springer International Publishing. 4, 6
- [HKP⁺17] Nico Hoffmann, Edmund Koch, Uwe Petersohn, Matthias Kirsch, and Gerald Steiner. Cerebral cortex classification by conditional random fields applied to intraoperative thermal imaging. *Current Directions in Biomedical Engineering*, 2(1):475–478, 2017. Functional Imaging. 1, 2, 3, 22
- [HT90] T. J. Hastie and R. J. Tibshirani. *Generalized additive models*. London: Chapman & Hall, 1990. 5
- [HW79] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *JSTOR: Applied Statistics*, 28(1):100–108, 1979. 17, 66
- [Ibm11] Ibm. *IBM ILOG CPLEX Optimization Studio CPLEX User’s Manual*, 2011. 13
- [JH13] Christian Schnabel Saskia Khler Stephan Sobottka Matthias Kirsch Gabriele Schackert Edmund Koch Gerald Steiner Julia Hollmach, Nico Hoffmann. Highly sensitive time-resolved thermography and multivariate image analysis of the cerebral cortex for intrasurgical diagnostics, 2013. 6
- [JH14] Nico Hoffmann Yordan Radev Stephan Sobottka Matthias Kirsch Gabriele Schackert Edmund Koch Gerald Steiner Julia Hollmach, Christian Schnabel. Intraoperative imaging of cortical perfusion by time-resolved thermography using cold bolus approach, 2014. 1, 2, 21
- [KAH⁺14] Jörg H. Kappes, Björn Andres, Fred A. Hamprecht, Christoph Schnörr, Sebastian Nowozin, Dhruv Batra, Sungwoong Kim, Bernhard X. Kausler, Thorben Kröger, Jan Lellmann, Nikos Komodakis, Bogdan Savchynskyy, and Carsten Rother. A comparative study of modern inference techniques for structured discrete energy minimization problems. *CoRR*, abs/1404.0533, 2014. 13
- [KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. 11
- [KFL06] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theor.*, 47(2):498–519, September 2006. 11
- [Kol06] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, Oct 2006. 11, 13, 14, 29
- [KS80] R. Kinderman and S.L. Snell. *Markov random fields and their applications*. American mathematical society, 1980. 9
- [KSH⁺13] Peter Kaatsch, Claudia Spix, Stefan Hentschel, Alexander Katalinic, Sabine Luttmann, Christa Stegmaier, Sandra Caspritz, Josef Cernaj, Anke Ernst, Juliane Folkerts, Jutta Hansmann, and Kristine Kranzhfer. Krebs in deutschland 2009/2010 - 9. ausgabe, 2013, 2013. 1

- [KW03] E. E. Kammann and M. P. Wand. Geoadditive models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 52(1):1–18, 2003. 7, 10
- [Lau96] S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, UK, 1996. 8
- [LD11] Dae-Jin Lee and Mara Durbn. P-spline anova-type interaction models for spatio-temporal smoothing. *Statistical Modelling*, 11(1):49–69, 2011. 7, 8
- [LF01] S. Lang and Ludwig Fahrmeir. Bayesian generalized additive mixed models. a simulation study, 2001. 10
- [Li09] Stan Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer Publishing Company, Incorporated, 3rd edition, 2009. 8, 9
- [Loe04] H. A. Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, Jan 2004. 11
- [LPR⁺16] David N. Louis, Arie Perry, Guido Reifenberger, Andreas von Deimling, Dominique Figarella-Branger, Webster K. Cavenee, Hiroko Ohgaki, Otmar D. Wiestler, Paul Kleihues, and David W. Ellison. The 2016 world health organization classification of tumors of the central nervous system: a summary. *Acta Neuropathologica*, 131(6):803–820, Jun 2016. 1
- [LZ99] X. Lin and D. Zhang. Inference in generalized additive mixed models by using smoothing splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(2):381–400, 1999. 10
- [Mah36] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936. 6
- [Mar95] Brian D. Marx. Multivariate statistical modelling based on generalized linear models (ludwig fahrmeir and gerhard tutz). *SIAM Review*, 37(4):633–634, 1995. 10
- [ME05] Brian D Marx and Paul H.C Eilers. Multidimensional penalized signal regression. *Technometrics*, 47(1):13–22, 2005. 6
- [MW14] David Miller and Simon Wood. Finite area smoothing with generalized distance splines. *Environmental & Ecological Statistics*, 21(4):715 – 731, 2014. 7
- [MWJ13] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. *CoRR*, abs/1301.6725, 2013. 12
- [Ng04] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML ’04, pages 78–, New York, NY, USA, 2004. ACM. 5
- [NH18] Gabriele Schackert Edmund Koch Stefan Gumhold Matthias Kirsch Nico Hoffmann, Uwe Petersohn. Fast mapping of the eloquent cortex by learning l2 penalties. *The Medical Image Computing and Computer Assisted Intervention Society, MICCAI*, 2018. 21, 22, 25, 52
- [NTC13] Claudia Nieuwenhuis, Eno Töpke, and Daniel Cremers. A survey and comparison of discrete and continuous multi-label optimization approaches for the potts model. *International Journal of Computer Vision*, 104(3):223–240, Sep 2013. 11
- [NW72] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972. 5

- [OYJ86] Finbarr O’sullivan, Brian S. Yandell, and William J. Raynor Jr. Automatic smoothing of regression functions in generalized linear models. *Journal of the American Statistical Association*, 81(393):96–103, 1986. 5
- [Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. 11, 12, 13
- [PF08] B. N. Pasley and R. D. Freeman. Neurovascular coupling. *Scholarpedia*, 3(3):5340, 2008. revision #91581. 4
- [PH06] K. Poeck and W. Hacke. *Neurologie*. Springer-Lehrbuch. Springer, 2006. 1
- [Pit79] Arthur O. Pittenger. Stochastic processes: A survey of the mathematical theory (john lamperti). *SIAM Review*, 21(3):421–422, 1979. 4
- [RA12] E F J Ring and K Ammer. Infrared thermal imaging in medicine. *Physiological Measurement*, 33(3):R33, 2012. 4
- [Ram02] Tim Ramsay. Spline smoothing over difficult regions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2):307–319, 2002. 7
- [Rey09] Douglas A. Reynolds. Gaussian mixture models. In Stan Z. Li and Anil K. Jain, editors, *Encyclopedia of Biometrics*, pages 659–663. Springer US, 2009. 17
- [RWC03] David Ruppert, M. P. Wand, and R. J. Carroll. *Semiparametric Regression*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2003. 5, 25, 44
- [SBS12] A. Schick, M. Buml, and R. Stiefelhagen. Improving foreground segmentations with probabilistic superpixel markov random fields. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 27–31, June 2012. 55
- [Shi94] Solomon Eyal Shimony. Finding maps for belief networks is np-hard. *Artificial Intelligence*, 68(2):399 – 410, 1994. 12
- [Sin93] Alistair Sinclair. *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Birkhauser Verlag, Basel, Switzerland, 1993. 11
- [SS06] Daniel Smith and Michael Smith. Estimation of binary markov random fields using markov chain monte carlo. *Journal of Computational and Graphical Statistics*, 15(1):207–227, 2006. 10, 11
- [STG⁺93] Igor A. Shevelev, Eugeny N. Tsicalov, Alexander M. Gorbach, Konstantin P. Budko, and George A. Sharaev. Thermoimaging of the brain. *Journal of Neuroscience Methods*, 46(1):49 – 57, 1993. 1, 4
- [Tho53] Robert L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, Dec 1953. 43
- [TKD⁺16] Dustin Tran, Alp Kucukelbir, Adji B. Dieng, Maja Rudolph, Dawen Liang, and David M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016. 55
- [TP09] H. Takagi and Denis Pallez. Paired comparison-based interactive differential evolution. In *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, pages 475–480, Dec 2009. 18

- [UAE93] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing. ii. efficiency design and applications. *Trans. Sig. Proc.*, 41(2):834–848, February 1993. 16
- [Ver64] A. M. Vershik. Some characteristic properties of gaussian stochastic processes. *Theory of Probability & Its Applications*, 9(2):353–356, 1964. 19
- [VPRK04] G. Varju, C. F. Pieper, J. B. Renner, and V. B. Kraus. Assessment of hand osteoarthritis: correlation between thermographic and radiographic methods. *Rheumatology*, 43(7):915–919, 2004. 3
- [WBH⁺08] Simon N. Wood, Mark V. Bravington, Sharon L. Hedley, Mathematical Sciences, and Bath Ba Ay U. K. Soap film smoothing. *Journal of the Royal Statistical Society B*, pages 931–955, 2008. 7
- [WF06] Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Trans. Inf. Theor.*, 47(2):736–744, September 2006. 12
- [WJ08] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, January 2008. 12, 13
- [WJW05] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, Nov 2005. 12, 13, 14
- [WM04] David C. Wilson and Bernard A. Mair. *Thin-Plate Spline Interpolation*, pages 311–340. Birkhäuser Boston, Boston, MA, 2004. 6
- [Woo03] Simon N. Wood. Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114, 2003. 6
- [Woo04] Simon N. Wood. Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, 99(467):673–686, 2004. 6
- [WR07] Haonan Wang and M. Giovanna Ranalli. Low-rank smoothing splines on complicated domains. *Biometrics*, 63(1):209–217, 2007. 7
- [WTH09] Daniela M. Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 7 2009. 5

Appendix A

OpenGM implementation of Graphical Model

The OpenGM framework [ATK12] is solely based on discrete factor graph models, and it can handle these models efficiently since functions that occur repeatedly need to be stored only once. Every factor is saved, and later the factor object is passed to the inference algorithm. As an initial step for constructing a graphical model using OpenGM, we need to define how many variables are there in the graph and how many labels, each variable can attain [ATK12]. Since our intraoperative thermal imaging data is of $m = 640 * 480$ dimension, there are 307200 variables to be defined in our graphical model setup. The optimum number of Gaussian components calculated after the discretization of spline coefficients acts as the number of labels that each variable can attain. The number of variables and the number of labels are defined in a discrete label space. Each of the variables is listed by integers ranging from $[0, 1, \dots, k - 1]$, where k is the number of labels.

To define a function such as pairwise potential θ_{pq} , one needs to indicate how many labels p and q could attain. Once a function has been added to the model, it can be connected to several factors and thus assigned to different sets of variables [ATK12]. This procedure is always the same, regardless of the number and type of classes used to encode functions [ATK12]. Algorithms for optimization and inference are classes in OpenGM. To run an algorithm, one instantiates an object of the class, providing a model and optional control parameters, and calls the member function, either without parameters or with one parameter indicating a visitor [ATK12]. Visitors are a powerful tool for monitoring and controlling algorithms by code injection [ATK12]. Once an algorithm has terminated, results such as optima and bounds can be obtained via member functions [ATK12].

An important aspect to be considered for our TRWS inference algorithm is the stopping condition. We stop the TRWS inference whichever among the following comes first during the runtime [ATK12]:

- (a) If an algorithm has run for 1 hour
- (b) If the gap between the energy and the lower bound is smaller than 10^{-6} .
- (c) If the numerical changes within the data are smaller than 10^{-7} .

These conditions provide better numerical results for TRWS as compared to a large number of iterations as the sole stopping condition.

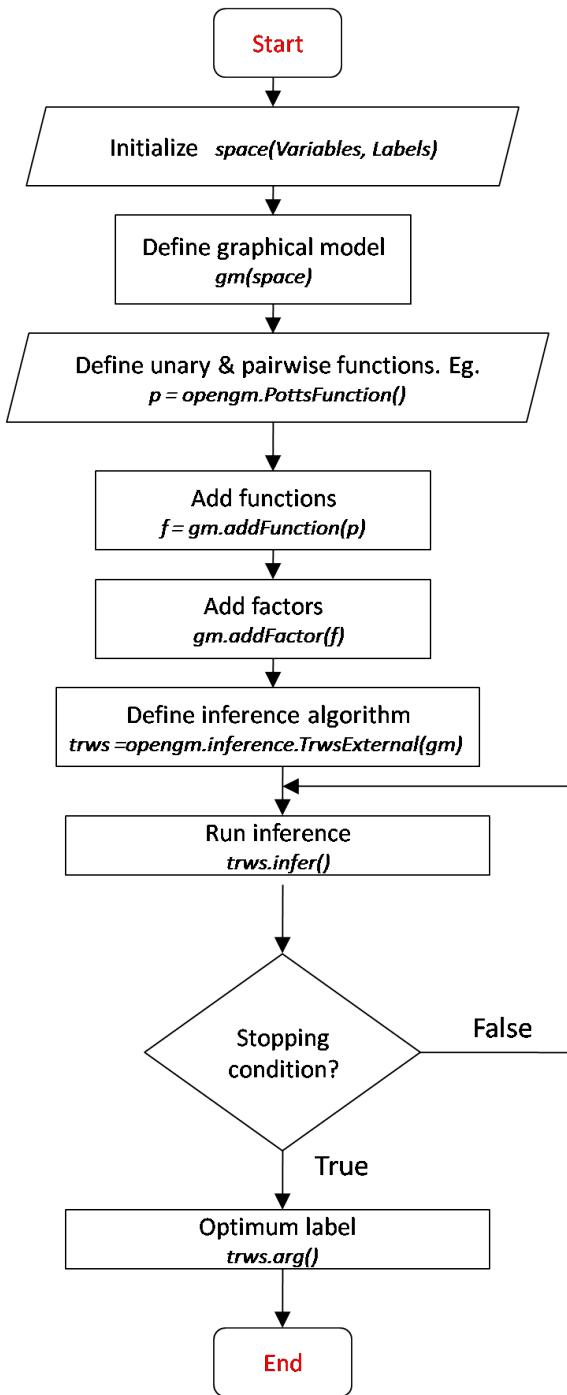


Figure A.1: Defining graphical model and running inference in OpenGM

Appendix B

Principal component analysis (PCA)

Principal Component Analysis (PCA) is a multivariate statistical procedure that dates back to works from [F.R01]. The procedure orthogonally transforms the p features of the data into a new p coordinates called principal components with the first principal component having the largest possible variance [F.R01]. Each of the following component has the highest possible variance under the constraint that it is orthogonal with the preceding components. The method helps in keeping only the first $q < p$ components while retaining most of the information, i.e., the variation in the data [F.R01]. The principal components reduce to solving an eigenvalue/eigenvector problem, and the new features are defined by the data at hand, not apriori, hence making PCA an efficient unsupervised data analysis technique. The spline coefficients $\beta \in \mathbb{R}^{p*m}$ contains m as the number of pixels, and p as the number of features (or spline control points). Then the covariance matrix $\mathbf{C} \in \mathbb{R}^{m*m}$ is given by [F.R01]:

$$\mathbf{C} = \beta^T \beta / (m - 1) \quad (\text{B.1})$$

It is to be noted that \mathbf{C} is a symmetric matrix and therefore is diagonalized to form the below equation:

$$\mathbf{C} = \mathbf{E} \mathbf{D} \mathbf{E}^T \quad (\text{B.2})$$

where \mathbf{E} is the eigenvector matrix and \mathbf{D} is the diagonal matrix with eigenvalues λ_i in the decreasing order on the diagonal matrix. These eigenvectors are the principal axes of our data. The projections of the data on the principal axes are called principal components [F.R01]. Hence these principal components act as the newly transformed features of our data. The singular value decomposition of β gives the following decomposition equation [F.R01]:

$$\beta = \mathbf{U} \mathbf{S} \mathbf{E}^T \quad (\text{B.3})$$

where $\mathbf{U} \in \mathbb{R}^{m*m}$ is the unitary matrix, \mathbf{S} is the diagonal matrix of singular values s_i . From above equations, \mathbf{C} matrix is written as [F.R01]:

$$\mathbf{C} = \mathbf{E} \mathbf{S} \mathbf{U}^T \mathbf{U} \mathbf{S} \mathbf{E}^T / (m - 1) = \mathbf{E} \frac{\mathbf{S}^2}{m - 1} \mathbf{E}^T \quad (\text{B.4})$$

meaning that the right singular vectors \mathbf{E} are the principal directions and the singular values are related to the eigenvalues of covariance matrix. Finally, the principal components are given by $\beta \mathbf{E} = \mathbf{U} \mathbf{S} \mathbf{E}^T \mathbf{E} = \mathbf{U} \mathbf{S}$ where columns of $\mathbf{U} \mathbf{S}$ are the principal components. To reduce the dimensionality of the data from p to $q < p$, we can select first q columns of \mathbf{U} and the $q * q$ upper left part of \mathbf{S} .

Appendix C

K-means clustering

The K-Means algorithm starts by randomly initializing the 'k' cluster centers which are usually chosen to be far apart from each other spatially, in the Euclidean distance, to be able to produce effective results [HW79]. Each cluster has a center, called the centroid, and a data point is clustered into a certain cluster based on how close the data points are to the centroid [HW79]. Then, each point in the data is taken and assigned to the nearest cluster. Finally, the centroid of the cluster is updated by taking the mean of the points that are newly assigned to it. This process is performed iteratively till convergence. In other words, cluster centroids should not move anymore [HW79]. Overall, the k-means algorithm aims at minimizing squared error function given by:

$$L = \sum_{j=1}^k \sum_{i=1}^n \| x_i^{(j)} - \mu_j \|^2 \quad (\text{C.1})$$

where $x_i^{(j)} \in \mathbb{R}$ is the datapoint i for cluster j , $\mu \in \mathbb{R}^k$, k are the number of clusters and n are the number of data points, $\| x_i^{(j)} - \mu_j \|^2$ is a chosen Euclidean distance measure between a data point $x_i^{(j)}$ and the cluster centroid μ_j and L is the squared error function. It is an indicator of the distance of the n data points from their respective cluster centroids.

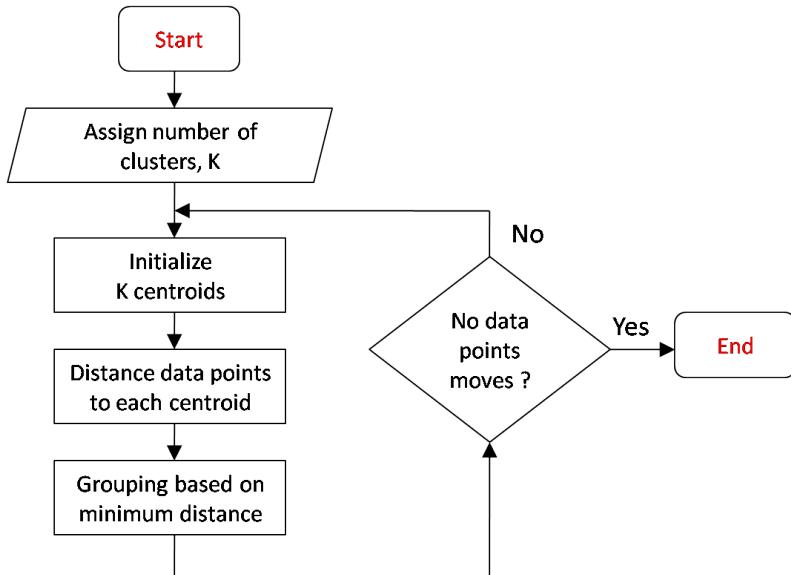


Figure C.1: K-Means Algorithm

Although the algorithm seems quite simple, finding the optimal solution to the problem is

NP-hard if either the number of clusters 'k' or the number of features is not fixed. It is noticeable from the Figure C.2 that group of pixels belonging to the same cluster has been assigned the same color. Each of these clusters possesses centroid values. Another point to be noted in the figure is that k-means discretization suffered from attaining local minimum and therefore results in a non-smooth discretized image.

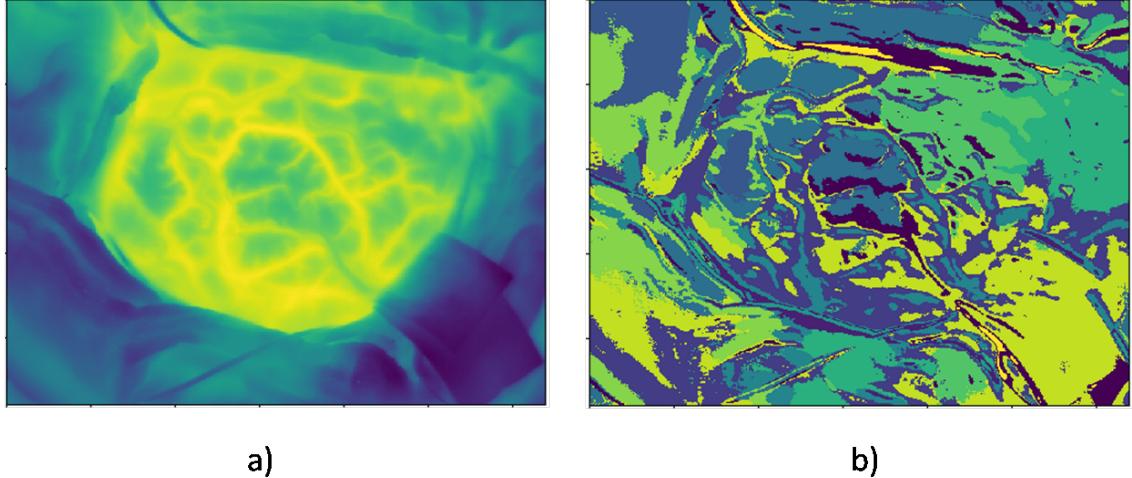


Figure C.2: a) shows the image of our raw data at 0^{th} time point. b) shows the discretization map of $m = 307200$ pixels using K-means algorithm with $k = 12$ clusters. Each color represents one of the 12 cluster labels assigned to each pixel. The result is obtained using $p = 431$ control points.