DARTMOUTH

DEPARTMENT OF MATHEMATICS,
DARTMOUTH COLLEGE

UNDERGRADUATE HONORS THESIS

Exploration into Reducing Uncertainty in Inverse Problems Using Markov Chain Monte Carlo Methods

Jiahui Zhang

supervised by
Dr. Anne GELB

June 3, 2020

Acknowledgements

First and foremost, I would like to express my sincere thanks to Dr. Anne Gelb, who has been one of the greatest mentors of my life. She taught me how to appreciate and love mathematics both in the outside of the classroom. Her indispensable support has not only made this thesis possible but also allowed me to pursue a future in mathematics. I may say without a doubt that her mentorship has and will continue to be a great source guidance in my life.

I would also like to show my gratitude to Dr. Theresa Scarnati for being such a role model while I had the amazing opportunity to intern at Air Force Research Laboratory. She taught me the power of mathematics in tackling real-world problems. I am very grateful for all the help she has given me in understanding the topic of this thesis and assisting me in the writing and editing of this thesis.

Lastly, I would like to thank my family for their unwavering support at every step of my life.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

Abstract

Image and function recovery from typically noisy and often under-sampled data is important in a wide variety of applications. Algorithms based on an inverse problem approach are often used in both the applied mathematics and statistics communities. In some cases, the resulting methods look quite similar. Specifically, deterministic techniques using regularization are standard in both communities, although they go by different names. For example, the celebrated LASSO technique developed in the statistics community [24] is very similar to the compressive sensing algorithms often used in sparse signal reconstruction [10]. We note that the type of regularization used is always problem dependent, and more detailed analysis of such techniques may be found in [24].

Sometimes the methodology differs substantially. For example, the Markov Chain Monte Carlo (MCMC) methods often employed by the statistics community takes a Bayesian approach, [17, 27]. A distinguishing feature of MCMC is that an entire *distribution* is recovered, rather than just a point estimate.

This thesis discusses how the deterministic and Bayesian approaches can be effectively combined to obtain a more accurate recovery process given noisy and under-sampled data. By combining these methodologies we are able to reduce the uncertainty of the solution distribution. The new method is also more efficient than the standard MCMC. Although the technique introduced here is inherently multi-dimensional, for ease of presentation this thesis considers only one-dimensional problems.

Chapter 1

Introduction

Image and function recovery from typically noisy and often under-sampled data is important in a wide variety of applications. Algorithms based on an inverse problem approach are often used in both the applied mathematics and statistics communities. In some cases, the resulting methods look quite similar. Specifically, deterministic techniques using regularization are standard in both communities, although they go by different names. For example, the celebrated LASSO technique developed in the statistics community [24] is very similar to the compressive sensing algorithms often used in sparse signal reconstruction [10]. We note that the type of regularization used is always problem dependent, and more detailed analysis of such techniques may be found in [24].

Sometimes the methodology differs substantially. For example, the Markov Chain Monte Carlo (MCMC) methods often employed by the statistics community takes a Bayesian approach, [17, 27]. A distinguishing feature of MCMC is that an entire *distribution* is recovered, rather than just a point estimate.

This thesis discusses how to effectively combine the deterministic and Bayesian approaches to obtain a more accurate recovery process given noisy and under-sampled data. By combining these methodologies we are able to reduce the uncertainty of the solution distribution. The new method is also more efficient than the standard MCMC.

To further motivate this investigation, we describe some characteristic features of deterministic and Bayesian approaches. For instance, when using a deterministic approach, one expects to recover a point estimate solution which can be viewed as the *maximum a-posteriori estimate (MAP)*. These techniques include regularization terms that help to penalize the solution to favor some prior information that is not data related. This is an active field of research that often falls under the general heading of *compressive sensing* [23], although there are many variations. Many efficient numerical algorithms have been developed such as ADMM, focal underdetermined system solvers (FOCUSS) and matching pursuit algorithms [13]. Theoretical and computational results demonstrate that it is possible to accurately recover functions and images even when the data are sparsely sampled. However, there are several major drawbacks inherent to this methodology. A major complaint among practitioners is that choosing the regularization parameters (e.g., how the data fidelity is weighed compared to the regularization terms) is highly application dependent. Further, it may not be robust even within the same application. This problem has been addressed in several ways, one of which is by using the *variance based joint sparsity* (VBJS) approach, which will be described shortly. A second drawback is limited by its very construction – a MAP estimate is a *point* estimate. While this might be acceptable in a variety applications, it is limiting in the sense that any uncertainty quantification about the solution is inaccessible [6]. For this reason it is desirable to consider the Bayesian framework, which enables the solution to be sampled from a *distribution*. The workhorse under this framework is the Markov Chain Monte Carlo (MCMC) method, which also has several variants, some of which will be described in this thesis. However, while being able to sample from a solution distribution is appealing, it is also often costly to construct, especially when a large search space must be explored. Moreover, in the Bayesian approach it is important to accurately depict the corresponding prior, likelihood and thus posterior estimates a-priori. This thesis leverages the VBJS approach in [18, 3] to more accurately describe the posterior. As a result, the overall uncertainty in the solution distribution is reduced (as depicted by tighter confidence intervals), and the MCMC becomes well mixed with fewer iterations.

Past works on leveraging the power of MCMC to solve inverse problems include [33, 7, 6]. In [6], The Metropolis-Hastings algorithm [17, 27] is employed to both recover the unknown distribution and also quantify uncertainty of the recovered distribution. As previously mentioned, in this thesis, the posterior probability density is modified by utilizing the concept of Variance Based Joint Sparsity (VBJS) [18, 3]. Specifically, the one-dimensional inverse problem introduced in [6] is investigated. The reasoning for adopting VBJS in constructing the posterior probability density is because the edge domain of the one-dimensional unknown is assumed to be sparse. Therefore, assuming sparsity may improve recovery of the unknown while improve the uncertainty quantification of the posterior probability density.

The structure of this thesis is as follows. Chapter 2 presents background information in probability, statistical inversion, Variance Based Joint Sparsity, and the Metropolis-Hastings algorithm. In Chapter 2.7, relevant papers are analyzed to build the foundation for the problem investigated in this thesis. This pioneering research gives motivation to the power and potential of the subject investigated. Next, in Chapter 3, the specific construction of the one-dimensional problem is provided. In Chapter 4 we present some results of our new method and discussion. Finally, Chapter 5, provides some concluding remarks and ideas for future work.

Chapter 2

Background

The topics covered here are background information required to understand the scope of the research of this paper.

2.1 Inverse Problems and Regularization

In the study of inverse problems, one seeks to recover an unknown variable from some sampled data, often riddled with noise. Further, the problem may be under-sampled, and therefore ill-posed [24, 32]. Suppose the goal is to reconstruct an unknown $x \in \mathbb{R}^n$. We have

$$y = Ax + e, \tag{2.1}$$

where $y \in \mathbb{R}^n$ is the sample data collected, $A : \mathbb{R}^n \to \mathbb{R}^n$ is a forward operator, and $e \in \mathbb{R}^n$ is some additive noise. The naive approach to recover the variable x would be to attempt to solve the equation

$$x = A^{-1}(y - e) = A^{-1}(y) - A^{-1}(e). \tag{2.2}$$

Observe that in (2.2), $A^{-1}e$ may dominate the solution recovery. In particular, the noise corresponds to eigenspaces of A of small eigenvalues, and thus singular values. By taking the inverse of the operator, the noise will then correspond to very large singular values in A^{-1} in (2.2). This makes naively taking the inverse of the operator A a very poor method for recovering x [21]. Further, since the model may be ill-posed, the operator A may not be well-conditioned and its corresponding inverse operator may not exist. Therefore, x may not uniquely exist within the solution space. In order to approximate x, the least-squares solution is used to obtain

$$x = \underset{x}{\operatorname{argmin}} \, ||Ax - y||_2^2 \tag{2.3}$$

It is well documented that (2.3) may yield poor results due the presence of noise [21]. In particular, without imposing constraints based on assumed knowledge, the solution x will be over-fitted to the noisy data. Therefore, a regularization term is often added to impose information assumed about the model. One then optimizes

$$x = \underset{x}{\operatorname{argmin}} \{||Ax - y||_2^2 + \lambda ||\mathcal{L}x||_p^q\} \tag{2.4}$$

for some $0 \leq p < \infty$. Define $q = p$ if $1 \leq p < \infty$ and $q = 1$ otherwise.

The first term in (2.4) is often referred to as the fidelity term and the second term is the regularization. The operator $\mathcal{L}$ varies based on the assumptions placed on the variable x. The coefficient on the regularization term $\lambda > 0$ is a tuning parameter for the model, which varies depending on the problem at hand. Generally, an increase in noise requires the increase of the tuning parameter, λ. This is because the collected data are not as reliable as the assumed knowledge. To successfully approximate the variable x, (2.4) must balance the need for fidelity to the data through the least-squares term, but also maintain the assumptions imposed in the regularization term to prevent over-fitting to the noise.

2.2 Variance Based Joint Sparsity

Another important concept in this investigation on inverse problems is *joint sparsity* [18, 3]. In particular we employ the variance based joint sparsity (VBJS) method introduced in [3, 18], which is described below.

2.2.1 ℓ_1 regularization for recovering piecewise smooth functions.

Often the regularization term in (2.4) is used to promote sparsity in the sparse domain of the corresponding solution x. Indeed, many algorithms under the general area of *compressive sensing* are designed specifically for this purpose [23]. The sparse domain, by definition, is the domain in which the corresponding solution is presumably sparse, and is formally defined as follows:

Definition 1. Consider a vector $u \in \mathbb{R}^N$. This vector u is s-sparse for some $1 \leq s \leq N$ if

$$||u||_0 = |\text{supp}(u)| \leq s. \tag{2.5}$$

There are potentially a variety of sparse domains one could choose for this purpose. For example, when recovering piecewise smooth functions, a good choice for the sparsity domain is the edge domain, since there are only a few edges (equivalently jump discontinuities). Gradient and wavelet domains (and their variants) are also commonly used, as are total variation (TV), and high order total variation (HOTV). Indeed, these domains are inherently linked by construction. More information on general recovery algorithms that use ℓ_1 regularization to exploit sparsity can be found in [18, 3].

If we wanted to use (2.4) to regularize the solution, ideally we would choose the ℓ_0 pseudo-norm, which is not a norm in the classical sense [12]. However, this is well known to be an NP

hard problem [12] so instead the ℓ_1 norm is used as a surrogate. It's important to understand why the ℓ_1 rather than the ℓ_2 norm is used. This is because the ℓ_1 norm penalizes the the small values relatively more while it penalizes large values relatively less compared to the ℓ_2 norm. This is particularly important because the sparse domain is assumed to be mostly near-zero other than regions with edge information [24]. However, computing (2.4) with the ℓ_2 norm is more computationally efficient, so these trade offs must be considered in applications.

Suppose that $\mathcal{L}$ is the sparsifying operator. In ℓ_1-regularization, the penalty on the norm of $\mathcal{L}x$ is proportional to the magnitude of $\mathcal{L}x$. On the other hand, ℓ_2-regularization squares each entry value in $\mathcal{L}x$. As a result, the ℓ_1 norm penalizes smaller values more than ℓ_2 norms, and penalizes larger values less compared to the ℓ_2 norm [9]. Therefore, since the prior belief is that the edge domain of x is sparse, ℓ_1 is typically the preferred norm for the regularization term.

Since in this thesis we are considering the vector x to be values of a piecewise smooth function, we will use the edge domain as the sparsity domain. To this end, this thesis adopts the polynomial annihilation operator, [5], as the sparsity operator $\mathcal{L}$, as it has been demonstrated to effectively recover the edges of piecewise smooth functions from grid point data (here the vector x).[1] Therefore, since x is considered to be a piecewise smooth function, the edge domain in (2.5) would be $u = \mathcal{L}x$. We stress that other sparsity operators might be better suited for different types of data sets.

2.2.2 Recovery from multiple measurement vectors

Suppose now that we have multiple measurements vectors in their sparse domain, $U = [u_1, u_2, u_3, \ldots, u_J]$ where $u_j \in \mathbb{R}^N$, $j = 1, \ldots, J$. Then we say that U is s-joint sparse if

$$||U||_{2,0} = \left| \bigcup_{j=1}^{J} \text{supp}(u_j) \right| \leq s \tag{2.6}$$

[1]The polynomial annihilation operator is analogous to Higher-Order Total Variation, [4], with certain technical differences.

where each u_j is s-sparse according to (2.5).

In addition to incorporating sparsity into the recovery process, when provided multiple measurement vectors (MMV), joint sparsity can be used to leverage the assumption that all vectors must be similar in the edge domain [18, 3]. The VBJS approach exploits this notion by introducing a weight for the regularization term in (2.4) that enforces this shared sparsity presumption.

The idea is to enforce sparsity in the recovered solution (2.1) in regions of the domain where the variance is small in the sparse domain. The example below illustrates how VBJS is implemented to solve inverse problems [3, 18].

Recall the inverse model (2.1) where x is the variable one seeks to recover. Let $x = (x_i)_{i=1}^N \in \mathbb{R}^N$ be an N-dimensional vector that represents a piecewise smooth function. Assume we have J measurements of the underlying unknown variable x. Assume that $U = [u_1, u_2, u_3, \ldots, u_J]$ are the measurement vectors in the edge domain such that $u_i = \mathcal{L}x_i$. Then it is assumed that the multiple measurement vectors have similar support defined as

$$\text{supp}(u_1) \approx \text{supp}(u_2) \approx \text{supp}(u_3) \approx \ldots \approx \text{supp}(u_J).$$

It follows then that a variance vector $v = \{v_i\}_{i=1}^N$ can be constructed as

$$v_i = \frac{1}{J}\sum_{j=1}^{J}(u_i^j)^2 - \Big(\frac{1}{J}\sum_{j=1}^{J}u_i^j\Big)^2 \quad i = 1, \ldots, N, \tag{2.7}$$

such that $\text{supp}(u) \approx \bigcup_{j=1}^J \text{supp}(u_j)$ From the variance vector $\vec{v}$, a *spatially variant* weight vector w for the regularization term is then constructed. One idea of weight construction is to simply define

$$w_i := \frac{1}{v_i + \epsilon}, \quad i = 1, \ldots, N,$$

where $\epsilon \ll v_i$ and is inserted to prevent division by zero. In this thesis, this simple method of obtaining the inverse described above of the variance is not used. Instead, we use a more sophisti-

cated method described in [18] to ensures normalization. The construction is as follows. Suppose again there are J measurements of the same unknown x. Further, suppose $\mathcal{L}$ is a sparse operator such that $U = [u_1, u_2, u_3, \ldots, u_J]$ is jointly sparse such that $u_i = \mathcal{L}x_i$. Then each sample in the sparse domain is normalized so that

$$\tilde{u}_{i,j} = \frac{|u_{i,j}|}{\max_i |u_{i,j}|},$$

where $j = 1, \ldots, J$. Then a weighting scalar C is defined as the average ℓ_1 norm across all measurements of the normalized sparsifying transform of the measurements,

$$C = \frac{1}{J} \sum_{j=1}^{J} \sum_{i=1}^{N} \tilde{u}_{i,j}.$$

The purpose of the weighting scale is to ensure that small edges are not deemed insignificant relative to large ones. Finally, the weights $w \in \mathbb{R}^N$ is define as

$$w_i = \begin{cases} C(1 - \frac{v_i}{\max_i v_i}) & \text{if } i \notin I \\ \frac{1}{C}(1 - \frac{v_i}{\max_i v_i}) & \text{if } i \in I \end{cases}$$

where v_i is defined in (2.7). The set I is defined as

$$\frac{1}{J} \sum_{j=1}^{J} \tilde{u}_{i,j} > \tau.$$

The constant τ is a threshold such at when the above inequality is satisfied by some index i, the corresponding y_i is assumed to be an edge. By observing Figure 4.2, we can clearly see difference in weights between regions that satisfy and do not satisfy the corresponding threshold τ.

Remark: *We note that the construction of $\mathcal{L}x_j$, $j = 1, \ldots, J$, requires each solution x_j, to be known. Obviously this is not the case since we are given the measurement data y_j, $j = 1, \ldots, N$ in* (2.1) *and indeed we are seeking the solution x. In [3, 18], an approximation $\tilde{x}_j$ to each x_j was first done using an unweighted (standard) form of* (2.4) *with $p = q = 1$, with λ chosen randomly. Then $\mathcal{L}\tilde{x}_j$ was calculated to approximate the edge domain for each measurement vector. It is possible to do this approximation of the edges more efficiently and without first reconstruction x, and this has since been done for the measurement vectors being noisy Fourier data [3], but for the purpose of this investigation, we use the approach suggested in [3, 18].*

With these calculations in hand, we are now ready to construct the VBJS method:

Algorithm 1 VBJS Weights Construction

Data: Set of measurement vectors in their jointly sparse domain, $U = [u_1, u_2, u_3, \ldots, u_J]$

Result: Modified regularization formula

Calculate weights $w = [w_1, w_2, w_3, \ldots, w_J]$

Choose a sparsifying operator (e.g., the polynomial annihilation operator in [5])

Set matrix $W = \text{diag}(w)$

Modify (2.4) to be $x = \text{argmin}_x\{||Ax - y||_2^2 + \lambda||W\mathcal{L}x||_1\}$

This modification to the ℓ_1 regularization allows the greater enforcement of sparsity in regions of the domain where it is believed to be truly sparse. Specifically, regions of high variance in the sparse domain correspond to non-zero entries. Therefore, as seen in Figure 4.2, those regions are penalized less to enforce fidelity to the sample data. On the other hand, small variance regions correspond to truly sparse regions, so the penalty is correspondingly higher there.

2.3 Probability Preliminaries

We now introduce some fundamental probability concepts that are needed for this investigation. There are great texts that thoroughly explain these concepts such as [8, 24, 30]. First, the concept of the probability space is formalized through a measure-theoretic foundation. From there, the characterizations of a random variable, probability distribution, and probability density are defined. Further, the concept of joint probability density and conditional probability density are defined, culminating this section with the Bayes formula [32]. Suppose Ω is an abstract space and Γ is a collection of subsets of Ω. The collection Γ is said to be a *σ-algebra* if the following are true:

1. $\Omega \in \Gamma$,
2. If $A \in \Gamma$ then $A^c = \Omega \setminus A \in \Gamma$, and
3. If $A_i \in \Gamma$ for $i \in \mathbb{N}$, then $\bigcup_{i=1}^{\infty} A_i \in \Gamma$.

Further, a mapping $\mu : \Gamma \to \mathbb{R}$ is a *probability measure* if the following are true:

1. $\mu(A) \geq 0$ for all $A \in \Gamma$,
2. $\mu(\varnothing) = 0$, and
3. If $A_i \in \Gamma$ are disjoint ($A_i \cap A_j = \varnothing$ for $i \neq j$) for all $i \in$, then $\mu\Big(\bigcup_{i=1}^{\infty} A_i\Big) = \sum_{i=1}^{\infty} \mu(A_i)$.

A measure $\mu : \Gamma \to \mathbb{R}$ is *finite* if $\mu(\Omega) < \infty$. Suppose we have a measure $\mathbb{P} : \Gamma \to [0, 1]$ where $\mathbb{P}(\Omega) = 1$, then the $(\Omega, \Gamma, \mathbb{P})$ defines a *probability space* (measurable space), where Ω is the *sample space*, Γ is the set of *events*, and $\mathbb{P}$ is the *probability measure*.

For the purposes of this thesis, assume that the sample space $\mathbb{R}$ is equipped with the *Borel σ-algebra*, which is the smallest σ-algebra that contains the topology (collection of open sets) of $\mathbb{R}$. Further, assume the Lebesgue measure restricted to the Borel σ-algebra. An in-depth treatment of the foundations of measure theory can be found in [16].

Now consider a function $f : \Omega \to \mathcal{T}$ where $\mathcal{T}$ is any topological space. Then the function f is said to be *μ-measurable* if and only if for any open set A of $\mathcal{T}$, its pre-image $f^{-1}(A)$ is a μ-measurable set in Ω.

The next measure-theoretic concept needed is the *Radon-Nikodym Theorem* and the *Radon-Nikodym derivative* [16]. Suppose there are two σ-finite measures μ and ν on a measurable space (Ω, Γ) such that ν is absolutely continuous with respect to μ (note $\mu(A) = 0$ implies $\nu(A) = 0$ for any $A \in \Gamma$). Then for some μ-measurable function $f : \Omega \to \mathbb{R}$, we have $\nu = \int_A f d\mu$ where the integral is for any measurable set $A \in \Gamma$. Note that any μ-measurable function f that satisfies the equation $\nu = \int_A f d\mu$ are equal almost everywhere with respect to measure μ. Thus the equivalence class of functions f is called the Radon-Nikodym derivative of ν with respect to μ and may denoted as $f = \frac{d\nu}{d\mu}$.

The necessary concepts in probability are now delineated. Consider the probability space $(\Omega, \Gamma, \mathbb{P})$. We assume events $A, B \in \Gamma$, with $\mathbb{P}(A) > 0$ and $\mathbb{P}(B) > 0$.

- Events $A, B \in \Gamma$ are said to be *independent* if $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$.

- The probability of event A *conditioned* on event B is

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}. \tag{2.8}$$

- The *Law of Total Probability* is given by

$$\mathbb{P}(A) = \mathbb{P}(A|B)\mathbb{P}(B) + \mathbb{P}(A|B^c)\mathbb{P}(B^c). \tag{2.9}$$

The next important concept is the idea of random variables. Consider again a sample space $(\Omega, \Gamma, \mathbb{P})$. A *random variable* defined on this sample space is a measurable function $\boldsymbol{X} : \Omega \to \mathbb{R}$. A random variable $\boldsymbol{X}$ may generate a measure $\mu_{\boldsymbol{X}}$ equipped with the Borel σ-algebra. This measure

is called the *probability distribution* of the random variable $\boldsymbol{X}$. For any event $A \in \Gamma$,

$$\mu_{\boldsymbol{X}}(A) = \mathbb{P}(\boldsymbol{X}^{-1}(A)) = \mathbb{P}(\boldsymbol{X} \in A).$$

A random variable $\boldsymbol{X}$ is absolutely continuous if it is absolutely continuous with respect to the Lebesgue measure [30]. Then by the definition given above of the Radon-Nikodym derivative, it is guaranteed that a measurable function $f_{\boldsymbol{X}} : \Omega \to \mathbb{R}$ exists such that for any $\mu_{\boldsymbol{X}}$-measurable set $A \in \Gamma$ we can express the probability distribution as

$$\mu_{\boldsymbol{X}}(A) = \int_A f_{\boldsymbol{X}}(x)dx.$$

Here, the function $f_{\boldsymbol{X}} : \Omega \to \mathbb{R}$ is the Radon-Nikodym derivative $\frac{d\mu_{\boldsymbol{X}}}{dx}$ and said to be the *probability density function*.

Finally, the concept of the probability density function leads to the definitions of the joint probability density and the conditional probability density. Suppose $\boldsymbol{X}_1 : \Omega \to \mathbb{R}$ and $\boldsymbol{X}_2 : \Omega \to \mathbb{R}$ are both random variables defined on the sample space $(\Omega, \Gamma, \mathbb{P})$. The *joint probability distribution* $\mu_{\boldsymbol{X}_1,\boldsymbol{X}_2}$ is defined as

$$\mu_{\boldsymbol{X}_1,\boldsymbol{X}_2}(A_1, A_2) = \mathbb{P}({\boldsymbol{X}_1}^{-1}(A_1) \cap {\boldsymbol{X}_2}^{-1}(A_2)), \quad A_1, A_2 \in \Gamma.$$

If both random variables are absolutely continuous with respect to the Lebesgue measure, then it follows that the *probability density function* denoted as $f_{\boldsymbol{X}_1,\boldsymbol{X}_2} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is defined as

$$\mu_{\boldsymbol{X}_1,\boldsymbol{X}_2}(A_1, A_2) = \int_{A_1 \times A_2} f_{\boldsymbol{X}_1,\boldsymbol{X}_2}(x_1, x_2)dx_1dx_2. \tag{2.10}$$

Further, if the random variables $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are independent then for any two outcomes $x_1, x_2 \in \Omega$,

$$\mu_{\boldsymbol{X}_1,\boldsymbol{X}_2}(x_1, x_2) = \mu_{\boldsymbol{X}_1}(x_1)\mu_{\boldsymbol{X}_2}(x_2).$$

This definition can be extended to their respective probability density functions to arrive at

$$f_{\boldsymbol{X}_1,\boldsymbol{X}_2}(x_1, x_2) = f_{\boldsymbol{X}_1}(x_1)f_{\boldsymbol{X}_2}(x_2).$$

Now the *marginal probability distribution* of $\boldsymbol{X}_1 \in \mathbb{R}^n$ is defined as

$$\mu_{\boldsymbol{X}_1}(A_1) = \int_{A_1\times\mathbb{R}^m} f_{\boldsymbol{X}_1,\boldsymbol{X}_2}(x_1, x_2)dx_1dx_2 = \int_{A_1} f_{\boldsymbol{X}_1}(x_1)dx_1, \tag{2.11}$$

and the *marginal probability density* is

$$f_{\boldsymbol{X}_1}(A_1) = \int_{\mathbb{R}^m} f_{\boldsymbol{X}_1,\boldsymbol{X}_2}(x_1, x_2)dx_2 \tag{2.12}$$

For any events $A_1, A_2 \in \Gamma$ with positive measures, the *conditional measure* $\mu_{\boldsymbol{X}_1|\boldsymbol{X}_2}$ is then given by

$$\mu_{\boldsymbol{X}_1|\boldsymbol{X}_2}(A_1|A_2) = \frac{\mu_{\boldsymbol{X}_1,\boldsymbol{X}_2}(A_1, A_2)}{\mu_{\boldsymbol{X}_2}(A_2)}.$$

Under certain conditions [8], the Law of Total Probability 2.9 may be expressed as

$$\mu_{\boldsymbol{X}_1|\boldsymbol{X}_2}(A_1|A_2) = \int_{A_2} \mu_{\boldsymbol{X}_1|\boldsymbol{X}_2}(A_1|x_2)d\mu(x_2), \tag{2.13}$$

and the probability density of $\boldsymbol{X}_1$ conditioned on $\boldsymbol{X}_2$ is

$$f_{\boldsymbol{X}_1|\boldsymbol{X}_2}(A_1|A_2) = \frac{1}{f_{\boldsymbol{X}_1|\boldsymbol{X}_2}(A_2)} \int_{A_1\times A_2} f_{\boldsymbol{X}_1,\boldsymbol{X}_2}(x_1, x_2)dx_1dx_2.$$

If the set A_2 consists of a single point x_2 then the above equation becomes

$$\mu_{\boldsymbol{X}_1|\boldsymbol{X}_2}(A_1|x_2) = \int_{A_1} \frac{f_{\boldsymbol{X}_1,\boldsymbol{X}_2}(x_1,x_2)}{f_{\boldsymbol{X}_1|\boldsymbol{X}_2}(x_2)} dx_1.$$

The probability density $f_{\boldsymbol{X}_1|\boldsymbol{X}_2}$ can then be defined as

$$f_{\boldsymbol{X}_1|\boldsymbol{X}_2}(x_1|x_2) = \frac{f_{\boldsymbol{X}_1|\boldsymbol{X}_2}(x_1,x_2)}{f_{\boldsymbol{X}_2}(x_2)}. \tag{2.14}$$

Finally, the Bayes formula is the *most* important concept to the statistical inversion perspective on regularization problems. It is defined as

$$f_{\boldsymbol{X}_2|\boldsymbol{X}_1}(x_2|x_1) = \frac{f_{\boldsymbol{X}_1|\boldsymbol{X}_2}(x_1|x_2) f_{\boldsymbol{X}_2}(x_2)}{f_{\boldsymbol{X}_1(x_1)}}, \tag{2.15}$$

where $f_{\boldsymbol{X}_2|\boldsymbol{X}_1}(x_2|x_1)$ is the *posterior density*, $f_{\boldsymbol{X}_1|\boldsymbol{X}_2}(x_1|x_2)$ is the *likelihood function*, $f_{\boldsymbol{X}_2}(x_2)$ is the *prior density*, and $f_{\boldsymbol{X}_1(x_1)}$ is a normalization constant.

2.4 Statistical Inversion

In inverse problems, one seeks to recover underlying unknown variables from noisy or under-sampled data. Classical inverse problems techniques recover single estimates of the unknown variables by removing ill-posed conditions. Prior information is often included through the use of regularization terms [32].

On the other hand, statistical inversion relies on a non-deterministic approach. Instead, the unknown is modeled as a random variable with the goal to recover information about the unknown variable's probability distribution. Statistical inversion is used to extract this information while also quantifying the uncertainty of the result. This approach relies on Bayes formula (2.15), in

which the distributions of the prior and noise are assumed to be known.

2.4.1 General Statistical Inversion Method

Assume that all random variables are sampled from the same probability space $(\Omega, \Gamma, \mathbb{P})$. Define $\boldsymbol{Y} : \Omega \to \mathbb{R}^m$ to be an observable random variable. Then the random variable $\boldsymbol{Y}$ can be sampled as

$$\boldsymbol{Y}(\omega) = y \tag{2.16}$$

where ω is an outcome in the sample space Ω and y is the observed data. The goal is to recover information about an un-observable, unknown random variable $\boldsymbol{X} : \Omega \to \mathbb{R}^n$. Further, assume the noise distribution describing the random variable $\boldsymbol{E} : \Omega \to \mathbb{R}^k$ is known. Although $\boldsymbol{E}$ models the noise and parameters that are not of interest, its approximate distribution is needed to recover the solution distribution describing the random variable $\boldsymbol{X}$.

Let $g : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^m$ be the forward operator which models the relationship between the un-observable and observable random variables. Then the statistical forward model can be realized as

$$\boldsymbol{Y} = g(\boldsymbol{X}, \boldsymbol{E}). \tag{2.17}$$

Recall from (2.10) the joint probability density of $\boldsymbol{X}$ and $\boldsymbol{Y}$, $f_{\boldsymbol{X},\boldsymbol{Y}}(x, y)$, where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$. Then the marginal density, (2.12) is then

$$\tilde{f}_{\boldsymbol{X}}(x) = \int_{\mathbb{R}^m} f_{\boldsymbol{X},\boldsymbol{Y}}(x, y) dy.$$

Further, the *likelihood function* is the conditional probability density (2.14) of $\boldsymbol{Y}$ given the un-

known $\boldsymbol{X}$:

$$f_{\boldsymbol{Y}|\boldsymbol{X}}(y|x) = \frac{f_{\boldsymbol{X},\boldsymbol{Y}}(x,y)}{\tilde{f}_{\boldsymbol{X}}(x)},$$

where $\tilde{f}_{\boldsymbol{X}}(x) \neq 0$.

Conversely, using Bayes formula, (2.15), the conditional probability density of $\boldsymbol{X}$ given the known data $\boldsymbol{Y}$ is the posterior probability density and encodes the desired information about the unknown random variable $\boldsymbol{X}$:

$$\hat{f}_{\boldsymbol{X}}(x) = f_{\boldsymbol{X}|\boldsymbol{Y}}(x|y) = \frac{f_{\boldsymbol{Y}|\boldsymbol{X}}(y|x)\tilde{f}_{\boldsymbol{X}}(x)}{f_{\boldsymbol{Y}}(y)}. \tag{2.18}$$

Finally, it important to note that the generalized Total Law of Probability, (2.13), provides that the probability density function of the observable random variable $\boldsymbol{Y}$ is

$$f_{\boldsymbol{Y}}(y) = \int_{\mathbb{R}^n} f_{\boldsymbol{X},\boldsymbol{Y}}(x,y)dx = \int_{\mathbb{R}^n} f_{\boldsymbol{Y}|\boldsymbol{X}}(y|x)\tilde{f}_{\boldsymbol{X}}(x)dx \neq 0.$$

We now have all of the tools needed to explain the Bayesian approach to solving inverse problems. Specifically, in the Bayesian approach, the goal is to use the given observation data modeled by (2.16) to characterize the conditional probability density $f_{\boldsymbol{X}|\boldsymbol{Y}}(x|y)$ of the random variable $\boldsymbol{X}$ by calculating (2.18).

This thesis uses the following to characterize the probability density function of the unknown random variable $\boldsymbol{X}$:

1. The likelihood function $f_{\boldsymbol{Y}|\boldsymbol{X}}$ must be defined to characterize the relationship between the observations and the unknown variable $\boldsymbol{X}$.

2. The prior density function $\tilde{f}_{\boldsymbol{X}}(x)$ must be chosen to reflect assumptions of the prior information about the unknown $\boldsymbol{X}$.

3. The posterior probability density $\hat{f}_{\boldsymbol{X}}(x) = f_{\boldsymbol{X}|\boldsymbol{Y}}(x|y)$ should be explored to interrogate the

distribution of $\boldsymbol{X}$.

2.4.2 Likelihood Function with Additive Noise

The construction of the likelihood function should be the most straightforward aspect of the statistical inversion model. The forward model $g(\boldsymbol{X}, \boldsymbol{E})$ from (2.17) is used along with the assumption that the noise random variable $\boldsymbol{E}$ is additive and independent from the unknown random variable $\boldsymbol{X}$. This independence is leveraged to arrive at the likelihood function, which is now described.

Since the noise is assumed to be additive, the forward model in (2.17) can be written as

$$\boldsymbol{Y} = g(\boldsymbol{X}) + \boldsymbol{E} \tag{2.19}$$

where the random variables are again defined as $\boldsymbol{X} : \Omega \to \mathbb{R}^n$, $\boldsymbol{Y} : \Omega \to \mathbb{R}^m$ and $\boldsymbol{E} : \Omega \to \mathbb{R}^m$. Now suppose we acquire realizations $y = \boldsymbol{Y}(\omega)$, $x = \boldsymbol{X}(\omega)$ and $e = \boldsymbol{E}(\omega)$. The realized model in (2.19) can then be expressed as

$$\boldsymbol{Y} = y = g(x) + e$$

with likelihood function

$$f_{\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{E}}(y|x, e) = \delta(y - g(x) - e)$$

where δ is the Dirac delta function. By marginalizing, the likelihood function becomes

$$\begin{aligned} f_{\boldsymbol{Y}|\boldsymbol{X}}(y|x) &= \int_{\mathbb{R}^m} f_{\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{E}}(y|x, e) f_{\boldsymbol{E}|\boldsymbol{X}}(e|x) de \\ &= \int_{\mathbb{R}^m} \delta(y - g(x) - e) f_{\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{E}}(y|x, e) f_{\boldsymbol{E}|\boldsymbol{X}}(e|x) de \\ &= f_{\boldsymbol{E}|\boldsymbol{X}}(y - g(x)|x) \\ &= f_{\boldsymbol{E}}(y - g(x)|x). \end{aligned} \tag{2.20}$$

The last equality above comes from the assumption that $\boldsymbol{X}$ and $\boldsymbol{E}$ are independent. Finally, we arrive at the posterior probability density of the model assuming the noise is additive:

$$\hat{f}_{\boldsymbol{X}}(x) \propto f_{\boldsymbol{E}}(y - g(y))\tilde{f}_{\boldsymbol{X}}(x). \tag{2.21}$$

2.4.3 Prior on the Unknown

Accurate construction of the appropriate prior density function tends to be the most crucial yet also most difficult because the prior belief of a model is often qualitative in nature [24]. In general, the prior density should inform the prior belief about the characterization of the unknown random variable $\boldsymbol{X}$. In particular, if S is a collection of expected vectors representing possible realizations of $\boldsymbol{X}$, and U is a collection of unexpected vectors, it should be the case that

$$\tilde{f}_{\boldsymbol{X}}(x) \gg \tilde{f}_{\boldsymbol{X}}(x'),$$

where $x \in S$ and $x' \in U$.

Here we consider the ℓ_1 prior and the ℓ_2 prior, as they are both consistent with our goal of function recovery:

1. The ℓ_1 prior: For $x \in \mathbb{R}^n$ and some $\alpha > 0$, the prior density is given as

$$\tilde{f}_{\boldsymbol{X}}(x) = \left(\frac{\alpha}{2}\right)^n \exp(-\alpha\|x\|_1). \tag{2.22}$$

   This ℓ_1 prior is also referred to as the *Laplacian prior*, and is known to enforce sparse solutions.

2. The ℓ_2 Prior: The ℓ_2 prior, also called the *Gaussian prior*, is often assumed when appropriate because it is easy to use and provides good approximations due to the *Central Limit Theorem*

[20]. Let $x_0 \in \mathbb{R}^n$ and $\Sigma \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Then the Gaussian prior for the random variable $\boldsymbol{X}$ with mean x_0 and covariance Σ is defined as

$$\tilde{f}_{\boldsymbol{X}}(x) = \Big(\frac{1}{2\pi\det(\Sigma)}^{\frac{n}{2}}\Big)\exp\Big(-\frac{1}{2}(x-x_0)^T\Sigma^{-1}(x-x_0)\Big) \tag{2.23}$$

The choice the prior distribution can determine whether the posterior probability density has a closed form solution. The ℓ_2 prior is Gaussian, which is a conjugate to itself [1]. This implies that if the likelihood is Gaussian and the prior is Gaussian, then the posterior probability density is also Gaussian. Therefore, a closed for solution may be derived from the means and variances of the likelihood and the prior distributions.

However, if the prior is ℓ_1, no conjugacy relations exists and the posterior probability density function has no closed form. This makes recovering the posterior density function particularly difficult as sampling is much less efficient.

2.4.4 Evaluating the Posterior Probability Density with Estimation

In order to find the statistical expectation of the random variable $\boldsymbol{X}$, the regularization in (2.4) is re-framed in a non-deterministic manner through Maximum a *Posteriori* (MAP) estimation. Given the posterior probability density function $f_{\boldsymbol{X}|\boldsymbol{Y}}$ of the random variable $\boldsymbol{X} : \Omega \to \mathbb{R}$, one attempts to obtain an estimate by maximizing the posterior as

$$x_{MAP} = \underset{x\in\mathbb{R}}{\operatorname{argmax}}\, f_{\boldsymbol{X}|\boldsymbol{Y}}(x|y) \tag{2.24}$$

It is important to realize that the (2.24) formulation may have similar pitfalls as the regularization in (2.4). The maximum x_{MAP} may either not exist or not be unique depending on the characterization of the posterior probability density function. Finally we point out that the MAP estimate in (2.24)

is the typical regularization problem in the deterministic approach. This will also be discussed again later in the thesis.

2.5 Markov Chain Monte Carlo (MCMC)

We now introduce the Markov Chain Monte Carlo (MCMC) method, which is at the heart of this thesis. MCMC is our choice of statistical technique through which statistical inversion is investigated. The following mathematical definitions and algorithms are drawn from [17]. We also include insights from the luminaries, Metropolis and Hastings [22, 25].

The delineation of the theory behind MCMC described below requires basic knowledge about Markov Chains, including their convergence and stationary properties. An overview of this necessary background information can be found in [14].

Consider a distribution p_x, $x \in S$ with $\sum_x p_x = 1$ where the state space S can be a subset of the line or even a d-dimensional subset of $\mathbb{R}^d$. The problem posed and solved by [25] was how to construct a Markov chain with stationary distribution π such that $\pi(x) = p_x$, $x \in S$.

Let Q be any irreducible transition matrix on S satisfying the symmetry condition $Q(x, y) = Q(y, x)$ for all $x, y \in S$. Define a Markov Chain $(\theta^{(n)})_{n\geq 0}$ as having transition from x to y proposed according to $Q(x, y)$.

This proposed value for $\theta^{(n+1)}$ is accepted with the probability $\min\{1, \frac{p_y}{p_x}\}$ and rejected otherwise, leaving the chain in state x. If we denote by TA the *transitions accepted*, we can write the transition probabilities $P(x, y)$ of the above chain $(\theta^{(n)})_{n\geq 0}$ as

$$\begin{aligned}
\mathbb{P}(x, y) &= \mathbb{P}(\theta^{(n+1)}) = \mathbb{P}(\theta^{(n+1)} = y, TA|\theta^{(n)} = x) \\
&= \mathbb{P}(\theta^{(n+1)} = y, \theta^{(n)} = x)\mathbb{P}(TA) \\
&= Q(x, y) \min\left\{1, \frac{p_y}{p_x}\right\}.
\end{aligned}$$

Further, for $y = x$ we have

$$\begin{aligned}
\mathbb{P}(x,x) &= \mathbb{P}(\theta^{(n+1)}) = x, TA|\theta^{(n)} = x) + \mathbb{P}(\theta^{(n=1)} \neq x, \overline{TA}|\theta^{(n)} = x) \\
&= \mathbb{P}(\theta^{(n+1)} = x|\theta^{(n)} = x)p(TA) + \sum_{y \neq x} \mathbb{P}(\theta^{(n+1)} = y, \overline{TA}|\theta^{(n)} = x) \\
&= Q(x,x) + \sum_{y \neq x} Q(x,y)\left[1 - \min\left\{1, \frac{p_y}{p_x}\right\}\right].
\end{aligned}$$

We now obtain the stationary distribution of this chain. First we show that p_x satisfies reversibility. For $x \neq y$, we have $\pi(x)\mathbb{P}(x,y) = \pi(y)\mathbb{P}(y,x)$ for all $x, y \in S$. For $x \neq y$, suppose without loss of generality, $p_y > p_x$, we have $p_x Pr(x,y) = p_x Q(x,y) = Q(x,y)\min\{1, \frac{p_y}{p_x}\}p_y = p_y\mathbb{P}(y,x)$.

Thus the chain is reversible and there is a stationary distribution. If Q is a-periodic, so is p and the stationary distribution is also the limiting distribution. Note that not all stationary distributions also constitute as limiting distributions. Therefore $P\pi = \pi \nRightarrow \lim_{n\to\infty} P^n(x,y) = \pi(y)$, where P is the transition probability.

2.5.1 Metropolis-Hastings Algorithms

Consider a distribution π from which a sample must be drawn via Markov chains. This is needed when the non-iterative generation of π is infeasible. In this case, a transition kernel $p(\theta, \phi)$ must be constructed so that π is the equilibrium distribution of the chain.

Now consider the reversible chains where the kernel p satisfies

$$\pi(\theta)p(\theta,\phi) = \pi(\phi)p(\phi,\theta) \tag{2.25}$$

for all (θ, ϕ). This is called *detailed balance* [14]. Detailed balance is a sufficient (but not necessary) condition to ensure convergence.

The kernel $p(\theta, \phi)$ consists of two elements: an arbitrary transition kernel $q(\theta, \phi)$ and a probability $\alpha(\theta, \phi)$ where

$$p(\theta, \phi) = q(\theta, \phi)\alpha(\theta, \phi)$$

for $\theta \neq \phi$. The transition kernel defines a density $p(\theta, \cdot)$, for every $\phi \neq \theta$ and defines a mixed distribution for the new state ϕ of the chain. In [22], the paper proposed an acceptance probability

$$\alpha(\theta, \phi) = \min\left\{1, \frac{\pi(\phi)q(\phi, \theta)}{\pi(\theta)q(\theta, \phi)}\right\} \tag{2.26}$$

Combined with some transition kernel, (2.26) should produce a reversible chain. Algorithms based on chains and transition kernels are referred to as Metropolis-Hastings algorithms.

In [29], it was shown that if q is irreducible and a-periodic, and $\alpha(\theta, \phi) > 0$ for all (θ, ϕ), then the Metropolis Hasting algorithm gives a irreducible, a-periodic chain with transition kernel p and limiting distribution π. The following is the algorithm for running a Markov chain $(X)^t$ for ℓ iterations:

Algorithm 2 Metropolis-Hastings Algorithm

Result: The chain $(X)_{i=1}^{\ell}$

initialize x_0 and $i = 1$

while $i \leq \ell$ **do**

propose $x^{cand} \sim q(x^k|x^{k-1})$

$\alpha(x^{cand}, x^{k-1}) = \min\left\{1, \frac{\pi(x^{cand})q(^{k-1},x^{cand})}{\pi(x^{k-1})q(x^{k-1},x^{cand})}\right\}$

sample $u \sim \mathcal{U}[0,1]$

if $u < \alpha$ **then**

accept the candidate so let $x^k = x^{cand}$

else

reject the candidate so let $x^k = x^{k-1}$

end

end

2.5.2 Convergence Characterization

A useful monitoring device of the method is given by the average percentage of iterations for which moves are accepted [27]. If $q(\cdot,\cdot)$ and $\pi(\cdot,\cdot)$ are continuous, generally smaller step sizes lead to greater acceptance rates but slower convergence rates. This is because the step size is too small to efficiently explore the domain of the target distribution.

On the other hand, if the proposal distribution has a large variance, then the step size tend to be large. This may lead to a relatively lower acceptance rate. However, a lower acceptance rate does not necessarily indicate that the domain is fully explored.

Therefore, the acceptance may be monitored as an indicator for how well the Markov chain is exploring the domain of the target distribution. The acceptance rate however cannot directly imply how well the Markov chain is converging to the target distribution.

There does not exist an ideal acceptance rate as it depends on the target distribution and the chosen proposal distribution. It has been proposed that target distributions in one or two dimensions should correspond to an acceptance ratio of approximately $\frac{1}{2}$ while target distributions of higher dimensions should correspond to acceptance rates of approximately $\frac{1}{4}$ [27]. These goal acceptance rates are not definitive and mainly correspond to Metropolis-Hastings algorithms equipped with a Gaussian proposal distribution.

2.6 Sample Auto-correlation Function

One important tool for analyzing stochastic processes is the sample auto-correlation function. In this thesis, auto-correlation is used to assess the Markov chains used in recovering the unknown. Therefore, the preliminary derivation and explanation of auto-correlation for time series analysis is included in this section. The material is drawn from [11].

The sample auto-correlation is a tool that assesses any correlation between sample observations of a stochastic process at an arbitrarily fixed distance apart. This distance is called the *lag*.

Let us first define the correlation between two stochastic processes $(\boldsymbol{X})^t = (x_1, x_2, x_3, \ldots, x_N)$ and $(\boldsymbol{Y})^t = (y_1, y_2, y_3, \ldots, y_N)$ as

$$r = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{N}(y_i - \overline{y})^2}}, \tag{2.27}$$

where $\overline{x}$ and $\overline{y}$ are the arithmetic mean of the stochastic processes $(\boldsymbol{X})^t$ and $(\boldsymbol{Y})^t$ respectively.

The correlation is non-existent if the above formula evaluates to 0. Intuitively, when $r \approx -1$, a high value of $(\boldsymbol{X})^t$ tends to correspond with a low value of $(\boldsymbol{Y})^t$, while a low value of $(\boldsymbol{X})^t$ tends to correspond to a high value of $(\boldsymbol{Y})^t$. On the other hand, for $r \approx 1$, a high value of $(\boldsymbol{X})^t$ tends to correspond to a high value of $(\boldsymbol{Y})^t$, and similarly a low value of $(\boldsymbol{X})^t$ tends to correspond to a

low value of $(\boldsymbol{Y})^t$.

This idea of correlation may be applied to a time series analysis in which one state in a stochastic process is compared to another state in the same stochastic process at a different time. The time difference defined to be the lag. For example, suppose we are given N observations $x_1, \ldots, x_N$, for which we want to compare the pairs $(x_1, x_2), (x_2, x_3), (x_3, x_4), \ldots, (x_{N-1}, x_N)$. In this case, the lag is 1 because the time difference between the two observations in a pair is 1. Then from (2.27), the auto-correlation of lag 1 is

$$r_1 = \frac{\sum_{i=1}^{N-1}(x_t - \overline{x_{(1)}})(x_i - \overline{x_{(2)}})}{\sqrt{\sum_{i=1}^{N-1}(x_i - \overline{x_{(1)}})^2 \sum_{t=1}^{N-1}(x_i - \overline{x_{(2)}})^2}}, \tag{2.28}$$

where

$$\overline{x_{(k)}} = \sum_{i=1}^{N-k} \frac{x_i}{N-1}.$$

For practical purposes, (2.28) may be approximated with

$$r_1 = \frac{\sum_{i=1}^{N-1}(x_i - \overline{x})(x_{i+1} - \overline{x})}{(N-1)\sum_{i=1}^{N} \frac{(x_i - \overline{x})^2}{N}}, \tag{2.29}$$

where

$$\overline{x} = \sum_{i=1}^{N} \frac{x_i}{N}.$$

For large N, $\frac{N-1}{N} \approx 1$, (2.29) becomes

$$r_1 = \frac{\sum_{i=1}^{N-1}(x_i - \overline{x})(x_{i+1} - \overline{x})}{\sum_{i=1}^{N}(x_i - \overline{x})^2}. \tag{2.30}$$

In general, the auto-correlation function for a lag of $k \in \mathbb{N}$ is

$$r_k = \frac{\sum_{i=1}^{N-k}(x_i - \overline{x})(x_{i+k} - \overline{x})}{\sum_{i=1}^{N}(x_i - \overline{x})^2}. \tag{2.31}$$

Then r_k is said to be the sample auto-correlation coefficient at a lag of k.

Finally, we note that the auto-correlation is usually visually assessed through a *correlogram*, which is a plot of the sample auto-correlation coefficients r_k against the lag k for $k = 1, \ldots, M$. Usually M is a much smaller number than N. This thesis will also use this visualization for assessment and comparison of methods (See Chapter 4). In particular we note that if a stochastic process only has a short-term correlation, then the value should start of close to -1 or 1 but decay to approximately 0 as the lag k increases.

2.7 The History of Markov Chain Monte Carlo

The papers discussed and analyzed here are relevant to the methodology presented in Chapter 2. Moreover, the historical and scientific context provided both explain and motivate the focus of this thesis.

The set of techniques known to the world as Monte Carlo originated from Los Alamos, New Mexico during World War II [28]. At the time, it was not yet associated with Markov Chan Monte Carlo, which was coined later. The idea of using statistical sampling to approximate distributions allowed physicists to compute intractable integrals in order to work on the atomic bomb.

It is believed that original Monte Carlo formulation was born out Mathematician Stanisław Marcin Ulam's desire to solve of the combinatorially intractable problem of computing the probability of winning the game solitaire in the 1940's. Ulam was a brilliant mathematician and nuclear physicist who worked on the Manhattan project. Ulam's colleague, John von Neumann utilized Ulam's Monte Carlo method to study neutron fusion.

2.7.1 Two significant papers by Metropolis et. al.

In 1952 Nicholas Metropolis published a ground-breaking paper in which he advanced the idea of Monte Carlo by using stochastic processes. Metropolis was credited with the invention of the first Markov Chain Monte Carlo (MCMC) algorithm. Through MCMC, Metropolis was able to compute integrals that in general do no possess closed form solutions.

In the next year, Metropolis and his co-authors [25] demonstrated the ergodicity of the Metropolis-Hastings algorithm (given by Algorithm 2). The significance of ergodicity lies in the fact that it corresponds to guarantees in convergence to a stationary distribution [14] in Markov chain theory.

2.7.2 The Hastings paper in 1970

The Metropolis algorithm was adopted by statistician W. K. Hastings as a statistical technique to overcome the "curse of dimensionality" problem [22]. Further, Hastings introduced the Gibbs sampling technique, where each component of the posterior probability density is updated one at a time in iterations.

From the 1953 to 1990, the advancements in MCMC was stagnant mostly because the computational power at the time was inadequate for efficient MCMC simulations. This stagnation finally ended when Gelfand and Smith published a paper using MCMC to calculate marginal densities. Their efforts engendered a shift on the statistics community to embrace the power of MCMC [19].

Since then, MCMC became a highly investigated research area in the 1990's as researchers were awed by the power of the Gibbs Sampler in approximating probability distributions and also stunned by its generality in applications. Techniques such as the Metropolis-Hastings algorithm are now commonplace in the statistical community. Finally, as discussed in this thesis, MCMC, and in particular the Metropolis-Hastings algorithm, can be also used and further enhanced for

function and image recovery using the numerical analysis perspective of solving inverse problems.

2.7.3 Quantifying Uncertainty with MCMC

In recent years, MCMC has become a point of interest to the inverse problems and uncertainty quantification communities. Among them is the works of Johnathan M. Bardsley, which have influenced the work of this thesis. In particular, the methods developed here build upon his work in applying MCMC to regularization through a Bayesian framework described in [6]. There, the linear inverse problem

$$b = Ax + \eta \tag{2.32}$$

is considered where $b \in \mathbb{R}^m$ is the observable data and $x \in \mathbb{R}^n$ is the variable to be recovered. The operator $A \in \mathbb{R}^{m \times n}$ is the forward matrix and $\eta \in \mathbb{R}^m$ is the the noise which for practical reasons is assumed to be independent, identical Gaussian random distribution. Each component of η has a variance of λ^{-1}, where λ is defined to be the precision.

The solution is then modeled as a standard regularization problem, given by

$$x_\alpha = \underset{x}{\operatorname{argmax}} \left\{ ||Ax - b||^2 + \alpha x^T L x \right\}. \tag{2.33}$$

In (2.33), the x_α is the solution to the inverse problem, L is the regularization operator and α is the regularization parameter. A more indepth discussion on the general formulation of L can be found in Section 2.1.

Viewed from the perspective of statistical inversion through Bayes' Theorem,

$$f_{\boldsymbol{x}|\boldsymbol{b}}(x|b) \propto f_{\boldsymbol{b}|\boldsymbol{x}}(b|x) f_{\boldsymbol{x}}(x),$$

the regularization problem (2.33) is derived as

$$f_{\boldsymbol{b}|\boldsymbol{x}}(b|x) \propto \exp\Big(-\frac{\lambda}{2}||Ax-b||^2\Big)$$
$$f_{\boldsymbol{x}}(x) \propto \exp\Big(-\frac{\delta}{2}x^T Lx\Big)$$
$$-\ln f_{\boldsymbol{x}|\boldsymbol{b}}(x|b) \propto ||Ax-b||^2 + \frac{\delta}{\lambda}x^T Lx,$$

which was discussed in Section 2.4. In addition to λ as the prior precision of the data, δ is introduced here as the prior precision. In [6] the regularization parameter is defined as $\alpha = \frac{\delta}{\lambda}$. This is a crucial aspect of that investigation as the regularization parameter must typically be tuned for an inverse problem. By contrast, in this investigation $\alpha = \frac{\delta}{\lambda}$ is treated as part of the posterior probability density to be recovered computationally.

With this formulation, one hopes to better characterize the solution to (2.33) than traditional methods can. Specifically, the standard approach to solving a regularization problem such as equation (2.33) is to find the MAP estimate explained in Section 2.4.4. However, the MAP estimate only finds the peak of the posterior probability density without any insight into higher moment information such as variance.

Therefore, instead of only computing the MAP estimate, in [6] Gibbs sampler [17, 27] is used to explore the posterior probability density to gain insights into the variance and thus the uncertainty of the regularization formulation in (2.33). It is assumed that a region with high variance also corresponds to high uncertainty in the variable x.

In particular δ and λ are also recovered along with the unknown variable x.

Some assumptions are need in order to proceed:

1. The prior distributions of δ and λ are assumed to be Gamma distributions.

2. The regularization operator L is defined with Gaussian Markov random fields as explained in [31].

3. Each entry in x should be similar to its neighbors (refer to [6] for details).

Then the conditional distributions needed to recover the full posterior are

$$\begin{aligned} f_{\boldsymbol{x}|\boldsymbol{\lambda},\boldsymbol{\delta},\boldsymbol{b}}(x) &\propto \exp\Big(\frac{\lambda}{2}||Ax-b||^2 - \frac{\delta}{2}x^T Lx\Big) \\ f_{\boldsymbol{\lambda}|\boldsymbol{x},\boldsymbol{\delta},\boldsymbol{b}}(\lambda) &\propto \lambda^{\frac{n}{2}+\alpha_\lambda - 1}\exp\Big(\big[-\frac{1}{2}||Ax-b||^2 - \beta_\lambda\big]\lambda\Big) \\ f_{\boldsymbol{\delta}|\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{b}}(\delta) &\propto \lambda^{\frac{n}{2}+\alpha_\delta - 1}\exp\Big(\big[-\frac{1}{2}||Ax-b||^2 - \beta_\delta\big]\delta\Big), \end{aligned} \tag{2.34}$$

in which $\alpha_\lambda = \alpha_\delta = 1$ and $\beta_\lambda = \beta_\delta = 10^{-4}$. The choice of these hyper-parameters makes then "uninformative" because the mean and variance of the corresponding Gamma distributions is $\frac{\alpha}{\beta} = 10^4$ and $\frac{\alpha}{\beta}^2 = 10^8$ [6].

Algorithm 3 A MCMC Method for Sampling from $p(x, \delta, \lambda|b)$

Result: The chains $(X)^n, (\lambda)^n$, and $(\delta)^n$

initialize δ_0, λ_0 and set $k = 0$

while $k \leq n$ **do**

compute x^k

compute λ_{k+1}

compute δ_{k+1}

set $k \leftarrow k + 1$

end

In Algorithm 3, each sample in the while-loop is correspondingly drawn from

$$\begin{aligned} x|\lambda,\delta,b &\sim \mathcal{N}\big((\lambda A^T A + \delta L)^{-1}\lambda A^T b, ({}^T A + \delta L)^{-1}\big) \\ \lambda|x,\delta,b &\sim \Gamma\big(\frac{n}{2} + \alpha_\lambda, \frac{1}{2}||Ax-b||^2 + \beta_\lambda\big) \\ \delta|x,\lambda,b &\sim \Gamma\big(\frac{n}{2} + \alpha_\delta, \frac{1}{2}x^T Lx + \beta_\delta\big). \end{aligned}$$

More details can be found in [6], where information can also be found about how the algorithm can be made computationally tractable by leveraging numerical linear algebra techniques. Finally, the algorithm warrants MCMC chain convergence anlaysis which is also included in [6].

The remainder of this thesis explores using a new technique for recovering the solution to (2.32). An important distinction between this investigation and the one in [6] is that here we use the Metropolis-Hastings algorithm, rather than the Gibbs Sampler. In [6], the Gibbs Sampler is feasible because the full conditional distributions (2.34) have closed-form solutions due to the deliberate choice of the of a Gaussian distribution for the prior and a Gamma distributions for the hyper-parameters. Here, however, the ℓ_1 prior is chosen to quantify the prior belief. As a consequence, there does not exist a closed form for the posterior probability density. Therefore, the Metropolis-Hastings algorithm is implemented instead in Algorithm 4.

To expand upon the findings in [6], and to reduce the uncertainty in the function recovery, the method introduced in this thesis uses an ℓ_1 prior to encourage sparsity in the edge domain. We note that ℓ_1 priors have been used before for this purpose. What is novel here is the introduction of a *weighted* ℓ_1 prior that takes advantage of the *joint sparsity* occurring across multiple measurement vectors. This is accomplished by using the *variance based joint sparsity* (VBJS) method, [3], which was previously described in Section 2.2.2. The new weighted ℓ_1 prior is directly incorporated into (2.21), which is later used to form (3.7). With these tools, we are able to both further quantify uncertainty as well as to reduce it.

Chapter 3

Methodology

3.1 Problem Set-up

We consider the one-dimensional additive model given by (2.19). We assume that the model is linear, i.e., $g(\boldsymbol{X}) = A\boldsymbol{X}$ with $A \in \mathbb{R}^{n\times n}$, and the noise $\boldsymbol{E}$ additive Gaussian white noise (AGWN) sampled from a Gaussian distribution, $\mathcal{N}[0, \sigma]$. These assumptions yield the resulting statistical model

$$\boldsymbol{Y} = A\boldsymbol{X} + \boldsymbol{E}, \tag{3.1}$$

where all random variables are sampled from a common probability space $(\Omega, \Gamma, \mathbb{P})$.

The underlying function we seek to recover is given by

$$h(s) = \begin{cases} 40, & 0.1 \leq s \leq 0.25 \\ 10, & 0.35 \leq s \leq 0.325 \\ \frac{2\pi}{0.05\sqrt{2\pi}} e^{-\left(\frac{s-0.75}{0.05}\right)^2}, & s > 0.5 \\ 0, & \text{otherwise,} \end{cases} \tag{3.2}$$

where $s \in [0, 1]$. We chose (3.2 because it is almost identical to the example used in [6], thus making our new method easier to compare to current state of the art.

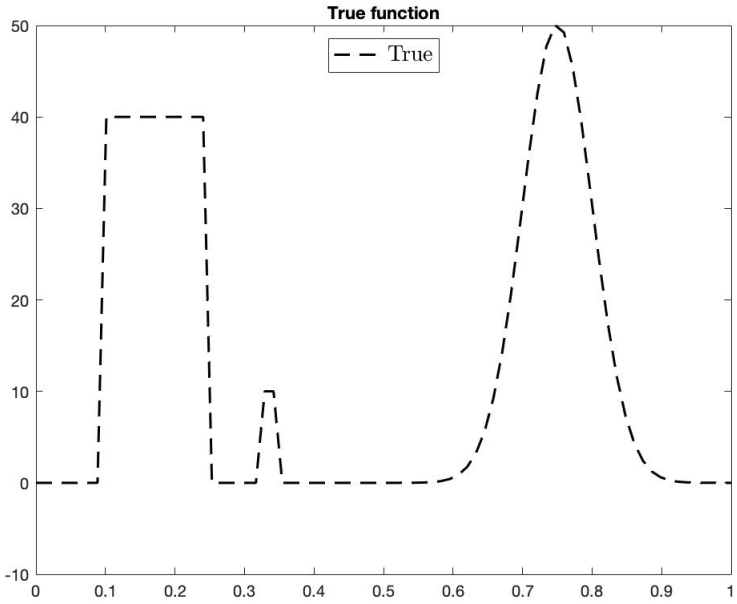


Figure 3.1: Test function given in (3.2).

Figure 3.1 displays the function given in (3.2). Observe that there are several challenging features to recover, including discontinuities, steep gradients and variation in scales. In particular, it is often difficult to recover small and narrow features, such as the one shown for $0.35 \leq s \leq 0.25$. The exponential function (Gaussian hump) also may require significant data for adequate resolution. Moreover, the recovery of each feature gets more difficult as we increase the variance σ of the AGWN $\boldsymbol{E}$ in (2.19).

Similar to [6], the domain Ω is discretized uniformly with $n = 80$ grid-points in the interval $[0, 1]$ and $A \in \mathbb{R}^{n \times n}$ is constructed to model Gaussian blur for given σ

$$\text{Blur}(s) = \frac{\exp\left(\frac{-s^2}{2\sigma^2}\right)}{\sqrt{\pi\sigma^2}}.$$

That is, the entries of A are given by

$$[A]_{ij} = \frac{1}{n}\frac{\exp(\frac{-(\frac{i-j}{n})^2}{2\sigma^2})}{\sqrt{\pi\sigma^2}}, \quad 1 \le i, j, \le n. \tag{3.3}$$

In the model described in (3.1), the underlying structure of $\boldsymbol{X}$ is assumed to be sparse in the edge domain, which we approximate using the polynomial annihilation (PA) technique introduced in [5]. As discussed in Section 2.2, the PA operator is akin to higher order total variation (HOTV) with nuanced differences in the derivation and normalization. For the remainder of this thesis, the regularization operator $\mathcal{L}$ will refer to the PA operator with a order to be specified [5].

Recall that the posterior density derived in (2.21) uses the assumption that the noise is additive and independent from the unknown random variable $\boldsymbol{X}$. Given that $\boldsymbol{E} \sim \mathcal{N}[0, \sigma]$, it follows that the distribution of the noise has probability density function

$$f_{\boldsymbol{E}}(e) = C_1 \exp\Big\{ -\frac{1}{2}(e-0)^T \frac{1}{\sigma^2}(e-0)\Big\} = C_1 \exp\Big\{ -\frac{\|e\|_2^2}{2\sigma^2}\Big\}, \tag{3.4}$$

for some constant $C_1 \in \mathbb{R}$. Thus, the likelihood (2.20) is

$$f_{\boldsymbol{Y}|\boldsymbol{X}}(y|x) = f_{\boldsymbol{E}}(y - Ax) = C_1 \exp\Big\{ -\frac{\|y - Ax\|_2^2}{2\sigma^2}\Big\}. \tag{3.5}$$

The ℓ_1 norm is chosen for the prior distribution because the prior assumption about the sampled vector $x = \boldsymbol{X}(\omega)$, $\omega \in \Omega$ is that it is sparse in the edge domain. Since the true function can be modeled as a piecewise polynomial (and specifically *not* as a piecewise constant), it is better to choose the second-order polynomial annihilation operator $\mathcal{L}$ on x to obtain $\mathcal{L}x$ that is sparse on the domain $[0, 1]$.[1] In this way, for most of the domain, $\mathcal{L}x$ can be assumed to closely approximate

[1] Indeed, an argument can be made for using a third order given the exponential term in (3.2). However, noise in the model starts to interfere with the ability to recover the sparse domain. It is important to note that the standard gradient (equivalently TV norm) is not appropriate, since the underlying assumption there would be that the gradient domain is sparse, which it is *not* in this case for (3.2).

zero, that is, there are only a few non-zero entries in the resulting vector $\mathcal{L}x$. Refer to 2.2.1 for a detailed explanation of the ℓ_1 norm in regularization. Analogous to the ℓ_1 prior defined in (2.22) for a sparse solution, we have

$$\tilde{f}_{\boldsymbol{X}}(x) = C_2 \exp\{-\alpha||\mathcal{L}x||_1\}, \tag{3.6}$$

for some $C_2, \alpha \in \mathbb{R}$ with $\alpha > 0$. Then the posterior distribution (2.21) becomes

$$\hat{f}_{\boldsymbol{X}}(x) = C_3 \exp\Big\{ - \alpha||\mathcal{L}x||_1 - \frac{1}{2\sigma^2}||y - Ax||_2^2\Big\}. \tag{3.7}$$

Finally, the goal is to recover samples of the posterior distribution described by the density $\hat{f}_{\boldsymbol{X}}(x)$ in (3.7) using the Metropolis-Hastings algorithm with a weighted regularization function described in Section 2.2 [3, 18] so that the weighted posterior distribution is

$$\hat{f}_{\boldsymbol{X},w}(x) = C_3 \exp\Big\{ - \alpha||W\mathcal{L}x||_1 - \frac{1}{2\sigma^2}||y - Ax||_2^2\Big\}, \tag{3.8}$$

in which the weighting matrix $W \in \mathbb{R}^{J\times J}$ is calculated with Algorithm 1.

From MCMC techniques, the uncertainty of the recoveries of (3.7) and (3.8) may be quantified using higher moment characterization of the MCMC chains and compared.

3.2 MAP Estimation

To initiate our MCMC sampling techniques and establish a more appropriate prior, we seek multiple measurement vectors. First, to obtain J measurement vectors, we sample from the random variable $\boldsymbol{Y}$ to obtain the vectors $\vec{y} = [y_1, y_2, y_3, \ldots, y_J]$.

In order to apply the Metropolis-Hastings algorithm effectively, the data $\vec{y}$ is pre-processed to reveal preliminary insights into the structure of the posterior density. To begin, we calculate

point estimations using the MAP estimation technique described in Section 2.4.4. This allows us to better interrogate the peak of the posterior distribution given in (3.7) and (3.8). Thus, for each measurement vector we calculate its corresponding MAP estimate as

$$\begin{aligned} x^i_{MAP} &\equiv \underset{x\in\mathbb{R}^n}{\operatorname{argmax}} \log\left[\hat{f}_{\boldsymbol{X}}(x)\right] \\ &= \underset{x\in\mathbb{R}^n}{\operatorname{argmax}} \left\{ \log(C_3) - \left(\alpha||\mathcal{L}x||_1 + \frac{1}{2\sigma^2}||y_i - Ax||_2^2\right)\right\} \\ &= \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \left\{\alpha||\mathcal{L}x||_1 + \frac{1}{2\sigma^2}||y_i - Ax||_2^2\right\}, \quad i = 1, ..., J. \end{aligned} \tag{3.9}$$

Because the constants C_3, α, and σ are assumed to be unknown, these values must be tuned. For this problem, $C_3 = 1$, $\alpha = 1$ and $\sigma = 1$. Consequently, $x_{\vec{MAP}} = [x^1_{MAP}, x^2_{MAP}, x^3_{MAP}, \ldots, x^J_{MAP}]$ offers information on the mode of the posterior distribution. Thus, the MCMC chain of the Metropolis-Hastings algorithm may be initialized at the arithmetic average of these J MAP estimates. For the numerical experiments considered in this thesis we fix $J = 20$ and note that a thorough exploration into the consequence of changing this parameter is left for future work.

3.3 Metropolis-Hastings Algorithm

To sample from either the unweighted or weighted posterior probability densities (3.7) and (3.8) using MCMC, the Metropolis-Hastings algorithm is needed. For more background and theoretical details of the Metropolis-Hastings algorithm, refer to Section 2.5. The history of the algorithm is further explained in Section 2.7.

The Metropolis-Hastings algorithm relies on an ergodic Markov chain that converges to the corresponding posterior probability density [20, 14]. In the end, the mean of the Markov chain is used to approximate a point estimate of the random variable $\boldsymbol{X}$, while higher order information, such as the variance of the chain, is used to build confidence intervals.

First, we construct a Markov kernel with stationary distribution that corresponds to the posterior probability density. This allows the generation of a Markov chain $(X^{(t)})$ using the kernel. The chain is constructed to satisfy ergodicity so that the the Markov chain converges to appropriate posterior probability density. Suppose that the vector $\vec{x}_0$ is the solution to the MAP estimate (3.9) and let ℓ be a predetermined length of the Markov chain.

The proposal distribution is denoted at q and the acceptance distribution is defined as α. In each iteration of the chain, the proposal distribution q proposes a candidate state based on the current state of the chain then the acceptance distribution is the probability of accepting the proposed candidate as the next state of the Markov chain. For details, refer to Section 2.5.

In this thesis, the proposal distribution is chosen to be the Gaussian distribution, in which $x^{cand} \sim q(x^k|x^{k-1})$ becomes $x^{cand} \sim \mathcal{N}(x^{k-1}, 0.1)$ which is the Gaussian distribution with mean x^{k-1} and standard deviation 0.1. Suppose that $\hat{f}_{\boldsymbol{X}}(x)$ is the posterior probability density. Notice that the Gaussian distribution is symmetric so the acceptance distribution simplifies to

$$\alpha(x^{cand}|x^{k-1}) = \min\left\{1, \frac{q(x^{k-1}|x^{cand})\hat{f}_{\boldsymbol{X}}(x^{cand})}{q(x^{cand}|x^{k-1})\hat{f}_{\boldsymbol{X}}(x^{k-1})}\right\} = \min\left\{1, \frac{\hat{f}_{\boldsymbol{X}}(x^{cand})}{\hat{f}_{\boldsymbol{X}}(x^{k-1})}\right\} \tag{3.10}$$

because then $q(x^{k-1}|x^{cand}) = q(x^{cand}|x^{k-1})$. Refer to Section 2.5 for details.

Algorithm 4 Metropolis-Hastings Algorithm

Data: $\vec{x}_0$ is arithmetic mean of the MAP estimates

Result: Markov chain $M \in \mathbb{R}^{n\times\ell}$

initialize $\vec{x}_0$ and $i = 1$

while $i \leq \ell$ **do**

propose $x^{cand} \sim q(x^k|x^{k-1})$

$\alpha(x^{cand}|x^{k-1}) = \min\left\{1, \frac{\hat{f}_{\boldsymbol{X}}(x^{cand})}{\hat{f}_{\boldsymbol{X}}(x^{k-1})}\right\}$

sample $u \sim \mathcal{U}[0,1]$

if $u < \alpha$ **then**

accept the candidate so let $x^k = x^{cand}$

else

reject the candidate so let $x^k = x^{k-1}$

end

end

The iterations of the Markov chain from the above algorithm may be represented by a matrix $M \in \mathbb{R}^{n\times\ell}$ where n is the number of grid points and ℓ is the length chosen for the Markov chain.

$$M = \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & \dots & x_1^\ell \\ x_2^0 & x_2^1 & \dots & x_2^3 & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n^0 & x_n^1 & x_n^2 & \dots & x_n^\ell \end{bmatrix}$$

Each column of the matrix M represents one state in the Markov chain. An entry x_j^i is the j^{th} grid point of the i^{th} state in the Markov chain. Suppose that the constant $B < \ell$ is the burn-in rate then the approximated posterior distribution $\hat{f}_{\boldsymbol{X}}(x)$ is

$$\hat{f}(x_i) = \frac{1}{\ell - B}\sum_{t=B+1}^{\ell} x_i^t \tag{3.11}$$

Thus, after discarding the first B states in the Markov chain, the mean is taken of the remaining chain to approximate the posterior distribution. For this problem, the number of iterations is chosen to be 50000 iterations and the first half of the states are discarded. Therefore, $\ell = 50000$ and $B = 25000$.

In total, the Metropolis-Hastings algorithm described in Algorithm 4 is performed for both the unweighted posterior probability density $\hat{f}_{\boldsymbol{X}}(x)$ described in (3.7) and the weighted posterior probability density $\hat{f}_{\boldsymbol{X},w}(x)$ described in (3.8).

3.3.1 Convergence Analysis

Theoretically, the chains in the Metropolis-Hastings algorithm should converge to their respective posterior probability densities (3.7) and (3.8) (refer to Subsection 2.5.2 for the theoretical details of convergence analysis). However, in practice, the convergence of the Metropolis-Hastings algorithm is difficult to guarantee or even quantify.

One issue that becomes immediately apparent is where the Markov chain should be initiated. Often times, one must approximately know the support of the posterior distribution to run a Markov chain. Otherwise, the chain may appear to be converging to the stationary distribution, but in reality miss the mode of the distribution of interest. Therefore, it is auspicious for the chain to begin within the mode of the posterior probability density.

In Algorithm 4, the beginning state of the chain is the MAP estimate (2.24). the MAP estimate is the peak of the mode of the posterior probability density. This starting state ensures that the Markov chain is exploring the correct region of the posterior probability density [27].

In order to diagnose the convergence of the Markov chain, the trace plot and the acceptance rate over time are used as indicators [15]. The trace plot is a plot of the state of the chain over time of iterations. It is an indication of the convergence to the posterior probability density. Theoretically,

if the Markov chain is at the posterior probability density, the states of the chain should not be correlated and the trace plot should exhibit behavior as if the states are drawn from independent identical distributions. Therefore, if the at a time of the Markov chain, there is high auto-correlation between states, it implies that the Markov chain is not at the stationary distribution.

To quantify the convergence to the posterior probability density, the correlogram of the trace plot is computed. The MATLAB package Autocorrelation Function (ACF) is used with the lag to be 10 iterations [26]. Refer to Section 2.6 for the derivation of auto-correlation and the construction of the correlogram. In this problem, the correlogram plots the lag from 1 to 5000.

Finally, the acceptance ratio is also used to indicate the convergence of the Markov chain. As pointed out in Subsection 2.5.2, an acceptance rate that is too high may indicate the Markov chain is stuck in a certain region of the posterior probability density and have not explored the entire support of the density. In contrast, an acceptance rate that is too low may indicate that the proposed states are too small compared to the current state. The chain may be moving too quickly on the domain of density and may be caused by the variance of the proposal being too large. Further, even with a low acceptance rate, it remains possible that the chain does not explore certain isolated modes of the posterior probability density [27].

It is debatable what the ideal acceptance rate should be for the Metropolis-Hastings algorithm. According to Roberts et al. (1997), the acceptance rate should be close to $\frac{1}{2}$ for a one-dimensional problem [27]. This metric was primarily derived from the use of the Gaussian proposal distribution.

3.4 MCMC with Unweighted and Weighted ℓ_1 Regularization

The true function to be recovered is (3.2) discretized into the vector $x \in \mathbb{R}^{80}$. In total, 20 samples of $\boldsymbol{Y} = A\boldsymbol{X} + \boldsymbol{E}$ are drawn as described to compute the MAP estimates with each of the AWGN standard deviation: $0.25, 0.50, 0.75$ and 1.00. The arithmetic mean of the signal-to-noise ratio (SNR) in decibels of samples are $36.75, 32.20, 27.43$, and 25.13, respectively. The SNR of a signal $y \in \mathbb{R}^N$ is obtained by calculating the ratio of its summed squared magnitude of the sample to that of the noise $e \in \mathbb{R}^N$ [2]. For each noise level, using the arithmetic mean of the MAP estimates as the starting state, the Metropolis-Hastings algorithm in Algorithm 4 is ran to recover the unweighted posterior probability density and another Metropolis-Hastings algorithm in Algorithm 4 is ran to recover the weighted posterior probability density. When constructing the weights for the posterior probability density, the hyper-parameter τ must be tuned [18]. Refer to Section 2.2 for the definition of the threshold τ. In general, an increase in the SNR of the sample corresponds to the need for a greater τ value because a greater threshold for noise is needed. In this one-dimensional problem, the τ values are chosen to be $0.05, 0.10, 0.15$, and 0.20, corresponding to increasing AWGN standard deviation.

To quantify the convergence of the MCMC chains of the Metropolis-Hastings algorithm, several convergence checking plots are presented in 4. They compare the convergence rates and the distributions of the MCMC chains through standard MCMC analysis described in Section 2.5.

Chapter 4

Results and Discussion

4.1 Numerical experiments

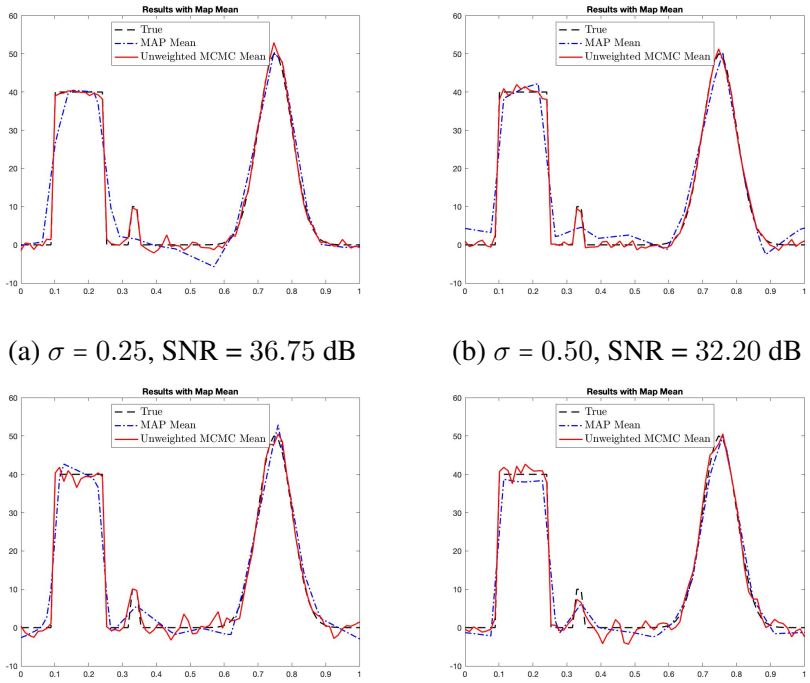
We now present results for recovering the function in (3.2) using the MAP estimate, the MCMC method using the (unweighted) posterior in (3.7), and the newly proposed weighted MCMC technique which uses the weighted posterior in (3.8) . For clarity of presentation, solid red will always be used to plot the MCMC recovery without weights (which we will refer to as the unweighted MCMC), while solid blue will denote the weighted MCMC.

4.1.1 Recovery of Unknown Posterior Probability Density

Figure 4.1 shows the means of the MAP estimates and MCMC with the unweighted posterior probability density (3.7). The MAP estimate mean is used as the first state to initialize the Markov chains in Algorithm 2. Observe that both the MAP estimate and the unweighted MCMC mean become less accurate as the standard deviation of the noise random variable is increased.

The unweighted MCMC is able to explore the region of the posterior probability density (3.7)

that captures the peak $x = 10$ better than the MAP estimate for all tested levels of noise. On the other hand, the unweighted MCMC mean appears to be influenced by the increase in noise more than the MAP estimates. This can be observed in the regions where $x = 0$, where the unweighted MCMC mean has relatively greater oscillatory behavior when the noise standard deviation is increased to $\sigma = 0.75$ and $\sigma = 1.00$.



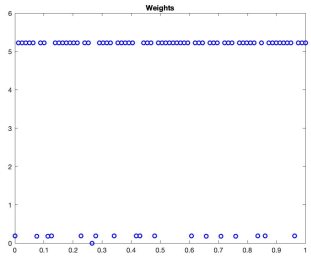
(a) $\sigma = 0.25$, SNR = 36.75 dB (b) $\sigma = 0.50$, SNR = 32.20 dB

(c) $\sigma = 0.75$, SNR = 27.43 dB (d) $\sigma = 1.00$, SNR = 25.13 dB

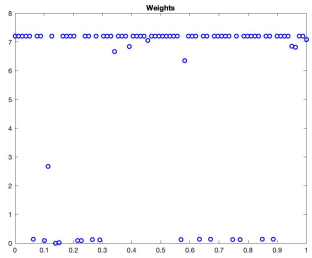
Figure 4.1: MAP mean and the Unweighted MCMC

The main premise of this investigation is to use information that comes from the numerical convergence properties of the sparse domain estimate to design weights that distinguish smooth regions from areas where jump discontinuities occur. It is the weight vector in (3.8) that allows a more accurate posterior for the MCMC. Figure 4.2 shows the weights constructed by Algorithm 1 in Section 2.2. The regions where the edge domain is assumed to be near-zero correspond to the higher weights while the regions where the edge domain is assumed to be non-zero correspond to near-zero weights. As is expected, the weights become less meaningful as the amount of noise is

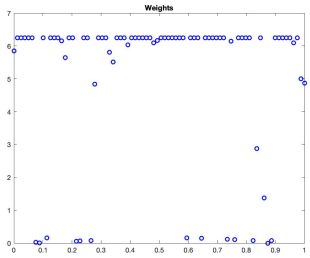
increased, or equivalently as the SNR decreases. In particular observe that for standard deviation $\sigma = 3$, the constructed weights are no longer as helpful in determining sparse regions. Hence the weighted and unweighted MCMC reconstructions become correspondingly less distinctive. Although not a part of this investigation, it should be noted that with increased resolution (i.e. using more than 80 grid-points between $[0, 1]$) will improve the approximation of the sparse domain. Other non-linear enhancements, such as those discussed in [3], also show promise in reducing the impact of noise on the sparse domain regularization term. These ideas will be considered in future work.



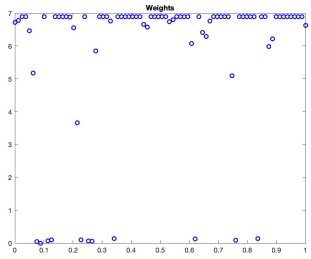
(a) $\sigma = 0.00$, SNR = ∞ dB



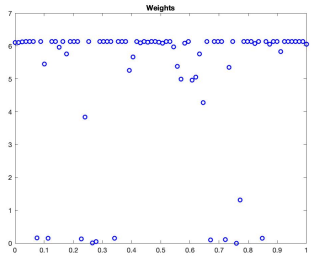
(b) $\sigma = 0.25$, SNR = 36.75 dB



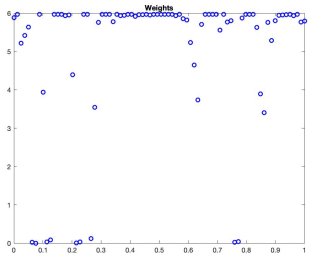
(c) $\sigma = 0.50$, SNR = 32.20 dB



(d) $\sigma = 0.75$, SNR = 27.43 dB



(e) $\sigma = 1.00$, SNR = 25.13 dB

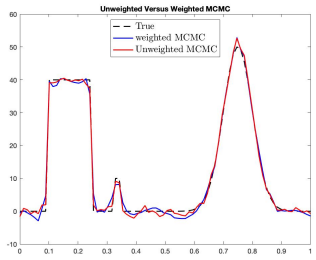


(f) $\sigma = 3.00$, SNR = 15.76 dB

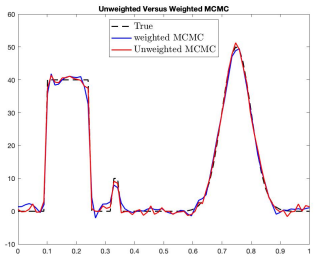
Figure 4.2: Weights constructed using the Algorithm 1

Figure 4.3 shows the mean values of the unweighted and weighted chain. Although both formulations see worse recovery with higher noise, the weighted MCMC show a better mean convergence to the true function. This difference is particularly pronounced in the regions that are sparse in the edge domain, which is not surprising when the weights plotted in Figure 4.2 are taken into consideration. In the regions without edges, the weights are relatively high, and thus smoothness is

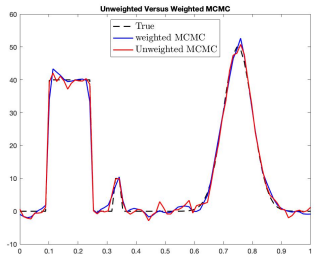
heavily enforced. Therefore, the mean of the weighted MCMC is smoother in regions corresponding to the smooth regions in $h(x)$ in (3.2).



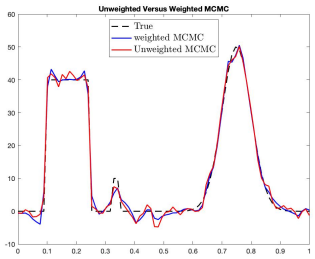
(a) σ = 0.25, SNR = 36.75 dB



(b) σ = 0.50, SNR = 32.20 dB



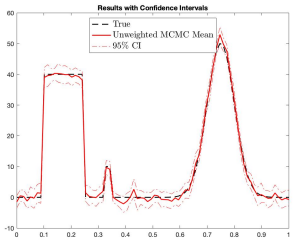
(c) σ = 0.75, SNR = 27.43 dB



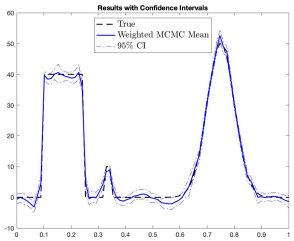
(d) σ = 1.00, SNR = 25.13 dB

Figure 4.3: Unweighted MCMC and weighted MCMC

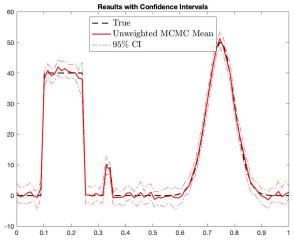
The 95% confidence intervals of of the chains are calculated and displayed in Figure 4.4. Comparatively, the confidence interval in for the weighted MCMC is much tighter. In particular, this difference is again more pronounced in regions where the edge domain of the true function is sparse.



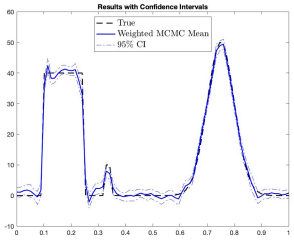
(a) σ = 0.25, SNR = 36.75 dB



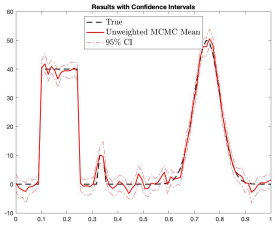
(b) σ = 0.25, SNR = 36.75 dB



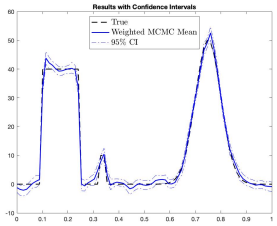
(c) σ = 0.50, SNR = 32.20 dB



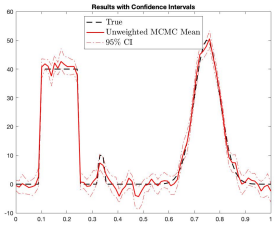
(d) σ = 0.50, SNR = 32.20 dB



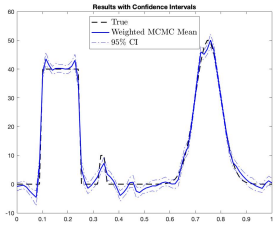
(e) σ = 0.75, SNR = 27.43 dB



(f) σ = 0.75, SNR = 27.43 dB



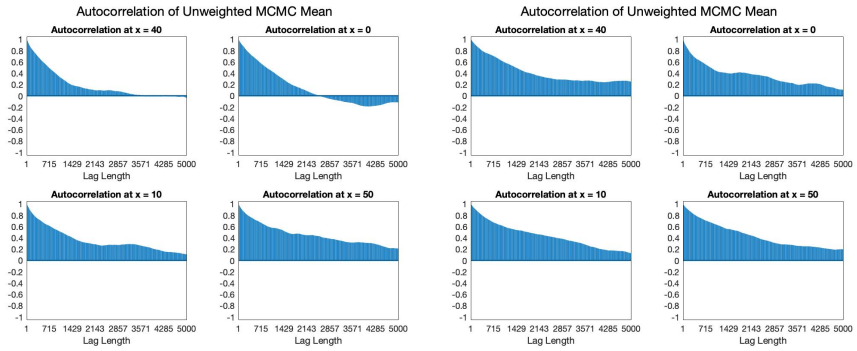
(g) σ = 1.00, SNR = 25.13 dB



(h) σ = 1.00, SNR = 25.13 dB

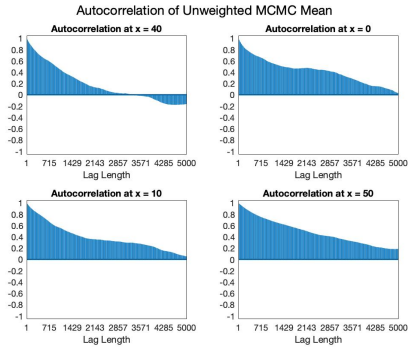
Figure 4.4: Unweighted MCMC (left) and weighted MCMC (right) with the 95% their respective confidence intervals

4.1.2 Convergence Analysis of MCMC

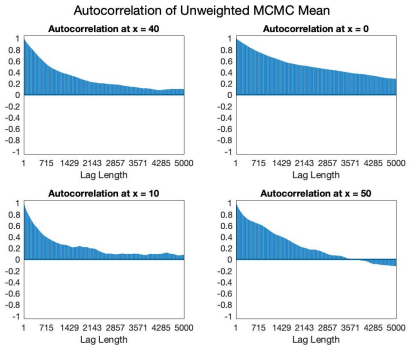


(a) σ = 0.25, SNR = 36.75 dB

(b) σ = 0.50, SNR = 32.20 dB



(c) σ = 0.75, SNR = 27.43 dB



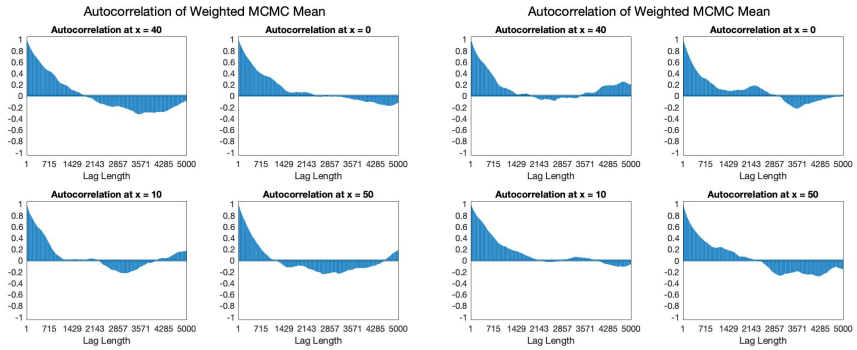
(d) σ = 1.00, SNR = 25.13 dB

Figure 4.5: Auto-correlation for unweighted MCMC.

In Figure 4.5, Figure 4.6, Figure 4.7, and Figure 4.8, different entries of the recovered $x \in \mathbb{R}^{80}$ were chosen for each plot. A plot label of $x = v \in \mathbb{R}$ represents that the plot corresponds to the mean value of the entries that represent region $h^{inv}(v)$ in the domain where the function h is the true function (3.2).

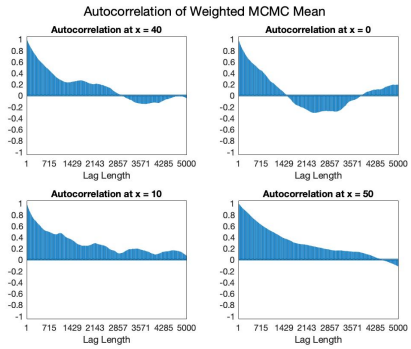
Figure 4.5 plots the auto-correlation of the unweighted MCMC against the lag described in Section 2.6, while Figure 4.6 does the same plot for the weighted MCMC. In all chosen regions of the domain, the auto-correlation of the weighted MCMC is clearly smaller in magnitude as the lag increases. Note in particular that the auto-correlation decays to zero and begins oscillating around

zero much earlier for the weighted MCMC, indicating that the weighted MCMC may converges to the respective posterior probability density earlier as well. This will also be the topic of future investigations, as it may be possible to develop more efficient and adaptive algorithms that take advantage of this apparent faster convergence in smoother regions.

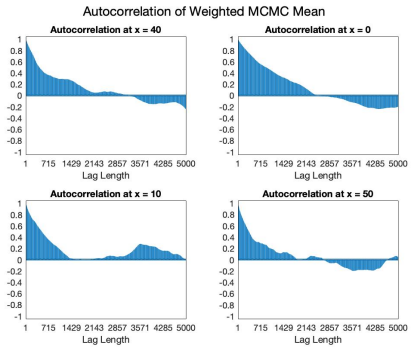


(a) σ = 0.25, SNR = 36.75 dB

(b) σ = 0.50, SNR = 32.20 dB



(c) σ = 0.75, SNR = 27.43 dB



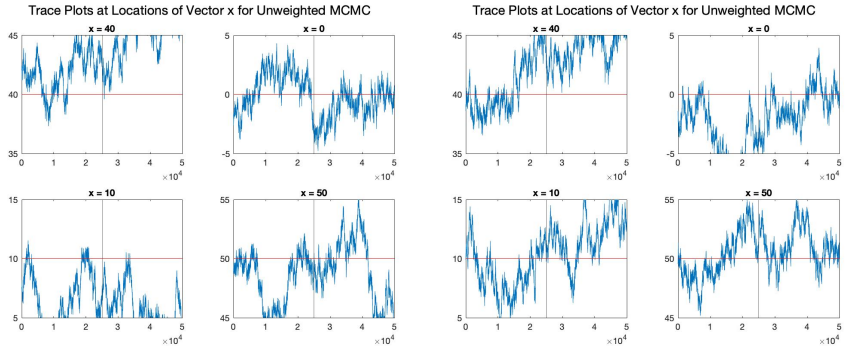
(d) σ = 1.00, SNR = 25.13 dB

Figure 4.6: Auto-correlation for weighted MCMC

Figures 4.7 and 4.8 display trace plots at chosen regions of the domain. One indication that a Markov chain is converging well to the posterior is that its corresponding trace plots appear to contain values sampled from independent, identical distributions. Similarly, an indication that the posterior probability density provides a good recovery to the true function is that the the points in the trace plots hover around the true value [15].

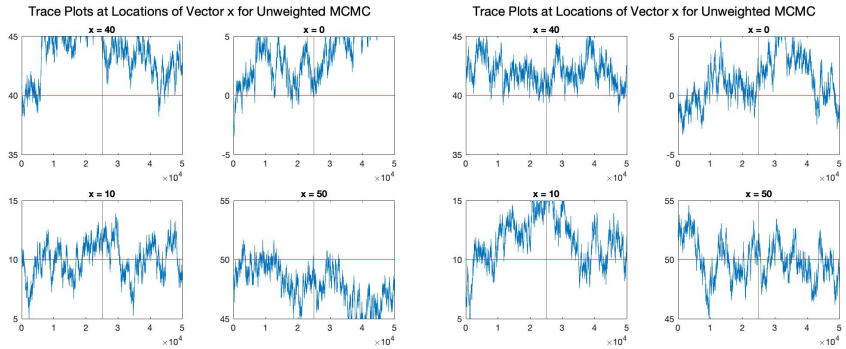
Observe that the trace plots of the unweighted MCMC appear to have much greater correlation between neighboring samples. By contrast the trace plots for the weighted MCMC exhibit less

correlation, in particular in the smooth regions of the domain. It is also evident that the trace plots of the weighted MCMC yield much closer values to the true function (given by the red line on each graph). This result corroborates both the MCMC mean plots and the auto-correlation plots.



(a) σ = 0.25, SNR = 36.75 dB

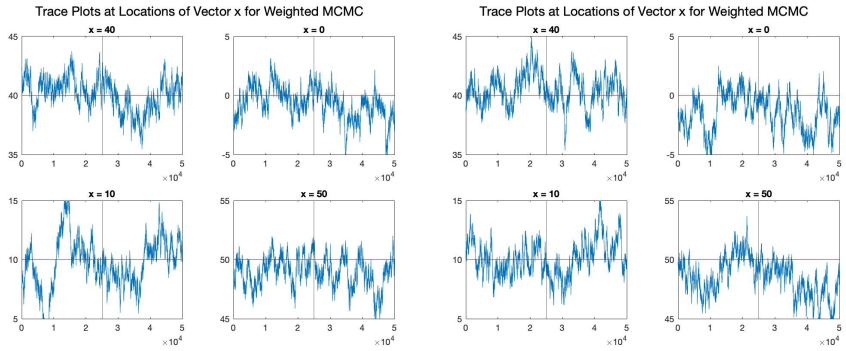
(b) σ = 0.50, SNR = 32.20 dB



(c) σ = 0.75, SNR = 27.43 dB

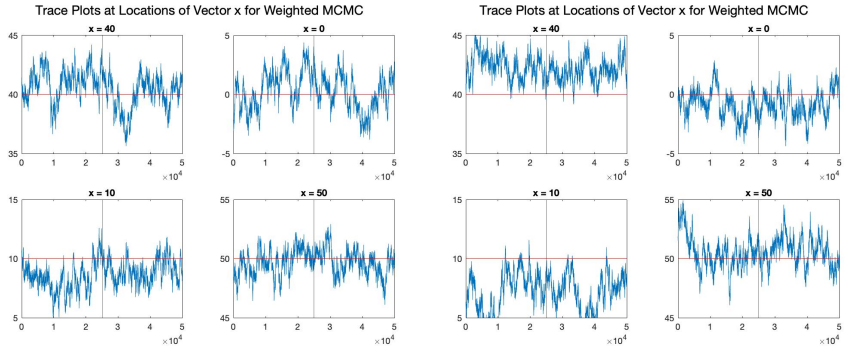
(d) σ = 1.00, SNR = 25.13 dB

Figure 4.7: Trace plots with unweighted MCMC



(a) σ = 0.25, SNR = 36.75 dB

(b) σ = 0.50, SNR = 32.20 dB

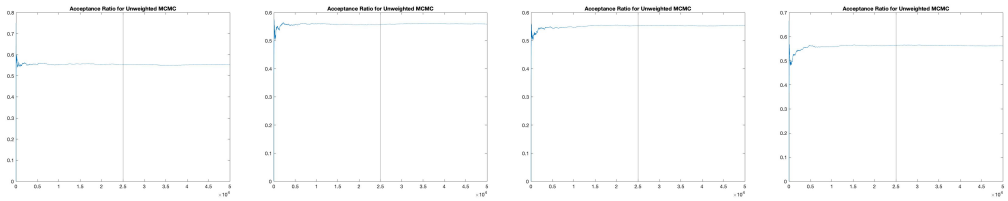


(c) σ = 0.75, SNR = 27.43 dB

(d) σ = 1.00, SNR = 25.13 dB

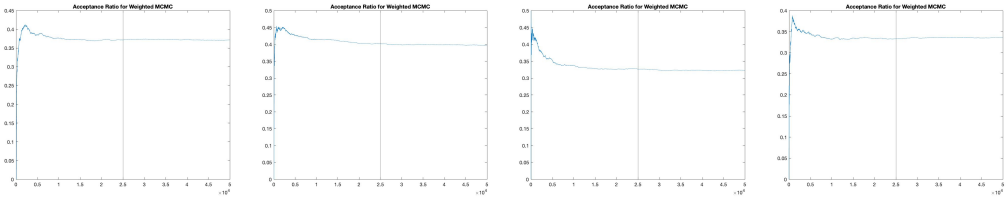
Figure 4.8: Trace plots with weighted MCMC

Lastly, Figures 4.9 and 4.10 show the acceptance rates of the unweighted MCMC and the weighted MCMC. Both acceptance rates have little variation and appear to hover around some constant. However, the acceptance rates for all selected regions of the weighted MCMC is lower. In Section 2.5, it was posited that a one-dimensional problem with a Guassian proposal distribution should ideally obtain an acceptance rate of about $\frac{1}{2}$ [27]. This benchmark, however, is far from conclusive. Factors such as the posterior probability density and proposal distribution's may influence the acceptance rate.



(a) $\sigma = 0.25$, SNR = 36.75 dB (b) $\sigma = 0.50$, SNR = 32.20 dB (c) $\sigma = 0.75$, SNR = 27.43 dB (d) $\sigma = 1.00$, SNR = 25.13 dB

Figure 4.9: Acceptance ratio of unweighted MCMC



(a) $\sigma = 0.25$, SNR = 36.75 dB (b) $\sigma = 0.50$, SNR = 32.20 dB (c) $\sigma = 0.75$, SNR = 27.43 dB (d) $\sigma = 1.00$, SNR = 25.13 dB

Figure 4.10: Acceptance ratio of weighted MCMC

4.2 Discussion

The numerical results demonstrate that adding the spatially varying weights to the prior in (3.6) to form (3.8) not only improve the recovery of the unknown function but also increased the confidence of the recovery. Based on these results, we would anticipate similar improvements in multi-dimensions.

In particular we note that in Figure 4.3, the recovery using the weighted is noticeably better than that from the unweighted version, and that this is true for *all* noise levels that were tested. The improvement is the most pronounced in the flat regions, corresponding to 0 values in the edge domain [3]. This is exactly what the weighted MCMC is intended to do – in smooth regions the posterior has a larger penalty on the prior since we expect it is a more reliable source than the data.

In addition to determining if the new method is able to improve function recovery, that is, the point estimate reconstruction, another important metric to measure is the variance of the resulting MCMC chain of the recovered posterior probability density. The variance of the MCMC chain provides insight into the rate of convergence as well as (of course) the variance of the posterior probability density itself. From Figure 4.4 it is evident that the weighted MCMC chain yields lower variance than the unweighted MCMC chain does. The 95% confidence interval constructed for the weighted MCMC chain appears to be much tighter than that of the unweighted MCMC at all noise levels. This difference is especially pronounced in the flat regions where the values are zero in the edge domain. This is particularly significant since a main reason practitioners choose to perform MCMC rather than use a traditional MAP estimate for solving inverse problem is so that they are able to quantify uncertainty. This is especially important when the the solution must be later used for inference or other downstream processing. Hence improving the confidence of the recovered solution should be of significant interest to a wide range of scientific applications.

This improvement is further corroborated by the convergence analysis. Comparing the auto-correlation plots in Figures 4.5 and 4.6, it is apparent that the auto-correlation approaches 0 at a relatively much faster rate for the weighted MCMC method. This suggests that the weights allow the MCMC to converge to the posterior probability density faster and has potentially important computational efficiency consequences. For example, a tolerable auto-correlation can be incorporated into an algorithm telling a practitioner when the solution is effectively recovered and in what region, so more resources can be spent interrogating other more critical regions of interest. According to Figures 4.7 and 4.8, the trace plots of the weighted MCMC appear to have noticeably lower correlation between the states. Ideally, the trace plots should appear to have each point sampled from identical, independent distributions (i.i.d.) centered around the mean, which suggests that the solution is "well-mixed" [15]. Thus, these plots provide another piece of evidence that insinuates the weighted MCMC method yields better convergence properties.

Chapter 5

Conclusion and Future Work

This thesis developed a method to ascertain the unknown recovery and uncertainty quantification of a one-dimensional inverse problem through the Bayesian framework, specifically by using the Metropolis-Hastings Algorithm. However instead of using the standard (unweighted) sparsity prior to create the posterior probability density, a new weighted sparsity prior was generated using the variance based joint sparsity (VBJS) method. The proposed weights are intended to variably penalize distinct regions of the domain depending on the variance of the unknown samples in the edge domain as described in Section 2.2. By imposing these weights on the penalty in the edge domain, this thesis demonstrated that marked improvements can be made to the convergence properties of MCMC for a one-dimensional inverse problem. In turn, this lead to improved computational efficiency, as fewer iterations are needed. The new method also yielded tighter confidence intervals constructed from the Markov chains constructed by the Metropolis-Hastings algorithm.

Our new method is promising and many new research ideas should be explored to further improve function (or image) recovery as well as provide more information about the uncertainty quantification of the inverse model. For instance, a similar methodology could be applied to problems of higher dimensions, in which ensuring the convergence of the MCMC is traditionally very difficult. Adding weights to the prior may allow better convergence to the posterior probability

density, which is crucial in problems of higher dimensions.

Another idea that should be explored is using first-moment information such as expected value to improve the construction of the weights. In this thesis, the variance among different MAP estimates is used to construct the weights. However, it should also be the case that zero regions in the edge domain should have an expected value near zero. Thus a non-zero mean of a region in the edge domain may indicate an edge while a near-zero mean may indicate that the variance in that region stems from noise.

Moreover, the use of the ℓ_2 norm for the prior should be experimented with in the weighted posterior probability density. The reasoning provided in Chapter 3 for choosing the ℓ_1 norm is to enforce sparsity in the edge domain. However, if the weights are chosen so that they sufficiently enforce sparsity, it should be acceptable to use the ℓ_2 norm, which typically yields more computationally efficient methods. This may also provide further insights into the unknown function.

Finally, it is important to try this method on real-world problems. As noted previously there are many applications for which this methodology would be useful. For example, in synthetic aperture radar (SAR) imaging, multiple measurements are collected in the phase history domain (essentially treated as Fourier data). This would yield a different transform matrix A in the model. Photoacoustic and ultrasound imaging provide other interesting applications. In all cases, it is often useful to know more than a point estimate, and specifically to be able to obtain a solution distribution. All of these topics will be explored in future work.

Bibliography

[1] *Conjugate priors: Beta and normal*. https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT$18_05S14_Reading15a.pdf$.

[2] *Mathworks help center*. `https://www.mathworks.com/help/signal/ref/snr.html#bt7c55c-1`. Accessed: 2020-05-16.

[3] B. ADCOCK, A. GELB, G. SONG, AND Y. SUI, *Joint sparse recovery based on variances*, SIAM Journal on Scientific Computing, 41 (2019), pp. A246–A268.

[4] R. ARCHIBALD, A. GELB, AND R. PLATTE, *Image reconstruction from undersampled fourier data using the polynomial annihilation transform*, Journal of Scientific Computing, (2015).

[5] R. ARCHIBALD, A. GELB, AND J. YOON, *Polynomial fitting for edge detection in irregularly sampled signals and images*, SIAM Journal on Numerical Analysis, 43 (2005), pp. 259–279.

[6] J. M. BARDSLEY, *Mcmc-based image reconstruction with uncertainty quantification*, SIAM Journal on Scientific Computing, 34 (2012), pp. A1316–A1332.

[7] J. M. BARDSLEY AND T. CUI, *A Metropolis-Hastings-Within-Gibbs Sampler for Nonlinear Hierarchical-Bayesian Inverse Problems*, Springer International Publishing, Cham, 2019, pp. 3–12.

[8] P. BILLINGSLEY, *Probability and measure*, Wiley series in probability and mathematical statistics, Wiley, 1986.

[9] S. BOYD, S. BOYD, L. VANDENBERGHE, AND C. U. PRESS, *Convex Optimization*, no. pt. 1 in Berichte über verteilte messysteme, Cambridge University Press, 2004.

[10] E. J. CANDES, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information*, IEEE Transactions on Information Theory, 52 (2006), pp. 489–509.

[11] C. CHATFIELD AND H. XING, *The Analysis of Time Series: An Introduction with R*, Chapman & Hall/CRC Texts in Statistical Science, CRC Press, 2019.

[12] Y. CHEN, Y. YE, AND M. WANG, *Approximation hardness for a class of sparse optimization problems*, Journal of Machine Learning Research, 20 (2019), pp. 1–27.

[13] S. F. COTTER, B. D. RAO, KJERSTI ENGAN, AND K. KREUTZ-DELGADO, *Sparse solutions to linear inverse problems with multiple measurement vectors*, IEEE Transactions on Signal Processing, 53 (2005), pp. 2477–2488.

[14] R. DURRETT, *Essentials of Stochastic Processes*, Springer Texts in Statistics, Springer New York, 2012.

[15] J. ELLIS, *A Practical Guide to MCMC Part 1: MCMC Basics*, 2018 (accessed May 7, 2020).

[16] G. FOLLAND, *Real Analysis: Modern Techniques and Their Applications*, Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts, Wiley, 1999.

[17] D. GAMERMAN AND H. F. LOPES, *Morkov chain monte carlo*, 68 (2006). MCMC.

[18] A. GELB AND T. SCARNATI, *Reducing effects of bad data using variance based joint sparsity recovery*, Journal of Scientific Computing, 78 (2019), pp. 94–120.

[19] A. E. GELFAND AND A. F. M. SMITH, *Sampling-based approaches to calculating marginal densities*, Journal of the American Statistical Association, 85 (1990), pp. 398–409.

[20] C. GRINSTEAD AND J. SNELL, *Grinstead and Snell's Introduction to Probability*, Titolo collana, University Press of Florida, 2009.

[21] P. HANSEN, J. NAGY, AND D. O'LEARY, *Deblurring Images: Matrices, Spectra, and Filtering*, Fundamentals of Algorithms, SIAM, Society for Industrial and Applied Mathematics, 2006.

[22] W. K. HASTINGS, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.

[23] S. JI, Y. XUE, AND L. CARIN, *Bayesian compressive sensing*, IEEE Transactions on Signal Processing, 56 (2008), pp. 2346–2356.

[24] J. KAIPIO AND E. SOMERSALO, *Statistical and Computational Inverse Problems*, Applied Mathematical Sciences, Springer New York, 2006.

[25] N. METROPOLIS, A. W. ROSENBLUTH, M. N. ROSENBLUTH, A. H. TELLER, AND E. TELLER, *Equation of state calculations by fast computing machines*, The Journal of Chemical Physics, 21 (1953), pp. 1087–1092.

[26] C. PRICE, *Autocorrelation function (acf)*, 2020. The MathWorks, Natick, MA, USA.

[27] C. ROBERT AND G. CASELLA, *Introducing Monte Carlo Methods with R*, Use R!, Springer, 2010.

[28] C. ROBERT AND G. CASELLA, *A short history of markov chain monte carlo: Subjective recollections from incomplete data*, Statist. Sci., 26 (2011), pp. 102–115.

[29] C. P. ROBERT, *Discussion: Markov chains for exploring posterior distributions*, The Annals of Statistics, 22 (1994), pp. 1742–1747.

[30] W. RUDIN, W. RUDIN, AND T. M.-H. P. COMPANY, *Real and Complex Analysis*, Higher Mathematics Series, McGraw-Hill Education, 1987.

[31] H. RUE AND L. HELD, *Gaussian Markov Random Fields: Theory and Applications*, Chapman & Hall/CRC Monographs on Statistics & Applied Probability, CRC Press, 2005.

[32] T. SCARNATI, *Inverse problems in a bayesian framework*, December 2019.

[33] Z. WANG, J. M. BARDSLEY, A. SOLONEN, T. CUI, AND Y. M. MARZOUK, *Bayesian inverse problems with ℓ_1 priors: A randomize-then-optimize approach*, SIAM Journal on Scientific Computing, 39 (2017), pp. S140–S166.