Summer 7-13-2021

Optimal Transport Driven Bayesian Inversion with Application to Signal Processing

Elijah F. Perez
University of New Mexico

Elijah Perez

Candidate

Mathematics and Statistics

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Dr. Mohammad Motamed , Chairperson

Dr. Daniel Appelo

Dr. Gabriel Huerta

Dr. Stephen Lau

Optimal Transport Driven Bayesian Inversion with Application to Signal Processing

by

Elijah Perez

B.S., Mathemtics, University of New Mexico, 2019

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Mathematics

The University of New Mexico

Albuquerque, New Mexico

July, 2021

Dedication

To my parents, Phillip and Martha, and my wife, Jessica. For their love and support.

"If I have seen further it is by standing on the shoulders of Giants." – Isaac Newton

Optimal Transport Driven Bayesian Inversion with Application to Signal Processing

by

Elijah Perez

B.S., Mathemtics, University of New Mexico, 2019

M.S., Mathematics, University of New Mexico, 2021

Abstract

This paper will outline a Debiased Sinkhorn Divergence driven Bayesian inversion framework. Conventionally, a Gaussian Driven Bayesian framework is used when performing Bayesian inversion. A major issue with this Gaussian framework is that the Gaussian likelihood, driven by the L_2 norm, is not affected by phase shift in a given signal. This issue has been addressed in [1] using a Wasserstein framework. However, the Wasserstein framework still has an issue because it assumes statistical independence when multidimensional signals are analyzed. This assumption of statistical independence cannot always be made when analyzing signals where multiple detectors are recording one event, say from a seismic event. The Wasserstein metric can be generalized to multidimensional signals, but implementation of the multidimensional Wasserstein metric is very computationally expensive. This means that it is unreasonable for Bayesian inversion. Debiased Sinkhorn Divergence offers an alternative to the multidimensional Wasserstein metric while remaining relatively cheap computationally. This allows for the creation of a Debiased Sinkhorn Divergence driven Bayesian framework that will be formulated and analyzed in this paper.

Contents

List of Figures

Chapter 1

Introduction

In most applications of Bayesian inversion, a Gaussian likelihood (sometimes called Normal likelihood) function is used to formulate a Bayesian inversion framework. The Gaussian likelihood implements the L_2 norm. This likelihood function is a standard choice in certain specific applications, but fails to account for phase differences in a given signal. Because of this, the Gaussian likelihood can produce many false optima that a Bayesian inversion algorithm can become trapped in, producing an incorrect posterior distribution. Through the work in [1] we can see that a way to avoid this issue all together is to use a different likelihood, namely a quadratic Wasserstein quasi-likelihood (loss function). The advantage that this quasi-likelihood has is that the Wasserstein metric not only measures the difference in amplitude of two signals, but also the difference in phase. This means that depending on the application, the Wasserstein quasi-likelihood can produce a better posterior compared to the Gaussian likelihood. This idea is explored in detail in [1]. The Wasserstein metric is used to create the Wasserstein quasi-likelihood and in [1], the one-dimensional Wasserstein metric is used. This is because implementation of a multidimensional Wasserstein metric would be very computationally expensive. So, instead of implementing the multidimensional Wasserstein metric, [1] uses the one-dimensional Wasserstein met-

ric and makes some assumptions on the signal being analyzed. The main assumption is that the signals being analyzed are statistically independent. This assumption is made so that we can assume that the quasi-likelihood of the signals is a product of the quasi-likelihoods of the one-dimensional signals. This assumption allows the one-dimensional Wasserstein quasi-likelihood to be applied to multidimensional signals. However, this assumption of statistical independence cannot always be made for a given set of signals. This is where Debiased Sinkhorn Divergence has a major advantage. Sinkhorn Divergence can be viewed as an entropically regularized Wasserstein distance[2][17], allowing for the creation of a Debiased Sinkhorn Divergence driven Bayesian framework. This Sinkhorn framework has the advantage that it does not require the assumption of statistical independence to work. Sinkhorn has an advantage over the multidimensional Wasserstein metric in that it is far less computationally expensive [18]. Because Sinkhorn Divergence costs less computationally than the multidimensional Wasserstein metric, it is a viable option in a Bayesian framework. Also, Sinkhorn Divergence does not need the statistical independence assumption, so it can be a better choice when analyzing a multidimensional signal compared to the one-dimensional Wasserstein metric. In this paper a Sinkhorn Divergence Bayesian framework is created by modifying the Wasserstein framework from [1], using a Markov Chain Monte Carlo method. Specifically, Metropolis-Hastings within Gibbs sampling algorithm will be used to formulate a numerical algorithm (see chapter 5 for details).

The rest of the paper will explore this new Sinkhorn-Bayesian framework. Chapter 2 will outline the general Bayesian inversion Problem. Chapter 3 will explore how to derive Sinkhorn divergence from the general optimal transport problem. Chapter 4 will outline quasi-likelihood structures based on Sinkhorn and Wasserstein. Chapter 5 will describe a numerical method using the new Sinkhorn quasi-likelihood. Chapter 6 will explore examples implementing this new Sinkhorn-Bayesian framework.

Chapter 2

Bayesian Inversion

2.1 Bayes' Theorem

In many stochastic processes, we are tasked with finding the conditional probability of the model parameter given an observed quantity. This is done through Bayes' Theorem which allows us to calculate the conditional probability of an event occurring. In Bayesian inversion, we are using Bayes' Theorem to calculate the conditional probability of the model parameter vector $\boldsymbol{\theta} = (\theta_1, ..., \theta_m) \in \Theta \subset \mathbb{R}^m$. Let $\boldsymbol{g} = (g_1, \ldots, g_n) \in \mathbb{R}^n$ be vector of n observed quantities and let $\boldsymbol{f} = (f_1, \ldots, f_n) \in \mathbb{R}^n$ be a vector of n predicted quantities created by a forward predictive model depending on the parameter vector $\boldsymbol{\theta}$. That is:

$$\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{\theta}) : \Theta \subset \mathbb{R}^m \to \mathbb{R}^n$$

Now, applying Bayes' Theorem to solve for the conditional probability of $\boldsymbol{\theta}$ given $\boldsymbol{g}$, written as $\pi(\boldsymbol{\theta}|\boldsymbol{g})$, we have [3]:

$$\pi(\boldsymbol{\theta}|\boldsymbol{g}) = \frac{\pi(\boldsymbol{g}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int_\Theta \pi(\boldsymbol{g}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{1}$$

Where $\pi(\boldsymbol{\theta}|\boldsymbol{g})$ is the posterior, $\pi(\boldsymbol{g}|\boldsymbol{\theta})$ is the likelihood, and $\pi(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$. Note that sometimes $\int_\Theta \pi(\boldsymbol{g}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$ is written as $\pi(\boldsymbol{g})$. Since $\pi(\boldsymbol{g})$ is independent of $\boldsymbol{\theta}$, it can be viewed as a scaling constant to ensure that the posterior obtained from (1) is consistent with the definition of a probability density function (i.e. $\int_\Theta \pi(\boldsymbol{\theta}|\boldsymbol{g})d\theta = 1$ and is positive for $\forall \boldsymbol{\theta} \in \Theta$). This means that $\pi(\boldsymbol{\theta}|\boldsymbol{g}) \propto \pi(\boldsymbol{g}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$, and that the proportionality constant is precisely $\frac{1}{\pi(\boldsymbol{g})}$. This now leaves the task of finding distributions for the prior and likelihood. The prior, $\pi(\boldsymbol{\theta})$, is a distribution that we obtain from prior knowledge about the model parameter $\boldsymbol{\theta}$. As an example, if we know that $\boldsymbol{\theta} \in (1, 3)$ then one choice for a prior could be $\boldsymbol{\theta} \sim Uniform(1, 3)$.

Now, if we have a quasi-likelihood instead of a true likelihood, we call the resulting posterior distribution a quasi-posterior or Gibbs posterior. This Gibbs posterior still gives accurate information on the probability of $\boldsymbol{\theta}$ given $\boldsymbol{g}$, but is defined using a quasi-likelihood and thus cannot be called a true posterior [22]. These quasi-likelihoods are defined by loss functions and still give information about goodness of fit, and can be a more realistic choice in Bayesian processes because the true underlying likelihood function may be unknown. This means that the Gibbs posterior results in near identical results to the true posterior, while offering a more general framework that works for applications for which the true likelihood function is unknown [22] (see section 6.2 and Figure 6.2).

2.2 Likelihood Structure

The choice of likelihood is a fundamental step in Bayesian inversion and is one of the main aspects of Bayesian inversion that can be changed. Often, the choice of a likelihood function is based on the noise structure that the analyzed signal is expected

to have. Under the assumption that we have a simple additive noise structure, the Gaussian likelihood is the most common choice. Assuming that we have measurement noise $(\epsilon_1, \ldots, \epsilon_n)$ that appears in the measured quantities $(g_1, \ldots, g_n)$ the additive noise is assumed to be normally distributed with mean zero and standard deviation σ. That is to say:

$$g_i = f_i(\boldsymbol{\theta}) + \epsilon_i, \qquad \epsilon_i \sim Normal(0, \sigma), \qquad i = 1, \ldots, n$$

Where n is the number of measured quantities. This noise structure can be handled well by the Gaussian likelihood [1]:

$$L_{norm}(\boldsymbol{\theta}) = \pi_{norm}(\boldsymbol{g}|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{n/2}\sigma^n}\exp(\tfrac{-1}{2\sigma^2}\textstyle\sum_{i=1}^{n}|g_i - f_i(\boldsymbol{\theta})|^2) \tag{2}$$

Such a likelihood structure also requires an assumption of statistical independence for the signals being analyzed. This is seen in the fact that the overall likelihood, $\pi(\boldsymbol{g}|\boldsymbol{\theta})$ is equal to the product of the individual likelihoods of g_i [1]:

$$L(\boldsymbol{\theta}) = \pi(\boldsymbol{g}|\boldsymbol{\theta}) = \textstyle\prod_{i=1}^{n}\pi(g_i|\boldsymbol{\theta})$$

This illuminates two major issues with the Gaussian likelihood. One is that the noise structure may not be realistic because of its simplicity. For example, if we have a set of two dimensional signals $\boldsymbol{g}(x,t)$ with an additive Gaussian noise structure

$$g(x_i, t_j) = f(x_i, t_j; \boldsymbol{\theta}) + \epsilon_{i,j}, \qquad \epsilon_{i,j} \sim Norm(0, \sigma)$$

we can easily show that the Gaussian Likelihood fails to predict the correct values of $\boldsymbol{\theta}$ for certain applications that contain this noise structure (see chapter 6 for numerical examples). Another issue with the Gaussian likelihood is that the assumption of statistical independence may not be mathematically consistent, again depending on the application. Statistical independence exists when the probability of two

events occurring is equal to the product of the individual probabilities of each event occurring independently [4]:

$$P(A \bigcap B) = P(A)P(B) \tag{3}$$

Another way to say this is that statistical independence exists when the occurrence of one event does not affect the probability of the other event occurring. Many applications in stochastic processes do not have statistical independence, leading to a desire to create a Bayesian framework that does not assume statistical independence and that can handle more complicated noise structures.

Chapter 3

Concepts from Optimal Transport

In this section, three concepts from optimal transport will be explored. These concepts are the key to creating a new Bayesian inversion framework that satisfies the need to have a framework that does not assume statistical independence and handles complicated noise structure well. First, the quadratic Wasserstein metric will be explored. Then, Sinkhorn Divergence and Debiased Sinkhorn Divergence (DSD) will be explored and will be shown to be a regularized Wasserstein distance. DSD will then be used in the next section to create a new likelihood function that can be used in Bayesian inversion.

One of the main desirable traits for a likelihood function used in Bayesian inversion is convexity for the type of problems the framework is applied to. This is because a convex function has a more well-defined minimum value compared to a non-convex function, allowing the Bayesian inversion algorithm to converge to the correct minimum and not fall into a false minimum. The likelihood functions in this paper that employ the Wasserstein metric and the DSD have this convexity property with respect to the phase shift, phase dilation, and amplitude change in the simulated and measured signals [1]. This is the motivation behind using the Wasserstein metric and DSD in a quasi-likelihood function.

3.1 Kantorovich Formulation of Optimal Transport

Suppose we have probability vectors $\boldsymbol{f}, \boldsymbol{g} \in \mathbb{R}_+^n$ defined on sets of n points $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \boldsymbol{X} \subset \mathbb{R}^d$ and $\{\boldsymbol{y}_1, ..., \boldsymbol{y}_n\} \subset \boldsymbol{Y} \subset \mathbb{R}^d$ respectively. Define $U(\boldsymbol{f}, \boldsymbol{g})$ to be the transport polytype of $\boldsymbol{f}$ and $\boldsymbol{g}$ [2]:

$$U(\boldsymbol{f}, \boldsymbol{g}) = \{P \in \mathbb{R}_+^{n \times n} : P\mathbf{1}_{\boldsymbol{n}} = \boldsymbol{f}, P^\top \mathbf{1}_{\boldsymbol{n}} = \boldsymbol{g}\} \tag{4}$$

where $\mathbf{1}_{\boldsymbol{n}}$ is a vector of length n with all entries equal to 1, and each matrix $P = [P_{ij}] \in \mathbb{R}_+^{n \times n}$ in $U(\boldsymbol{f}, \boldsymbol{g})$ is a transport matrix that encodes a transport plan. Each element P_{ij} describes the amount of mass transported from point $\boldsymbol{x}_i \in \boldsymbol{X}$ to point $\boldsymbol{y}_j \in \boldsymbol{Y}$ where $i, j = 1, ..., n$. Now, let $c : \boldsymbol{X} \times \boldsymbol{Y} \to \mathbb{R}_+$ be a non-negative cost function on $\boldsymbol{X} \times \boldsymbol{Y}$ so that $\forall(\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{X} \times \boldsymbol{Y}$, $c(\boldsymbol{x}, \boldsymbol{y})$ represents the cost of transporting one unit of mass from $\boldsymbol{x} \in \boldsymbol{X}$ to a target point $\boldsymbol{y} \in \boldsymbol{Y}$. This allows for a cost matrix, C, to be defined as:

$$C = [C_{ij}] \in \mathbb{R}_+^{n \times n}, \quad C_{ij} = c(\boldsymbol{x}_i, \boldsymbol{y}_j), \quad i, j = 1, ..., n. \tag{5}$$

The optimal transport problem can now be viewed as minimizing the Forbenius inner product of P and C. That is to say the transport cost T_C is:

$$T_C(\boldsymbol{f}, \boldsymbol{g}) = \min_{P \in U(\boldsymbol{f}, \boldsymbol{g})} \langle P, C \rangle \tag{6}$$

where $\langle P, C \rangle$ is the Forbenius inner product $\langle P, C \rangle = \sum_{i,j} P_{ij} C_{ij}$. One key assumption when formulating the Wasserstein metric and Sinkhorn divergence is that the cost matrix C is defined for a distance function [12]. That is to say for a distance function d:

$$C = [C_{ij}] \in \mathbb{R}_+^{n \times n}, \quad C_{ij} = d(\boldsymbol{x}_i, \boldsymbol{y}_j)^p, \quad i, j = 1, ..., n, \, p \in [1, \infty) \tag{8}$$

This now allows us to define the Wasserstein metric of order p induced by the

optimal cost T_C [12][13][14][15]

$$W_p(\boldsymbol{f}, \boldsymbol{g}) = (T_C(\boldsymbol{f}, \boldsymbol{g}))^{1/p} \tag{9}$$

Note that the Wasserstein metric discussed in section 3.2 is a discrete one-dimensional squared Wasserstein metric with $p = 2$.

3.2 Wasserstein Metric

The Wasserstein distance in this section is a one-dimensional squared Wasserstein distance of order $p = 2$. This metric is simple compared to a higher dimensional Wasserstein metric, and thus is suitable for a Bayesian framework. Because of computational cost, it is infeasible to use a higher dimensional Wasserstein metric as a basis for a Bayesian framework.

The Wasserstein metric is a distance function defined by the minimization of the cost of turning one probability distribution into the other [1]. Suppose we have two discrete time signals $\boldsymbol{f}, \boldsymbol{g} \in \mathbb{R}^n$ with discrete time steps $(t_1, \ldots, t_n)$. The two signals $\boldsymbol{f}$ and $\boldsymbol{g}$ need some preliminary altering before they can be implemented into the Wasserstein metric. Since the Wasserstein metric is a measure of the distance between two probability distributions, we need to alter the signals since they are likely not probability distributions. We need to ensure that the signals are always non-negative and that the $\sum_{i=1}^{n} f_i = 1$ and $\sum_{i=1}^{n} g_i = 1$ to remain consistent with the definition of a probability mass function (PMF). This can be done several different ways, but the one that will be used in this paper is to shift the signals by some constant, and then normalize.

First, choose a constant value c such that $c > min(\boldsymbol{g}, \boldsymbol{f})$. This ensures that $f_i + c > 0$ and $g_i + c > 0$ for all $i = 1, \ldots, n$. Next, normalize the two signals which creates two new signals that are now in the form of a probability distribution [7][8]:

$$\overline{\boldsymbol{f}}=\frac{(f_i+c)}{\sum(f_i+c)} \ , \qquad \overline{\boldsymbol{g}}=\frac{(g_i+c)}{\sum(g_i+c)} \tag{10}$$

Now we have two PMFs which can be used to create two discrete cumulative density functions (CDFs).

$$F_i = \sum_{k=1}^{i} \overline{f}_k, \qquad G_i = \sum_{k=1}^{i} \overline{g}_k, \qquad i = 1, .., n \tag{11}$$

where $\overline{f}_k$ and $\overline{g}_k$ are the k^{th} component of their respective functions. This allows us to define the discrete quadratic Wasserstein distance between two signals, $\boldsymbol{f}$ and $\boldsymbol{g}$ [12][13][14][15]:

$$d_W(\boldsymbol{f}, \boldsymbol{g}) = W_2^2(\boldsymbol{f}, \boldsymbol{g}) \approx \sum_{i=1}^{n} |t_i - T_i|^2 \overline{f}_i, \qquad T = G^{-1} \circ F \tag{12}$$

Where $T = G^{-1} \circ F$ is the optimal map from $\overline{\boldsymbol{f}}$ to $\overline{\boldsymbol{g}}$ [1]. Note that this is a formulation for the single dimensional discrete Wasserstein metric. Applications of this single dimensional metric in multidimensional inversion problems will be addressed in chapter 4.

3.3 Sinkhorn Divergence

Suppose that the transport problem is now regularized by adding an entropic penalty term to the total transport cost [2]:

$$T_C^{\lambda}(\boldsymbol{f}, \boldsymbol{g}) = \min_{P \in U(\boldsymbol{f}, \boldsymbol{g})} \langle P, C \rangle - \tfrac{1}{\lambda} H(P), \tag{13}$$

where $\lambda > 0$ is a regularization parameter and $H(P)$ is the discrete entropy of the transport matrix [16],

$$H(P) = -\sum_{i,j} P_{ij}(\log P_{ij} - 1) \tag{14}$$

This regularized problem with $C \in \boldsymbol{M}^{n \times n}$ has a unique solution, P_λ. As λ

increases, $T_C^\lambda \to T_C$ because P_λ approaches the solution with maximum entropy within the set of all optimal solutions of the original Kantorovich's [17]. Sinkhorn divergence of order p between $\boldsymbol{f}$ and $\boldsymbol{g}$ is defined as [2]:

$$S_{p,\lambda}(\boldsymbol{f}, \boldsymbol{g}) = \langle P_\lambda, C \rangle^{1/p}. \tag{15}$$

Sinkhorn Divergence can be viewed as a regularized Wasserstein distance since P_λ is the solution to the regularized Kantorovich problem [2]. Since $P_\lambda \in U(\boldsymbol{f}, \boldsymbol{g})$ and $T_C(\boldsymbol{f}, \boldsymbol{g}) = \min_{P \in U(\boldsymbol{f}, \boldsymbol{g})} \langle P, C \rangle$, it is straightforward to see that [2]

$$S_{p,\lambda}(\boldsymbol{f}, \boldsymbol{g}) \geq W_p(\boldsymbol{f}, \boldsymbol{g}) \tag{16}$$

Unlike the Wasserstein metric, Sinkhorn divergence does not fully satisfy the definition of a metric [16]. However, with some careful manipulation we can create a metric using Sinkhorn Divergence called Debiased Sinkhorn Divergence (DSD) of order p as

$$d_{dsd,p}(\boldsymbol{f}, \boldsymbol{g}) = |S_{p,\lambda}(\boldsymbol{f}, \boldsymbol{g}) - \tfrac{1}{2}[S_{p,\lambda}(\boldsymbol{f}, \boldsymbol{f}) + S_{p,\lambda}(\boldsymbol{g}, \boldsymbol{g})]|. \tag{17}$$

This means that $d_{dsd,2}^2$ (from here on simply called d_{dsd}) can be used instead of W_2^2 as a measure of dissimilarity while still being a distance function. Note that in practice, λ is not chosen to be very large, but rather chosen to balance accuracy with cost. See [2] for details.

3.4 Sinkhorn's Algorithm

We can now write the Lagrangian for the regularized optimal transport problem by introducing two dual variables $\widehat{\boldsymbol{f}} \in \mathbb{R}^n$ and $\widehat{\boldsymbol{g}} \in \mathbb{R}^n$ for the marginal constraints $P\mathbf{1}_{\boldsymbol{n}} = \boldsymbol{f}$ and $P^\top \mathbf{1}_{\boldsymbol{n}} = \boldsymbol{g}$ [2]

$$\mathcal{L}(P, \widehat{\boldsymbol{f}}, \widehat{\boldsymbol{g}}) = \langle P, C \rangle - \tfrac{1}{\lambda} H(P) - \widehat{\boldsymbol{f}}^\top (P\mathbf{1}_{\boldsymbol{n}} - \boldsymbol{f}) - \widehat{\boldsymbol{g}}^\top (P^\top \mathbf{1}_{\boldsymbol{n}} - \boldsymbol{g}). \tag{19}$$

Setting $\partial_{P_{ij}}\mathcal{L} = 0$ we obtain

$$P_{ij} = \boldsymbol{u}_i Q_{ij} \boldsymbol{v}_j, \quad Q_{ij} = exp(-\lambda C_{ij}), \quad \boldsymbol{u}_i = exp(\lambda \widehat{\boldsymbol{f}}_i), \quad \boldsymbol{v}_j = exp(\lambda \widehat{\boldsymbol{g}}_j) \tag{20}$$

or in matrix factorization form

$$P_\lambda = UQV, \quad U = diag(\boldsymbol{u}_1, ..., \boldsymbol{u}_n), \quad Q = [Q_{ij}], \quad V = diag(\boldsymbol{v}_1, ..., \boldsymbol{v}_n). \tag{21}$$

[19] Vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ can be obtained from

$$UQV\mathbf{1}_{\boldsymbol{n}} = \boldsymbol{f}, \qquad VQ^\top U\mathbf{1}_{\boldsymbol{n}} = \boldsymbol{g} \tag{22}$$

where $U\mathbf{1}_{\boldsymbol{n}} = \boldsymbol{u}$ and $V\mathbf{1}_{\boldsymbol{n}} = \boldsymbol{v}$ and we obtain equations for $(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+$ [20]

$$\boldsymbol{u} \odot (Q\boldsymbol{v}) = \boldsymbol{f}, \qquad \boldsymbol{v} \odot (Q^\top \boldsymbol{u}) = \boldsymbol{g} \tag{23}$$

where $\odot$ is the Hadamard (entrywise) product. We can now solve for $\boldsymbol{u}$ and $\boldsymbol{v}$ through an iterative method called Sinkhorn's algorithm [21]

$$\boldsymbol{u}^{(i)} = \boldsymbol{f} \oslash (Q\boldsymbol{v}^{(i-1)}), \; \boldsymbol{v}^{(i)} = \boldsymbol{g} \oslash (Q^\top \boldsymbol{u}^{(i)}), \; i = 1, ..., K \tag{24}$$

where $\oslash$ represents Hadamard (entrywise) division. In real applications a stopping criterion is needed. This is done by defining a tolarance $\epsilon_S > 0$, and continuing Sinkhorn iterations until we have [21]:

$$\max\{||\boldsymbol{u}^{(i)} \odot (Q\boldsymbol{v}^{(i)}) - \boldsymbol{f}||_\infty, ||\boldsymbol{v}^{(i)} \odot (Q^\top \boldsymbol{u}^{(i)}) - \boldsymbol{g}||_\infty\} \leq \epsilon_S. \tag{25}$$

After computing vectors $(\boldsymbol{u}, \boldsymbol{v})$ from Sinkhorn's algorithm, we obtain the Sinkhorn divergence of order p [17]:

$$S_{p,\lambda} = (\boldsymbol{u}^\top \widehat{Q} \boldsymbol{v})^{1/p}, \qquad \widehat{Q} = Q \odot C. \tag{26}$$

The cost of computing Sinkhorn divergence using Sinkhorn's algorithm is $\mathcal{O}(n^2 \log n)$ if C is chosen naively. This is still an improvement in cost over the Wasserstein metric if $d \geq 2$ which has a cost of $\mathcal{O}(n^3)$. However, if the data is given on a regular grid, the cost of Sinkhorn's algorithm can be reduced to $\mathcal{O}(n^{1+1/d} \log n)$ for specific

cost matrices [18][2]. One such cost matrix can be made by using the cost function [2]

$$c(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||_p^p = \sum_{k=1}^{d} |x^{(k)} - y^{(k)}|^p,\ \boldsymbol{x} = (x^{(1)}, ..., x^{(d)}) \in \mathbb{R}^d,$$
$$\boldsymbol{y} = (y^{(1)}, ..., y^{(d)}) \in \mathbb{R}^d,\ C_{ij} = c(\boldsymbol{x}_i, \boldsymbol{y}_j). \tag{27}$$

This reduction in cost is a key advantage that DSD has over the Wasserstein metric for $d \geq 2$. A multidimensional Wasserstein metric is more costly than a multidimensional DSD, especially if the choice of C is defined by (27).

Chapter 4

Optimal Transport Based Bayesian Inversion

4.1 DSD Quasi-Likelihood

We can now create an exponential quasi-likelihood function based on the DSD. We will first look at the quasi-likelihood function for the one-dimensional Wasserstein metric, and derive a new DSD quasi-likelihood function based on the Wasserstein quasi-likelihood function. The Wasserstein quasi-likelihood is [1]:

$$L_{wass}(\boldsymbol{\theta}) = \pi_{wass}(\boldsymbol{g}|\boldsymbol{\theta}) = s^N \exp(-s d_W(\boldsymbol{f}(\boldsymbol{\theta}), \boldsymbol{g})) \tag{28}$$

where s is a hyperparameter that will be found through a Markov Chain Monte Carlo (MCMC) sampling algorithm discussed later in this paper, and $d_W(\boldsymbol{f}(\boldsymbol{\theta}), \boldsymbol{g})$ is found using (12). This is a quasi-likelihood and connects the parameter vector $\boldsymbol{\theta}$ and observed quantities $\boldsymbol{g}$ via a loss function [22](in this case the Wasserstein distance). Note that the one-dimensional Wasserstein quasi-likelihood can be applied to multi-dimensional problems if statistical independence is assumed. This is the reason that

we have the term s^N in the Wasserstein quasi-likelihood, since this is considering the product of other one-dimensional exponential quasi-likelihoods. A numerical algorithm for computing the Wasserstein metric can be found in [5]. As discussed in the previous section, the DSD is a regularized multidimensional Wasserstein distance, meaning that we do not need the product of multiple quasi-likelihoods in order to analyze multidimensional problems. This ultimately simplifies the exponential quasi-likelihood, thus creating an exponential quasi-likelihood with the DSD:

$$L_{dsd}(\boldsymbol{\theta}) = \pi(\boldsymbol{g}|\boldsymbol{\theta}) = s[\exp(-sd_{dsd}(\boldsymbol{f}(\boldsymbol{\theta}), \boldsymbol{g}))] \tag{29}$$

where s is again a hyperparameter that will be found through the MCMC sampling algorithm discussed later in this paper, and $d_{dsd}(\boldsymbol{f}(\boldsymbol{\theta}), \boldsymbol{g}))$ is found using (17). This is a quasi-likelihood and connects the parameter vector $\boldsymbol{\theta}$ and observed quantities $\boldsymbol{g}$ via a loss function [22](in this case DSD). Note here that s is not raised to the power of N since we are no longer looking at the product of multiple quasi-likelihoods.

4.2 Convexity of DSD Quasi-Likelihood

As it has been stated earlier in this paper, one important feature of the DSD quasi-likelihood is the convexity with respect to phase shift, phase dilation, and amplitude change. This convexity will be tested by applying the DSD quasi-likelihood to a test problem. A comparison will be made between the DSD quasi-likelihood, Wasserstein quasi-likelihood, and Gaussian likelihood. Suppose that the original signal f is:

$$f(t) = e^{-(\frac{t-4}{\delta})^2} - e^{-(\frac{t-5}{\delta})^2} + e^{-(\frac{t-6}{\delta})^2}$$

And a shifted version of the signal (representative of noise perhaps) g is:

$$g(t) = e^{-(\frac{t-s-4}{\delta})^2} - e^{-(\frac{t-s-5}{\delta})^2} + e^{-(\frac{t-s-6}{\delta})^2}$$

Where s is the factor in which the signal is shifted.

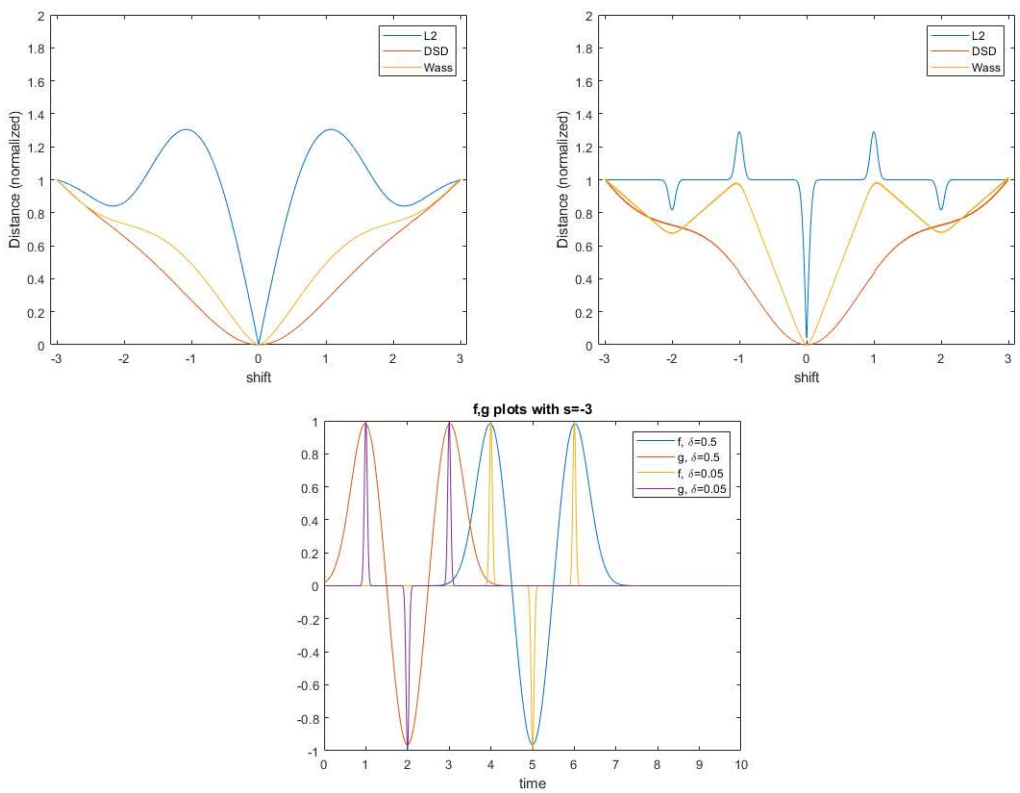

Figure 4.1: Plot of Convexity of DSD quasi-likelihood. Top plots normalized so when $s = -3$ the value on the plot is 1. Left plot is for wide signals ($\delta = 0.5$) and right plot is for narrow signals ($\delta = 0.05$).

Note that there are many ways to normalize the signals for the DSD quasi-likelihood and the Wasserstein quasi-likelihood. The option that has been used in this example is linear scaling, but other normalization protocols exist [1][7][8]. Linear scaling normalization is:

$$\widehat{\boldsymbol{f}}=\frac{\boldsymbol{f}+c}{<\boldsymbol{f}+c>} \quad \text{and} \quad \widehat{\boldsymbol{g}}=\frac{\boldsymbol{g}+c}{<\boldsymbol{g}+c>}, \; <\boldsymbol{f}>=\sum_{i=1}^{N} f_i \tag{30}$$

Where c is some constant chosen to ensure that both $\boldsymbol{f}+c > 0$ and $\boldsymbol{g}+c > 0$. The signals are then normalized to ensure that they can now be viewed as probability dis-

tributions. As seen in Figure 4.1, the DSD quasi-likelihood shows the best convexity for this specific example. The Gaussian likelihood produced many minima which, in a Bayesian inversion application, could produce an incorrect posterior. This example shows the advantage of using the DSD quasi-likelihood over the L_2 likelihood for inversion problems that involve phase shift. Note that it is possible to obtain better convexity from both DSD and Wasserstein quasi-likelihoods by choosing different ways of normalizing the initial signals $\boldsymbol{f}$ and $\boldsymbol{g}$.

Chapter 5

Numerical Algorithm

In this section a Markov Chain Monte Carlo (MCMC) algorithm, along with the DSD quasi-likelihood, are used to create a numerical algorithm for the DSD Bayesian framework. The specific MCMC algorithm that will be used is the Metropolis-Hastings-within-Gibbs [9] (MH within G) sampling algorithm. MH within G is a method that combines two MCMC algorithms, the Gibbs sampler and Metropolis-Hastings sampler. The Gibbs sampler solves for the posterior of the hyper-parameter s, and the Metropolis-Hastings sampler solves for the posterior of the parameter vector $\boldsymbol{\theta}$. MH-within-G updates samples based on a selection process and keeps samples with a probability, α (see section 5.3). Looking at the posterior in Bayesian inversion, note that:

$$\pi(\boldsymbol{\theta}|\boldsymbol{g}) \propto \pi(\boldsymbol{g}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$$

This means that for a Bayesian algorithm we need a likelihood and prior. This is where we decide to use the DSD quasi-likelihood and employ a know algorithm, MH within G.

5.1 Gibbs Sampler

Suppose that we have a gamma prior for s. That is, $s \sim Gamma(a, b)$, where $a, b > 0$ are the shape and rate parameters of the Gamma prior. Employing this prior along with the DSD quasi-likelihood we have:

$$\pi(s|\boldsymbol{\theta}, \boldsymbol{g}) \propto \pi_{dsd}(\boldsymbol{g}|\boldsymbol{\theta}, s)\ \pi_{prior}(s) \propto se^{-sd_{dsd}} s^{a-1} e^{-bs} = s^a e^{-s(b+d_{dsd})}$$

where $d_{dsd} = d_{dsd}(\boldsymbol{f}(\boldsymbol{\theta}),\boldsymbol{g})$. Note that the posterior is proportional to a Gamma distribution. To be exact:

$$s \sim Gamma(a^*, b^*), \qquad a^* = a + 1, \qquad b^* = b + d_{dsd}$$

This will be used to generate new values of s with a given $\boldsymbol{\theta}$. Note that the prior here is a conjugate prior since both the prior and posterior of s are Gamma distributions. This is not to say that the prior for s must be a Gamma distribution. This is just the choice of prior for s in this paper, and others may be used.

5.2 Metropolis-Hastings Sampler

The Gibbs sampler in the previous section assumes a fixed $\boldsymbol{\theta}$. The Metropolis-Hastings sampler [10][11] assumes a fixed s. Suppose then that we have a fixed value for s and a posterior for $\boldsymbol{\theta}$.

$$\pi(\boldsymbol{\theta}|s, \boldsymbol{g}) \propto \pi_{dsd}(\boldsymbol{g}|\boldsymbol{\theta}, s)\pi_{prior}(\boldsymbol{\theta})$$

Given a sample value of $\boldsymbol{\theta}$, call it $\boldsymbol{\theta^{(i)}}$, the goal is to generate a new sample, $\boldsymbol{\theta^{(i+1)}}$. Generate a candidate sample $\widetilde{\boldsymbol{\theta}}$ by sampling from a proposal distribution $q(\boldsymbol{\theta^{(i)}}, \widetilde{\boldsymbol{\theta}})$ from the current sample $\boldsymbol{\theta^{(i)}}$.

We then accept this candidate sample with probability [11]:

$$\alpha = \frac{\pi(\widetilde{\boldsymbol{\theta}}|s,\boldsymbol{g})q(\boldsymbol{\theta^{(i)}},\widetilde{\boldsymbol{\theta}})}{\pi(\boldsymbol{\theta^{(i)}}|s,\boldsymbol{g})q(\widetilde{\boldsymbol{\theta}},\boldsymbol{\theta^{(i)}})} = \frac{\pi_{dsd}(\boldsymbol{g}|\widetilde{\boldsymbol{\theta}},s)\pi_{prior}(\widetilde{\boldsymbol{\theta}})q(\boldsymbol{\theta^{(i)}},\widetilde{\boldsymbol{\theta}})}{\pi_{dsd}(\boldsymbol{g}|\boldsymbol{\theta^{(i)}},s)\pi_{prior}(\boldsymbol{\theta^{(i)}})q(\widetilde{\boldsymbol{\theta}},\boldsymbol{\theta^{(i)}})} \tag{31}$$

This still leaves the choice of prior and proposal distributions. The choice of prior is often left up to experts for the specific problem since we want an expert to decide what we already know about the parameter $\boldsymbol{\theta}$. Sometimes nothing may be known about the prior distribution of $\boldsymbol{\theta}$. This is called a non-informative prior and would mean that $\pi_{prior}(\boldsymbol{\theta}) = 1$. As for the proposal distribution of $\widetilde{\boldsymbol{\theta}}$, a Gaussian random walk is often used and will be used in the examples in chapter 6:

$$\widetilde{\boldsymbol{\theta}} \sim Normal(\boldsymbol{\theta^{(i)}}, \Sigma) \tag{32}$$

where Σ is the covariance matrix. In the MH-within-G algorithm, the covariance matrix has a noticeable affect on the posterior and is often changed multiple times to see what works best for a given problem. Note here that if the proposal distribution is symetric (i.e. $q(\boldsymbol{\theta^{(i)}}, \widetilde{\boldsymbol{\theta}}) = q(\widetilde{\boldsymbol{\theta}}, \boldsymbol{\theta^{(i)}})$), the terms with q cancel out in the α ratio and we are left with [10]:

$$\alpha = \frac{\pi_{dsd}(\boldsymbol{g}|\widetilde{\boldsymbol{\theta}},s)\pi_{prior}(\widetilde{\boldsymbol{\theta}})}{\pi_{dsd}(\boldsymbol{g}|\boldsymbol{\theta^{(i)}},s)\pi_{prior}(\boldsymbol{\theta^{(i)}})}$$

Symmetric proposals will be used in the examples in this paper so that this simplified α may be used.

5.3 The Algorithm: Metropolis-Hastings-within-Gibbs in the DSD-Bayesian Framework

1. *Initialization*: Select an initial starting point $(\boldsymbol{\theta^{(0)}}, s^{(0)})$ and set $i = 0$.

2. *Normalize*: Select a normalization protocol to ensure that signals $\boldsymbol{f}$ and $\boldsymbol{g}$ are consistent with the definition of probability distributions.

3. *Gibbs Sampler*: Generate $s^{(i+1)}$ from the posterior $\pi(s|\boldsymbol{\theta^{(i)}}, \boldsymbol{g})$ with Gamma distribution

$$s^{(i+1)} \sim Gamma(a^*, b^*), \qquad a^* = a + 1, \qquad b^* = b + d_{dsd}(\boldsymbol{f}(\boldsymbol{\theta^{(i)}}), \boldsymbol{g})$$

4. *Metropolis-Hastings Sampler*: Follow steps i-iii to generate $\boldsymbol{\theta^{(i+1)}}$:

i. Sample a candidate $\widetilde{\boldsymbol{\theta}}$ from the proposal distribution $q(\boldsymbol{\theta^{(i)}}, \widetilde{\boldsymbol{\theta}})$

ii. Compute the ratio:

$$\alpha(\boldsymbol{\theta^{(i)}}, \widetilde{\boldsymbol{\theta}}) = \frac{\pi_{dsd}(\boldsymbol{g}|\widetilde{\boldsymbol{\theta}}, s)\pi_{prior}(\widetilde{\boldsymbol{\theta}})q(\boldsymbol{\theta^{(i)}}, \widetilde{\boldsymbol{\theta}})}{\pi_{dsd}(\boldsymbol{g}|\boldsymbol{\theta^{(i)}}, s)\pi_{prior}(\boldsymbol{\theta^{(i)}})q(\widetilde{\boldsymbol{\theta}}, \boldsymbol{\theta^{(i)}})}$$

iii. Set

$$\boldsymbol{\theta^{(i+1)}} = \begin{cases} \widetilde{\boldsymbol{\theta}}, & Unif(0,1) \leq \alpha(\boldsymbol{\theta^{(i)}}, \widetilde{\boldsymbol{\theta}}) \\ \boldsymbol{\theta^{(i)}}, & otherwise \end{cases}$$

5. *Iterate*: Increment i by 1 and go to step 3.

Chapter 6

Numerical Examples from Seismic Inversion

This section will explore several examples that implement the above DSD-Bayesian framework. These examples exist in the form of wave propagation, something seen in applications like seismic inversion. These examples serve to show the accuracy of this Bayesian framework and to compare it to both a Wasserstein framework and Gaussian framework.

6.1 Problem Formulation

All the examples in this section will use the same general one-dimensional source inversion problem. This problem is of similar form to an example in [1], allowing for a direct comparison to work in [1]. This problem has a wave pulse that propagates at a constant speed. The main difference in each example will be the noise seen in the example. This will allow for direct comparison based on noise complexity as well as show the benefits of the DSD framework.

Suppose we have the Cauchy problem for the one-dimensional wave equation

$$u_{tt}(t,x) - u_{xx}(t,x) = 0, \qquad t \in [0,T], \qquad x \in \mathbb{R} \tag{33}$$

$$u(0,x) = h(x;x_0,a), \qquad u_t(0,x) = 0 \tag{34}$$

With initial data

$$h(x;x_0,a) = a(e^{-100(x-x_0-0.5)^2} + e^{-100(x-x_0)^2} + e^{-100(x-x_0+0.5)^2}) \tag{35}$$

This initial data acts as a source creating an initial wave pulse with given amplitude a and initial starting location x_0. The solution to this problem is given by d'Alembert's formula [1]

$$u(t,x;x_0,a) = \tfrac{1}{2}h(x-t;x_0,a) + \tfrac{1}{2}h(x+t;x_0,a) \tag{36}$$

6.2 Example 1: Known Posterior

This first example will show the accuracy of the DSD framework. Suppose that both the amplitude a and initial position x_0 are treated as parameters in our Bayesian framework. That is to say that $\boldsymbol{\theta} = (\theta_1, \theta_2)$ where $\theta_1 = x_0$ and $\theta_2 = a$ are both unknown parameters. Suppose that we also have $N_r = 7$ recievers each collectiong data located at 7 different positions:

$$x_1 = -3,\ x_2 = -2,\ x_3 = -1,\ x_4 = 0,\ x_5 = 1,\ x_6 = 2,\ x_7 = 3$$

where each reciever is located at x_r with $r = 1, ..., N_r$ and records noisy discrete-time data $g(t_k, x_r)$ over the time interval $[0,T]$ at N discrete time levels $t_k = (k-1)\Delta t$ with $\Delta t = T/(N-1)$ and $k = 1, ..., N$. Let $f(t_k, x_r; \boldsymbol{\theta})$ be the corresponding simulated

signal for a given $\boldsymbol{\theta}$ computed using $f(t_k, x_r; \boldsymbol{\theta}) = u(t, x; \theta_1, \theta_2)$. Now, define the fixed parameter:

$$\boldsymbol{\theta^*} = (\theta_1^*, \theta_2^*), \qquad \theta_1^* = 0.1, \qquad \theta_2^* = 5$$

and generate synthetic data $g(t_k, x_r)$ so that the posterior of $\boldsymbol{\theta}$ is known. This is done by first generating a true posterior distribution for $\boldsymbol{\theta}$.

$$\pi_{posterior}(\theta_1|\boldsymbol{g}) = Norm(\theta_1^*, 0.001), \qquad \pi_{posterior}(\theta_2|\boldsymbol{g}) = Norm(\theta_2^*, 0.01)$$

Define $\widehat{\theta_1}$ and $\widehat{\theta_2}$ so that:

$$\widehat{\theta_1} \sim Norm(\theta_1^*, 0.001) \text{ and } \widehat{\theta_2} \sim Norm(\theta_2^*, 0.01)$$

Now, sample values for $\widehat{\theta_1}$ and $\widehat{\theta_2}$ with $r = 1, ..., N_r$ and $k = 1, ..., N$ to produce

$$g(t_k, x_r) = f(t_k, x_r; \widehat{\theta_1}^{(rk)}, \widehat{\theta_2}^{(rk)})$$

where each $\widehat{\theta_1}^{(rk)} \sim Normal(\theta_1^*, 0.001)$ and $\widehat{\theta_1}^{(rk)} \sim Normal(\theta_2^*, 0.01)$. This allows us to test the accuracy of the DSD Bayesian framework since we can directly compare the results from the DSD framework to the true posteriors. Note that the data in this problem is two dimensional. This means that when storing the data $g(t_k, x_r)$ into a vector $\boldsymbol{g}$, we store it by defining

$$\boldsymbol{g} = (g(t_1, x_1), g(t_2, x_1), ..., g(t_N, x_1), g(t_1, x_2), g(t_2, x_2), ..., g(t_N, x_2), ..., g(t_N, x_{N_r}))$$

so that we translate the two dimensional data into a vector. This same procedure is done with $\boldsymbol{f}$. This means that vectors $\boldsymbol{f}$ and $\boldsymbol{g}$ have $n = N \times N_r$ entries.

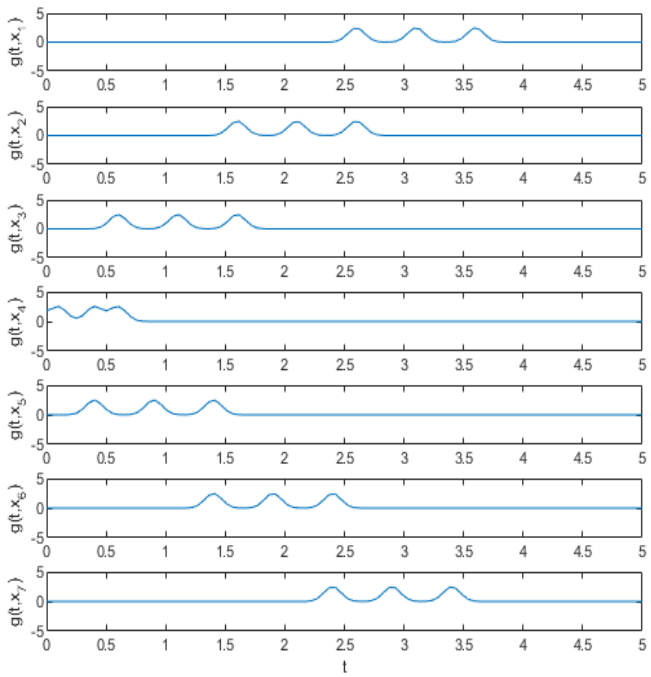

Figure 6.1: Observed signals with known posterior, recorded at seven receivers

Here are the following computations that were employed along with the DSD-Bayesian algorithm:

- *Likelihood*: $\pi_{dsd}(\boldsymbol{g}|\boldsymbol{\theta}) = s[\exp(-sd_{dsd}(\boldsymbol{f}(\boldsymbol{\theta}), \boldsymbol{g}))]$ (from equation 29)
- *Priors*: $\theta_1 \sim Unif(-3, 3), \quad \theta_2 \sim Unif(3, 7), \quad s \sim Gamma(1200, 2)$
- *Initial Data*: $\theta_1^{(0)} = 0.6, \quad \theta_2^{(0)} = 3, \quad s^{(0)} = 70$
- *Proposal*: $\widetilde{\boldsymbol{\theta}} \sim Normal(\boldsymbol{\theta^{(i)}}, \Sigma)$ with covariance matrix $\Sigma = diag(0.005, 0.005)$.

Now, the DSD-Bayesian algorithm is ran with $M = 500000$ iterations and remove the first $M_b = 250000$ samples in the burn-in period. A thinning period of $M_t = 4$ is used. That is to say that every 4^{th} is kept, and the rest are discarded.

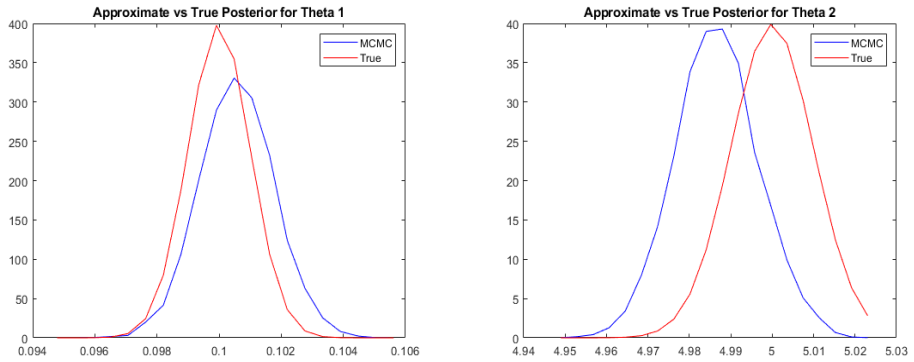

Figure 6.2: Approximate vs. true posteriors for θ_1 and θ_2. Appoximate posteriors found using DSD-Bayesian algorithm and are labeled MCMC (for Markov Chain Monte Carlo).

As seen in Figure 6.2, the approximate posterior converges to the true posterior. The error between the approximate and true posteriors can be measured using the Wasserstein distance, since this is a measure of dissimilarity between two probability distributions. The Wasserstein distance between the true and approximate posteriors for θ_1 is 8.9198×10^{-4} and the Wasserstein distance between the true and approximate posteriors for θ_2 is 1.446×10^{-2}. Now, we can measure the Wassertein distance for different values of M to get an idea about convergence. In Figure 6.3 we see how the Wasserstein distance between the true and approximate posteriors for θ_2 decrease with a larger number of iterations, M. In Figure 6.3 a plot of $\mathcal{O}(1/M^2)$ is shown to give context to this convergence.

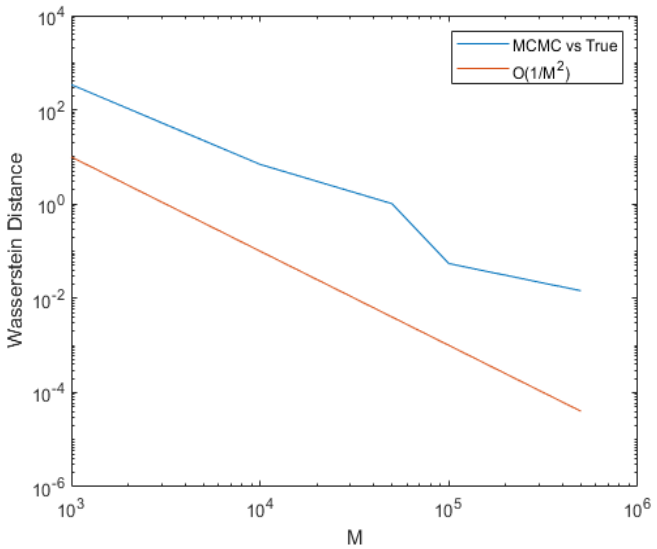

Figure 6.3: Wasserstein distance for the approximate vs true posteriors of θ_2 with varying number of iterations, M. MCMC solutions found using DSD-Bayesian framework.

6.3 Example 2: Additive Gaussian Noise with Unknown Phase and Amplitude

This example will compare the DSD framework to both the Wasserstein and Gaussian framework. This example will illustrate the advantage that optimal transport based Bayesian frameworks have over the standard Gaussian framework for problems involving additive Gaussian noise structures. Suppose that both amplitude a and initial position x_0 are treated as parameters in our Bayesian framework. That is to say that $\boldsymbol{\theta} = (\theta_1, \theta_2)$ where $\theta_1 = x_0$ and $\theta_2 = a$ are both unknown parameters. Suppose we have $N_r = 7$ receivers each collecting data located at 7 different positions:

$$x_1 = -3,\ x_2 = -2,\ x_3 = -1,\ x_4 = 0,\ x_5 = 1,\ x_6 = 2,\ x_7 = 3$$

where each reciever is located at x_r with $r = 1, ..., N_r$ and records noisy discrete-time data $g(t_k, x_r)$ over the time interval $[0, T]$ at N discrete time levels $t_k = (k-1)\Delta t$ with $\Delta t = T/(N-1)$ and $k = 1, ..., N$. Let $f(t_k, x_r; \boldsymbol{\theta})$ be the corresponding simulated signal for a given $\boldsymbol{\theta}$ computed using $f(t_k, x_r; \boldsymbol{\theta}) = u(t, x; \theta_1, \theta_2)$. Now, define the fixed parameter:

$$\boldsymbol{\theta^*} = (\theta_1^*, \theta_2^*), \qquad \theta_1^* = 0, \qquad \theta_2^* = 5$$

and generate synthetic data $g(t_k, x_r)$ by polluting $f(t_k, x_r; \boldsymbol{\theta^*})$ with an additive Gaussian noise:

$$g(t_k, x_r) = f(t_k, x_r; \boldsymbol{\theta^*}) + \epsilon_{rk}, \qquad \epsilon_{rk} \sim Normal(0, 0.1)$$

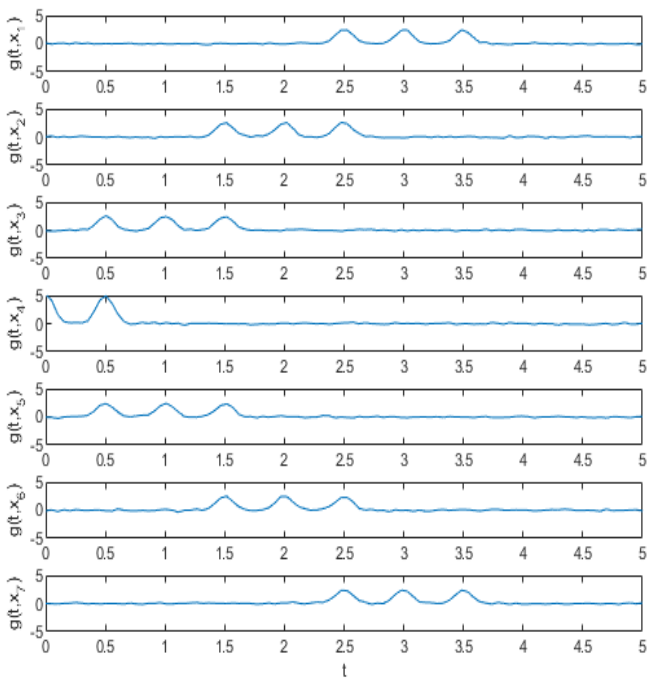

Figure 6.4: Observed signals with additive Gaussian noise, recorded at seven receivers

Here are the following computations that were employed along with the DSD-Bayesian algorithm:

- *Likelihood*: $\pi_{dsd}(\boldsymbol{g}|\boldsymbol{\theta}) = s[\exp(-sd_{dsd}(\boldsymbol{f}(\boldsymbol{\theta}), \boldsymbol{g}))]$ (from equation 29)

- *Priors*: $\theta_1 \sim Unif(-3, 3)$, $\theta_2 \sim Unif(3, 7)$, $s \sim Gamma(15000, 0.3)$

- *Initial Data*: $\theta_1^{(0)} = 0.6, \qquad \theta_2^{(0)} = 3, \qquad s^{(0)} = 70,$

- *Proposal*: $\widetilde{\boldsymbol{\theta}} \sim Normal(\boldsymbol{\theta^{(i)}}, \Sigma)$ with covariance matrix $\Sigma = diag(0.00001, 0.00001)$

Now, the DSD-Bayesian algorithm is ran with $M = 50000$ iterations and remove the first $M_b = 25000$ samples in the burn-in period. A thinning period of $M_t = 2$ is used. That is to say that every 2^{nd} entry is kept, and the rest are discarded.

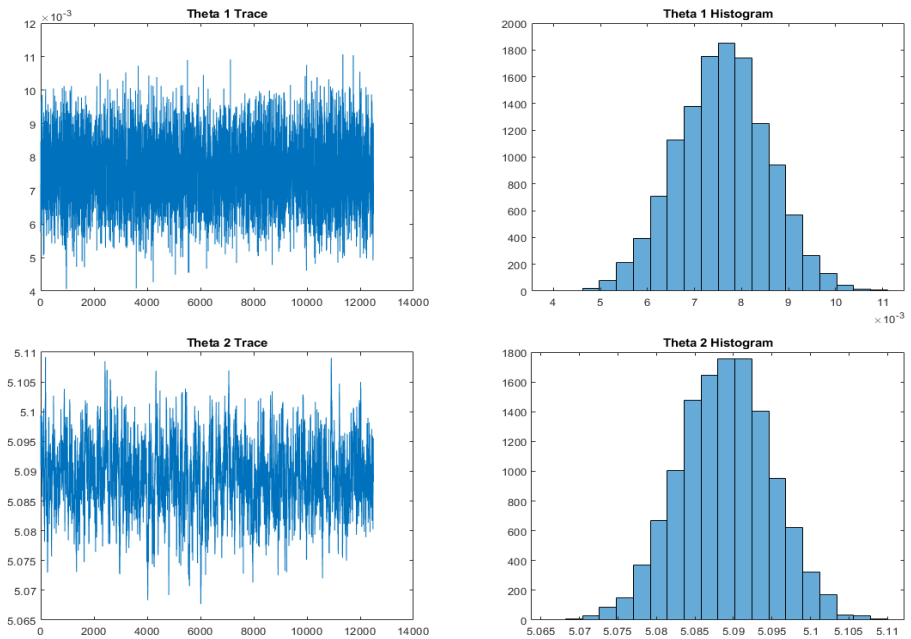

Figure 6.5: Trace and Histograms for θ_1 and θ_2 found using DSD-Bayesian framework

This same test is also ran using the Wasserstein quasi-likelihood and Gaussian likelihood (see figures 6.6 and 6.7). It is clear that the DSD and Wasserstein frameworks performed much better than the Gaussian framework. This is seen in the fact that the Gaussian framework produced posteriors that converge to incorrect values for θ_1 and θ_2. As stated in [1], the reason that this occurs is because the Gaussian framework produces many local minima, meaning that the Bayesian framework may fall into a false minimum and converge to an incorrect posterior. This is also what is seen in Figure 6.8 which shows that for this example the L_2 (or Gaussian) likelihood produces many extrema.

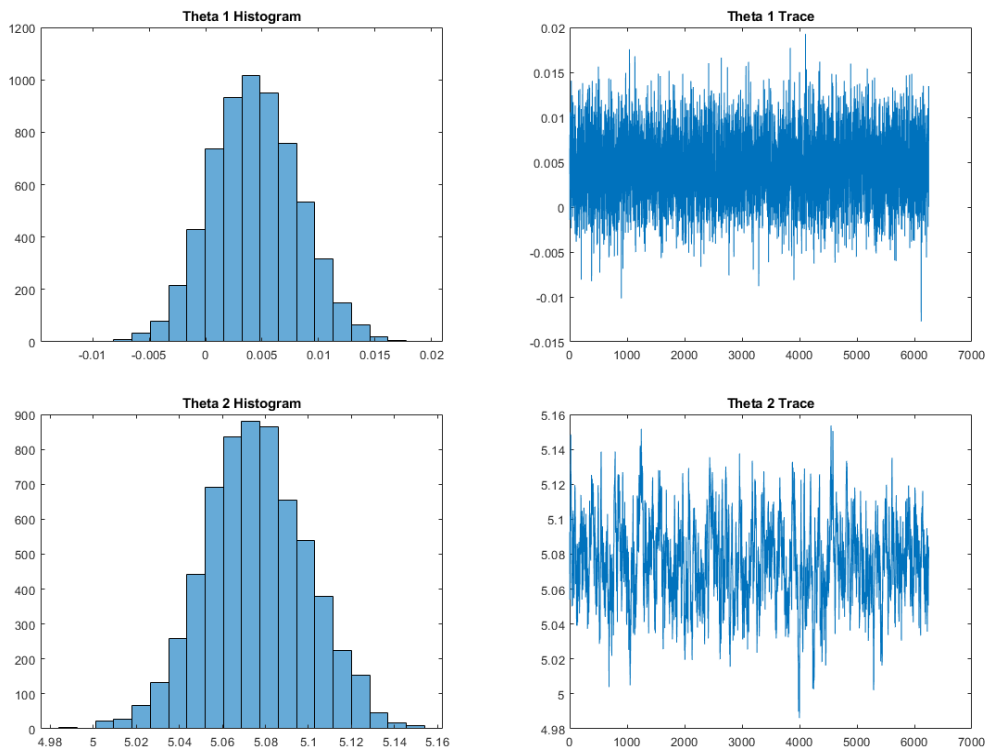

Figure 6.6: Trace and Histograms for θ_1 and θ_2 found using Wasserstein-Bayesian framework. As seen in this figure as well as Figure 6.5, it is clear to see that the Wasserstein and DSD frameworks are converging to the same posterior because the histograms are centered at nearly the same values and the spread of these histograms are also similar.

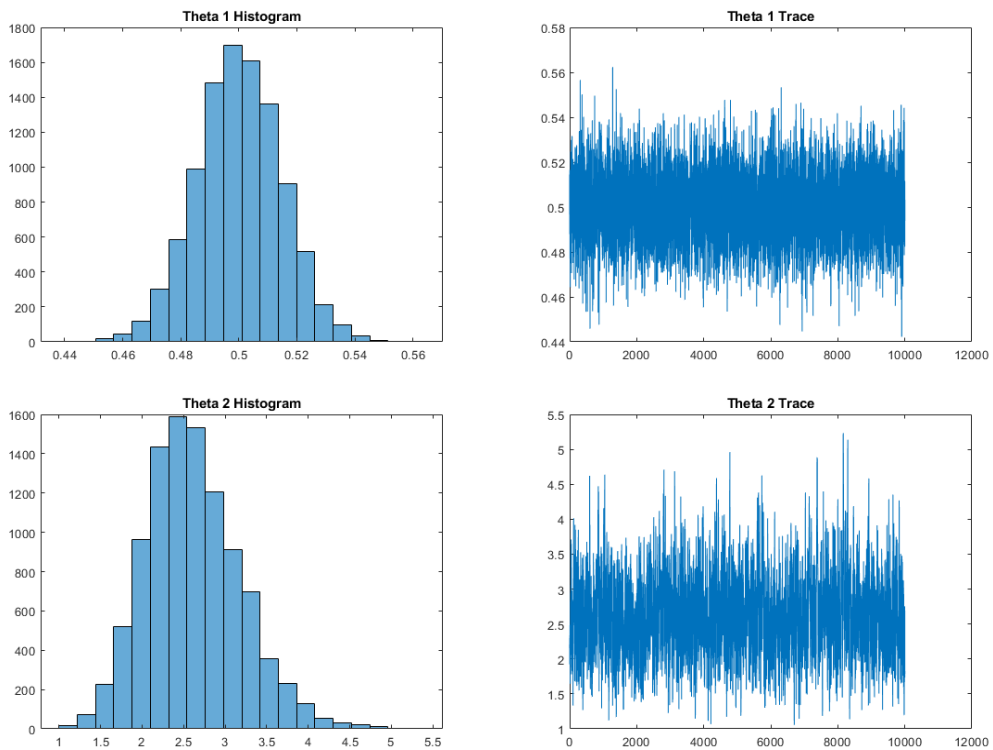

Figure 6.7: Trace and Histograms for θ_1 and θ_2 found using Gaussian-Bayesian framework. As clearly seen in the figures, the posteriors found from this Gaussian-Bayesian framework converged to incorrect values of θ_1 and θ_2.

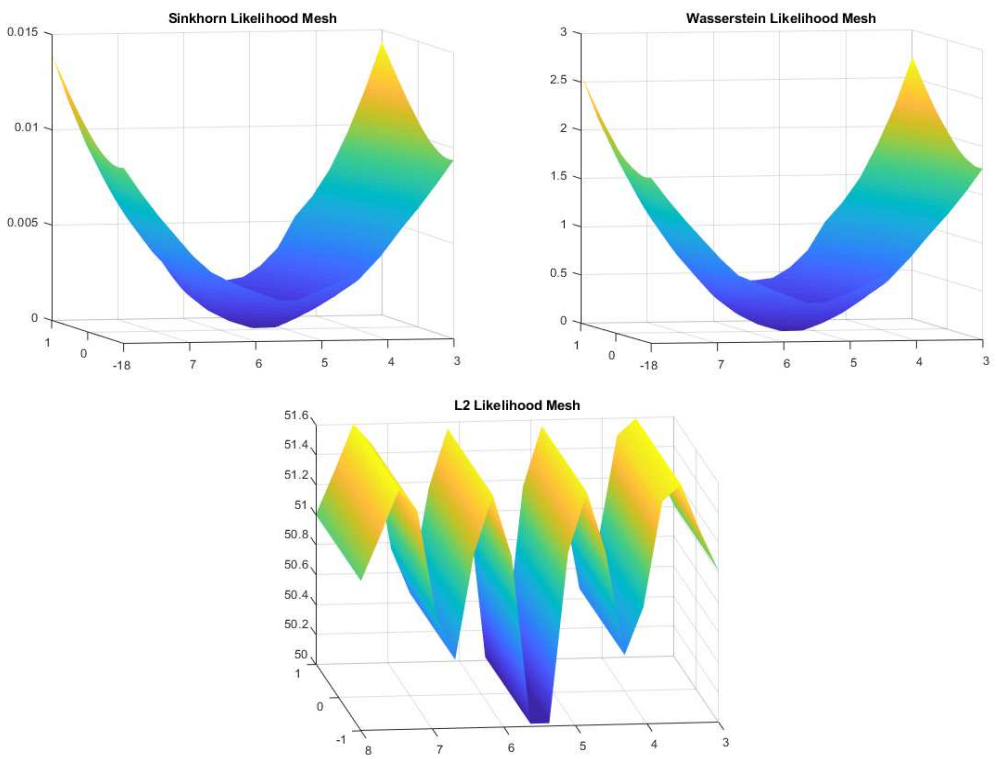

Figure 6.8: Surface plots of DSD (left), Wasserstein (right), and L_2 (bottom) likelihoods for example 2. This shows that the L_2 likelihood did not stay convex and that the optimal transport quasi-likelihoods did.

Chapter 7

Conclusion

Presented in this paper is a Bayesian framework based on Debiased Sinkhorn Divergence. This framework is based closely on the Wasserstein-Bayesian framework presented in [1], and preforms similarly in Example 2. The DSD-Bayesian framework allows for statistical independence, allowing for a more robust mathematical framework for Bayesian inversion for problems that do not involve statistical independence. This framework also has a computational time benefit over a higher dimensional Wasserstein framework, since Sinkhorn Divergence is a low cost regularized Wasserstein distance. This framework is well suited for inversion problems that involve phase shift, phase dilation, and amplitude change such as seismic events or other signal analysis problems.

References

1. M. Motamed and D. Appelo. Wasserstein metric-driven Bayesian inversion with applications to signal processing. *International J. for Uncertainty Quantification*, vol. 9, pp. 395-414, 2019.

2. M. Motamed. Hierarchical low-rank approximation of regularized Wasserstein distance. 2020. https://arxiv.org/abs/2004.12511

3. T. Bayes, R. Price, and J. Canton. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. *Philosophical Transactions of the Royal Society of London*, pages 370–418, 1763.

4. Gut, Allan (2013). Probability: A Graduate Course (Second ed.). New York, NY: Springer. ISBN 978-1-4614-4707-8.

5. J.-D. Benamou, B. D. Froese, and A. M. Oberman. Numerical solution of the Optimal Transportation problem using the Monge-Ampère equation. *Journal of Computational Physics*, 260:107–126, 2014.

6. B. Engquist, B.D. Froese Brittany, and Y. Yang. Optimal transport for seismic full waveform inversion. *Communications in Mathematical Sciences*, 14(8):2309–2330, 2016.

7. B. Engquist and Y. Yang. Seismic Imaging and Optimal Transport. *To appear in Communications in Information and Systems, see also arXiv e-prints*, page arXiv:1808.04801, August 2018.

8. L. Métivier, A. Allain, R. Brossier, Q. Mérigot, E. Oudet, and J. Virieux. Optimal transport for mitigating cycle skipping in full-waveform inversion: A graph-space transform approach. *GEOPHYSICS*, 83(5):R515–R540, 2018.

9. A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2004.

10. 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*, 21:1087–1092, 1953

11. S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. The American Statistician, 49:327–335, 1995.

12. C. Villani. Optimal Transport: Old and New, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer Verlag, 2009.

13. C. Villani. Topics in Optimal Transportation, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, 2003.

14. C. Villani. Optimal Transport: Old and New, volume 338 of *Comprehensive Studies in Mathematics*. Springer-Verlag, Berlin, 2009.

15. Y. Yang, B. Engquist, J. Sun, and B. F. Hamfeldt. Application of optimal transport and the quadratic Wasserstein metric to full-waveform inversion. *Geophysics*, 83(1):R43– R62, 2018.

16. M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26*, pages 2292–2300, 2013.

17. G. Peyré and M. Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11:355–607, 2019.

18. J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional Wasserstein distances: efficient optimal transportation on geometric domains. *ACM Transactions on Graphics*, 34:66:1–66:11, 2015.

19. A. Nemirovski and U. Rothblum. On complexity of matrix scaling. *Linear Algebra and its Applications*, 302:435–460, 1999.

20. B. Kalantari and L. Khachiyan. On the complexity of nonnegative-matrix scaling. *Linear Algebra and its Applications*, 240:87–103, 1996.

21. R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Annals of Mathematical Statististics*, 35:876-879, 1964

22. P.G. Bissiri, C.C. Holmes, and S.G. Walker. A General Framework for Updating Belief Distributions. *Journal of the Royal Statistical Society, Statistical Methodology Series B*, 78:1103-1130, 2016.