DOE/NV/03624--1086

BAYESIAN SPATIALLY VARYING MULTI-REGULARIZATION IMAGE DEBLURRING

by

Jessica Pillow

A Dissertation Submitted to the Faculty of the

GRADUATE INTERDISCIPLINARY
PROGRAM IN APPLIED MATHEMATICS

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2021

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation
prepared by: Jessica Pillow
titled: Bayesian Spatially Varying Multi-Regularization Image Deblurring

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Matthias Morzfeld
Matthias Morzfeld Date: May 6, 2021

Kevin Lin Date: May 6, 2021

Matthew Kupinski Date: May 6, 2021

Marylesa Howard
Marylesa Howard Date: May 6, 2021

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Matthias Morzfeld
Matthias Morzfeld
Mathematics
Date: May 6, 2021

Acknowledgements

This work would not have been possible without the patience and guidance of my co-advisors, Dr. Matthias Morzfeld and Dr. Marylesa Howard. A few years ago, when I was struggling to find a research advisor, Matti sought me out and presented me with an amazing opportunity to collaborate with Marylesa and her team at the Nevada National Security Site. In addition to being a part of an incredible research project, I also had the chance to spend two summers interning for the NNSS. I'm forever grateful to Matti and Marylesa for believing in me, and shaping me into the mathematician I am today.

I must also thank Dr. Matthew Kupinski and Dr. Jesse Adams for their advice, support, and encouragement over the last couple of years. Thank you, Matt, for serving on my comprehensive exam committee, and then stepping into Matti's administrative role when he transitioned to a new position at UCSD. Thank you for providing me with a huge, quiet office space next to yours in the Optical Sciences building. I wish I could have used it longer than the two months before the COVID-19 pandemic shut everything down. Thank you, Jesse, for initiating the collaboration between the University of Arizona and the NNSS while you were a graduate student. Thank you for welcoming me to the UA/NNSS team and for helping with the transition after you graduated.

I am also grateful to the UA applied mathematics program and the professors I had the pleasure of learning from. Thank you for challenging me and building my base-level of mathematical knowledge. A special thank you to Dr. Kevin Lin, from whom I've taken several courses, for serving on both my comprehensive exam committee and my dissertation committee.

I would also like to thank my friends and family for their love and support during my graduate career. I would not have made it without you. Thank you especially for your care during the COVID-19 pandemic. I never thought I would have to write my entire dissertation from my living room, but that was my reality this past year. Thank you for the texts, phone calls, and Zoom calls when I needed to talk. To my in-laws, thank you for hosting a weekly bake-off competition via Zoom. I know this wasn't for me specifically, but it brought me so much joy every week. To my wife, Liz, thank you for believing in me and taking care of me through it all. Thank you for your sacrifices, and thank you for encouraging me while I chased this dream.

Finally, I am grateful to have been supported by the NNSS through a Site Directed Research and Development grant, which allowed me to devote more time and energy to this work than I would have been able to otherwise.

This work was supported by Mission Support and Test Services, LLC, under Contract No. DE-NA0003624 with the U.S. Department of Energy and supported by the Site-Directed Research and Development Program. DOE/NV/03624–1086

Dedication

To my parents, Julie and Paul, and to my brother, Thomas. My life-long love of learning began with your guidance and encouragement. To my wife, Liz, for your laughter, love, and patience during the writing process.

Contents

List of Figures

List of Tables

List of Algorithms

Notation

Images

$\mathbf{B} \in \mathbb{R}^{n \times n}$	Image data
$\mathbf{X} \in \mathbb{R}^{n \times n}$	True Image
$\overline{\mathbf{X}} \in \mathbb{R}^{n \times n}$	Image Reconstruction
$\mathbf{b} = \operatorname{vec}(\mathbf{B}) \in \mathbb{R}^{n^2}$	Column stack of $\mathbf{B}$
$\mathbf{x} = \operatorname{vec}(\mathbf{X}) \in \mathbb{R}^{n^2}$	Column stack of $\mathbf{X}$
$\overline{\mathbf{x}} = \operatorname{vec}(\overline{\mathbf{X}}) \in \mathbb{R}^{n^2}$	Column stack of $\mathbf{X}$

Model Components

$\mathbf{a} \in \mathbb{R}^{n \times n}$	Convolution kernel
$\lambda \in \mathbb{R}_{>0}$	Likelihood precision
$\delta \in \mathbb{R}_{>0}$	Prior precision
$\boldsymbol{\varepsilon} \in \mathbb{R}^{n}$	Additive Gaussian noise

Matrices

$\mathbf{A} \in \mathbb{R}^{n \times n}$	Convolution matrix with zero boundary conditions
$\mathbf{D}_1 \in \mathbb{R}^{n \times n}$	Discrete 1st derivative
$\mathbf{L}_{\text{TIK}} := \mathbf{D}_2 \in \mathbb{R}^{n \times n}$	Discrete 2nd derivative, and chosen operator for Tikhonov
$\mathbf{L}_{\text{TV}} := \mathbf{D}_1^T \Psi \mathbf{D}_1 \in \mathbb{R}^{n \times n}$	Approximated TV operator
$\mathbf{D}_s := \mathbf{I} \otimes \mathbf{D}_1 \in \mathbb{R}^{n^2 \times n^2}$	Discrete 1st derivative in s-direction of (s,t)-plane
$\mathbf{D}_t := \mathbf{D}_1 \otimes \mathbf{I} \in \mathbb{R}^{n^2 \times n^2}$	Discrete 1st derivative in t-direction of (s,t)-plane
$\mathbf{L}_{\text{TIK}}^s := \mathbf{I} \otimes \mathbf{L}_{\text{TIK}} \in \mathbb{R}^{n^2 \times n^2}$	Discrete 2nd derivative in s-direction of (s,t)-plane
$\mathbf{L}_{\text{TIK}}^t := \mathbf{L}_{\text{TIK}} \otimes \mathbf{I} \in \mathbb{R}^{n^2 \times n^2}$	Discrete 2nd derivative in t-direction of (s,t)-plane
$\mathbf{L}_{\text{TV}}^s := \mathbf{D}_s^T \Psi \mathbf{D}_s \in \mathbb{R}^{n^2 \times n^2}$	$\mathbf{L}_{\text{TV}}$ operator in s-direction of (s,t)-plane
$\mathbf{L}_{\text{TV}}^t := \mathbf{D}_t^T \Psi \mathbf{D}_t \in \mathbb{R}^{n^2 \times n^2}$	$\mathbf{L}_{\text{TV}}$ operator in t-direction of (s,t)-plane

Matrix Operations

$\otimes$	Kronecker product
$\operatorname{vec}(\cdot)$	Column-stacking operation

Abstract

Many scientific experiments such as those found in astronomy, geology, microbiology, and X-ray radiography require the use of high-energy instruments to capture images. Due to the imaging system, blur and added noise are inevitably present. Oftentimes the captured images must be deblurred to extract valuable information. In the presence of noise, image deblurring is an ill-posed inverse problem in which regularization is required to obtain useful reconstructions. Choosing the appropriate strength of the regularization, however, is difficult. Moreover, many images contain some mixture of smooth features and edges which requires the use of multi-regularization, i.e., the type of regularization (total variation or Tikhonov) varies across the image. We address these two issues by formulating the image deblurring problem within a hierarchical Bayesian framework, varying both the strength of the regularization, as well as the regularization type across the image. In this way, both the image and the strength of the regularization, which varies across the image, are described by a hierarchical posterior distribution which we can sample by Markov chain Monte Carlo (MCMC), in particular Gibbs samplers that make use of conditional distributions for efficient sampling. We compute the means of the image and parameter samples for simulated test images, and we compare our results with existing non-spatially-varying Bayesian methods to show that our new method both increases the quality and decreases the error of the image reconstruction.

Chapter 1

Introduction

Deblurring images is necessary in many applications, including astronomy [14], geology [31, 36], microscopy [49], and X-ray radiography [20, 21]. Depending on the application, blurred images can result from atmospheric turbulence, radiation, the object of interest being in motion, or the imaging system itself [4, 10, 19, 29]. In this work, we are interested in deblurring as it applies to images captured by high-energy X-ray systems, such as *Cygnus*, located at the Nevada National Security Site (NNSS) [35]. Cygnus is used to observe dynamic subcritical experiments in support of the Stockpile Stewardship program, which is in place to ensure the nation's remaining nuclear weapons stockpile remains safe, secure, and effective. Cygnus produces high resolution (4,096×4,096) images that experience the effects of blur and added noise from the system. Deblurring images can be challenging, as typical reconstruction methods require the user to tune parameters to obtain useful reconstructions. Further, deblurring Cygnus images is computationally challenging due to the images' large size. In this work, we develop a new method using a Bayesian framework with spatially varying hyperparameters to better deblur X-ray images. The Bayesian paradigm allows us to eliminate the hand-tuning procedure of the parameters in the traditional reconstruction methods.

Image blurring is formulated as the convolution $\mathbf{B} = \mathbf{a} * \mathbf{X}$, where $\mathbf{X}$ is the original image, $\mathbf{a}$ is the blurring kernel, and '$*$' is the convolution operator. The act of retrieving $\mathbf{X}$ given $\mathbf{B}$ and $\mathbf{a}$ is called *deconvolution.* Additional noise is inevitably present in captured images, due to small errors in the imaging system [29]. It is, perhaps, more appropriate to write the forward problem of blurring an image as $\mathbf{B} = \mathbf{a} * \mathbf{X} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is added noise. The inverse problem of interest is to estimate the true image $\mathbf{X}$ given the corrupted image $\mathbf{B}$, the blurring kernel $\mathbf{a}$, and assumptions about the statistics of the noise. Deconvolution is an ill-posed inverse problem in the sense that small perturbations to the data can produce drastically different solutions [18, 26, 45].

A common starting point in linear models is to obtain the *least squares* solution. The least squares solution $\mathbf{X}^*$ minimizes the error between the data $\mathbf{B}$ and the convolution $\mathbf{a} * \mathbf{X}$ as measured by the 2-norm. Due to added noise, least squares solutions to the deblurring problem are highly unstable, producing large and oscillatory reconstructions. To combat this instability, we *regularize* the problem by bounding the size of the least squares solution and making use of prior information. For example, in the context of image deblurring we may know the scene has smooth features. We would then solve the deconvolution problem with an additional requirement that solutions be smooth and bounded. This is a common technique called *Tikhonov* regularization [3, 19, 22, 45]. If, on the other hand, we know that the scene has sharp edges and constant features, we would use *Total Variation (TV)* regularization, which is known to preserve edges in signal and image reconstructions [20, 38, 45].

In either case, the main challenge in applying regularization to the deconvolution problem is determining how much regularization is required. Let $\mathbf{R}(\mathbf{X})$ represent regularization on the unknown image of interest. The deconvolution problem with regularization amounts to finding the minimizer $\mathbf{X}^*$ of the function

$$\mathcal{J}(\mathbf{X}) := ||\mathbf{a} * \mathbf{X} - \mathbf{B}||_2^2 + \alpha \mathbf{R}(\mathbf{X}),$$

where $||\cdot||_2$ is the 2-norm, and α is a *regularization parameter*. The regularization parameter is a non-negative weight attached to the added constraint of $\mathbf{R}(\mathbf{X})$. Notice that choosing $\alpha = 0$ eliminates the regularization term, and the minimizer of $\mathcal{J}(\mathbf{X})$ is the least squares solution. Selecting appropriate values for the regularization parameter is a delicate process, as it determines a trade-off between the solution's fidelity to the data and the desired features set by the choice of regularization. If the parameter value is too large then more weight is placed on regularization, and solutions will deviate from the true image. On the other hand, if the parameter value is too small then not enough regularization is present, and the added noise will once again affect the solution, causing unwanted artifacts. Recently work has been done to reformulate the deconvolution problem under a Bayesian framework, where the unknown parameters can be treated as random variables [3].

The Bayesian approach to deblurring requires a *prior distribution* on the unknown true image and a *likelihood function*. The prior distribution encodes prior information one might know about the true image, while the likelihood function measures how well a given sample fits the associated inverse model. The product of the likelihood function and prior distribution yields the *posterior distribution*, a distribution of the unknown true image conditioned on known measurements. Under this model, the best approximation of the data is the sample that maximizes the posterior distribution, called the maximum a posteriori (MAP) estimator. For more information on the theory of the Bayesian approach to inverse problems, we reference [22, 26, 27, 42].

In the case of reconstructing a blurred and noisy image $\mathbf{B}$ with regularization, we define a prior that incorporates assumptions about the true image $\mathbf{X}$ (e.g. smoothness, presence of edges, etc.). Referring back to the function $\mathcal{J}(\mathbf{X})$, the prior typically includes the added regularization term $\alpha\mathbf{R}(\mathbf{X})$ in the classical optimization problem, while the likelihood function involves the least-squares difference $||\mathbf{a} * \mathbf{X} - \mathbf{B}||_2^2$. Assuming the added noise is Gaussian, it is appropriate to define the likelihood as a

Gaussian function and the prior as the exponentiated regularization term. The MAP estimator is then the best approximation of $\mathbf{X}$ under this model for a given regularization parameter value.

In the Bayesian paradigm, any unknown quantities should be treated as random variables. Therefore, we can include the unknown regularization parameter, α, in the Bayesian deconvolution model. Along with the prior on $\mathbf{X}$, we now must define a prior for α. Since α is a parameter of the prior distribution, we call it a *hyperparameter* and its distribution a *hyperprior*. This produces a hierarchical model, and the posterior distribution is now the product of the likelihood, prior, and hyperprior(s). Computing the MAP estimator is much more difficult with a hierarchical model; however, we can still obtain approximations of $\mathbf{X}$ by sampling the posterior. If the likelihood and prior are both Gaussian, and the hyperprior(s) is conjugate to a Gaussian, then a hierarchical Gibbs sampler can be used to obtain samples from the posterior. The "best" approximation of $\mathbf{X}$ in this case is the mean of the samples. We put "best" in quotations, because the mean of the samples better approximates the mean of the posterior as the number of samples increases.

Hierarchical sampling of Bayesian models has been applied to applications in X-ray imaging [3, 20, 21]. In [20], for example, the authors use a hierarchical Bayesian framework on another linear inverse problem, Abel inversion. The authors apply Abel inversion with TV regularization to an X-ray image of a radially symmetric object in order to measure the object's density. Samples are drawn from the constructed posterior using a hierarchical Gibbs sampler, and estimations are given by the mean of the samples. In simulations where the true density profile is known, the approximated density profile exhibits "staircasing" artifacts and intervals of over-regularization common with TV regularization.

Extending on the existing hierarchical Bayesian framework for deblurring, we propose that hyperpriors should vary pixel-wise over the image. The authors in [6] have a similar construction where they assume a conditional Gaussian prior for the image with a spatially varying variance. However, the authors employ iterative solvers instead of hierarchical sampling. Hierarchical sampling can be a more expensive process; however, it allows us to quantify uncertainties, which can be valuable information for real-life applications. Our model also incorporates a predetermined spatial partitioning of Tikhonov and TV regularization. Images of interest often include both smooth and solid features with edges, making this incorporation of the two techniques necessary.

In Chapter 2, we present more detailed background information necessary to understand the deblurring problem. We first formulate the problem in the simpler 1D case, and we then extend to 2D. We also provide information on Tikhonov and TV regularization before formulating the problem in the Bayesian framework. We define the prior distribution, hyperprior distributions, and likelihood function that make up the desired posterior distribution. We then introduce the basic theory of Markov chain Monte Carlo (MCMC) methods, which are commonly employed to sample from posterior distributions. In particular, we focus on hierarchical Gibbs sampling, a special kind of MCMC method. In Chapter 3, we present the hierarchical Bayesian model that includes spatially varying hyperparameters. In Chapter 4, we extend on the work in Chapter 3 to include an effective switching between Tikhonov and TV regularization. We present numerical results in Chapter 5, applying our method to simulated blurred and noisy images, as well as a section of a test image captured by Cygnus. We conclude in Chapter 6, summarizing our method and results.

Chapter 2

Background

In this chapter, we provide background on discrete convolution, regularization, the Bayesian formulation of the deblurring problem, and numerical methods to obtain solutions. Discrete convolution is introduced in both the 1D and 2D case. We then discuss the need for regularization and how the regularized problem lends itself to a Bayesian formulation. Finally, we present Markov Chain Monte Carlo (MCMC) methods.

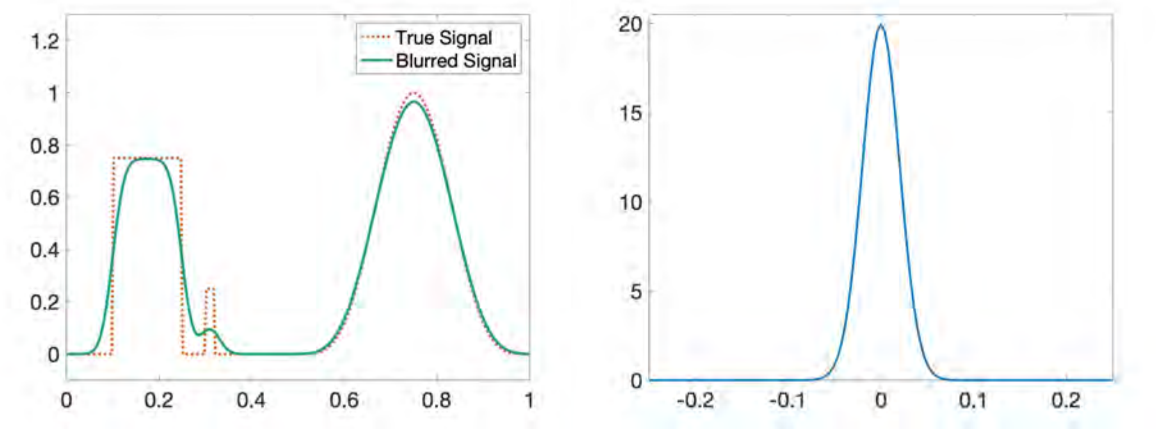
2.1 Discrete Convolution

The blurring process can be modeled by convolution. We start by formulating the simpler case of 1D convolution and then extend to 2D, providing the discretized version of both operators. We follow the notation used in [3].

2.1.1 Discrete Convolution in 1D

In the continuous case, the convolution of a function x with kernel a can be written as

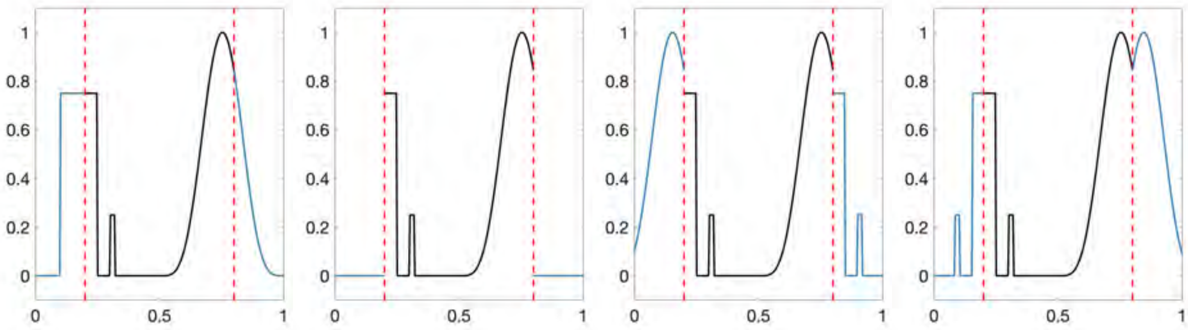
$$b(s) = \int_{-\infty}^{\infty} a(s-t)x(t)\ \mathrm{d}t. \tag{2.1.1}$$



(a) Blurred Signal (b) Blurring Kernel

Figure 2.1.1: The left figure shows a signal (dotted orange line) that has been blurred and corrupted by added noise (solid green line). The blurred signal is the result of convolving the true signal with Gaussian kernel shown in the right figure.

As an example, assume the dashed signal shown in Figure 2.1.1a is convolved with the Gaussian kernel in Figure 2.1.1b. The convolution of the true signal with the Gaussian kernel results in the blurred signal in Figure 2.1.1a. Following our notation, the original signal is denoted by $x(s)$ and the Gaussian kernel is denoted by $a(s)$, where $s \in \mathbb{R}$. As equation (2.1.1) suggests, the convolution of $x(s)$ with the kernel $a(s)$ amounts to "sweeping" the kernel from left to right across the signal, and averaging the areas underneath both curves. Within the interval $[0.1, 0.9]$, the convolution results in a signal $b(s)$ that is similar to $x(s)$, but with damped edges. However, during the "sweeping" process, the kernel extends outside of the signal's boundary. We, therefore, must make assumptions about the behavior of $x(s)$ outside our field of view, the interval $[0, 1]$. Since $x(s)$ is already zero at the boundaries, it is reasonable to assume zero boundary conditions, i.e., $x(s) = 0$ for all $s \notin [0, 1]$. If our field of view were restricted to the interval $[0.2, 0.8]$, then zero boundary conditions would be less appropriate. The most common choices in the literature are zero, periodic, or reflecting boundary conditions. The leftmost subfigure in Figure 2.1.2 shows the original signal with two vertical dashed lines. The dashed lines indicate a restriction



(a) Original Signal (b) Zero BC (c) Periodic BC (d) Reflecting BC

Figure 2.1.2: Three examples of imposed boundary conditions on the signal in Subfigure (a). The vertical dashed lines represent a restriction of our field of view, i.e., in our application we are only able to see the piece of the signal in between the vertical dashed line. The extension of the signal in Subfigure (a) beyond the dashed lines is the true behavior of the signal. Subfigure (b) shows the signal if we were to impose zero boundary conditions, i.e., the signal's value is zero beyond the vertical dashed lines. Subfigure (c) shows the signal if we were to assume periodic behavior beyond the dashed lines, and Subfigure (d) shows a reflection of the signal in our field of view across the dashed lines.

on our field of view, i.e., we only see the piece of the signal in between the two vertical dashed lines. The remaining three subfigures show the observed signal along with zero, periodic, and reflecting boundary conditions, in that order. Of course, none of the three assumptions on the boundaries match what we know to be the true behavior of the signal outside our field of view. Choosing the correct boundary conditions can be difficult and should be done with care. Side-stepping this difficulty, we generate simulated examples that have value zero at the boundaries, so that we may assume zero boundary conditions. We refer the reader to [3, 19, 45] for more information on constructing the discretized convolution problem under periodic and reflecting boundary conditions.

Returning to our example, we assume our field of view is $[0,1]$ and x has zero boundary conditions. Therefore, (2.1.1) becomes

$$b(s) = \int_0^1 a(s-t)x(t)\ \mathrm{d}t \tag{2.1.2}$$

To numerically apply convolution, we must discretize the continuous convolution equation using midpoint quadrature. Let n be the number of equidistant points in $[0,1]$ and $h = 1/n$ be the step size. Let $s = (i - 1/2)h$ and $t = (j - 1/2)h$, for $i, j = 1, \ldots, n$. Then (2.1.2) can be written as

$$b((i-1/2)h) = h\sum_{j=1}^{n} a((i-j)h)x((j-1/2)h). \tag{2.1.3}$$

For simplicity, let $b_i = b((i-1/2)h)$; $a_{i-j} = a((i-j)h)$; and $x_j = x((j-1/2)h)$. Then in matrix-vector form, (2.1.3) becomes

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} = h \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-n+1} \\ a_1 & a_0 & a_{-1} & \cdots & a_{-n+2} \\ a_2 & a_1 & a_0 & \cdots & a_{-n+3} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}. \tag{2.1.4}$$

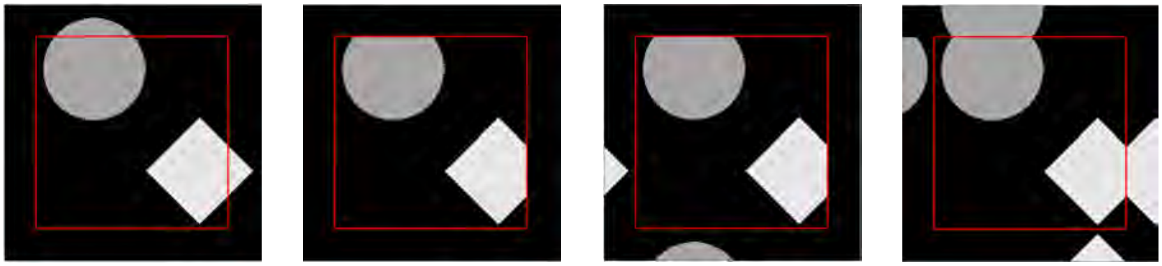
Notice that the blurring matrix has constant diagonals. Matrices with constant diagonals are called *Toeplitz* matrices [3, 45], and all blurring matrices for the 1D blurring problem with zero boundary conditions are of this form. Let $\mathbf{A}$ be the Toeplitz matrix in (2.1.4) multiplied by the step size h. Let $\mathbf{b}$ and $\mathbf{x}$ be the vectors of b_i and x_i values, respectively. Then the forward problem of blurring a signal simply becomes

$$\mathbf{b} = \mathbf{A}\mathbf{x}. \tag{2.1.5}$$

Of course, in applications there are inevitably errors in the measured data. Therefore, the forward problem (2.1.5) is more realistically written as

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\varepsilon}, \tag{2.1.6}$$

where $\boldsymbol{\varepsilon}$ is an $n \times 1$ vector such that $||\boldsymbol{\varepsilon}||_2 \ll 1$, representing small unknown measure-



(a) Original Image (b) Zero BC (c) Periodic BC (d) Reflecting BC

Figure 2.1.3: Three examples of imposed boundary conditions on the image in Subfigure (a). The red box is a restriction of our field of view. Subfigure (b) shows the image if we were to impose zero boundary conditions; Subfigure (c) shows the image if we assume periodic behavior; and Subfigure (d) shows a reflection of the image on the outside of the box.

ment errors.

2.1.2 Discrete Convolution in 2D

The construction of the discrete convolution problem in 2D is similar to that in 1D, just with more indices to account for. In 2D, the convolution of x with kernel a can be written as

$$b(s,u) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} a(s-t,u-v)x(t,v)\ \mathrm{d}t\ \mathrm{d}v. \tag{2.1.7}$$

Again, assumptions on the boundary conditions are essential in constructing the convolution problem because the kernel extends beyond the field of view. As in 1D, the most common assumptions are zero, periodic, and reflecting boundary conditions. Figure 2.1.3 shows an image in the leftmost subfigure, with a red box indicating our restricted field of view. The following three subfigures shows the image's restricted field of view along with zero, periodic, and reflecting boundary conditions. We once again avoid the challenge of choosing boundary conditions by using examples where zero boundary conditions are the appropriate choice.

Now, assuming $x(s,u) = 0$ for all $s,u \notin [0,1]\times[0,1]$, the 2D continuous convolution equation in (2.1.7) becomes

$$b(s,u) = \int_0^1 \int_0^1 a(s-t, u-v)x(t,v)\ \mathrm{d}t\ \mathrm{d}v. \tag{2.1.8}$$

For consistency, we keep the same discretization scheme on the interval $[0,1]$ as in the 1D case. Our region of interest is $[0,1] \times [0,1]$, so let $s = (i-1/2)h$; $u = (\ell - 1/2)h$; $t = (j-1/2)h$; and $v = (m-1/2)h$ for $i, j, \ell, m = 1, \ldots, n$. Define $b_{i,\ell} = b((i-1/2)h, (\ell-1/2)h)$; $a_{i-j,\ell-m} = a((i-j)h, (\ell-m)h)$; and $x_{j,m} = x((j-1/2)h, (m-1/2)h)$. Then we can write (2.1.8) as

$$b_{i,\ell} = h^2 \sum_{j=1}^{n} \sum_{m=1}^{n} a_{i-j,\ell-m} x_{j,m}. \tag{2.1.9}$$

We say a function $a(s,u)$ is *separable* if $a(s,u) = a_1(s)a_2(u)$, where a_1 and a_2 are single-variable functions. Assume the blurring kernel a is separable; and in the discretized case, let $a_{i,\ell} = a_i^{(1)} a_\ell^{(2)}$. Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be $n \times n$ Toeplitz matrices resembling the blurring matrix in (2.1.4), i.e.,

$$\mathbf{A}_k = \begin{bmatrix} a_0^{(k)} & a_{-1}^{(k)} & a_{-2}^{(k)} & \cdots & a_{-n+1}^{(k)} \\ a_1^{(k)} & a_0^{(k)} & a_{-1}^{(k)} & \cdots & a_{-n+2}^{(k)} \\ a_2^{(k)} & a_1^{(k)} & a_0^{(k)} & \cdots & a_{-n+3}^{(k)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n-1}^{(k)} & a_{n-2}^{(k)} & a_{n-3}^{(k)} & \cdots & a_0^{(k)} \end{bmatrix}, \tag{2.1.10}$$

for $k = 1, 2$. Then continuing from (2.1.9),

$$\begin{aligned} b_{i,\ell} &= h^2 \sum_{j=1}^{n} \sum_{m=1}^{n} a_{i-j,\ell-m} x_{j,m}, \\ &= h \sum_{j=1}^{n} a_{i-j}^{(1)} \left(h \sum_{m=1}^{n} a_{\ell-m}^{(2)} x_{j,m} \right), \\ &= h \sum_{j=1}^{n} a_{i-j}^{(1)} \left[\mathbf{A}_2 \mathbf{X}^T \right]_{\ell,j}, \\ &= h \sum_{j=1}^{n} a_{i-j}^{(1)} \left[\mathbf{X} \mathbf{A}_2^T \right]_{j,\ell}, \\ &= \left[\mathbf{A}_1 \mathbf{X} \mathbf{A}_2^T \right]_{i,\ell}. \end{aligned} \tag{2.1.11}$$

Therefore, letting $\mathbf{B}$ be the matrix of values $b_{i,\ell}$ for $i, \ell = 1, \ldots, n$, we have that the discrete forward problem of blurring an image is

$$\mathbf{B} = \mathbf{A}_1 \mathbf{X} \mathbf{A}_2^T. \tag{2.1.12}$$

Taking into account the presence of added noise, (2.1.12) is more appropriately written as

$$\mathbf{B} = \mathbf{A}_1 \mathbf{X} \mathbf{A}_2^T + \mathbf{E}, \tag{2.1.13}$$

where $\mathbf{E}$ is an $n \times n$ matrix representing noise. There is, however, a simpler way to write (2.1.13) by vectorizing and using properties of Kronecker products.

We define the act of *vectorizing* a matrix by stacking the columns into one vector. More formally, let $\mathbf{C}$ be an $m \times n$ matrix defined by

$$\mathbf{C} = \left[\mathbf{c}^{(1)} \;\middle|\; \mathbf{c}^{(2)} \;\middle|\; \cdots \;\middle|\; \mathbf{c}^{(n)} \right], \tag{2.1.14}$$

where $\mathbf{c}^{(i)}$ is the ith column vector. Define

$$\operatorname{vec}(\mathbf{C}) = \begin{bmatrix} \mathbf{c}^{(1)} \\ \mathbf{c}^{(2)} \\ \vdots \\ \mathbf{c}^{(n)} \end{bmatrix}. \tag{2.1.15}$$

The operator $\operatorname{vec}(\cdot)$ takes in a $m \times n$ matrix, stacks its column vectors, and returns a vector of length mn.

For two $m \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$, we define the *Kronecker product* as follows:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \cdots & \cdots & a_{1,n}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \cdots & \cdots & a_{2,n}\mathbf{B} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{m,1}\mathbf{B} & a_{m,2}\mathbf{B} & \cdots & \cdots & a_{m,n}\mathbf{B} \end{bmatrix}_{mn \times mn}. \tag{2.1.16}$$

It is known (see [3, 19]) for $\mathbf{C} = \mathbf{AXB}$,

$$\operatorname{vec}(\mathbf{C}) = \operatorname{vec}(\mathbf{AXB}) = \left(\mathbf{B}^T \otimes \mathbf{A}\right) \operatorname{vec}(\mathbf{X}). \tag{2.1.17}$$

Using this property, we vectorize equation (2.1.13):

$$\begin{aligned} \operatorname{vec}(\mathbf{B}) &= \operatorname{vec}(\mathbf{A}_1\mathbf{X}\mathbf{A}_2^T + \mathbf{E}) && (2.1.18) \\ &= \operatorname{vec}(\mathbf{A}_1\mathbf{X}\mathbf{A}_2^T) + \operatorname{vec}(\mathbf{E}) && (2.1.19) \\ &= (\mathbf{A}_2 \otimes \mathbf{A}_1) \operatorname{vec}(\mathbf{X}) + \operatorname{vec}(\mathbf{E}). && (2.1.20) \end{aligned}$$

Let $\widetilde{\mathbf{b}} = \operatorname{vec}(\mathbf{B}); \widetilde{\mathbf{x}} = \operatorname{vec}(\mathbf{X}); \widetilde{\boldsymbol{\varepsilon}} = \operatorname{vec}(\mathbf{E})$; and $\widetilde{\mathbf{A}} = \mathbf{A}_2 \otimes \mathbf{A}_1$. Then (2.1.13) becomes

$$\widetilde{\mathbf{b}} = \widetilde{\mathbf{A}}\widetilde{\mathbf{x}} + \widetilde{\boldsymbol{\varepsilon}}, \tag{2.1.21}$$

exactly the same form as in the 1D case.

2.2 Deconvolution and Regularization

Deconvolution is an ill-posed inverse problem, in the sense that small perturbations to the data can yield drastically different solutions [18, 26]. By adding an additional term to the problem where we minimize the least squares difference between the model and the observation, we obtain more stable solutions. This process is called *regularization*, and the additional term we introduce is called a *regularization term*. Therefore, we are interested in minimizing functions of the form

$$\mathcal{J}_\delta(\mathbf{x}) = \frac{1}{2}\left|\left|\mathbf{A}\mathbf{x} - \mathbf{b}\right|\right|_2^2 + \delta\mathcal{R}(\mathbf{x}), \tag{2.2.1}$$

where $\mathbf{x}$ is a variable representing an $N \times 1$ vectorized image, δ is a nonnegative regularization parameter, and $\mathcal{R} : \mathbb{R}^n \to \mathbb{R}$ is a penalty function based on prior information. A common technique is called *Tikhonov* regularization, where the regularization term is defined by the squared Euclidean norm of $\mathbf{x}$ or some function of $\mathbf{x}$ [3, 45]. In this case, the function $\mathcal{R}(\mathbf{x}) = \frac{1}{2}\left|\left|\mathbf{L}_{\text{TIK}}\mathbf{x}\right|\right|_2^2$, where $\mathbf{L}_{\text{TIK}}$ is commonly chosen to be the identity $\mathbf{I}$ or discrete gradient operators. Another popular technique is to let the regularization term be the 1-norm of the gradient of the image. Minimizing this term is equivalent to minimizing the total variation of the image; hence, this technique is called *total variation* or (TV) regularization (first introduced in [38]). With TV regularization, $\mathcal{R}(\mathbf{x}) = \left|\left|\mathbf{D}\mathbf{x}\right|\right|_1$, where in 1D the matrix $\mathbf{D}$ is the discrete first derivative operator, and in the 2D (s,t)-coordinate space $\mathbf{D} = [\mathbf{D}_s\ \mathbf{D}_t]^T$. The choice of regularization depends on prior information about the image's features. Tikhonov regularization is ideal for producing smooth reconstructions, while TV regularization is well-suited for preserving edges in the reconstructions.

No matter which regularization technique is used, the delicate process of assigning a value to the parameter, δ, must be performed. The value determines the trade-off between how well the reconstruction matches the forward model of the original data and the extent to which the chosen regularization term is enforced. If the regularization parameter is too small, then the regularization is not enforced enough, causing noise to affect the reconstruction. On the other hand, if the parameter is too large then there is too much regularization, and the reconstruction loses fidelity to the original data.

Examples of common parameter selection methods are generalized cross validation (GCV) [16], the discrepancy principle (DP) [34], and the L-curve method [17, 30]. However, these methods are also flawed in that (1) they sometimes require information about the added noise level which (as in our case) is not always available and (2) they fail to converge when the noise level approaches zero [3, 19]. Each method has its pros and cons, and the user must decide which selection method is most appropriate for the given problem. Not only is choosing the regularization parameter a delicate process, but so is choosing the parameter selection method. By using a Bayesian framework, we can bypass this issue altogether by allowing the parameter to be a random variable.

2.3 Bayesian Formulation

In this section, we rewrite the deblurring problem using a Bayesian framework. This requires an assumption on the statistical properties of the added noise. For this work, we assume the added noise is Gaussian with variance λ^{-1}. We start by assuming λ is known, along with the regularization parameter δ. We then assume the two parameters are unknown, and following the Bayesian paradigm, we let them be random variables. This leads to the hierarchical Bayesian model that is the foundation of this work.

2.3.1 Fixed Parameters

Assume $\lambda, \delta > 0$ are known. The Bayesian framework for inverse problems calls for all unknowns to be random variables. Therefore, our unknown is the image $\mathbf{x} \in \mathbb{R}^n$. In both 1D and 2D, our problem formulation is to estimate $\mathbf{x} \in \mathbb{R}^n$ given $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{n\times n}$ where

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\varepsilon}. \tag{2.3.1}$$

Assume $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I}_n)$. Note that if $\mathbf{x}$ were given, then $\mathbf{b} \sim \mathcal{N}(\mathbf{A}\mathbf{x}, \lambda^{-1}\mathbf{I}_n)$. We define the likelihood, i.e., the function measures a given $\mathbf{x}$'s goodness of fit to the observation $\mathbf{b}$, as

$$p_{\mathrm{LH}}(\mathbf{b}|\mathbf{x}) \propto \exp\left(-\frac{\lambda}{2}\,||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2\right). \tag{2.3.2}$$

The prior distribution contains prior information about the unknown image $\mathbf{x}$. Prior distributions are commonly of the form

$$p_0(\mathbf{x}) \propto \exp\left(-\delta\mathcal{R}(\mathbf{x})\right). \tag{2.3.3}$$

The posterior distribution is constructed using Bayes' theorem, and is made up of the likelihood and prior in the following way:

$$p(\mathbf{x}|\mathbf{b}) \propto p_{\mathrm{LH}}(\mathbf{b}|\mathbf{x})p_0(\mathbf{x}) \propto \exp\left(-\frac{\lambda}{2}\,||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 - \delta\mathcal{R}(\mathbf{x})\right). \tag{2.3.4}$$

Notice that the maximum a posteriori (MAP) estimator is

$$\begin{aligned}
\mathbf{x}_{MAP} &= \arg\max_{\mathbf{x}} \left\{p(\mathbf{x}|\mathbf{b})\right\} && (2.3.5)\\
&= \arg\min_{\mathbf{x}} \left\{-\ln(p(\mathbf{x}|\mathbf{b}))\right\} && (2.3.6)\\
&= \arg\min_{\mathbf{x}} \left\{\frac{\lambda}{2}\,||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 + \delta\mathcal{R}(\mathbf{x})\right\}. && (2.3.7)
\end{aligned}$$

The advantage to using a Bayesian framework in the image deblurring problem is that the solution is not one reconstruction, but a *distribution* of reconstructions. The posterior distribution is a distribution of image reconstructions $\mathbf{x}$, and to each $\mathbf{x}$ is attached a probability on how likely it is to be the optimal reconstruction based on how well it matches the given data and prior information.

2.3.2 Hierarchical Formulation

Now assume λ and δ are unknown. Following [3], we assume the parameters have Gamma prior distributions

$$p_{\mathrm{H1}}(\lambda) \sim \mathrm{Gamma}(\alpha_\lambda, \beta_\lambda) \propto \lambda^{\alpha_\lambda - 1} \exp(-\beta_\lambda \lambda), \tag{2.3.8}$$

$$p_{\mathrm{H2}}(\delta) \sim \mathrm{Gamma}(\alpha_\delta, \beta_\delta) \propto \delta^{\alpha_\delta - 1} \exp(-\beta_\delta \delta), \tag{2.3.9}$$

where $\alpha_\lambda, \alpha_\delta$ and $\beta_\lambda, \beta_\delta$ are shape and scale parameters, respectively. In this work, we choose $\alpha_\lambda = \alpha_\delta = 1$ and $\beta_\lambda = \beta_\delta = 10^{-4}$, following [3]. The reasoning is that for these shape and scale parameter values, the Gamma distribution behaves like a uniform distribution. This formulation, therefore, gives us conjugate priors as well as (approximately) uniform distributions from which to sample the unknown λ and δ parameters.

The distributions $p_{H1}(\lambda)$ and $p_{H2}(\delta)$ are called hyperpriors, as they are distributions for parameters included in the likelihood and prior distributions. Bayes' theorem allows us to incorporate the hyperpriors in the following way:

$$\begin{aligned} p(\mathbf{x},\lambda, \delta|\mathbf{b}) &\propto p_{\mathrm{LH}}(\mathbf{b}|\mathbf{x}, \lambda) p_0(\mathbf{x}|\delta) p_{\mathrm{H1}}(\lambda) p_{\mathrm{H2}}(\delta), \\ &\propto \lambda^{n/2+\alpha_\lambda-1} \delta^{n/2+\alpha_\delta-1} \exp\left(-\frac{\lambda}{2} ||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 - \delta \mathcal{R}(\mathbf{x}) - \beta_\lambda \lambda - \beta_\delta \delta\right). \end{aligned} \tag{2.3.10}$$

The posterior distribution not only contains possible reconstructions $\mathbf{x}$, but also possible values for the parameters λ and δ.

2.4 Numerical Methods

In the Bayesian framework, obtaining image reconstructions involves sampling from the posterior distribution, which is not always an easy task. In this section, we introduce Markov chain Monte Carlo (MCMC) methods, a class of algorithms that allow us to sample from more complicated distributions. We highlight one MCMC method in particular – Gibbs sampling.

2.4.1 MCMC and Gibbs Sampling

Suppose we want to sample from a general multivariate posterior distribution $p\left(\mathbf{y}\right)$. A Markov chain Monte Carlo (MCMC) method constructs a Markov chain that, under the right conditions [5, 39], will converge to the distribution of interest. A Markov chain is a sequence of random variables $\left\{\mathbf{y}^{(k)}\right\}$ that satisfies the Markov property:

$$p\left(\mathbf{y}^{(k)}|\mathbf{y}^{(1)},\mathbf{y}^{(2)},\ldots,\mathbf{y}^{(k-1)}\right)=p\left(\mathbf{y}^{(k)}|\mathbf{y}^{(k-1)}\right), \tag{2.4.1}$$

for $k \geq 2$. The Markov property states that the current state $\mathbf{y}^{(k)}$ depends only on the previous state $\mathbf{y}^{(k-1)}$.

In MCMC, the Markov chain is built with the convergence to the equilibrium distribution in mind. However, it may take a variable amount of time (called a burn-in period) for the chain to reach equilibrium. Samples taken before equilibrium should be discarded, as they are not true samples of the posterior distribution. It is also important to note that the samples generated from an MCMC method are correlated due to the Markov property, which can affect the efficiency of the algorithm. The integrated autocorrelation time (IACT), denoted by τ_{int}, is a useful measure of efficiency. For a chain of MCMC samples $\mathbf{y}^{(1)},\mathbf{y}^{(2)},\ldots,\mathbf{y}^{(K)}$, the IACT is computed by

$$\tau_{\text{int}} = 1 + 2\sum_{k=1}^{K} C_k/C_0, \tag{2.4.2}$$

where C_j is the autocovariance function

$$C_j = \frac{1}{K}\sum_{i=1}^{K-j}\left(\mathbf{y}^{(i)} - \mu_K\right)\left(\mathbf{y}^{(i+j)} - \mu_K\right), \tag{2.4.3}$$

with μ_K as the estimated mean of the K samples. The function C_j/C_0 is called the autocorrelation function. It is shown in [39] that the correlation of samples causes the statistical error to be a factor of τ_{int} larger than that of independent sampling. This means the number of effectively independent samples is given by

$$K_{\text{eff}} = \frac{K}{\tau_{\text{int}}}. \tag{2.4.4}$$

In other words, the closer τ_{int} is to 1, the more efficient the MCMC method.

Suppose we want to generate samples from a multivariate distribution $p(\mathbf{y}, \mathbf{z})$, but the distribution is difficult to sample using standard methods. Assume the conditional distributions $p(\mathbf{y}|\mathbf{z})$ and $p(\mathbf{z}|\mathbf{y})$ are easy to sample and are conjugate distributions, i.e. they belong in the same probability distribution family. In the event that sampling from $p(\mathbf{y}, \mathbf{z})$ is difficult, but sampling from the conditionals $p(\mathbf{y}|\mathbf{z})$ and $p(\mathbf{z}|\mathbf{y})$ is easy, one might consider using a Gibbs sampler [37]. Gibbs sampling (first named in [15]) is a popular MCMC method that constructs the Markov chain by sampling the conditional distributions of the multivariate distribution. Algorithm 1 shows the basic schematic of the sampler. Given an initial guess $\mathbf{z}^{(0)}$, we sample $\mathbf{y}^{(1)} \sim p\left(\mathbf{y}|\mathbf{z}^{(0)}\right)$, and then we sample $\mathbf{z}^{(1)}$ from its conditional distribution conditioned on the sample we just obtained, $\mathbf{y}^{(1)}$. This continues for K iterations, i.e., for $1 \leq k \leq K$, $\mathbf{y}^{(k)} \sim p\left(\mathbf{y}|\mathbf{z}^{(k-1)}\right)$ and $\mathbf{z}^{(k)} \sim p\left(\mathbf{z}|\mathbf{y}^{(k)}\right)$. We note that for large K, the order in which we sample from the conditionals does not matter [3, 39]. We could obtain similar results by initializing $\mathbf{y}^{(0)}$, first sampling $\mathbf{z}^{(k)} \sim p\left(\mathbf{z}|\mathbf{y}^{(k-1)}\right)$ and then $\mathbf{y}^{(k)} \sim p\left(\mathbf{y}|\mathbf{z}^{(k)}\right)$. After a burn-in

Algorithm 1: Gibbs Sampler

```
input: Initial approximation $\mathbf{z}^{(0)}$; maximum number of iterations $K$; and
       conditional distributions $p(\mathbf{y}|\mathbf{z})$ and $p(\mathbf{z}|\mathbf{y})$

for $k = 1, \ldots, K$ do
    $\mathbf{y}^{(k)} \sim p\left(\mathbf{y}|\mathbf{z}^{(k-1)}\right)$
    $\mathbf{z}^{(k)} \sim p\left(\mathbf{z}|\mathbf{y}^{(k)}\right)$
end
```

period, the Gibbs sampler will converge to the equilibrium distribution $p(\mathbf{y}, \mathbf{z})$.

2.4.2 Gibbs Sampling for Hierarchical Inverse Problems

In order to use a Gibbs sampler to sample from the posterior distribution $p\left(\mathbf{x}, \lambda, \delta|\mathbf{b}\right)$ in (2.3.10), the full conditional distributions $p\left(\lambda|\mathbf{b}, \mathbf{x}, \delta\right)$, $p\left(\delta|\mathbf{b}, \mathbf{x}, \lambda\right)$, $p\left(\mathbf{x}|\mathbf{b}, \lambda, \delta\right)$ must exist and be easy to sample. We have

$$\begin{aligned} p\left(\lambda|\mathbf{b}, \mathbf{x}, \delta\right) &\propto \lambda^{n/2+\alpha_\lambda-1} \exp\left(-\lambda\left(\frac{1}{2}\left|\left|\mathbf{A}\mathbf{x}-\mathbf{b}\right|\right|_2^2+\beta_\lambda\right)\right) && (2.4.5) \\ &\sim \text{Gamma}\left(n/2+\alpha_\lambda, \frac{1}{2}\left|\left|\mathbf{A}\mathbf{x}-\mathbf{b}\right|\right|_2^2+\beta_\lambda\right), \\ p\left(\delta|\mathbf{b}, \mathbf{x}, \lambda\right) &\propto \delta^{n/2+\alpha_\delta-1} \exp\left(-\delta\left(\mathcal{R}(\mathbf{x})+\beta_\delta\right)\right) && (2.4.6) \\ &\sim \text{Gamma}\left(n/2+\alpha_\delta, \mathcal{R}(\mathbf{x})+\beta_\delta\right), \end{aligned}$$

which can easily be sampled. The final conditional distribution

$$p\left(\mathbf{x}|\mathbf{b}, \lambda, \delta\right) \propto \exp\left(-\frac{\lambda}{2}\left|\left|\mathbf{A}\mathbf{x}-\mathbf{b}\right|\right|_2^2-\delta\mathcal{R}(\mathbf{x})\right) \tag{2.4.7}$$

can be easily sampled if $\mathcal{R}(\mathbf{x}) = \frac{1}{2}\left|\left|\mathbf{L}_{\text{TIK}}\mathbf{x}\right|\right|_2^2$. In this case, $p\left(\mathbf{x}|\mathbf{b},\lambda,\delta\right)$ is Gaussian. The mean can be determined by finding the minimum of $-\ln\left(p\left(\mathbf{x}|\mathbf{b},\lambda,\delta\right)\right)$. We have

$$\frac{d}{d\mathbf{x}}\left[-\ln\left(p\left(\mathbf{x}|\mathbf{b},\lambda,\delta\right)\right)\right] = \frac{d}{d\mathbf{x}}\left[\frac{\lambda}{2}\left|\left|\mathbf{A}\mathbf{x}-\mathbf{b}\right|\right|_2^2 + \frac{\delta}{2}\left|\left|\mathbf{L}_{\text{TIK}}\mathbf{x}\right|\right|_2^2\right], \tag{2.4.8}$$

$$= \lambda\mathbf{A}^T\left(\mathbf{A}\mathbf{x}-\mathbf{b}\right) + \delta\mathbf{L}_{\text{TIK}}^T\mathbf{L}_{\text{TIK}}\mathbf{x}, \tag{2.4.9}$$

$$= \left(\lambda\mathbf{A}^T\mathbf{A} + \delta\mathbf{L}_{\text{TIK}}^T\mathbf{L}_{\text{TIK}}\right)\mathbf{x} - \lambda\mathbf{A}^T\mathbf{b}, \tag{2.4.10}$$

which equals zero when $\mathbf{x} = \lambda\left(\lambda\mathbf{A}^T\mathbf{A} + \delta\mathbf{L}_{\text{TIK}}^T\mathbf{L}_{\text{TIK}}\right)^{-1}\mathbf{A}^T\mathbf{b}$. The precision matrix can be obtained by the second derivative of the negative log-likelihood with respect to $\mathbf{x}$, which is $\lambda\mathbf{A}^T\mathbf{A} + \delta\mathbf{L}_{\text{TIK}}^T\mathbf{L}_{\text{TIK}}$. Therefore, the conditional distribution is

$$p\left(\mathbf{x}|\mathbf{b},\lambda,\delta\right) \sim \mathcal{N}\left((\lambda\mathbf{A}^T\mathbf{A} + \delta\mathbf{L}_{\text{TIK}}^T\mathbf{L}_{\text{TIK}})^{-1}\mathbf{A}^T\mathbf{b}, (\lambda\mathbf{A}^T\mathbf{A} + \delta\mathbf{L}_{\text{TIK}}^T\mathbf{L}_{\text{TIK}})^{-1}\right). \tag{2.4.11}$$

If we were to instead choose $\mathcal{R}(\mathbf{x}) = \left|\left|\mathbf{D}\mathbf{x}\right|\right|_1$, then $p(\mathbf{x}|\mathbf{b},\lambda,\delta)$ in (2.4.7) would not be Gaussian. However, we can approximate $\left|\left|\mathbf{D}\mathbf{x}\right|\right|_1$ by the squared 2-norm of a different operator applied to $\mathbf{x}$, thereby making $p(\mathbf{x}|\mathbf{b},\lambda,\delta)$ a Gaussian distribution. For some small value η, notice

$$|x| \approx \sqrt{x^2+\eta}\ . \tag{2.4.12}$$

Although the absolute value function is non-differentiable, we can approximate the derivative by

$$\frac{d}{dx}|x| \approx \frac{x}{\sqrt{x^2+\eta}}\ . \tag{2.4.13}$$

We extend this to $\left|\left|\mathbf{D}\mathbf{x}\right|\right|_1$. We have

$$\left|\left|\mathbf{D}\mathbf{x}\right|\right|_1 = \sum_{i=1}^{n}\left|(\mathbf{D}\mathbf{x})_i\right| \approx \sum_{i=1}^{n}\sqrt{(\mathbf{D}\mathbf{x})_i^2+\eta}\ . \tag{2.4.14}$$

The approximated derivative is then

$$\begin{aligned}\frac{d}{d\mathbf{x}}\left|\left|\mathbf{D}\mathbf{x}\right|\right|_1 &= \sum_{i=1}^{n}\frac{d}{d\mathbf{x}}\left|(\mathbf{D}\mathbf{x})_i\right| \\ &\approx \sum_{i=1}^{n}\frac{d}{d\mathbf{x}}\sqrt{(\mathbf{D}\mathbf{x})_i^2+\eta}\ , \\ &= \sum_{i=1}^{n}\frac{\left(\mathbf{D}^T\mathbf{D}\mathbf{x}\right)_i}{\sqrt{(\mathbf{D}\mathbf{x})_i^2+\eta}} \\ &= \mathbf{D}^T\Phi(\mathbf{x})\mathbf{D}\mathbf{x}, \end{aligned} \tag{2.4.15}$$

where $\Phi(\mathbf{x})$ is an $n \times n$ diagonal matrix with entries $\left((\mathbf{D}\mathbf{x})_i^2+\eta\right)^{-1/2}$ for $i = 1, \ldots, n$. This motivates the definition of the operator

$$\mathbf{L}_{\text{TV}}(\mathbf{x}) := \Psi(\mathbf{x})\mathbf{D}, \tag{2.4.16}$$

where $\Psi(\mathbf{x})$ is the element-wise square root of $\Phi(\mathbf{x})$, i.e., $\Psi(\mathbf{x})$ is an $n \times n$ diagonal matrix with entries $\left((\mathbf{D}\mathbf{x})_i^2+\eta\right)^{-1/4}$ for $i = 1, \ldots, n$. With this definition, we have that $\left(\mathbf{L}_{\text{TV}}(\mathbf{x})\right)^T\mathbf{L}_{\text{TV}}(\mathbf{x})\mathbf{x} \approx \frac{d}{d\mathbf{x}}\left[\left|\left|\mathbf{D}\mathbf{x}\right|\right|_1\right]$. Notice the operator $\mathbf{L}_{\text{TV}}$ depends on $\mathbf{x}$. In this work, we let $\mathbf{L}_{\text{TV}} := \mathbf{L}_{\text{TV}}(\mathbf{x}_{\text{TV}})$ where $\mathbf{x}_{\text{TV}}$ is an approximated TV reconstruction generated by an iterative solver called the Lagged Diffusivity Fixed Point Method (Algorithm 2), as shown in [45]. The approximated solution $\mathbf{x}_{\text{TV}}$ provides an initial estimation of edge location, making it an appropriate fixed input for the $\mathbf{L}_{\text{TV}}$ matrix. From our experiments, we find that $\mathbf{x}_{\text{TV}}$ needs to be "reasonably good," in that it is not too under-regularized (too many edges) or over-regularized (expected edges are removed). This method of using $\mathbf{x}_{\text{TV}}$ to fix the $\mathbf{L}_{\text{TV}}$ matrix is also used in [20, 21].

Since $\mathbf{L}_{\text{TV}}^T\mathbf{L}_{\text{TV}}\mathbf{x} \approx \frac{d}{d\mathbf{x}}\left[\left|\left|\mathbf{D}\mathbf{x}\right|\right|_1\right]$ and $\mathbf{L}_{\text{TV}}^T\mathbf{L}_{\text{TV}}\mathbf{x} = \frac{d}{d\mathbf{x}}\left[\frac{1}{2}\left|\left|\mathbf{L}_{\text{TV}}\mathbf{x}\right|\right|_2^2\right]$, by transitivity we can write the approximated conditional distribution as

$$p\left(\mathbf{x}|\mathbf{b},\lambda,\delta\right) \propto \exp\left(-\frac{\lambda}{2}\left|\left|\mathbf{A}\mathbf{x}-\mathbf{b}\right|\right|_2^2 - \frac{\delta}{2}\left|\left|\mathbf{L}_{\text{TV}}\mathbf{x}\right|\right|_2^2\right), \tag{2.4.17}$$

Algorithm 2: Lagged Diffusivity Fixed Point Method

```
input: Initial approximation $\mathbf{x}^{(0)}$; regularization parameter $\alpha$; maximum number of
    iterations $K$; tolerance parameter $c > 0$

for $k = 1, \dots, K$ do
    $\mathbf{L}^{(k)} = \left(\mathbf{L}_{\text{TV}}\left(\mathbf{x}^{(k-1)}\right)\right)^T \mathbf{L}_{\text{TV}}\left(\mathbf{x}^{(k-1)}\right)$
    $\mathbf{g}^{(k)} = \mathbf{A}^T\left(\mathbf{A}\mathbf{x}^{(k-1)} - \mathbf{b}\right) + \alpha\mathbf{L}^{(k)}\mathbf{x}^{(k-1)}$
    $\mathbf{H}^{(k)} = \mathbf{A}^T\mathbf{A} + \alpha\mathbf{L}^{(k)}$
    $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \left(\mathbf{H}^{(k)}\right)^{-1}\mathbf{g}^{(k)}$
    if $\left|\left|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right|\right|_2 < c$ then
        break
    end
end
```

which follows the normal distribution

$$p\left(\mathbf{x}|\mathbf{b}, \lambda, \delta\right) \sim \mathcal{N}\left((\lambda\mathbf{A}^T\mathbf{A} + \delta\mathbf{L}_{\text{TV}}^T\mathbf{L}_{\text{TV}})^{-1}\mathbf{A}^T\mathbf{b}, (\lambda\mathbf{A}^T\mathbf{A} + \delta\mathbf{L}_{\text{TV}}^T\mathbf{L}_{\text{TV}})^{-1}\right). \quad (2.4.18)$$

Hence, the prior for the Bayesian 1D deconvolution problem for both Tikhonov and TV is a Gaussian of the form

$$p(\mathbf{x}|\delta) \propto \exp\left(-\frac{\delta}{2}\left|\left|\mathbf{R}\mathbf{x}\right|\right|_2^2\right). \quad (2.4.19)$$

For 2D deconvolution, we choose priors of the form

$$p(\mathbf{x}|\delta) \propto \exp\left(-\frac{\delta}{2}\left|\left|\left[\mathbf{R}_s\ \mathbf{R}_t\right]^T\mathbf{x}\right|\right|_2^2\right), \quad (2.4.20)$$

$$= \exp\left(-\frac{\delta}{2}\Big[\left|\left|\mathbf{R}_s\mathbf{x}\right|\right|_2^2 + \left|\left|\mathbf{R}_t\mathbf{x}\right|\right|_2^2\Big]\right), \quad (2.4.21)$$

where $\mathbf{x}$ is an $n^2 \times 1$ vectorization of the $n \times n$ image, and both $\mathbf{R}_s$ and $\mathbf{R}_t$ are $n^2 \times n^2$ matrices that enforce regularization in the s and t direction. For Tikhonov we let $\mathbf{R}_s = \mathbf{L}_{\text{TIK}}^s := \mathbf{I} \otimes \mathbf{L}_{\text{TIK}}$ and $\mathbf{R}_t = \mathbf{L}_{\text{TIK}}^t := \mathbf{L}_{\text{TIK}} \otimes \mathbf{I}$; and for TV we let $\mathbf{R}_s = \mathbf{L}_{\text{TV}}^s := \Psi_{\text{2D}}\mathbf{D}_s$ and $\mathbf{R}_t = \mathbf{L}_{\text{TV}}^t := \Psi_{\text{2D}}\mathbf{D}_t$, where the matrix Ψ_{2D} is an $n^2 \times n^2$ diagonal matrix with entries $\left((\mathbf{D}_s\mathbf{x})_i^2 + (\mathbf{D}_t\mathbf{x})_i^2 + \eta\right)^{-1/4}$ for $i = 1, \dots, n^2$. As in the 1D case, the matrix Ψ_{2D} depends on an approximated $\mathbf{x}$. In application, we use

Algorithm 3: Hierarchical Gibbs Sampler

```
input: Initial approximation $\mathbf{x}^{(0)}$; maximum number of iterations $K$; and
       conditional distributions $p(\mathbf{x}|\mathbf{b},\lambda,\delta), p(\lambda|\mathbf{b},\mathbf{x})$, and $p(\delta|\mathbf{x})$

for $k = 1, \ldots, K$ do
    $\lambda^{(k)} \sim p(\lambda|\mathbf{b},\mathbf{x}^{(k-1)})$
    $\delta^{(k)} \sim p(\delta|\mathbf{x}^{(k-1)})$
    $\mathbf{x}^{(k)} \sim p(\mathbf{x}|\mathbf{b},\lambda^{(k)},\delta^{(k)})$
end
```

Algorithm 2 with $\mathbf{L}_{\text{TV}}^T\mathbf{L}_{\text{TV}} := (\mathbf{L}_{\text{TV}}^s)^T\mathbf{L}_{\text{TV}}^s + (\mathbf{L}_{\text{TV}}^t)^T\mathbf{L}_{\text{TV}}^t$ to generate an initial TV reconstruction $\mathbf{x}_{\text{TV}}^{\text{2D}}$. We then fix $\Psi_{\text{2D}} = \Psi_{\text{2D}}\left(\mathbf{x}_{\text{TV}}^{\text{2D}}\right)$. This method of regularization where rows and columns are treated separately is discussed in [19], and incorporating this technique into the prior will be especially useful when implementing methods where the parameter δ varies across the image.

Whether the deconvolution problem is 1D or 2D, uses Tikhonov or TV regularization, the way we have constructed the posterior distribution allows us to easily write the conditional distributions and sample using a Gibbs sampler. Algorithm 3 shows the hierarchical Gibbs sampler used to generate samples from the posterior distribution.

2.5 Deconvolution Examples in 1D

We present two examples of deblurring signals of length $n = 256$ using the hierarchical Gibbs sampler. The left images in Figures 2.5.1 and 2.5.2 show blurred, noisy signals (solid green line) along with their respective true underlying signals (dotted orange line). In both cases, the blurred signals are generated by convolving the true signals with the kernel

$$a(s) = \frac{1}{\gamma\sqrt{2\pi}}\exp\left(-\frac{s^2}{2\gamma^2}\right), \tag{2.5.1}$$

with $\gamma = 0.02$. Assuming zero boundary conditions, we then create a 256×256 Toeplitz matrix by discretizing $a(s)$ as in equation (2.1.4). The blurring matrix $\mathbf{A}$ results by multiplying the Toeplitz matrix by the step size $h = 1/256$. The blurred, noisy signals in Figures 2.5.1 and 2.5.2 are given by

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\varepsilon}, \tag{2.5.2}$$

where $\mathbf{x}$ is the true signal, and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{256})$ is a 256×1 random vector with variance σ^2 chosen so that the signal-to-noise ratio (SNR), defined in [3] as

$$||\mathbf{A}\mathbf{x}||_2 / \sqrt{n\sigma^2}, \tag{2.5.3}$$

is 25. In other words, the standard deviation is 4% of the 2-norm of $\mathbf{A}\mathbf{x}$. We run the Gibbs sampler in Algorithm 3, initializing $\mathbf{x}^{(0)} = \mathbf{b}$ and setting the maximum number of iterations to be $K = 10{,}000$.

Figure 2.5.1 shows a blurred, noisy signal in the top left image (solid green line) where the appropriate regularization technique is Tikhonov. In this example, we let

$$\mathbf{L}_{\text{TIK}} = \mathbf{D}_2 := \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}_{n \times n} . \tag{2.5.4}$$

Once again, the Tikhonov-regularized reconstruction is the solution of the optimization problem

$$\mathbf{x}_{\lambda,\delta} = \arg\min_{\mathbf{x}} \left(\frac{\lambda}{2} ||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 + \frac{\delta}{2} ||\mathbf{L}_{\text{TIK}}\mathbf{x}||_2^2 \right), \tag{2.5.5}$$

where λ and δ are nonnegative parameters that control the balance of regularization strength versus fidelity to the data. Using the hierarchical Gibbs sampler, we obtain the reconstruction shown in the top right subfigure in Figure 2.5.1 without having to

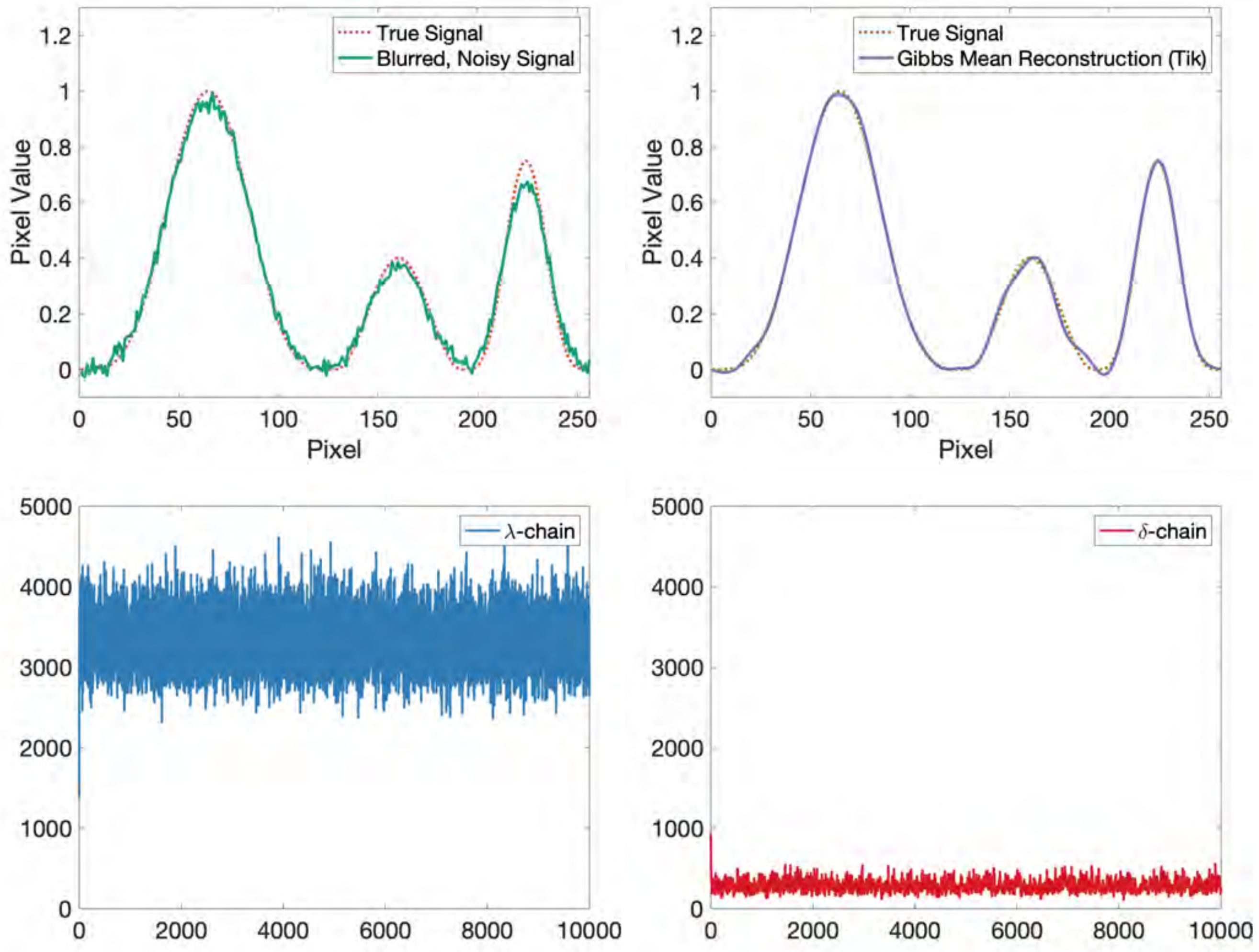


Figure 2.5.1: The top left figure shows a smooth underlying signal (dotted orange line) along with a blurred, noisy version of the signal (solid green line). The top right figure shows a reconstruction (solid purple line) of the blurred signal using the Gibbs sampler with Tikhonov regularization. The bottom row shows the chains of λ and δ samples.

choose values for λ and δ. The sampled λ- and δ-chains are shown in the bottom row of Figure 2.5.1.

Notice with both chains, it does not take too many samples for the chain to reach equilibrium. However, out of an abundance of caution, we choose a burn-in of 1,000 samples. After burn-in, the mean values of the regularization parameters are $\overline{\lambda} \approx$ 3,362 and $\overline{\delta} \approx$ 18,881. We compute the integrated autocorrelation time (IACT) of the λ- and δ-chains to assess the efficiency of the Gibbs sampler. The IACT's for the

λ-chain and δ-chain after burn-in are $\tau_\lambda \approx 1.24$ and $\tau_\delta \approx 17.62$. Referring to (2.4.4), the effective number of samples is the total number of samples (which after burn-in is 9,000) divided by the computed IACT. Hence, there are approximately 7,258 effective samples for λ and 510 effective samples for δ.

The mean reconstruction from the Gibbs sampler with Tikhonov regularization in Figure 2.5.1 seems to match the true underlying signal pretty well. Of course, we cannot rely on visual inspection alone in order to determine the quality of a reconstruction. To measure the differences between two signals or column-stacked images $\mathbf{u} = (u_1, \ldots, u_N)$ and $\mathbf{v} = (v_1, \ldots, v_N)$, we use root mean square error (RMSE) defined by

$$\text{RMSE}(\mathbf{u}, \mathbf{v}) = \sqrt{\frac{\sum_{i=1}^{N} (u_i - v_i)^2}{N}}. \tag{2.5.6}$$

From the example in Figure 2.5.1, let $\mathbf{x}^{(1)}$ be the true signal, $\mathbf{b}^{(1)}$ be the blurred and noisy signal, and $\overline{\mathbf{x}}^{(1)}$ be the mean Gibbs reconstruction. We have that $\text{RMSE}(\mathbf{x}^{(1)}, \mathbf{b}^{(1)}) = 0.0306$ and $\text{RMSE}(\mathbf{x}^{(1)}, \overline{\mathbf{x}}^{(1)}) = 0.017$, an improvement of about 44%.

We now turn to the example in Figure 2.5.2, where we use the Gibbs sampler with TV regularization. The top left subfigure shows the true signal (dashed orange line) along with the blurred, noisy signal (solid green line), and the bottom right figure shows the mean reconstruction resulting from running the hierarchical Gibbs sampler for 10,000 samples with a burn-in of 1,000. Again, we choose a burn-in of 1,000 samples after plotting the λ- and δ-chains (bottom row in Figure 2.5.2) and determining the length of time the chains take to reach equilibrium. A burn-in of 1,000 samples is perhaps too large, but we simply want to be especially sure we are sampling from the equilibrium distribution. The mean λ and δ values after burn-in are $\overline{\lambda} = 4{,}207$ and $\overline{\delta} = 6.76$. The small sampled values of δ's are expected, because the matrix norm of the operator $\mathbf{L}_{\text{TV}}$ can be very large. This is due to the definition of the diagonal matrix Ψ, where the denominators of the entries can be close to zero. The IACT's for the λ-chain and δ-chain are $\tau_\lambda \approx 1.16$ and $\tau_\delta \approx 25.68$, which means

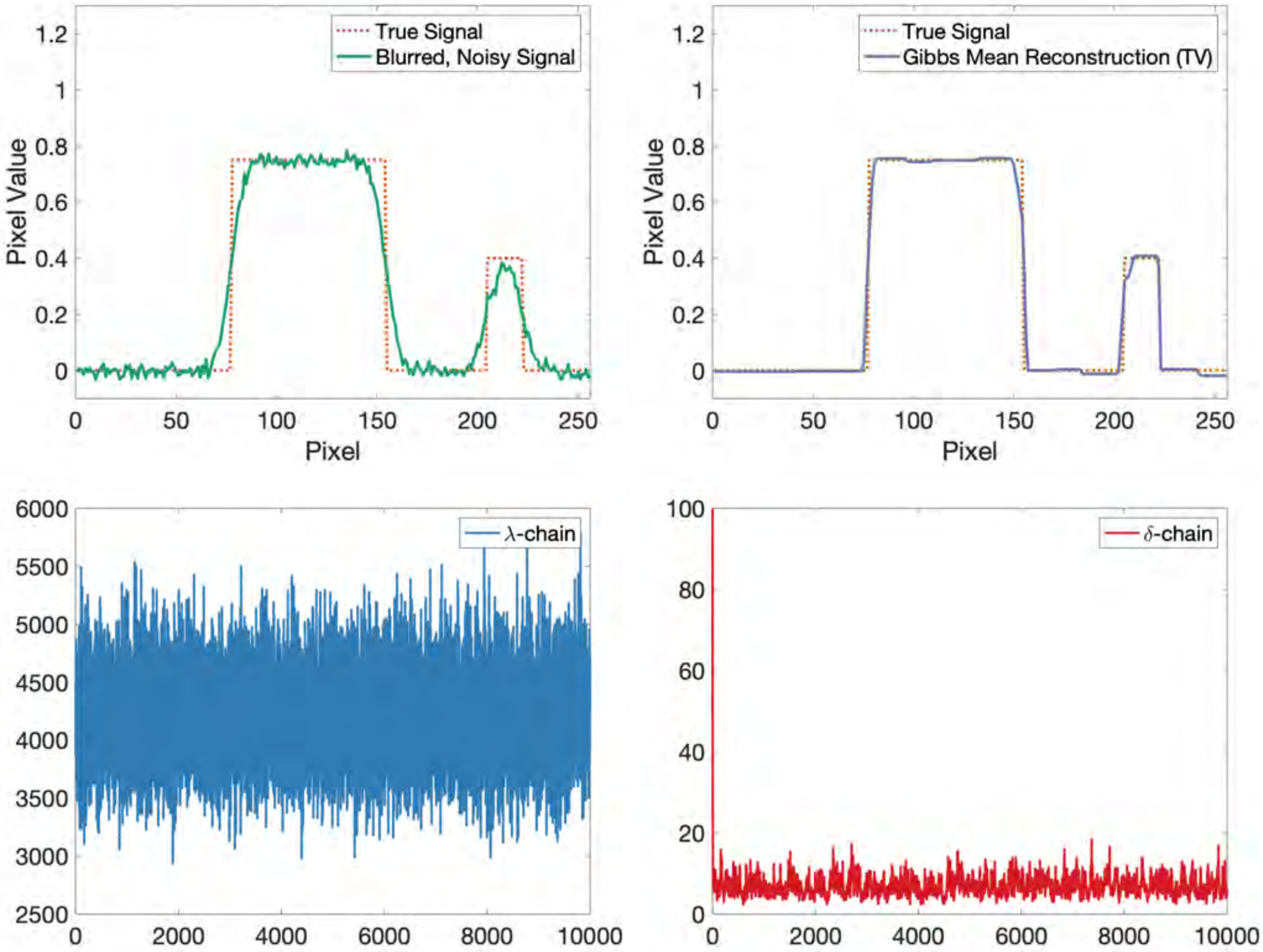


Figure 2.5.2: The top left figure shows a piecewise constant underlying signal (dotted orange line) along with a blurred, noisy version of the signal (solid green line). The top right figure shows a reconstruction (solid purple line) of the blurred signal using the Gibbs sampler with TV regularization. The bottom row shows the chains of λ and δ samples.

there were approximately 7,758 effective samples for λ and 350 effective samples for δ.

The mean Gibbs reconstruction seems to match the underlying signal well; however, it does have "staircasing" artifacts in several areas. Staircasing artifacts are common in TV-regularized reconstructions and are defined as regions where the reconstruction's value changes sharply from one constant value to another. To compute the RMSE, let $\mathbf{x}^{(2)}$ be the true signal, $\mathbf{b}^{(2)}$ be the blurred and noisy signal, and $\overline{\mathbf{x}}^{(2)}$

be the mean Gibbs reconstruction. We have that $\text{RMSE}(\mathbf{x}^{(2)}, \mathbf{b}^{(2)}) = 0.0851$ and $\text{RMSE}(\mathbf{x}^{(2)}, \overline{\mathbf{x}}^{(2)}) = 0.0456$. In this example, the mean Gibbs reconstruction cuts the error between the given corrupted signal and the true signal by almost half.

The hierarchical Bayesian approach to the deblurring problem seems to produce good reconstructions, both visually and quantitatively. However, we believe better reconstructions can be obtained when the hyperparameters λ and δ vary across the data. While the hierarchical Gibbs sampler yields an optimal ratio of δ to λ, it is reasonable to assume that the regularization strength in one part of the image may need to be greater than that of another part. The added noise $\boldsymbol{\varepsilon}$ affects the underlying image in a non-uniform way; therefore, we should let the regularization strength be non-uniform, as well. We investigate this in the next chapter by building a hierarchical model with spatially varying parameters, and writing an algorithm for a hierarchical Gibbs sampler that will effectively sample from the much more complex posterior distribution.

2.6 Summary

We have provided necessary background material to understand this work, such as discrete 1D and 2D convolution, regularization techniques, the Bayesian formulation of inverse problems, and Gibbs sampling. Deblurring is an ill-posed inverse problem, and requires regularization to obtain useful solutions. Traditional methods require finding minimizers of functions involving the least squares difference between the data and the model, with an additional weighted term called the regularization term. The weight on the regularization term is the regularization parameter. It is challenging to determine the value of the regularization parameter, as it represents a trade-off between fidelity to the data and enforcing regularization. By using a Bayesian framework and letting the parameters be random variables, we no longer have the challenge of hand-tuning the parameters. The resulting posterior distribution is difficult to sample; however, its conditional distributions are easy to sample. Therefore, a hierarchical Gibbs sampler is used to obtain image reconstructions. We concluded the chapter by deblurring 1D examples using the existing hierarchical Bayesian model from [3]. We measure the efficiency of the sampler by computing the integrated autocorrelation time.

Chapter 3

Hierarchical Gibbs Sampler with Spatially Varying Parameters

In this chapter, we present a hierarchical Gibbs sampler that allows the parameters λ and δ to vary across each pixel in the image. We call the sampler HGSV, for a (h)ierarchical (G)ibbs sampler with (s)patially (v)arying parameters. The motivation for the HGSV sampler is that the added noise affects the data unevenly; therefore, it is unreasonable to fix the strength of the regularization for the entire image. A certain strength may be appropriate for one section of the image, but not another. We adapt the existing hierarchical model to include hyperparameters that vary pixel-to-pixel, and present the HGSV sampler.

3.1 A Hierarchical Model with Spatially Varying Parameters

We investigate if by letting the regularization parameter values vary from pixel-to-pixel, better reconstructions can be obtained. Instead of having two hyperpriors, λ and δ, we now have $2N$ hyperpriors, where $N = n$ in 1D and $N = n^2$ in 2D.

3.1.1 1D Model

We adapt the hyperpriors in (2.3.8) and (2.3.9) to accommodate our spatially varying formulation. For $i = 1, \dots, N$, assume λ_i and δ_i have Gamma distributions:

$$p(\lambda_i) \sim \text{Gamma}(\alpha_\lambda, \beta_\lambda) \propto \lambda_i^{\alpha_\lambda - 1} \exp(-\beta_\lambda \lambda_i), \tag{3.1.1}$$

$$p(\delta_i) \sim \text{Gamma}(\alpha_\delta, \beta_\delta) \propto \delta_i^{\alpha_\delta - 1} \exp(-\beta_\delta \delta_i), \tag{3.1.2}$$

where $\alpha_\lambda, \alpha_\delta$ are Gamma shape parameters and $\beta_\lambda, \beta_\delta$ are Gamma rate parameters. Let $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_N)$ and $\boldsymbol{\delta} = (\delta_1, \dots, \delta_N)$ be N-dimensional vectors containing the parameter's value at each pixel. We assume $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ are independent, i.e.,

$$p(\boldsymbol{\lambda}) = \prod_{i=1}^{N} p(\lambda_i) \ , \qquad p(\boldsymbol{\delta}) = \prod_{i=1}^{N} p(\delta_i) \ , \qquad p(\boldsymbol{\lambda}, \boldsymbol{\delta}) = p(\boldsymbol{\lambda}) p(\boldsymbol{\delta}) \ . \tag{3.1.3}$$

The likelihood and prior densities are respectively defined as

$$p(\mathbf{b}|\mathbf{x}, \boldsymbol{\lambda}) \propto \left(\prod_{i=1}^{N} \lambda_i \right)^{1/2} \exp\left(-\frac{1}{2} \sum_{i=1}^{N} \lambda_i \big[(\mathbf{A}\mathbf{x})_i - b_i \big]^2 \right), \tag{3.1.4}$$

$$p(\mathbf{x}|\boldsymbol{\delta}) \propto \left(\prod_{i=1}^{N} \delta_i \right)^{1/2} \exp\left(-\frac{1}{2} \sum_{i=1}^{N} \delta_i \left(\mathbf{L}\mathbf{x} \right)_i^2 \right), \tag{3.1.5}$$

where $\mathbf{L}$ is either $\mathbf{L}_{\text{TIK}}$ or $\mathbf{L}_{\text{TV}}$, depending on which type of regularization is required.

Combining the likelihood (3.1.4), the prior (3.1.5), and the hyperpriors $p(\lambda_i), p(\delta_i)$, we can write the posterior distribution

$$\begin{aligned} p(\mathbf{x}, &\boldsymbol{\lambda}, \boldsymbol{\delta}|\mathbf{b}) \propto p(\mathbf{b}|\mathbf{x}, \boldsymbol{\lambda}) p(\mathbf{x}|\boldsymbol{\delta}) p(\boldsymbol{\lambda}) p(\boldsymbol{\delta}), \\ &\propto \prod_{i=1}^{N} \lambda_i^{\alpha_\lambda - 1/2} \delta_i^{\alpha_\delta - 1/2} \\ &\quad \times \exp\left(-\sum_{i=1}^{N} \lambda_i \left(\frac{1}{2} \big[(\mathbf{A}\mathbf{x})_i - b_i \big]^2 + \beta_\lambda \right) - \sum_{i=1}^{N} \delta_i \left(\frac{1}{2} \left(\mathbf{L}\mathbf{x} \right)_i^2 + \beta_\delta \right) \right). \end{aligned} \tag{3.1.6}$$

The conditional posterior distributions for the $2N$ hyperpriors are Gamma distributions:

$$\lambda_i|\mathbf{b},\mathbf{x} \sim \text{Gamma}\left(\frac{1}{2}+\alpha_\lambda, \frac{1}{2}\big[(\mathbf{A}\mathbf{x})_i - b_i\big]^2 + \beta_\lambda\right), \tag{3.1.7}$$

$$\delta_i|\mathbf{x} \sim \text{Gamma}\left(\frac{1}{2}+\alpha_\delta, \frac{1}{2}\left(\mathbf{L}\mathbf{x}\right)_i^2 + \beta_\delta\right). \tag{3.1.8}$$

The conditional distribution for $\mathbf{x}$ is given by

$$p(\mathbf{x}|\mathbf{b},\boldsymbol{\lambda},\boldsymbol{\delta}) \propto \exp\left(-\sum_{i=1}^{N}\frac{\lambda_i}{2}\big[(\mathbf{A}\mathbf{x})_i - b_i\big]^2 - \sum_{i=1}^{N}\frac{\delta_i}{2}\left(\mathbf{L}\mathbf{x}\right)_i^2\right), \tag{3.1.9}$$

$$= \exp\left(-\frac{1}{2}\left|\left|\boldsymbol{\Lambda}^{1/2}\left(\mathbf{A}\mathbf{x}-\mathbf{b}\right)\right|\right|_2^2 - \frac{1}{2}\left|\left|\boldsymbol{\Delta}^{1/2}\mathbf{L}\mathbf{x}\right|\right|_2^2\right), \tag{3.1.10}$$

where $\boldsymbol{\Lambda}^{1/2} = \text{diag}\left(\boldsymbol{\lambda}^{1/2}\right)$ and $\boldsymbol{\Delta}^{1/2} = \text{diag}\left(\boldsymbol{\delta}^{1/2}\right)$ with $\boldsymbol{\lambda}^{1/2} = \left(\lambda_1^{1/2}, \lambda_2^{1/2}, \ldots, \lambda_N^{1/2}\right)^T$ and $\boldsymbol{\delta}^{1/2} = \left(\delta_1^{1/2}, \delta_2^{1/2}, \ldots, \delta_N^{1/2}\right)^T$. Therefore, the conditional distribution follows the normal distribution

$$p(\mathbf{x}|\mathbf{b},\boldsymbol{\lambda},\boldsymbol{\delta}) \sim \mathcal{N}\left(\mathbf{H}^{-1}\mathbf{A}^T\boldsymbol{\Lambda}\mathbf{b}, \mathbf{H}^{-1}\right), \tag{3.1.11}$$

where $\mathbf{H} = \mathbf{A}^T\boldsymbol{\Lambda}\mathbf{A} + \mathbf{L}^T\boldsymbol{\Delta}\mathbf{L}$ with $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})$ and $\boldsymbol{\Delta} = \text{diag}(\boldsymbol{\delta})$.

3.1.2 2D Model

Assuming $\boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}^{-1}\right)$ where $\boldsymbol{\Lambda} = \text{diag}\left(\left(\lambda_1, \ldots, \lambda_N\right)^T\right)$ with unknown parameter values $\lambda_1, \ldots, \lambda_N$, the likelihood function is

$$p_{\text{LH}}\left(\mathbf{b}|\mathbf{x},\boldsymbol{\Lambda}\right) \propto \left|\boldsymbol{\Lambda}\right|^{1/2}\exp\left(-\frac{1}{2}\left|\left|\boldsymbol{\Lambda}^{1/2}\left(\mathbf{A}\mathbf{x}-\mathbf{b}\right)\right|\right|_2^2\right), \tag{3.1.12}$$

where $|\cdot|$ is the determinant operation. In the case of Tikhonov regularization with spatially varying parameters represented by $\boldsymbol{\Delta} = \text{diag}\left(\left(\delta_1, \ldots, \delta_N\right)^T\right)$, we use the prior

$$p_{\text{TIK}}\left(\mathbf{x}|\boldsymbol{\Delta}\right) \propto \left|\boldsymbol{\Delta}\right|^{1/2} \exp\left(-\frac{1}{2}\left|\left|\boldsymbol{\Delta}^{1/2}\left[\mathbf{L}^{s}_{\text{TIK}}\ \mathbf{L}^{t}_{\text{TIK}}\right]^{T}\mathbf{x}\right|\right|_{2}^{2}\right), \tag{3.1.13}$$

$$= \left|\boldsymbol{\Delta}\right|^{1/2} \exp\left(-\frac{1}{2}\left|\left|\boldsymbol{\Delta}^{1/2}\mathbf{L}^{s}_{\text{TIK}}\mathbf{x}\right|\right|_{2}^{2} - \frac{1}{2}\left|\left|\boldsymbol{\Delta}^{1/2}\mathbf{L}^{t}_{\text{TIK}}\mathbf{x}\right|\right|_{2}^{2}\right), \tag{3.1.14}$$

where $\mathbf{L}^{s}_{\text{TIK}} = \mathbf{I}_N \otimes \mathbf{D}_2$ and $\mathbf{L}^{t}_{\text{TIK}} = \mathbf{D}_2 \otimes \mathbf{I}_N$, with $\mathbf{I}_N$ as the $N \times N$ identity matrix and

$$\mathbf{D}_2 := \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}_{N\times N}. \tag{3.1.15}$$

We now compute HGSV reconstruction with a TV prior. We define the prior as

$$p_{\text{TV}}\left(\mathbf{x}|\boldsymbol{\Delta}\right) \propto \left|\boldsymbol{\Delta}\right|^{1/2} \exp\left(-\frac{1}{2}\left|\left|\boldsymbol{\Delta}^{1/2}\left[\mathbf{L}^{s+}_{\text{TV}}\ \mathbf{L}^{t+}_{\text{TV}}\ \mathbf{L}^{s-}_{\text{TV}}\ \mathbf{L}^{t-}_{\text{TV}}\right]^{T}\mathbf{x}\right|\right|_{2}^{2}\right), \tag{3.1.16}$$

$$\begin{aligned} &= \left|\boldsymbol{\Delta}\right|^{1/2} \exp\left(-\frac{1}{2}\left|\left|\boldsymbol{\Delta}^{1/2}\mathbf{L}^{s+}_{\text{TV}}\mathbf{x}\right|\right|_{2}^{2} - \frac{1}{2}\left|\left|\boldsymbol{\Delta}^{1/2}\mathbf{L}^{t+}_{\text{TV}}\mathbf{x}\right|\right|_{2}^{2}\right. \\ &\qquad \left. -\frac{1}{2}\left|\left|\boldsymbol{\Delta}^{1/2}\mathbf{L}^{s-}_{\text{TV}}\mathbf{x}\right|\right|_{2}^{2} - \frac{1}{2}\left|\left|\boldsymbol{\Delta}^{1/2}\mathbf{L}^{t-}_{\text{TV}}\mathbf{x}\right|\right|_{2}^{2}\right), \end{aligned} \tag{3.1.17}$$

where $\mathbf{L}^{s+}_{\text{TV}} = \Psi^{s+}_{\text{2D}}\mathbf{D}_s$; $\mathbf{L}^{t+}_{\text{TV}} = \Psi^{t+}_{\text{2D}}\mathbf{D}_t$; $\mathbf{L}^{s-}_{\text{TV}} = \Psi^{s-}_{\text{2D}}\mathbf{D}_s^T$; and $\mathbf{L}^{t-}_{\text{TV}} = \Psi^{t-}_{\text{2D}}\mathbf{D}_t^T$. We define

$$\Psi^{s+}_{\text{2D}}(\mathbf{x}) = \text{diag}\left(\left[(\mathbf{D}_s\mathbf{x}) \odot (\mathbf{D}_s\mathbf{x}) + \eta\mathbb{1}\right]^{-1/4}\right), \tag{3.1.18}$$

$$\Psi^{t+}_{\text{2D}}(\mathbf{x}) = \text{diag}\left(\left[(\mathbf{D}_t\mathbf{x}) \odot (\mathbf{D}_t\mathbf{x}) + \eta\mathbb{1}\right]^{-1/4}\right), \tag{3.1.19}$$

$$\Psi^{s-}_{\text{2D}}(\mathbf{x}) = \text{diag}\left(\left[(\mathbf{D}_s^T\mathbf{x}) \odot (\mathbf{D}_s^T\mathbf{x}) + \eta\mathbb{1}\right]^{-1/4}\right), \tag{3.1.20}$$

$$\Psi^{t-}_{\text{2D}}(\mathbf{x}) = \text{diag}\left(\left[(\mathbf{D}_t^T\mathbf{x}) \odot (\mathbf{D}_t^T\mathbf{x}) + \eta\mathbb{1}\right]^{-1/4}\right), \tag{3.1.21}$$

where $\odot$ denotes pointwise multiplication. We also assume the power $-1/4$ is applied pointwise.

Recall that $\mathbf{D}_s = \mathbf{I}_N \otimes \mathbf{D}_1$ and $\mathbf{D}_t = \mathbf{D}_1 \otimes \mathbf{I}_N$, where

$$\mathbf{D}_1 := \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & 1 & -1 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}_{N \times N} . \tag{3.1.22}$$

Therefore, by the properties of Kronecker products, $\mathbf{D}_s^T = \mathbf{I}_N \otimes \mathbf{D}_1^T$ and $\mathbf{D}_t^T = \mathbf{D}_1^T \otimes \mathbf{I}_N$. The purpose of the four operators in the TV prior is so that each pixel $\mathbf{x}_{i,j}$ is regularized with respect to the pixels to the left, right, above, and below, i.e., $\mathbf{x}_{i-1,j}, \mathbf{x}_{i+1,j}, \mathbf{x}_{i,j+1}$, and $\mathbf{x}_{i,j-1}$.

3.2 The HGSV Sampler

Since the conditional distributions of the posterior are easy to sample, we use a hierarchical Gibbs sampler to sample the posterior with spatially varying hyperparameters. We call this sampler the HGSV sampler, for a (h)ierarchical (G)ibbs sampler for (s)patially (v)arying parameters. Algorithm 4 shows the sampler, where the spatially varying parameters are sampled from their conditional distributions, conditioned on the previous sample $\mathbf{x}^{(k-1)}$. After all $2N$ samples have been collected, the new reconstruction $\mathbf{x}^{(k)}$ is drawn from the conditional distribution (4.1.10). An easy way to sample from this distribution is to sample $\boldsymbol{\varepsilon}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_N)$, then compute

$$\mathbf{H}^{-1/2}\left(\mathbf{H}^{-T/2}\mathbf{A}^T\boldsymbol{\Lambda}\mathbf{b} + \boldsymbol{\varepsilon}_1\right). \tag{3.2.1}$$

Notice the precision matrix $\mathbf{H}$ must be raised to the 1/2 power. Computing the "square root" of $\mathbf{H}$ can be performed using Cholesky factorization. We continue sampling until the desired number of samples K has been drawn. Since Gibbs samplers take time to converge to the posterior, one should discard an initial number of samples

Algorithm 4: HGSV Sampler

```
input: Initial approximation $\mathbf{x}^{(0)}$; maximum number of iterations $K$; and
       conditional distributions $p(\mathbf{x}|\mathbf{b}, \boldsymbol{\lambda}, \boldsymbol{\delta}), p(\lambda_i|\mathbf{b}, \mathbf{x})$, and $p(\delta_i|\mathbf{x})$

for $k = 1, \dots, K$ do
    for $i = 1, \dots, N$ do
        $\lambda_i^{(k)} \sim p(\lambda_i|\mathbf{b}, \mathbf{x}^{(k-1)})$
        $\delta_i^{(k)} \sim p(\delta_i|\mathbf{x}^{(k-1)})$
    end
    $\mathbf{x}^{(k)} \sim p(\mathbf{x}|\mathbf{b}, \boldsymbol{\lambda}^{(k)}, \boldsymbol{\delta}^{(k)})$
end
```

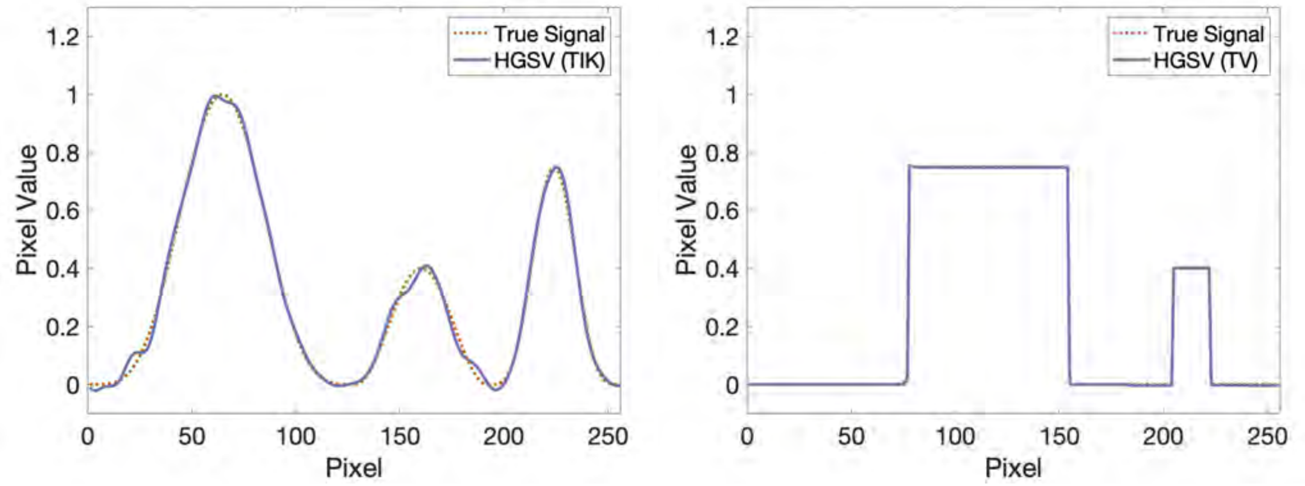


Figure 3.3.1: Two reconstructions (solid purple line) of the previous corrupted signals using the HGSV sampler. The reconstruction in the left figure uses Tikhonov regularization, and the reconstruction in the right figure uses TV regularization.

c. The optimal reconstruction is given by the mean of the $K - c$ samples.

3.3 Deconvolution Examples in 1D

To illustrate the effectiveness of the new hierarchical model, we continue with the same examples presented in Section 2.5. The blurring matrix $\mathbf{A}$ and the regularization matrices $\mathbf{L}_{\text{TIK}}$ and $\mathbf{L}_{\text{TV}}$ are the same as in the previous example. The only change is that the hyperparameters λ and δ are varying pixel-to-pixel. Figure 3.3.1 shows the two reconstructions using the HGSV sampler. In each case, 10,000 samples were drawn. The first 1,000 samples were discarded to account for burn-in. The Tikhonov-based reconstruction on the left slightly deviates from the true signal in several places, but overall fits the data's trend well. The average IACT for $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^T$ is $\overline{\tau_{\boldsymbol{\lambda}}} \approx 1.3224$, and the average IACT for $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)^T$ is $\overline{\tau_{\boldsymbol{\delta}}} \approx 1.4639$. The TV-based reconstruction on the right is almost indistinguishable from the true signal. Here, the average IACT's for $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ are $\overline{\tau_{\boldsymbol{\lambda}}} \approx 1.0802$ and $\overline{\tau_{\boldsymbol{\delta}}} \approx 1.0092$. Past burn-in, this means that almost every sample of the hyperparameters for the TV-based reconstruction is independent.

Table 3.3.1 compares the RMSE values from the mean reconstructions in the previous chapter and those obtained here using the HGSV sampler. (TIK/TV) refers to the Tikhonov or TV reconstructions from the standard hierarchical model, and HGSV (TIK/TV) refers to those from the hierarchical model with spatially varying hyperparameters. There is not much difference in the RMSE's of the Tikhonov-based reconstruction from the standard model versus the spatially varying model. However, the TV-based reconstruction under the spatially varying model reduces the initial error due to blur and added noise by an astounding 97.3%.

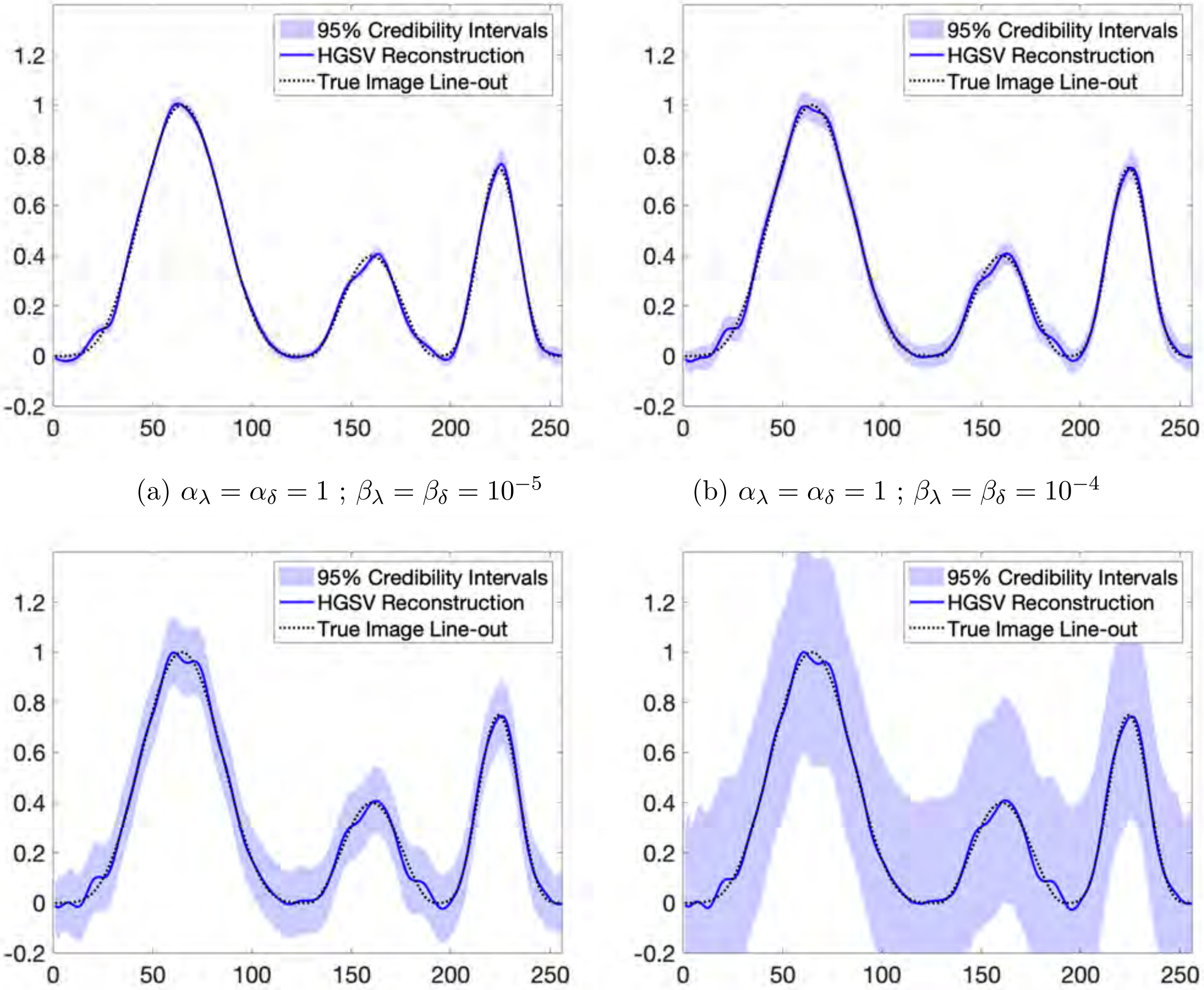
Table 3.3.1: A comparison of errors between reconstructions from the two hierarchical models. The reconstructions HG (TIK/TV) refers to those obtained from the standard hierarchical Gibbs sampler, and HGSV (TIK/TV) refers to the reconstructions obtained from the HGSV sampler.

	RMSE $(\mathbf{x}, \mathbf{b})$	RMSE $(\mathbf{x}, \overline{\mathbf{x}})$	% Decrease in RMSE
HG (TIK)	0.0306	0.0119	61.1%
HGSV (TIK)	0.0306	0.0170	44.4%
HG (TV)	0.0851	0.0456	46.4%
HGSV (TV)	0.0851	0.0023	97.3%

3.4 A Discussion on Uncertainty Quantification

A benefit of using a Bayesian model for deblurring is that one can compute credibility intervals to quantify uncertainty in the reconstruction. Another way to compare the HGSV method to the standard non-spatially-varying method would be to analyze the widths of credibility intervals for the different reconstructions. However, we notice the computed credibility intervals are sensitive to the choice of gamma parameters $\alpha_\lambda, \alpha_\delta, \beta_\lambda, \beta_\delta$ in (3.1.1) and (3.1.2). Recall that we choose the values of the parameters in such a way that the gamma PDF behaves like that of a uniform distribution. We do not wish to change $\alpha_\lambda = \alpha_\delta = 1$, as this choice eliminates contribution from the factor of λ_i in (3.1.1) and δ_i in (3.1.2). We do, however, have more freedom in choosing the β_λ and β_δ parameters.

Figure 3.4.1 shows four different HGSV reconstructions (solid purple lines) of the previous Tikhonov-based example along with their computed 95% credibility intervals (shaded purple regions) with fixed $\alpha_\lambda = \alpha_\delta = 1$ and varying β_λ and β_δ. We see that the smaller the choice of β_λ and β_δ, the narrower the widths of the credibility intervals. At the same time, we see the mean reconstructions are not significantly affected by the choice of gamma parameters. This gives us confidence in our hierarchical model; however, we cannot trust any computed credibility intervals until this sensitivity issue is addressed. For this work, we will fix $\beta_\lambda = \beta_\delta = 10^{-4}$.



(a) $\alpha_\lambda = \alpha_\delta = 1$; $\beta_\lambda = \beta_\delta = 10^{-5}$

(b) $\alpha_\lambda = \alpha_\delta = 1$; $\beta_\lambda = \beta_\delta = 10^{-4}$

(c) $\alpha_\lambda = \alpha_\delta = 1$; $\beta_\lambda = \beta_\delta = 10^{-3}$

(d) $\alpha_\lambda = \alpha_\delta = 1$; $\beta_\lambda = \beta_\delta = 10^{-2}$

Figure 3.4.1: Four HGSV reconstructions (solid purple lines) with their respective 95% credibility intervals (shaded purple regions) with gamma shape parameters $\alpha_\lambda = \alpha_\delta = 1$ and varying rate parameters β_λ and β_δ.

3.5 Summary

In this chapter, we introduced the idea of letting the regularization parameters λ and δ vary pixel-to-pixel. We adapted the existing hierarchical Bayesian model and Gibbs sampler to include the spatially varying parameters. We named the new model HGSV. We applied the HGSV model to two 1D deconvolution problems. One problem required Tikhonov regularization, and the other problem required TV regularization.

There is not much difference in RMSE's between the standard model and HGSV model with Tikhonov regularization. However, the HGSV model with TV regularization does much better than the standard model both qualitatively and quantitatively. We also discussed why we are unable to trust credibility intervals from reconstructions due to sensitivity to the gamma rate parameters β_λ and β_δ.

Chapter 4

Hierarchical Gibbs Sampler with Spatially Varying Parameters and Mixed Regularization

In this chapter, we extend the HGSV algorithm to incorporate a spatial mixing of Tikhonov and TV regularization. We call this extension the (h)ierarchical (G)ibbs sampler with (s)patially (v)arying parameters and (m)ixed regularization, or the HGSVM algorithm. We develop this algorithm because most images (especially Cygnus images) include a combination of smooth and sharp features. Therefore, effectively combining the two regularization techniques could result in better reconstructions. For now, we assume there exists a partitioning based on needed regularization. Under this assumption we incorporate spatially varying parameters, and build the hierarchical model. We show an example of deblurring a signal using the HGSVM sampler. In the last section, we include a discussion on possible ways to partition a blurred, noisy signal or image into regions where Tikhonov or TV regularization is most appropriate.

4.1 A Hierarchical Model for Deconvolution with Mixed Regularization

Assume an observation $\mathbf{b}$ is an $N \times 1$ blurred, noisy signal or vectorized image. Assume it has been determined that certain regions should use Tikhonov regularization, while the remaining regions should use TV. Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)^T$ be an $N \times 1$ logical vector that determines such a partitioning where for each $i = 1, \ldots, N$, $\pi_i \in \{0, 1\}$. Let the Tikhonov regions correspond to $\pi_i = 0$ and the TV regions correspond to $\pi_i = 1$. We assume $\boldsymbol{\pi}$ is known, and incorporate it into the hierarchical model with spatially varying parameters. We discuss how one might construct $\boldsymbol{\pi}$ later in Section 4.4.

For $i = 1, \ldots, N$, let λ_i be the inverse variance of the added Gaussian noise, and δ_i be the strength of the imposed regularization. These parameters are unknown and therefore, in the Bayesian paradigm, are treated as random variables. We assume λ_i and δ_i, are independent for all $i = 1, \ldots, N$; and we assign Gamma distributions to the two sets of parameters:

$$p(\lambda_i) \sim \text{Gamma}(\alpha_\lambda, \beta_\lambda) \propto \lambda_i^{\alpha_\lambda - 1} \exp(-\beta_\lambda \lambda_i), \tag{4.1.1}$$

$$p(\delta_i) \sim \text{Gamma}(\alpha_\delta, \beta_\delta) \propto \delta_i^{\alpha_\delta - 1} \exp(-\beta_\delta \delta_i), \tag{4.1.2}$$

where $\alpha_\lambda, \alpha_\delta$ are Gamma shape parameters and $\beta_\lambda, \beta_\delta$ are Gamma rate parameters. Let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^T$ and $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)^T$. The prior density now depends on the partitioning vector $\boldsymbol{\pi}$. In 1D, the prior is

$$p_{1\text{D}}(\mathbf{x}|\boldsymbol{\delta}, \boldsymbol{\pi}) \propto \left(\prod_{i=1}^N \delta_i \right)^{1/2} \exp\left(-\frac{\delta_i}{2} \sum_{i=1}^N \left[(1 - \pi_i) \left(\mathbf{L}_{\text{TIK}}\mathbf{x}\right)_i^2 + \pi_i \left(\mathbf{L}_{\text{TV}}\mathbf{x}\right)_i^2 \right] \right). \tag{4.1.3}$$

The likelihood density does not depend on regularization terms and, therefore, does not depend on $\boldsymbol{\pi}$. Hence, the likelihood remains unchanged from the spatially varying parameters model in the previous chapter as

$$p(\mathbf{b}|\mathbf{x},\boldsymbol{\lambda}) \propto \left(\prod_{i=1}^{N} \lambda_i\right)^{1/2} \exp\left(-\frac{1}{2}\sum_{i=1}^{N} \lambda_i \left[(\mathbf{A}\mathbf{x})_i - \mathbf{b}_i\right]^2\right). \tag{4.1.4}$$

The posterior distribution from which we need to sample is

$$\begin{aligned} p(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\delta}|\mathbf{b},\boldsymbol{\pi}) &\propto p(\mathbf{b}|\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\delta},\boldsymbol{\pi}) p_{\mathrm{1D}}(\mathbf{x}|\boldsymbol{\delta},\boldsymbol{\pi}) p(\boldsymbol{\lambda}) p(\boldsymbol{\delta}), & (4.1.5)\\ &\propto \left(\prod_{i=1}^{N} \lambda_i\right)^{\alpha_\lambda - 1/2} \left(\prod_{i=1}^{N} \delta_i\right)^{\alpha_\delta - 1/2} \\ &\quad \times \exp\left(-\sum_{i=1}^{N} \lambda_i \left(\frac{1}{2}\left[(\mathbf{A}\mathbf{x})_i - \mathbf{b}_i\right]^2 + \beta_\lambda\right)\right) \\ &\quad \times \exp\left(-\sum_{i=1}^{N} \delta_i \left(\frac{(1-\pi_i)}{2}\left(\mathbf{L}_{\mathrm{TIK}}\mathbf{x}\right)_i^2 + \frac{\pi_i}{2}\left(\mathbf{L}_{\mathrm{TV}}\mathbf{x}\right)_i^2 + \beta_\delta\right)\right). & (4.1.6) \end{aligned}$$

The posterior for 2D deconvolution is similar. We note that the conditional distributions on the individual hyperpriors are Gamma distributions:

$$\lambda_i|\mathbf{b},\mathbf{x} \sim \text{Gamma}\left(\alpha_\lambda + \frac{1}{2},\ \frac{1}{2}\left[(\mathbf{A}\mathbf{x})_i - \mathbf{b}_i\right]^2 + \beta_\lambda\right), \tag{4.1.7}$$

$$\delta_i|\mathbf{x},\pi \sim \text{Gamma}\left(\alpha_\delta + \frac{1}{2},\ \frac{(1-\pi_i)}{2}\left(\mathbf{L}_{\mathrm{TIK}}\mathbf{x}\right)_i^2 + \frac{\pi_i}{2}\left(\mathbf{L}_{\mathrm{TV}}\mathbf{x}\right)_i^2 + \beta_\delta\right). \tag{4.1.8}$$

The conditional distribution $p(\mathbf{x}|\mathbf{b},\boldsymbol{\lambda},\boldsymbol{\delta},\boldsymbol{\pi})$ is

$$\begin{aligned} p(\mathbf{x}|\mathbf{b},\boldsymbol{\lambda},\boldsymbol{\delta},\boldsymbol{\pi}) \propto \exp\Bigg(&-\frac{1}{2}\left|\left|\boldsymbol{\Lambda}^{1/2}\left(\mathbf{A}\mathbf{x} - \mathbf{b}\right)\right|\right|_2^2 \\ &-\frac{1}{2}\left|\left|\boldsymbol{\Delta}^{1/2}\left(\mathbf{I} - \boldsymbol{\Pi}\right)\mathbf{L}_{\mathrm{TIK}}\mathbf{x} + \boldsymbol{\Delta}^{1/2}\boldsymbol{\Pi}\mathbf{L}_{\mathrm{TV}}\mathbf{x}\right|\right|_2^2\Bigg) \end{aligned} \tag{4.1.9}$$

where $\boldsymbol{\Lambda}^{1/2} = \operatorname{diag}\left(\boldsymbol{\lambda}^{1/2}\right)$, $\boldsymbol{\Delta}^{1/2} = \operatorname{diag}\left(\boldsymbol{\delta}^{1/2}\right)$, and $\boldsymbol{\Pi} = \operatorname{diag}\left(\boldsymbol{\pi}\right)$. Therefore, the $\mathbf{x}|\mathbf{b},\boldsymbol{\lambda},\boldsymbol{\delta},\boldsymbol{\pi}$ follows a normal distribution:

$$p(\mathbf{x}|\mathbf{b},\boldsymbol{\lambda},\boldsymbol{\delta},\boldsymbol{\pi}) \sim \mathcal{N}\left(\mathbf{H}^{-1}\mathbf{A}^T\boldsymbol{\Lambda}\mathbf{b}, \mathbf{H}^{-1}\right), \tag{4.1.10}$$

where

$$\mathbf{H} = \mathbf{A}^T\boldsymbol{\Lambda}\mathbf{A} + \mathbf{L}_{\mathrm{TIK}}^T\left(\mathbf{I} - \boldsymbol{\Pi}\right)\boldsymbol{\Delta}\left(\mathbf{I} - \boldsymbol{\Pi}\right)\mathbf{L}_{\mathrm{TIK}} + \mathbf{L}_{\mathrm{TV}}^T\boldsymbol{\Pi}\boldsymbol{\Delta}\boldsymbol{\Pi}\mathbf{L}_{\mathrm{TV}}. \tag{4.1.11}$$

Since the conditional distributions are easy to sample, we can use a Gibbs sampler to generate samples from the posterior.

4.2 The HGSVM Sampler

The Gibbs sampler presented in Algorithm 5 is called the HGSVM sampler, short for the (h)ierarchical (G)ibbs sampler that uses (s)patially (v)arying parameters from (m)ixed regularization techniques. Along with the conditional distributions, the algorithm requires an initial approximation $\mathbf{x}^{(0)}$ to the true signal, the desired number of samples K, and the partitioning vector $\boldsymbol{\pi}$. We first sample the vectors of hyperparameters $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ from the gamma distributions in (4.1.7)–(4.1.8). We note that δ_i determines the strength of TV regularization whenever $\pi_i = 1$ and Tikhonov regularization whenever $\pi_i = 0$. We then use the sampled values of $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ to sample from the conditional distribution $p(\mathbf{x}|\mathbf{b}, \boldsymbol{\lambda}, \boldsymbol{\delta}, \boldsymbol{\pi})$. Once K samples have been drawn and the number c of initial samples to discard for burn-in has been determined, the optimal reconstruction is the mean of the remaining $K - c$ samples.

Algorithm 5: HGSVM Sampler

```
input: Initial approximation $\mathbf{x}^{(0)}$; desired number of samples $K$;
       logical vector $\boldsymbol{\pi}$; and conditional distributions $p(\mathbf{x}|\mathbf{b},\boldsymbol{\lambda},\boldsymbol{\delta}), p(\lambda_i|\mathbf{b},\mathbf{x}), p(\delta_i|\mathbf{x})$

for $k = 1,\ldots,K$ do
    for $i = 1,\ldots,N$ do
        $\lambda_i^{(k)} \sim p\left(\lambda_i|\mathbf{b},\mathbf{x}^{(k-1)}\right)$
        $\delta_i^{(k)} \sim p\left(\delta_i|\mathbf{x}^{(k-1)},\boldsymbol{\pi}\right)$
    end
    $\mathbf{x}^{(k)} \sim p\left(\mathbf{x}|\mathbf{b},\boldsymbol{\lambda}^{(k)},\boldsymbol{\delta}^{(k)},\boldsymbol{\pi}\right)$
end
```

4.3 Deconvolution Example in 1D

Here we show an example of a corrupted signal we wish to deblur using a combination of Tikhonov and TV regularization. The true and corrupted signals are shown in Figure 4.3.1. The true signal (dotted orange line) is composed of a piecewise constant region on the left half of the signal, as well as a smooth region on the right half. If we were to use only Tikhonov regularization, then the right half of the reconstructed signal would match well with the true signal. However, the features in the right half of the true signal would be lost due to enforced smoothing on the reconstruction. Similarly, if we were to use only TV regularization, then we would recover a good reconstruction of the left half of the signal. This would be at the expense of the right side of the true signal, as the reconstruction would not be able to capture the smooth bump. For a given logical vector $\boldsymbol{\pi}$, we are able to effectively combine the two regularization techniques in the hierarchical Gibbs sampler in Algorithm 5 for an optimal reconstruction.

We generate the corrupted signal (solid green line in Figure 4.3.1) from the true signal by first applying the blurring kernel

$$a(s) = \frac{1}{\gamma\sqrt{2\pi}} \exp\left(-\frac{s^2}{2\gamma^2}\right), \tag{4.3.1}$$

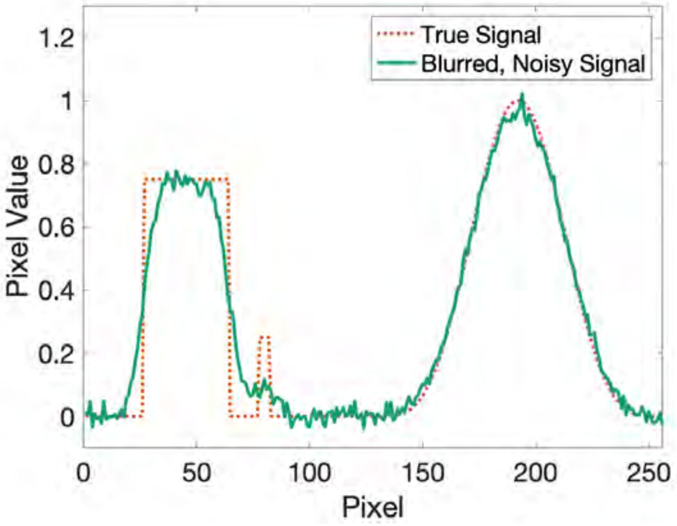
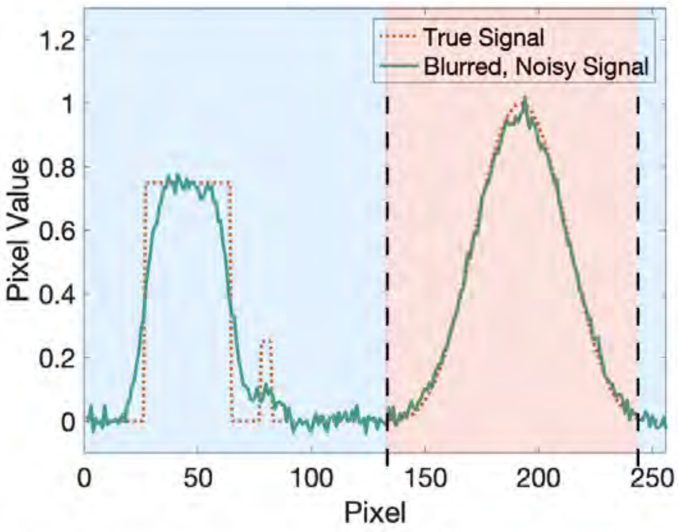


Figure 4.3.1: A true signal (dotted orange line) and its blurred signal with added noise (solid green line). The true signal contains steps on the left half and a smooth bump on the right half. Reconstructions must incorporate both Tikhonov and TV regularization to capture the true image's features.

with $\gamma = 0.02$ and assuming zero boundary conditions. After the true signal has been blurred, we add random Gaussian noise with SNR $= 25$. Since this is a simulated example, we are fortunate to know the true features of the underlying signal. This leads us to construct $\boldsymbol{\pi}$ such that for all $i = 1, \ldots, 141$, and $i = 246, \ldots, 256$, the entries $\pi_i = 1$. The remaining entries are set to 0. Figure 4.3.2 illustrates the division of the signal based on needed regularization, as well as the construction of the logical vector $\boldsymbol{\pi}$. One may wonder why we've chosen the last sliver of the signal (pixels 247 to 256) to use TV regularization, even though it seems to be a leveling off of the smooth bump. We discovered that when Tikhonov regularization is used in this small region, the reconstruction becomes oscillatory as the signal levels off to zero. Therefore, we decided TV regularization in this last piece of the signal would give reconstructions that better represented the true signal.

We run the HGSVM sampler with $\boldsymbol{\pi}$, an initial approximation $\mathbf{x}^{(0)} = \mathbf{b}$ (the blurred, noisy signal), and a maximum number of iterations $K = 10{,}000$. We discard the first 1,000 samples to account for burn-in, and take the mean of the remaining



$$\boldsymbol{\pi} = (1,1,\ldots\ldots\ldots\ldots\ldots,1,0,0,\ldots\ldots\ldots,0,1,\ldots,1)^T$$

Figure 4.3.2: An illustration of the logical vector $\boldsymbol{\pi}$. The ith entry of $\boldsymbol{\pi}$ is either 0 or 1, depending on if the ith pixel is in an area where Tikhonov (0) or TV (1) should be used. The image from Figure 4.3.1 is shown along with vertical dashed lines indicating the transition from one regularization technique to the other. The blue shaded regions require TV regularization, and the red shaded regions require Tikhonov. The logical vector $\boldsymbol{\pi}$ is shown beneath the picture.

samples to obtain the HGSVM reconstruction. The mean IACT for all λ_i's is $\overline{\boldsymbol{\tau}}_{\boldsymbol{\lambda}} \approx 1.213$; and the mean IACT for all δ_i's is $\overline{\boldsymbol{\tau}}_{\boldsymbol{\delta}} \approx 2.038$. Therefore, of the 9,000 samples taken, on average 7,419 $\boldsymbol{\lambda}$-samples and 4,416 $\boldsymbol{\delta}$-samples are effective.

Figure 4.3.3 shows the blurred, noisy signal (top left), the HGSVM reconstruction (top right), the HGSV reconstruction with Tikhonov regularization (bottom left), and the HGSV reconstruction with TV regularization (bottom right). We see that the HGSVM reconstruction matches the true signal very well, capturing the steps on the left half and the smooth bump on the right half. The only noticeable difference between the two is that the second, smaller step is shorter than the true step. However, as seen in the top left subfigure the blurring process and noise heavily affected this region, and it is amazing we see this step feature come through at all. Now, to quantify the goodness of fit of this reconstruction we compare the RMSE between

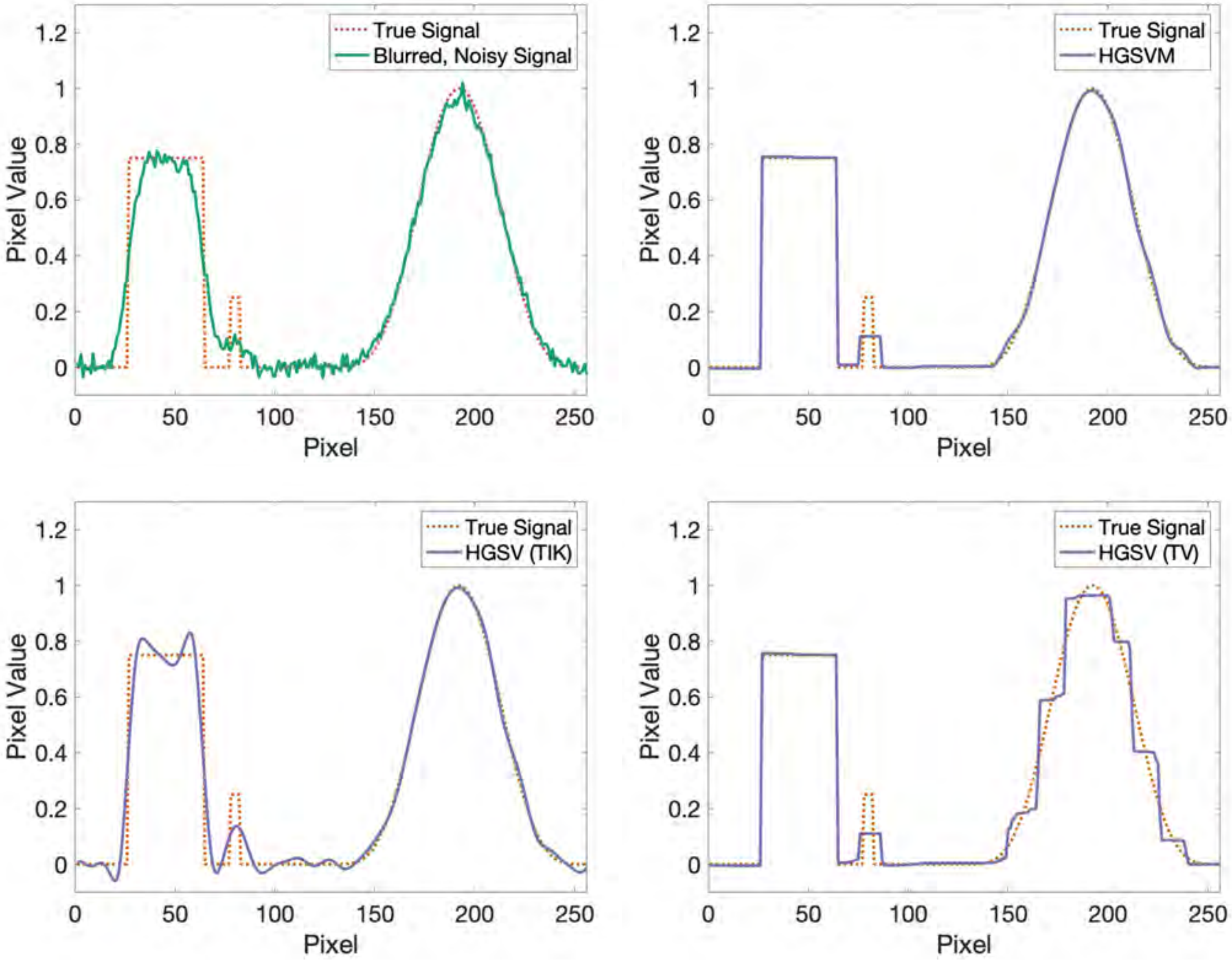


Figure 4.3.3: (Top left) The blurred, noisy signal. (Top right) The HGSVM reconstruction. (Bottom left) The HGSV reconstruction with Tikhonov regularization. (Bottom right) The HGSV reconstruction with TV regularization. The HGSVM reconstruction effectively combines the best parts of the two HGSV reconstructions, and it matches the true underlying signal well.

the true signal and the corrupted signal with that of the true signal and mean reconstruction. Let $\mathbf{x}$ be the true signal, $\mathbf{b}$ be the corrupted signal, and $\overline{\mathbf{x}}$ be the mean reconstruction. We have $\mathrm{RMSE}(\mathbf{x}, \mathbf{b}) = 0.0798$ and $\mathrm{RMSE}(\mathbf{x}, \overline{\mathbf{x}}) = 0.0274$, a 65.66% decrease in error.

4.4 A Heuristic Approach to Constructing the Partitioning Vector

In our model, we have assumed the partitioning vector $\boldsymbol{\pi}$ is known. However, determining such a partitioning scheme can be difficult, as one must make assumptions on the underlying features of a corrupted signal. The idea behind our heuristic approach is to apply the HGSV sampler (Algorithm 4) with Tikhonov regularization and then with TV regularization in order to search for indications within the reconstructions that the regularization technique is ill-suited. We have determined such indications after many experiments where we intentionally use the wrong regularization technique for a signal. Our findings show when the HGSV sampler with Tikhonov regularization is applied to corrupted piecewise constant signals, the reconstruction exhibits oscillatory behavior; and when the HGSV sampler with TV regularization is applied to corrupted smooth signals, the reconstruction exhibits a “staircasing” behavior, where the reconstruction attempts to fit the data by generating multiple irregular steps. We use the HGSV sampler as opposed to the standard Gibbs sampler because of the HGSV algorithm’s sensitivity to every piece of the signal. For instance, if a signal were made up of a mix of smooth and piecewise constant regions, the reconstruction generated by the HGSV algorithm will be more likely to show irregularities in the pieces where the chosen regularization technique is inappropriate.

Figure 4.4.1 shows two blurred and noisy signals (left column) and their corresponding HGSV reconstructions (right column) where the wrong regularization technique is implemented. The signal in the top left subfigure originates from a step function (dotted orange line in the top right subfigure) and thus requires TV regularization. However, if one were unsure and used the HGSV algorithm with Tikhonov regularization, the resulting reconstruction would be oscillatory (solid purple line in top right subfigure). Similarly, if we apply the HGSV algorithm with TV regulariza-

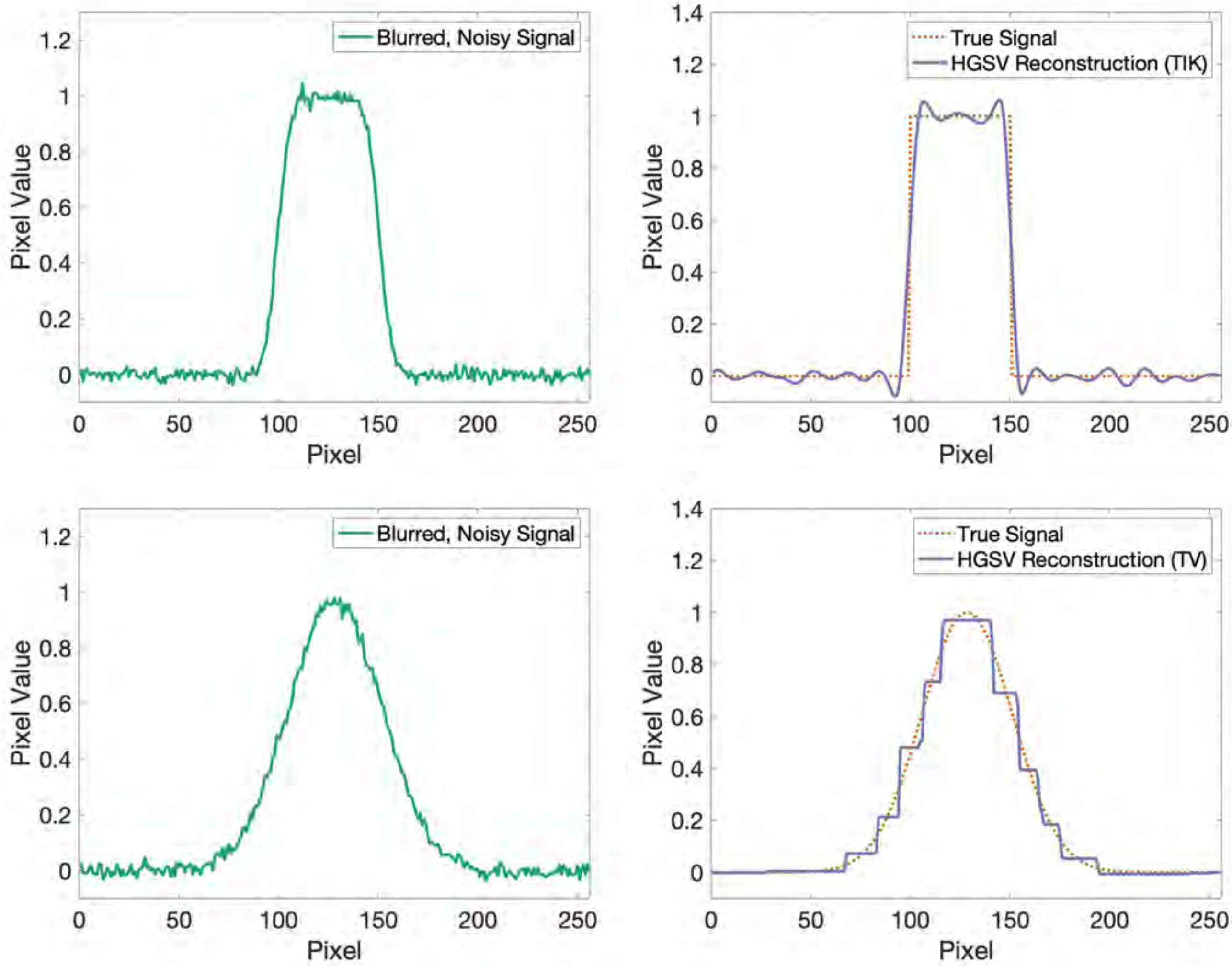


Figure 4.4.1: The left column shows two blurred signals with added noise. The top left signal requires TV regularization, and the bottom left signal requires Tikhonov regularization. However, we intentionally apply the HGSV algorithm with the wrong regularization technique in order to discover resulting artifacts. Tikhonov regularization is applied to the top left signal, and the reconstruction on the top right (solid purple line) is oscillatory. TV regularization is applied to the bottom left signal, and the reconstruction on the bottom right exhibits the staircasing effect.

tion to the bottom left signal, the reconstruction would have a staircasing effect, as shown in the bottom right subfigure.

Now consider the corrupted signal in Figure 4.4.2. We have no prior knowledge of the true signal's features; therefore, it is not clear whether to use Tikhonov or TV regularization or a combination of the two. We start by using the HGSV sampler with

Tikhonov and then TV regularization. Figure 4.4.3 shows the resulting reconstructions where the left subfigure used Tikhonov regularization, and the right subfigure used TV. We begin our analysis by splitting the reconstructions into three sections – **A**, **B**, and **C**. In section **A**, the Tikhonov-based reconstruction is oscillatory, while the TV-based reconstruction is a simple step function, with no staircasing effects. This comparison indicates that TV regularization is the appropriate technique in this region. In section **B**, we note the irregular staircasing pattern in the TV HGSV reconstruction. While the Tikhonov reconstruction is wiggly in certain parts, it is not oscillatory. Therefore, Tikhonov regularization is the appropriate technique in this region. Finally, section **C** should use TV regularization, as the TV HGSV reconstruction is completely flat and the Tikhonov HGSV reconstruction is oscillatory. Hence, we construct $\boldsymbol{\pi}$ such that its entries are 0's in section **B** (for Tikhonov) and 1's in sections **A** and **C** (for TV). Figure 4.4.4 shows the mean reconstruction from using the HGSVM sampler with our constructed $\boldsymbol{\pi}$. Since this is a simulated example, we do have access to the true signal and include it in the figure, as well (dotted orange line). We see that our choice of $\boldsymbol{\pi}$ was an appropriate one, as the reconstruction captures the true signal's features.

We end by noting a similar heuristic method can be applied in 2D, with images instead of signals. If we were given a corrupted image with no prior information about the true image, we would run the HGSV sampler with Tikhonov and TV regularization separately, then analyze the reconstructions to determine regions of irregular behavior. Staircasing effects are more easily spotted in 2D, as sharp jumps from one color to another within a small area is easier to detect with the naked eye. Once a partition is determined, one would construct a logical matrix $\mathbf{P} \in N \times N$ of 0's and 1's and then vectorize (or column-stack) $\mathbf{P}$. Therefore, we could run the HGSVM sampler with $\boldsymbol{\pi} = \text{vec}(\mathbf{P})$.

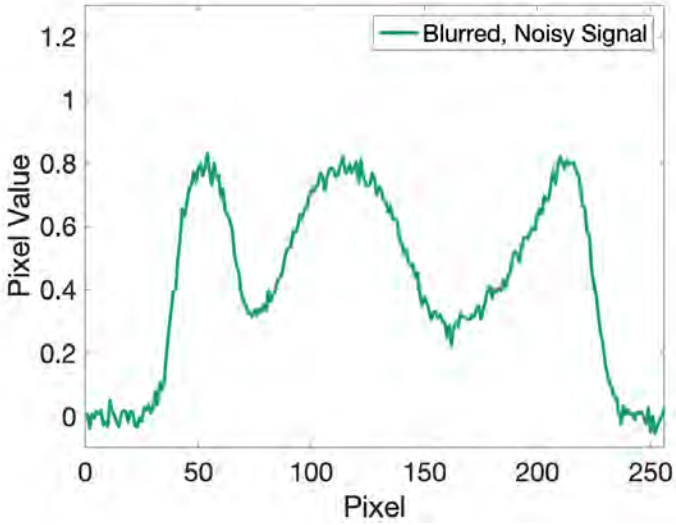


Figure 4.4.2: A blurred and noisy signal. We do not have prior knowledge of the underlying signal and, therefore, cannot determine which regularization technique (Tikhonov, TV, or a combination) is most appropriate.

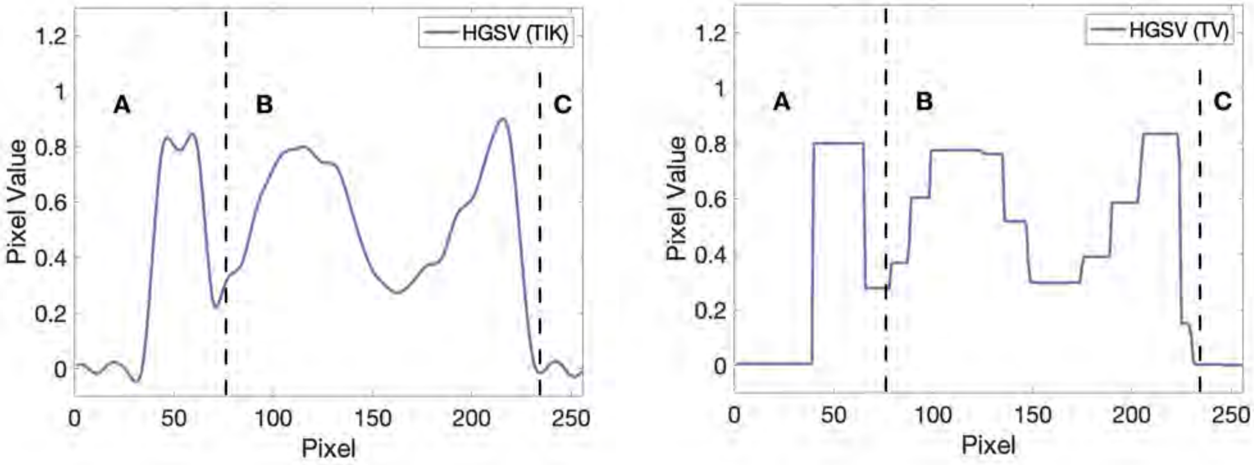


Figure 4.4.3: Two mean reconstructions from the HGSV sampler with Tikhonov (left) and TV (right) regularization. In both cases 10,000 samples were drawn with a burn-in of 1,000 samples. Comparing the two reconstructions provides us a way to construct the logical vector $\boldsymbol{\pi}$ by locating irregular behaviors. For the Tikhonov reconstruction, irregular behavior means oscillations; and for TV, irregular behavior means staircasing. Therefore, it can be seen that Tikhonov should be used in section **B**, and TV should be used in sections **A** and **C**.

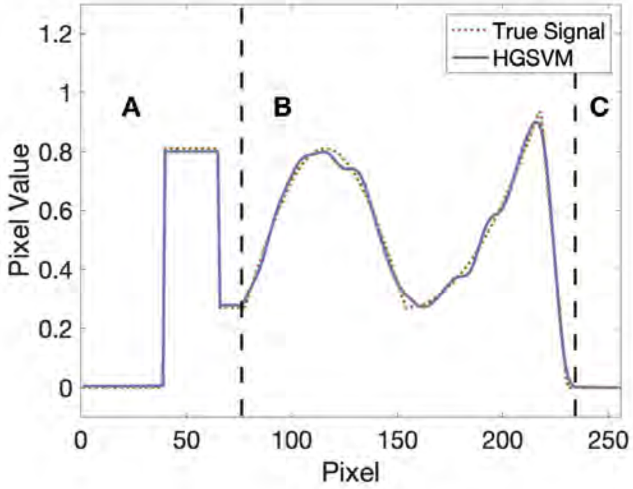


Figure 4.4.4: The mean reconstruction (solid purple line) from the HGSVM sampler with $\boldsymbol{\pi}$ such that $\pi_i = 0$ in section **B**, and $\pi_i = 1$ in sections **A** and **C**. We drew a sample of 10,000 and discarded the first 1,000 samples for burn-in. The true signal (dotted orange line) is included as well to justify our choice of partitioning for $\boldsymbol{\pi}$.

4.5 Summary

In this chapter, we adapted the HGSV model to include a partitioning vector $\boldsymbol{\pi}$, made up of 0's and 1's. If the ith entry of $\boldsymbol{\pi}$ is 1, then the ith pixel requires TV regularization; and if the entry is 0, the ith pixel requires Tikhonov regularization. We also discussed a heuristic method for determining the partition vector based on HGSV reconstructions. By computing two HGSV reconstructions, one with a Tikhonov prior and the other with a TV prior, one will likely see flaws in certain regions of the reconstructions. If the Tikhonov HGSV reconstruction is oscillatory in a region while the TV HGSV reconstruction is well-behaved (no stair-casing artifacts), then we conclude TV regularization is the appropriate choice in that region. Likewise, if the TV HGSV reconstruction exhibits stair-casing artifacts in a region while the Tikhonov HGSV reconstruction is well-behaved, then Tikhonov regularization should be used in the region.

Chapter 5

Numerical Results

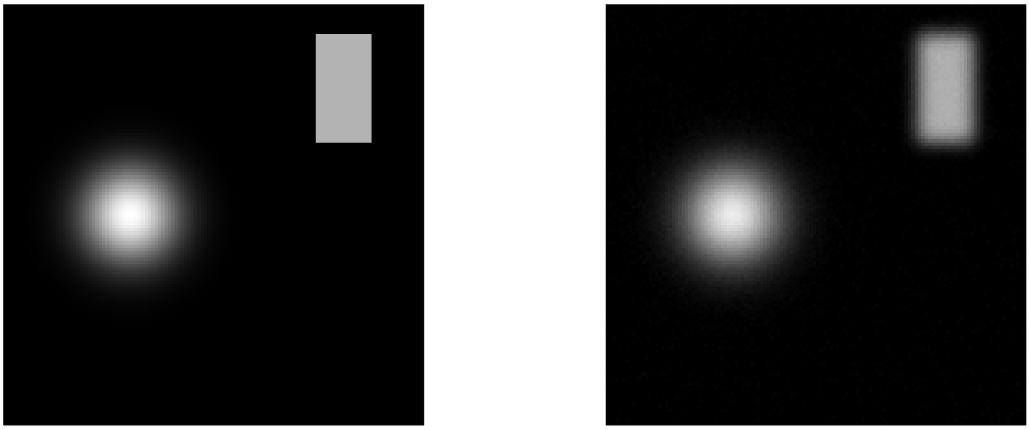
In this chapter, we apply both HGSV and HGSVM models to two-dimensional test problems. We first use two simulated test images, and we conclude by showing the methods' results on a test image from Cygnus.

5.1 Simulated Deconvolution Examples in 2D

Figures 5.1.1 and 5.1.2 show two 128 × 128 test images that have been corrupted by blur and added noise. The true images are on the left, and the blurred, noisy images are on the right. The test image in Figure 5.1.1 shows a smooth sphere-like object in the left center region of the scene and a solid rectangle in the top right. We will refer to this image as the "sphere/rectangle image". The test image in Figure 5.1.2 is a cartoon picture of a satellite with stars and planets in the background. This image is more complex, as the body of the satellite contains several interconnected pieces, with a glaring effect on the top left and bottom right panels of the satellite. We will refer to this image as the "satellite image."

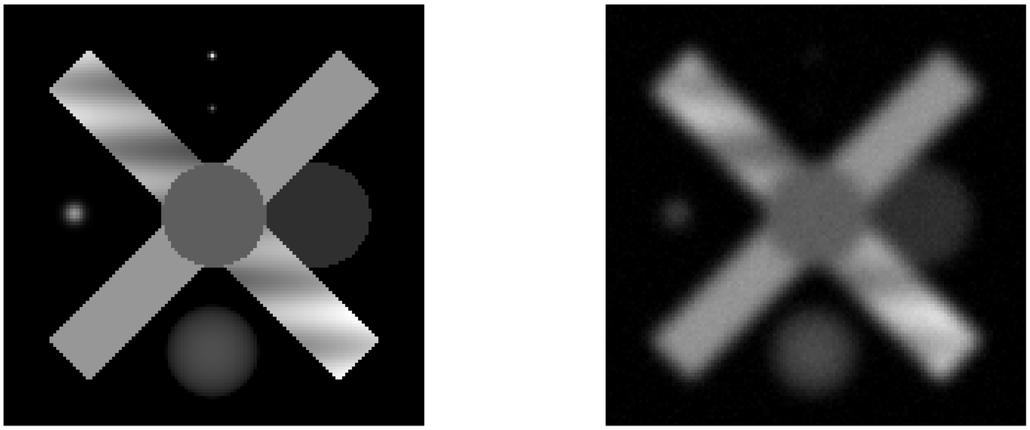
We blur both test images with the two-dimensional Gaussian kernel

$$a(s,t) = \frac{1}{2\pi\gamma^2} \exp\left(-\frac{s^2}{2\gamma^2} - \frac{t^2}{2\gamma^2}\right), \tag{5.1.1}$$



(a) True image (b) Blurred and noisy image

Figure 5.1.1: The sphere/rectangle test image. The left subfigure shows the true image, and the right subfigure shows the blurred image with added noise.



(a) True image (b) Blurred and noisy image

Figure 5.1.2: The satellite test image. The left subfigure shows the true image, and the right subfigure shows the blurred image with added noise.

assuming zero boundary conditions. Since the kernel separates into the product of two equivalent one-dimensional Gaussian functions, we can write the $128^2 \times 128^2$ blurring matrix as $\mathbf{A} = \mathbf{A}_{1\mathrm{D}} \otimes \mathbf{A}_{1\mathrm{D}}$, where $\mathbf{A}_{1\mathrm{D}}$ is an $n \times n$ Toeplitz matrix representing the one-dimensional kernel

$$a(u) = \frac{1}{\gamma\sqrt{2\pi}} \exp\left(-\frac{u^2}{2\gamma^2}\right). \tag{5.1.2}$$

Let $\mathbf{X}$ represent the original 128×128 image. We must vectorize $\mathbf{X}$ into a $128^2 \times 1$ vector in order to apply the $128^2 \times 128^2$ blurring matrix $\mathbf{A}$. We let $\mathbf{x} = \text{vec}(\mathbf{X})$, where $\text{vec}(\cdot)$ turns an $n \times n$ matrix into an $n^2 \times 1$ vector by stacking the columns of the matrix. Therefore, $\mathbf{Ax}$ is produces a blurred image that is column-stacked. We add Gaussian noise with SNR equal to 25, represented by the $128^2 \times 1$ vector $\boldsymbol{\varepsilon}$. Hence, the column-stacked blurred images are produced by the familiar linear equation

$$\mathbf{b} = \mathbf{Ax} + \boldsymbol{\varepsilon}. \tag{5.1.3}$$

5.1.1 Sphere/Rectangle Test Image

Figure 5.1.3 shows the true sphere/rectangle test image along with five reconstructions. The reconstructions labeled HG (TIK/TV) use the standard non-spatially-varying parameters hierarchical Gibbs sampler found in the Background chapter. The bottom row shows two HGSV reconstructions with each regularization technique and the final image is the HGSVM reconstruction. In all cases, 1,000 samples were drawn from the posteriors with the initial 100 samples discarded for burn-in. We note that for the HG reconstructions, there is only one λ-chain and δ-chain for the entire image; whereas for the HGSV and HGSVM reconstructions, there are 128^2 λ-chains and δ-chains. Table 5.1.1 shows the mean (in the HG case, single) IACT values for the chains. Table 5.1.2 shows the RMSE values for the reconstructions.

The HG (TIK) and HG (TV) reconstructions are still experiencing the effects of blur and noise. We see oscillations in the background of the HG (TIK) reconstruction and blurred edges in the HG (TV) reconstruction. The IACT's for the δ-parameters are high, at 21.9359 for the HG (TIK) sampler and 56.4518 for the HG (TV) sampler. This means for one sample of the δ hyperparameter, the HG (TIK) sampler would

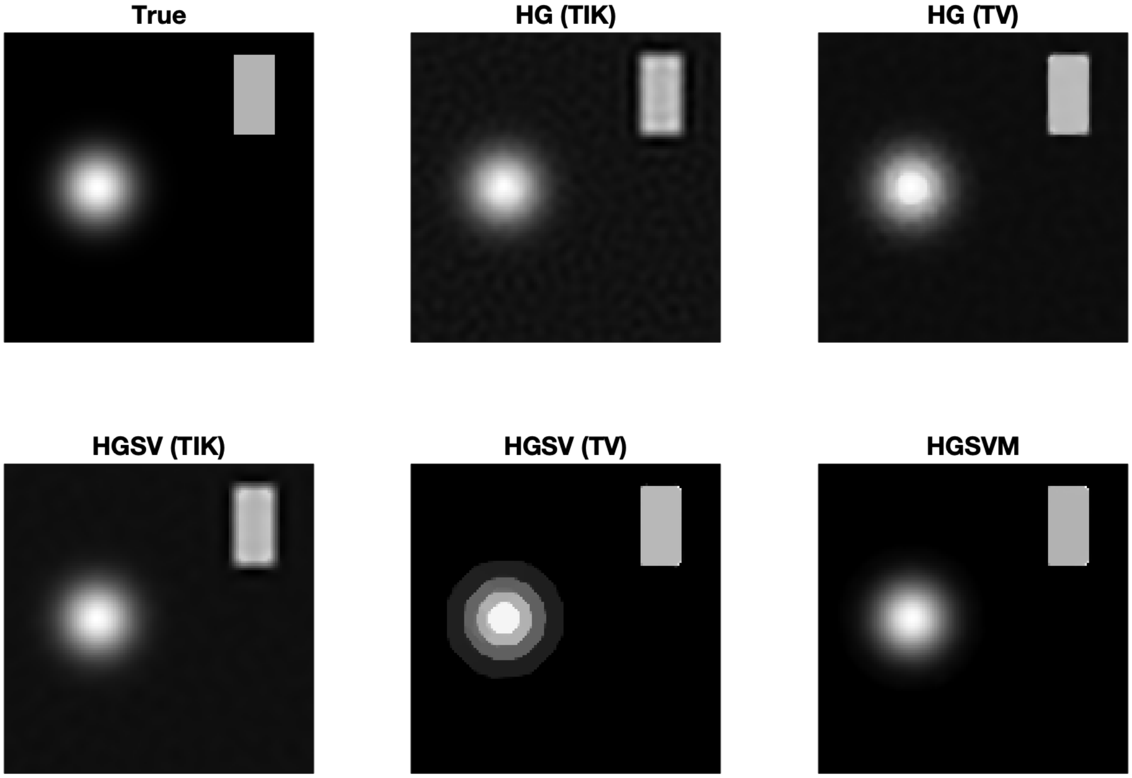


Figure 5.1.3: (Top row) The true sphere/rectangle test image; the mean reconstruction using the standard hierarchical Gibbs sampler with Tikhonov regularization; the mean reconstruction using the standard hierarchical Gibbs sampler with TV regularization. (Bottom row) HGSV reconstruction with Tikhonov regularization; HGSV reconstruction with TV regularization; HGSVM reconstruction with Tikhonov regularization applied to the sphere and TV regularization applied everywhere else.

need to generate approximately 22 more samples before obtaining an independent sample. The HG (TV) sampler would need to generate about 57 samples before obtaining an independent sample. The IACT's for the chains may improve for longer run-times and a larger burn-in. The reconstructions, however, reduce the error between the true image and blurred image by 24.88% for HG (TIK) and 62.21% for HG (TV).

The HGSV (TIK), HGSV (TV), and HGSVM samplers produce λ- and δ-chains with an average IACT's close to one, indicating that almost every sample is independent. The HGSV (TIK) reconstruction reduces the RMSE between the true and blurred image about the same amount as the HG (TIK) reconstruction, at 24.88%. The HGSV (TV) reconstruction successfully sharpens the edges of the solid rectangle; however, it also forces the smooth bump to transform into concentric circles. Despite this, the RMSE is still reduced by 45.85%. Finally, the HGSVM sampler produces a reconstruction that is the closest qualitatively and quantitatively to the true image. The sampler uses a partitioning scheme where Tikhonov regularization is used for the sphere, and TV regularization is used everywhere else. The RMSE is reduced by 80.18%.

Table 5.1.1: A comparison of mean IACT's for the parameter vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ in the sphere/rectangle reconstructions. The reconstructions HG (TIK/TV) refers to those obtained from the standard hierarchical Gibbs sampler, and HGSV (TIK/TV) refers to the reconstructions obtained from the HGSV sampler.

	$\overline{\tau}_{\boldsymbol{\lambda}}$	$\overline{\tau}_{\boldsymbol{\delta}}$
HG (TIK)	2.5946	21.9359
HGSV (TIK)	1.0191	1.1061
HG (TV)	2.1107	56.4518
HGSV (TV)	1.0100	1.9294
HGSVM	0.9967	1.1856

Table 5.1.2: A comparison of RMSE's for the sphere/rectangle reconstructions.

	RMSE $(\mathbf{x}, \mathbf{b})$	RMSE $(\mathbf{x}, \overline{\mathbf{x}})$	% Decrease in RMSE
HG (TIK)	0.0434	0.0326	24.88%
HGSV (TIK)	–	0.0330	23.96%
HG (TV)	–	0.0164	62.21%
HGSV (TV)	–	0.0235	45.85%
HGSVM	–	0.0086	80.18%

5.1.2 Satellite Test Image

Figure 5.1.4 shows the true satellite image in the left subfigure on the top row, along with HG (TIK/TV), HGSV (TIK/TV), and HGSVM reconstructions. For all samplers, 1,000 samples were drawn with the initial 100 discarded for burn-in. Table

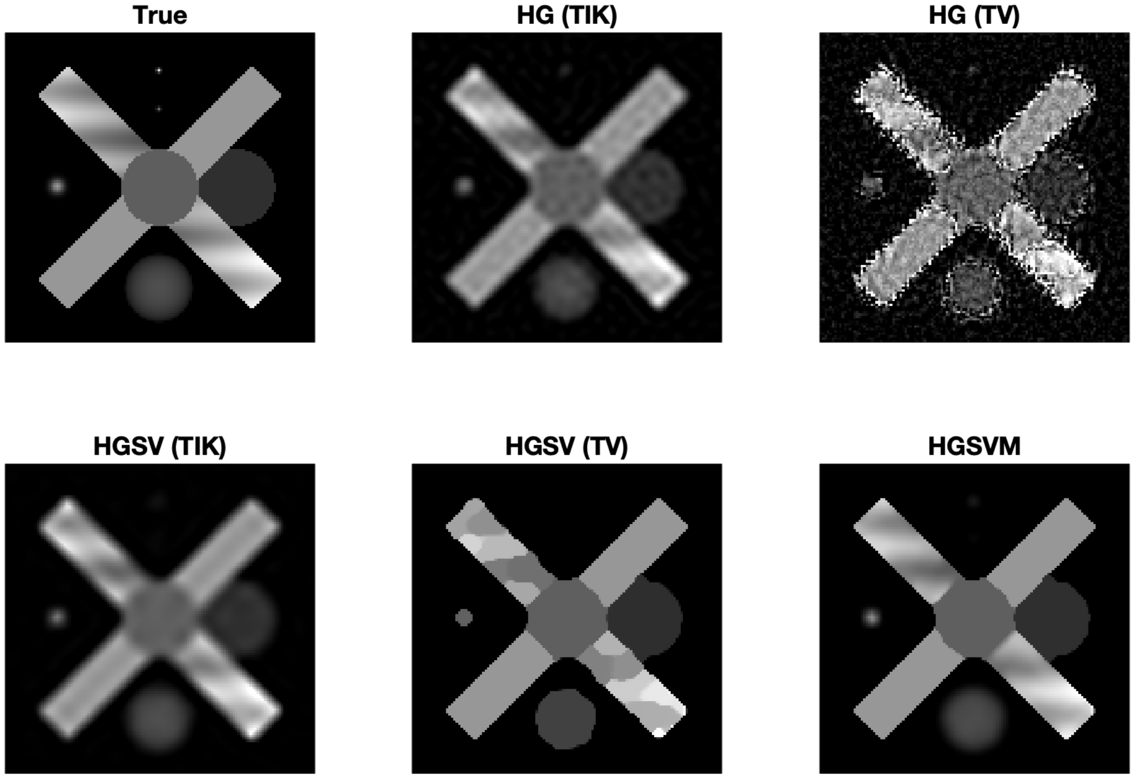


Figure 5.1.4: (Top row) The true satellite test image; the mean reconstruction using the standard hierarchical Gibbs sampler with Tikhonov regularization; the mean reconstruction using the standard hierarchical Gibbs sampler with TV regularization. (Bottom row) HGSV reconstruction with Tikhonov regularization; HGSV reconstruction with TV regularization; HGSVM reconstruction with Tikhonov regularization applied to the planets along with the top left and bottom right panels, and TV regularization applied everywhere else.

5.1.3 shows the mean IACT values for the λ- and δ-chains. Table 5.1.4 compares the errors of the reconstructions to that between the true and blurred images.

The HG (TIK) and HG (TV) reconstructions are still experiencing the effects of blur and noise. In particular, the HG (TV) reconstruction is especially noisy and fails to reduce the RMSE at all. The HG (TIK) sampler produces λ- and δ-chains with IACT's $\tau_\lambda = 3.1679$ and $\tau_\delta = 19.8757$, and the reconstruction reduces the RMSE by 26.67%, even though it is qualitatively poor. The HGSV (TIK), HGSV

(TV), and HGSVM reconstructions again produce chains with IACT values close to one, indicating that almost every sample is independent after burn-in. The HGSVM reconstruction applies Tikhonov regularization to the planets in the background, as well as the top left and bottom right satellite panels. TV regularization is applied everywhere else. The HGSVM reconstruction qualitatively matches the true satellite image well, and reduces the original RMSE value by approximately 77.93%.

Table 5.1.3: A comparison of mean IACT's for the parameter vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ in the satellite reconstructions. The reconstructions HG (TIK/TV) refers to those obtained from the standard hierarchical Gibbs sampler, and HGSV (TIK/TV) refers to the reconstructions obtained from the HGSV sampler.

	$\overline{\tau}_{\boldsymbol{\lambda}}$	$\overline{\tau}_{\boldsymbol{\delta}}$
HG (TIK)	3.1679	19.8757
HGSV (TIK)	1.0591	1.3457
HG (TV)	123.1706	40.0378
HGSV (TV)	1.0431	1.8841
HGSVM	1.0122	1.4138

Table 5.1.4: A comparison of RMSE's for the satellite reconstructions.

	RMSE $(\mathbf{x}, \mathbf{b})$	RMSE $(\mathbf{x}, \overline{\mathbf{x}})$	% Decrease in RMSE
HG (TIK)	0.0870	0.0638	26.67%
HGSV (TIK)	–	0.0674	22.53%
HG (TV)	–	0.2054	-136.09%
HGSV (TV)	–	0.0428	50.80%
HGSVM	–	0.0192	77.93%

5.2 Deconvolution with Cygnus Test Data

The left subfigure in Figure 5.2.1 shows a scene captured by the Cygnus X-ray system at the Nevada National Security Site. The objects in the scene are used for calibrating the system. The step wedge is on the left, causing the stacked solid blocks. The Abel cylinder is in the top middle of the scene, and the cylinder consists of concentric cylinders of varying densities. The "Pacman" object on the right is made of tungsten,

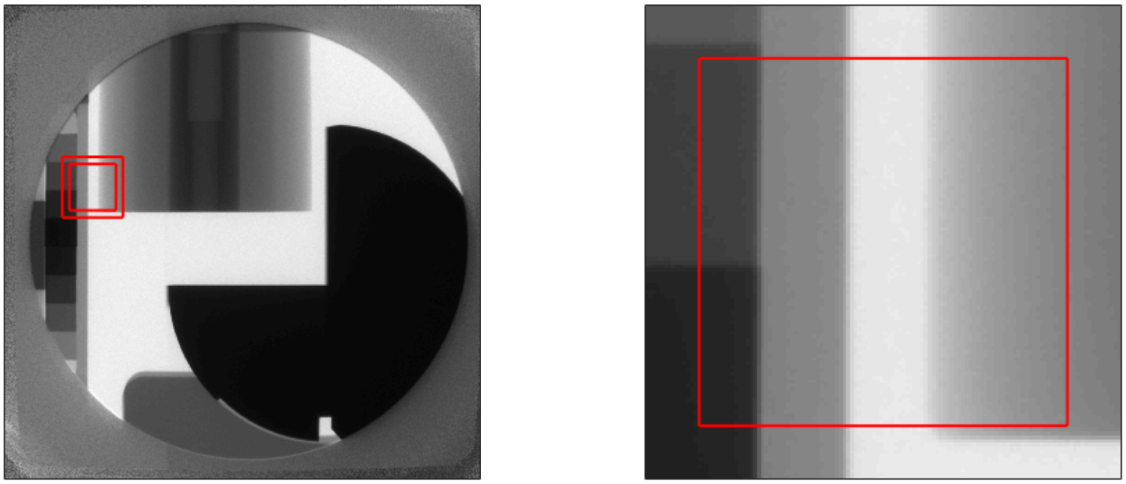


Figure 5.2.1: (Left) The 4K×4K Cygnus test image. The inner red box highlights a 400×400 sub-image of interest, and the outer red box highlights a 512×512 region containing the sub-image of interest. (Right) The zoomed in 512×512 region with the 400×400 sub-image of interest within the inner red box. For our computations, we use the larger region which will incur artifacts along its boundaries. We then crop out the boundaries and take this as the reconstruction for the smaller sub-image.

which blocks all incoming X-rays. This is why the object is completely black in the captured scene. This image is a 4K×4K resolution image, which is too large for our current methods to handle. Therefore, we consider a smaller section to apply the HGSV and HGSVM algorithms. The smaller red box in the left image is our region of interest.

The box contains a 400×400 sub-image, and is still too large for the algorithms. While we may not be able to run the algorithms with the full 400×400 image, we can downsample to a coarser version of the image and apply the algorithms to the approximated sub-image. Another issue is the non-zero boundaries of the sub-image. Our model assumes zero boundary conditions, so we expect there to be artifacts along the boundaries as a result. However, we can get around this issue by taking a larger window that contains the 400×400 sub-image, downsample, apply the algorithms to get reconstructions, and then stripping the boundaries of the reconstruction. We

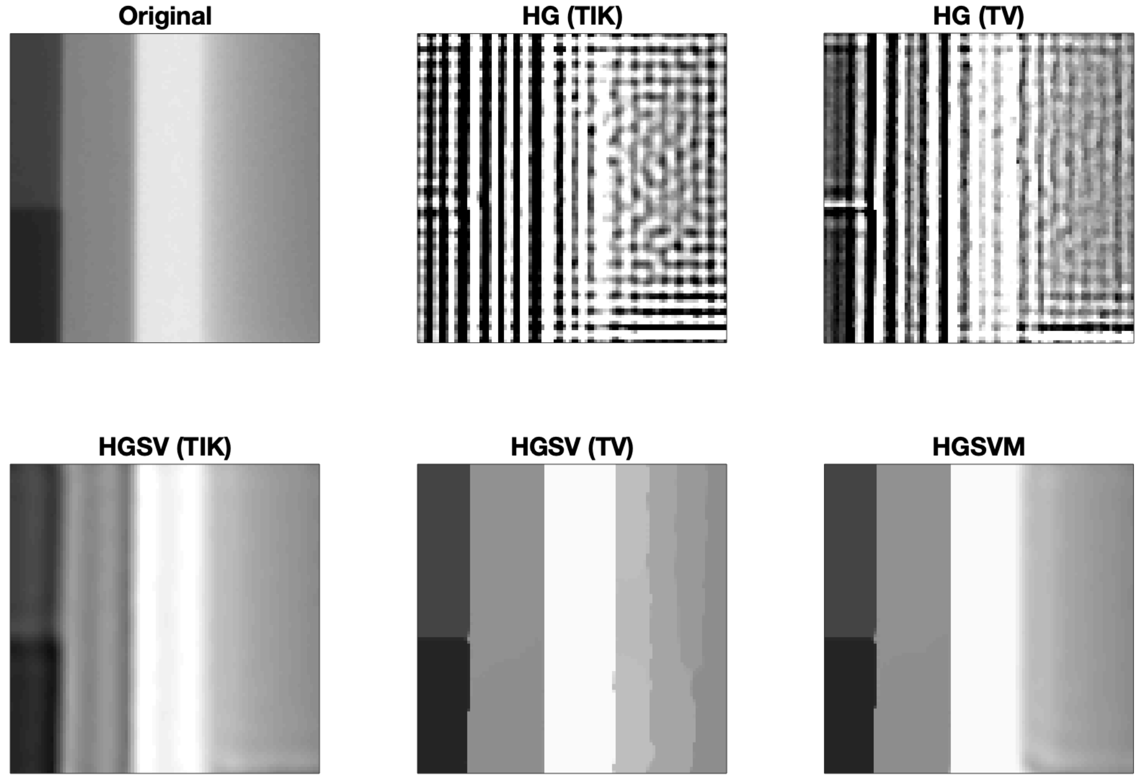


Figure 5.2.2: (Top row) The scaled-down original 100 × 100 Cygnus sub-image; the mean reconstruction using the standard hierarchical Gibbs sampler with Tikhonov regularization; the mean reconstruction using the standard hierarchical Gibbs sampler with TV regularization. (Bottom row) HGSV reconstruction with Tikhonov regularization; HGSV reconstruction with TV regularization; HGSVM reconstruction with Tikhonov regularization applied to the Abel cylinder section and TV regularization applied everywhere else.

expand the 400×400 window to a 512×512 window (outer red box in left subfigure), and then we downsample the 512×512 image to a 128×128 image for which our algorithm is better suited. When scaled down, the sub-image contained in the inner red box (right subfigure in Figure 5.2.1) is 100×100.

Figure 5.2.2 shows the original scaled-down sub-image of interest (top row, first subfigure) along with HG (TIK/TV), HGSV (TIK/TV), and HGSVM reconstructions. The blurring matrix is built by estimating the point spread function using the methods in [25]. The HG (TIK/TV) samplers fail to produce useful reconstructions, likely due to the non-zero boundary conditions. However, the HGSV (TIK/TV) samplers are successful at producing quality reconstructions within the 100 × 100 sub-image. The HGSVM sampler uses Tikhonov in the region containing the Abel cylinder and TV everywhere else. This choice is based on prior knowledge about the calibration objects. The Abel cylinder is made up of concentric cylinders; therefore when X-rayed, the image should show smooth transitions. Table 5.2.1 shows the mean IACT's for the λ- and δ-chains. The HGSV (TIK/TV) and HGSVM samplers produce chains that are on average close to one, with the exception of the δ-chains for the HGSV (TV) sampler. The mean IACT for the δ-chains are approximately 2.15, but this is due to the sampler struggling within the Abel cylinder region.

Table 5.2.1: A comparison of mean IACT's for the parameter vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ in the Cygnus reconstructions. The reconstructions HG (TIK/TV) refers to those obtained from the standard hierarchical Gibbs sampler, and HGSV (TIK/TV) refers to the reconstructions obtained from the HGSV sampler.

	$\overline{\tau}_{\boldsymbol{\lambda}}$	$\overline{\tau}_{\boldsymbol{\delta}}$
HG (TIK)	3.8247	6.2562
HGSV (TIK)	1.2673	1.5303
HG (TV)	8.1138	3.6519
HGSV (TV)	1.0759	2.1510
HGSVM	1.0817	1.5276

5.3 Summary

In this chapter we applied the HGSVM model to effectively deblur two simulated images and a sub-image captured by Cygnus. We determined the partitioning scheme for the images based on prior knowledge along with the inspection of artifacts in HGSV reconstructions with Tikhonov or TV priors. For all three images, we generated 1,000 samples and discarded the first 100 for burn-in. The HGSVM reconstructions for the sphere/rectangle and satellite test images proved to be the most successful, both qualitatively and quantitatively. For the Cygnus sub-image, the HGSVM reconstruction seems to be the most successful qualitatively, given our prior knowledge of the calibration objects.

Chapter 6

Conclusion

In this work, we have introduced a new method (the HGSVM method) that allows not only the strength, but the type of regularization to vary across an image. We do this by formulating the deblurring problem in a hierarchical Bayesian framework. Using the Bayesian paradigm, we are able to let the unknown regularization parameters be random variables, thus eliminating the need to hand-tune the parameters or choose an appropriate parameter selection method. The spatially varying parameters, especially in the case of TV regularization, produces successful reconstructions because the added noise forces certain regions of the image to require stronger regularization than others. Having the parameters vary also allows us to effectively mix regularization methods, which is necessary in the common case of deblurring an image with both smooth features and edges. The mixing of Tikhonov and TV regularization in an image is done by creating a logical partitioning vector of 1's and 0's. The partitioning vector is built based on a combination of prior knowledge and heuristics, but we aim to automate this process in the future.

We were able to successfully implement the HGSVM model on a 400×400 sub-image (scaled down to a 100×100 matrix) of a 4K×4K image captured by Cygnus. We generated 1,000 samples and discarded the initial 100 samples for burn-in. The computed IACT for the spatially varying parameters were, on average, between 1 and 2. This means about every other sample generated from the HGSVM sampler is independent.

The HGSVM method has proven to be useful in deblurring Cygnus sub-images. Future work includes fixing the sensitivity of the computed credibility intervals to the choice of gamma rate parameters from the hyperprior distributions. It would be incredibly useful to quantify uncertainties in our deblurred images. We also are interested in adjusting the HGSVM method so it can work on larger scales. To clarify, we are confident our method can successfully deblur high-resolution images when given a large storage capacity, as with a supercomputer. We would, however, like the sampler to work for larger images using a standard laptop. Currently, due to limited storage capacity, the HGSVM sampler cannot handle images larger than approximately 200×200 on a 2017 Macbook Pro.

We are confident the HGSVM method can be useful in other applications, as well. The method was built with deblurring Cygnus images in mind; however, it can be used to solve any linear inverse problem that requires a Tikhonov-type regularization method, i.e. minimizing the least squares difference between the model and the data plus a squared 2-norm regularization term. The HGSVM method can provide meaningful solutions to these types of inverse problems.

Bibliography

[1] Adams, J., Morzfeld, M., Joyce, K., Howard, M., and Luttman, A. A blocking scheme for dimension-robust gibbs sampling in large-scale image deblurring. *Inverse Problems in Science and Engineering* (2021).

[2] Barbu, A., and Zhu, S.-C. *Monte Carlo Methods.* Springer Nature, Singapore, 2020.

[3] Bardsley, J. M. *Computational uncertainty quantification for inverse problems*, vol. 19 of *Computational Science & Engineering.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2018.

[4] Bertero, M., and Boccacci, P. *Introduction to Inverse Problems in Imaging.* IOP Publishing, Bristol, UK, 1998.

[5] Brooks, S., Gelman, A., Jones, G. L., and Meng, X.-L. *Handbook of Markov Chain Monte Carlo.* Handbooks of Modern Statistical Methods. Chapman & Hall/CRC, 2011.

[6] Calvetti, D., Pragliola, M., Somersalo, E., and Strang, A. Sparse reconstructions from few noisy data: Analysis of hierarchical bayesian models with generalized gamma hyperpriors. *Inverse Problems 36*, 2 (2020).

[7] Calvetti, D., and Somersalo, E. *Introduction to Bayesian scientific computing*, vol. 2 of *Surveys and Tutorials in the Applied Mathematical Sciences.* Springer, New York, 2007. Ten lectures on subjective computing.

[8] Calvetti, D., and Somersalo, E. Hypermodels in the Bayesian imaging framework. *Inverse Problems 24*, 3 (2008).

[9] Casella, G., and Berger, R. L. *Statistical Inference.* Duxbury Press, 2001.

[10] Chan, T. F., and Shen, J. *Image processing and analysis.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005. Variational, PDE, wavelet, and stochastic methods.

[11] Cullum, J. The effective choice of the smoothing norm in regularization. *Mathematics of Computation 33*, 145 (1979), 149–170.

[12] Dogrusoz, Y. S., and Gavgani, A. M. Genetic algorithm-based regularization parameter estimation for the inverse electrocardiography problem using multiple constraints. *Med. Biol. Eng. Comput. 51* (2013), 367–375.

[13] Doicu, A., Trautmann, T., and Schreier, F. *Numerical regularization for atmospheric inverse problems.* Springer, Heidelberg; Published in association with Praxis Publishing, Chichester, 2010.

[14] Fiandrotti, A., Fosson, S. M., Ravazzi, C., and Magli, E. Gpu-accelerated algorithms for compressed signals recovery with application to astronomical imagery deblurring. *International Journal of Remote Sensing 39*, 7 (2018), 2043–2065.

[15] Geman, S., and Geman, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 6 (1984).

[16] Golub, G., Heath, M., and Wahba, G. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics 21* (1979), 215–223.

[17] Hansen, P. Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Rev. 34* (1992), 561–580.

[18] Hansen, P. C. *Discrete Inverse Problems: Insight and Algorithms*, vol. 7 of *Fundamentals of Algorithms.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2010.

[19] Hansen, P. C., Nagy, J. G., and O'Leary, D. P. *Deblurring images*, vol. 3 of *Fundamentals of Algorithms.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006. Matrices, spectra, and filtering.

[20] Howard, M., Fowler, M., Luttman, A., Mitchell, S. E., and Hock, M. C. Bayesian Abel inversion in quantitative X-ray radiography. *SIAM J. Sci. Comput. 38*, 3 (2016), B396–B413.

[21] Howard, M., Luttman, A., and Fowler, M. Sampling-based uncertainty quantification in deconvolution of X-ray radiographs. *J. Comput. Appl. Math. 270* (2014), 43–51.

[22] Idier, J. *Bayesian Approach to Inverse Problems.* Digital Signal and Image Processing Series. Wiley-ISTE, 2008.

[23] Ito, K., Jin, B., and Takeuchi, T. Multi-parameter Tikhonov regularization. *Methods Appl. Anal. 18*, 1 (2011), 31–46.

[24] Ito, K., Jin, B., and Takeuchi, T. Multi-parameter Tikhonov regularization—an augmented approach. *Chin. Ann. Math. Ser. B 35*, 3 (2014), 383–398.

[25] Joyce, K. T., Bardsley, J. M., and Luttman, A. Point spread function estimation in X-ray imaging with partially collapsed Gibbs sampling. *SIAM J. Sci. Comput. 40*, 3 (2018), B766–B787.

[26] Kabanikhin, S. I. *Inverse and Ill-Posed Problems: Theory and Applications*, vol. 55 of *Inverse and Ill-Posed Problems Series.* De Gruyter, Inc., 2012.

[27] Kaipio, J., and Somersalo, E. *Statistical and computational inverse problems*, vol. 160 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 2005.

[28] Kyung, M., Gill, J., Ghosh, M., and Casella, G. Penalized regression, standard errors, and Bayesian lassos. *Bayesian Anal. 5*, 2 (2010), 369–411.

[29] Lagendijk, R. L., and Biemond, J. *Iterative Identification and Restoration of Images.* The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, 1991.

[30] Lawson, C., and Hanson, R. *Solving Least Squares Problems.* Prentice-Hall, Englewood Cliffs, NJ, 1974, Reprinted SIAM, Philadelphia, 1995.

[31] Lee, J., and Kitanidis, P. Bayesian inversion with total variation prior for discrete geologic structure identification. *Water Resources Research 49* (2013), 7658–7669.

[32] Li, Q., and Lin, N. The Bayesian elastic net. *Bayesian Anal. 5*, 1 (2010), 151–170.

[33] Liu, J. S. *Monte Carlo strategies in scientific computing.* Springer Series in Statistics. Springer, New York, 2008.

[34] Morozov, V. On the solution of functional equations by the method of regularization. *Soviet Math. Dokl. 7* (1966), 414–417.

[35] Nelson, D. S., Ormond, E. C., Molina, I., et al. Cygnus trigger system. *IEEE Transactions on Plasma Science 36*, 5 (2008).

[36] Picozzi, M., Parolai, S., and Bindi, D. Deblurring of frequency-wavenumber images from small-scale seismic arrays. *Geophysical Journal International 181*, 1 (2010), 357–368.

[37] Rubinstein, R. Y., and Kroese, D. P. *Simulation and the Monte Carlo Method.* Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 2017.

[38] Rudin, L. I., Osher, S., and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D. 60* (1992), 259–268.

[39] Sokal, A. Monte Carlo methods in statistical mechanics: foundations and new algorithms. In *Functional integration (Cargèse, 1996)*, vol. 361 of *NATO Adv. Sci. Inst. Ser. B Phys.* Plenum, New York, 1997, pp. 131–192.

[40] Sonka, M., Hlavac, V., and Boyle, R. *Image Processing, Analysis and Machine Vision.* Chapman & Hall Computing Series. Chapman & Hall, 1993.

[41] Tang, S., Zheng, W., Xie, X., He, T., Yang, P., Luo, L., Li, Z., Hu, Y., and Zhao, H. Multi-regularization-constrained blur kernel estimation method for blind motion deblurring. *IEEE Access 7* (2019), 5296–5311.

[42] Tarantola, A. *Inverse problem theory and methods for model parameter estimation.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.

[43] Tierney, L. Markov chains for exploring posterior distributions. *The Annals of Statistics 22*, 4 (1994).

[44] Vapnik, V. N. *The nature of statistical learning theory*, second ed. Statistics for Engineering and Information Science. Springer-Verlag, New York, 2000.

[45] Vogel, C. R. *Computational methods for inverse problems*, vol. 23 of *Frontiers in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002. With a foreword by H. T. Banks.

[46] Wang, Z. Multi-parameter Tikhonov regularization and model function approach to the damped Morozov principle for choosing regularization parameters. *J. Comput. Appl. Math. 236*, 7 (2012), 1815–1832.

[47] Winkler, G. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*, vol. 27 of *Applications of Mathematics: Stochastic Modelling and Applied Probability.* Springer, 2003.

[48] Yap, K.-H., Guan, L., Perry, S. W., and Wong, H.-S. *Adaptive Image Processing: A Computational Intelligence Perspective.* Image Processing Series. CRC Press, Boca Raton, FL, 2010.

[49] Zhao, H., Ke, Z., et al. A new deep learning method for image deblurring in optical microscopic systems. *Journal of Biophotonics 13*, 3 (2020).

[50] Zou, H., and Hastie, T. Addendum: “Regularization and variable selection via the elastic net” [J. R. Stat. Soc. Ser. B Stat. Methodol. **67** (2005), no. 2, 301–320; mr2137327]. *J. R. Stat. Soc. Ser. B Stat. Methodol. 67*, 5 (2005), 768.