MAT-52506 Inverse Problems

Samuli Siltanen
Version 13

February 9, 2010

Contents

Chapter 1

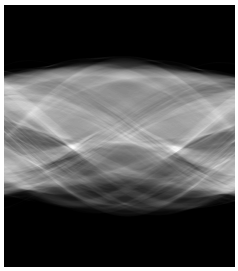
Introduction

Inverse problems are the opposites of direct problems. Informally, in a direct problem one finds an effect from a cause, and in an inverse problem one is given the effect and wants to recover the cause. The most usual situation giving rise to an inverse problem is the need to interpret indirect physical measurements of an unknown object of interest.

For example in medical X-ray tomography the direct problem would be to find out what kind of X-ray projection images would we get from a patient whose internal organs we know precisely. The corresponding inverse problem is to reconstruct the three-dimensional structure of the patient's insides given a collection of X-ray images taken from different directions.

Direct problem →

← Inverse problem



Here the patient is the cause and the collection of X-ray images is the effect.

Another example comes from image processing. Define the direct problem as finding out how a given sharp photograph would look like if the camera was incorrectly focused. The inverse problem known as *deblurring* is finding the sharp photograph from a given blurry image.

Direct problem →

← Inverse problem

Here the cause is the sharp image and the effect is the blurred image.

There is an apparent symmetry in the above explanation: without further restriction of the definitions, direct problem and inverse problem would be in identical relation with each other. For example, we might take as the direct problem the determination of a positive photograph from the knowledge of the negative photograph.

Direct problem

"Inverse problem"

In this case the corresponding "inverse problem" would be inverting a given photograph to arrive at the negative. Here both problems are easy and stable, and one can move between them repeatedly.

However, inverse problems research concentrates on situations where the inverse problem is more difficult to solve than the direct problem. More precisely, let us recall the notion of a *well-posed problem* introduced by Jacques Hadamard (1865-1963):

The problem must have a solution (existence). (1.1)

The problem must have at most one solution (uniqueness). (1.2)

The solution must depend continuously on input data (stability). (1.3)

An inverse problem, in other words an *ill-posed problem*, is any problem that is not well-posed. Thus at least one of the conditions (1.1)–(1.3) must fail in order for a problem to be an inverse problem. This rules out the positive-negative example above.

To make the above explanation more precise, let us introduce the linear measurement model discussed throughout the document:

$$m = Ax + \varepsilon,$$

where $x \in \mathbb{R}^n$ and $m \in \mathbb{R}^k$ are vectors, A is a $k \times n$ matrix, and ε is a random vector taking values in $\mathbb{R}^k$. Here m is the data provided by a measurement device, x is a discrete representation of the unknown object, A is a matrix modeling the measurement process, and ε is random error. The inverse problem is

Given m, find an approximation to x.

We look for a reconstruction procedure $R : \mathbb{R}^k \to \mathbb{R}^n$ that would satisfy $R(m) \approx x$, the approximation being better when the size $\|\varepsilon\|$ of the noise is small. The connection between R and Hadamard's notions is as follows: m is the input and

$R(m)$ is the output. Now (1.1) means that the function R should be defined on all of $\mathbb{R}^k$, condition (1.2) states that R should be a single-valued mapping, and (1.3) requires that R should be continuous. For well-posed problems one can simply take $R(m) = A^{-1}m$, but for ill-posed problems that straightforward approach will fail.

This document is written to serve as lecture notes for my course *Inverse Problems* given at Department of Mathematics of Tampere University of Technology. Since more than half the students major in engineering, the course is designed to be very application-oriented. Computational examples abound, and the corresponding Matlab routines are available at the course web site. Several discrete models of continuum measurements are constructed for testing purposes. We restrict to linear inverse problems only to avoid unnecessary technical difficulties. Special emphasis is placed on extending the reconstruction methods to practical large-scale situations; the motivation for this stems from the author's experience of research and development work on medical X-ray imaging devices at Instrumentarium Imaging, GE Healthcare, and Palodex Group.

Important sources of both inspiration and material include the Finnish lecture notes created and used by Erkki Somersalo in the 1990's, the books by Jari Kaipio and Erkki Somersalo [13], Andreas Kirsch [15], Curt Vogel [28] and Per Christian Hansen [9], and the lecture notes of Guillaume Bal [1].

I thank the students of the fall 2008 course for valuable comments that improved these lecture notes (special thanks to Esa Niemi, who did an excellent job in editing parts of the manuscript, and to Lauri Harhanen for his work on the Gibbs sampler computation).

Chapter 2

Linear measurement models

The basic model for indirect measurements used in this course is the following matrix equation:

$$m = Ax + \varepsilon, \tag{2.1}$$

where $x \in \mathbb{R}^n$ and $m \in \mathbb{R}^k$ are vectors, A is a $k \times n$ matrix, and ε is a random vector taking values in $\mathbb{R}^k$. In Sections 2.2 and 2.3 below we will construct a couple of models of the form (2.1) explicitly by discretizing continuum models of physical situations. We will restrict ourselves to white Gaussian noise only and give a brief discussion of noise statistics in Section 2.1.

2.1 Measurement noise

In this course we restrict ourselves to Gaussian white noise. In terms of the random noise vector

$$\varepsilon = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k]^T$$

appearing in the basic equation (2.1) we require that each random variable $\varepsilon_j : \Omega \to \mathbb{R}$ with $1 \leq j \leq k$ is independently distributed according to the normal distribution: $\varepsilon_j \sim \mathcal{N}(0, \sigma^2)$, where $\sigma > 0$ is the standard deviation of ε_j. In other words, the probability density function of ε_j is

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-s^2/(2\sigma^2)}.$$

We will call σ the *noise level* in the sequel.

In many examples the noise may be multiplicative instead of additive, and the noise statistics may differ from Gaussian. For instance, photon counting instruments typically have Poisson distributed noise. As mentioned above, these cases will not be discussed in this treatise.

2.2 Convolution

Linear convolution is a useful process for modeling a variety of practical measurements. The one-dimensional case with a short and simple point spread function will serve us as a basic example that can be analyzed in many ways.

When the point spread function is longer and more complicated, the one-dimensional deconvolution (*deconvolution* is the inverse problem corresponding to convolution understood as direct problem) can model a variety of practical engineering problems, including removing blurring by device functions in physical measurements or inverting finite impulse response (FIR) filters in signal processing.

Two-dimensional deconvolution is useful model for deblurring images; in other words removing errors caused by imperfections in an imaging system. The dimension of inverse problems appearing in image processing can be very large; thus two-dimensional deconvolution acts as a test bench for large-scale inversion methods.

2.2.1 One-dimensional case

We build a model for one-dimensional deconvolution. The continuum situation concerns a signal $\mathcal{X} : [0,1] \to \mathbb{R}$ that is blurred by a *point spread function* (PSF) $\psi : \mathbb{R} \to \mathbb{R}$. (Other names for the point spread function include *device function, impulse response, blurring kernel, convolution kernel and transfer function.*) We assume that the PSF is strictly positive: $\psi(s) \geq 0$ for all $s \in \mathbb{R}$. Furthermore, we require that ψ is symmetric ($\psi(s) = \psi(-s)$ for all $s \in \mathbb{R}$) and compactly supported ($\psi(s) \equiv 0$ for $|s| > a > 0$). Also, we use the following normalization for the PSF:

$$\int_{\mathbb{R}} \psi(\lambda)\, d\lambda = 1. \tag{2.2}$$

The continuum measurement $\mathcal{M} : [0,1] \to \mathbb{R}$ is defined with the convolution integral

$$\mathcal{M}(s) = (\psi * \mathcal{X})(s) = \int_{\mathbb{R}} \psi(\lambda)\mathcal{X}(s-\lambda)\, d\lambda, \qquad s \in [0,1], \tag{2.3}$$

where we substitute the value $\mathcal{X}(s) = 0$ for $s < 0$ and $s > 1$.

Note that we do not include random measurement noise in this continuum model. This is just to avoid technical considerations related to infinite-dimensional stochastic processes.

Let us build a simple example to illustrate our construction. Define the signal and PSF by the following formulas:

$$\mathcal{X}(s) = \begin{cases} 1 \text{ for } \frac{1}{4} \leq s \leq \frac{1}{2}, \\ 2 \text{ for } \frac{3}{4} \leq s \leq \frac{7}{8}, \\ 0 \text{ otherwise,} \end{cases} \qquad \psi(s) = \begin{cases} 10 \text{ for } |s| \leq \frac{1}{20}, \\ 0 \text{ otherwise.} \end{cases} \tag{2.4}$$

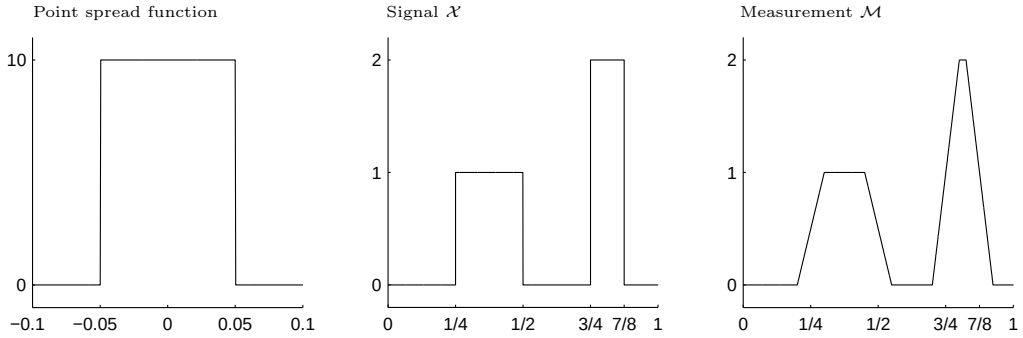


Figure 2.1: Continuum example of one-dimensional convolution.

Note that ψ satisfies the requirement (2.2). See Figure 2.1 for plots of the point spread function ψ and the signal $\mathcal{X}$ and the measurement $\mathcal{M}$.

Next we need to discretize the continuum problem to arrive at a finite-dimensional measurement model of the form (2.1). Throughout the course the dimension of the discrete measurement m is denoted by k, so let us define $m = [m_1, m_2, \ldots, m_k]^T \in \mathbb{R}^k$. We choose in this section to represent the unknown as a vector x with the same dimension as the measurement m. The reason for this is just the convenience of demonstrating inversion with a square-shaped measurement matrix; in general the dimension of x can be chosen freely. Define

$$s_j := (j-1)\Delta s \quad \text{for} \quad j = 1, 2, \ldots, k, \quad \text{where} \quad \Delta s := \frac{1}{k-1}.$$

Now vector $x = [x_1, x_2, \ldots, x_k]^T \in \mathbb{R}^k$ represents the signal $\mathcal{X}$ as $x_j = \mathcal{X}(s_j)$.

We point out that the construction of Riemann integral implies that for any reasonably well-behaving function $f : [0,1] \to \mathbb{R}$ we have

$$\int_0^1 f(s)\, ds \approx \Delta s \sum_{j=1}^{k} f(s_j), \tag{2.5}$$

the approximation becoming better when k grows. We will repeatedly make use of (2.5) in the sequel.

Let us construct a matrix A so that Ax approximates the integration in (2.3). We define a discrete PSF denoted by

$$p = [p_{-N}, p_{-N+1}, \ldots, p_{-1}, p_0, p_1, \ldots, p_{N-1}, p_N]^T$$

as follows. Recall that $\psi(s) \equiv 0$ for $|s| > a > 0$. Take $N > 0$ to be the smallest integer satisfying the inequality $N\Delta s > a$ and set

$$\tilde{p}_j = \psi(j\Delta s) \text{ for } j = -N, \ldots, N.$$

By (2.5) the normalization condition (2.2) almost holds: $\Delta s \sum_{j=-N}^{N} \tilde{p}_j \approx 1$. However, in practice it is a good idea to normalize the discrete PSF explicitly by the

formula

$$p = \left(\Delta s \sum_{j=-N}^{N} \tilde{p}_j \right)^{-1} \tilde{p}; \tag{2.6}$$

then it follows that

$$\Delta s \sum_{j=-N}^{N} p_j = 1.$$

Discrete convolution is defined by the formula

$$(p * x)_j = \sum_{\nu=-N}^{N} p_\nu x_{j-\nu}, \tag{2.7}$$

where we implicitly take $x_j = 0$ for $j < 1$ and $j > k$. Then we define the measurement by

$$m_j = \Delta s (p * x)_j + \varepsilon_j, \tag{2.8}$$

The measurement noise ε is explained in Section 2.1. Now (2.7) is a Riemann sum approximation to the integral (2.3) in the sense of (2.5), so we have

$$m_j \approx \mathcal{M}(s_j) + \varepsilon_j.$$

But how to write formula (2.8) using a matrix A so that we would arrive at the desired model (2.1)? More specifically, we would like to write

$$\begin{bmatrix} m_1 \\ \vdots \\ m_k \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_k \end{bmatrix}.$$

The answer is to build a circulant matrix having the elements of p appearing systematically on every row of A. We explain this by example.

We take $N = 2$ so the PSF takes the form $p = [p_{-2} \quad p_{-1} \quad p_0 \quad p_1 \quad p_2]^T$. According to (2.7) we have $(p*x)_1 = p_0x_1 + p_{-1}x_2 + p_{-2}x_3$. This can be visualized as follows:

p_2	p_1	p_0	p_{-1}	p_{-2}				
		x_1	x_2	x_3	x_4	x_5	x_6	. . .

It follows that the first row of matrix A is $[p_0 \quad p_1 \quad p_2 \quad 0 \quad \cdots \quad 0]$. The construction of the second row comes from the picture

p_2	p_1	p_0	p_{-1}	p_{-2}			
	x_1	x_2	x_3	x_4	x_5	x_6	. . .

and the third row from the picture

p_2	p_1	p_0	p_{-1}	p_{-2}		
x_1	x_2	x_3	x_4	x_5	x_6	. . .

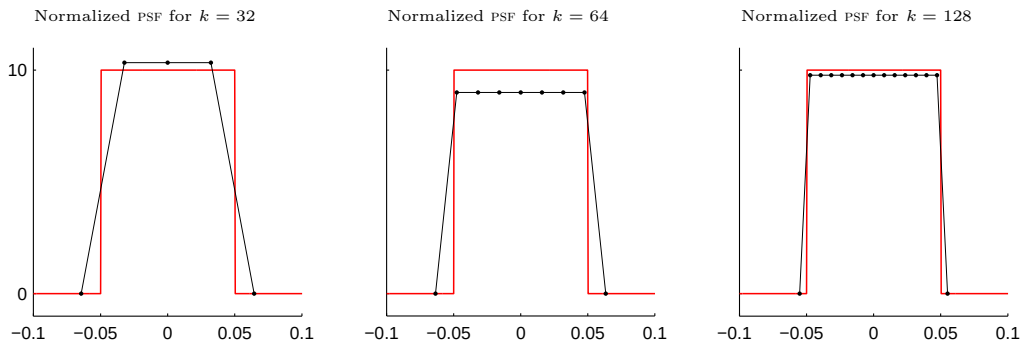


Figure 2.2: The dots denote values of discrete objects and the lines show the continuum PSF 2.1 for comparison. Note that the continuum and discrete PSFs do not coincide very accurately; this is due to the normalization (2.6).

The general construction is hopefully clear now, and the matrix A looks like this:

$$A = \begin{bmatrix} p_0 & p_{-1} & p_{-2} & 0 & 0 & 0 & \cdots & 0 \\ p_1 & p_0 & p_{-1} & p_{-2} & 0 & 0 & \cdots & 0 \\ p_2 & p_1 & p_0 & p_{-1} & p_{-2} & 0 & \cdots & 0 \\ 0 & p_2 & p_1 & p_0 & p_{-1} & p_{-2} & \cdots & 0 \\ \vdots & & & & \ddots & & & \\ \vdots & & & & & \ddots & & \\ 0 & & \cdots & & p_2 & p_1 & p_0 & p_{-1} & p_{-2} \\ 0 & & \cdots & & 0 & p_2 & p_1 & p_0 & p_{-1} \\ 0 & & \cdots & & 0 & 0 & p_2 & p_1 & p_0 \end{bmatrix}$$

The Matlab command `convmtx` can be used to construct such convolution matrices.

Let us return to example (2.4). See Figure 2.2 for plots of discretized PSFs corresponding to ψ with different choices of k. Further, see Figures 2.3 and 2.4 for examples of measurements $m = Ax + \varepsilon$ done using the above type convolution matrix with $k = 32$ and $k = 64$, respectively. We can see that with the proper normalization (2.6) of the discrete PSFs our discretized models do produce measurements that approximate the continuum situation.

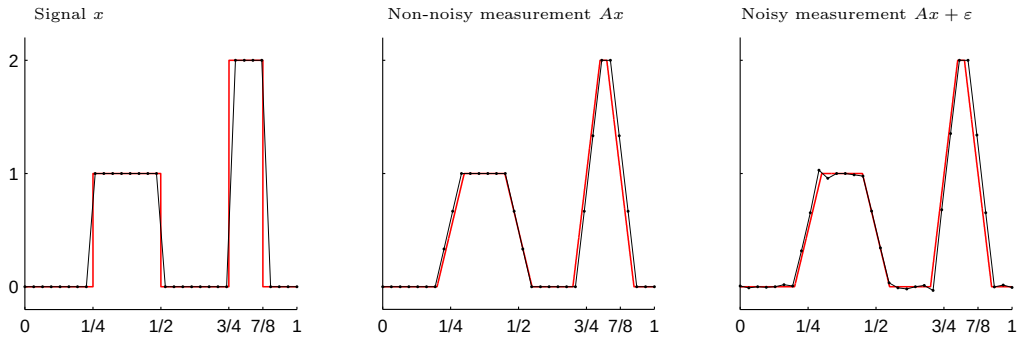


Figure 2.3: Discrete example of one-dimensional convolution. Here $k = n = 32$ and we use the discrete PSF shown in the leftmost plot of Figure 2.2. The dots denote values of discrete objects and the lines show the corresponding continuum objects for comparison. The noise level is $\sigma = 0.02$.

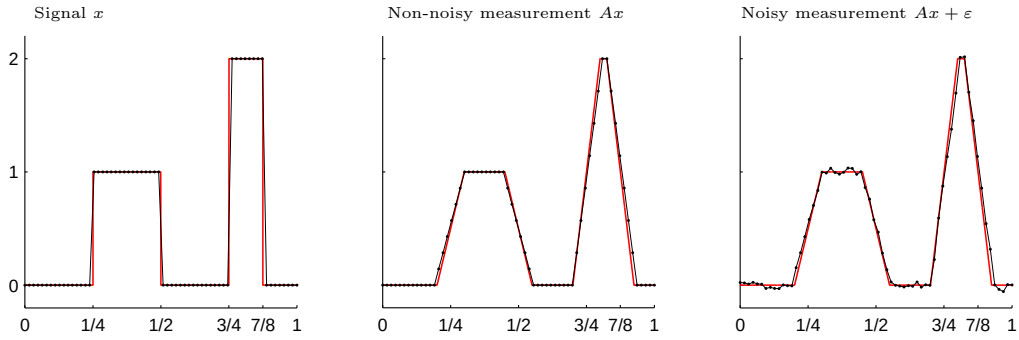


Figure 2.4: Discrete example of one-dimensional convolution. Here $k = n = 64$ and we use the discrete PSF shown in the middle plot of Figure 2.2. The dots denote values of discrete objects and the lines show the corresponding continuum objects for comparison. The noise level is $\sigma = 0.02$.

2.2.2 Two-dimensional case

Consider a pixel image X with K rows and L columns. We index the pixels according to MATLAB standard:

X_{11}	X_{12}	X_{13}	X_{14}	$\cdots$			X_{1L}
X_{21}	X_{22}	X_{23}					
X_{31}	X_{32}	X_{33}					
X_{41}			$\ddots$				
$\vdots$							
X_{K1}	$\cdots$						X_{KL}

.

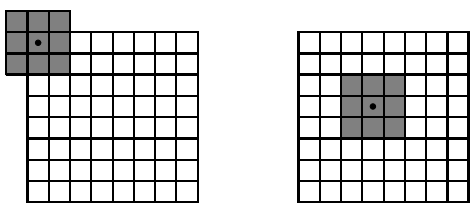
We introduce a two-dimensional point spread function (here 3×3 for ease of demonstration) p with the following naming convention:

$$p = \begin{array}{|c|c|c|} \hline p_{(-1)(-1)} & p_{(-1)0} & p_{(-1)1} \\ \hline p_{0(-1)} & p_{00} & p_{01} \\ \hline p_{1(-1)} & p_{10} & p_{11} \\ \hline \end{array} \quad . \tag{2.9}$$

The two-dimensional convolution $p * X$ is defined by

$$(p * X)_{k\ell} = \sum_{i=-1}^{1} \sum_{j=-1}^{1} X_{(k-i)(\ell-j)} p_{ij} \tag{2.10}$$

for $1 \leq k \leq K$ and $1 \leq \ell \leq L$ with the convention that $x_{k\ell} = 0$ whenever $k, \ell < 1$ or $k > K$ or $\ell > L$. The operation $p * X$ can be visualized by a mask p moving over the image X and taking weighted sums of pixels values:



Consider now the direct problem $X \mapsto p * X$. How to write it in the standard form $m = Ax$? Express the pixel image X as a vector $x \in \mathbb{R}^{KL}$ by renumerating

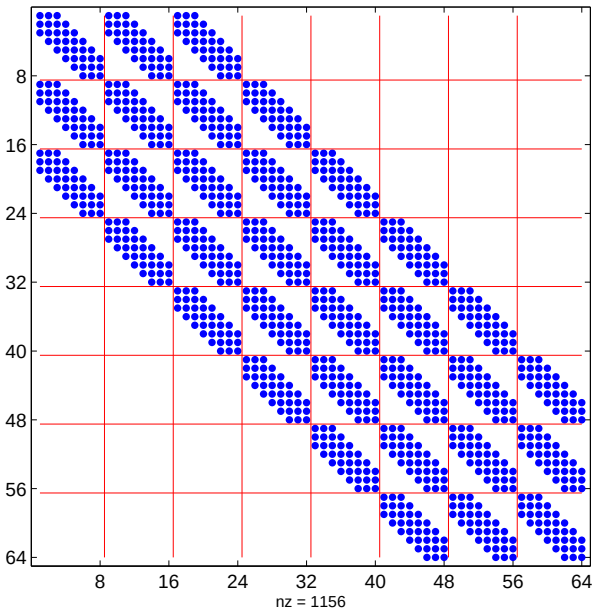


Figure 2.5: Nonzero elements (blue dots) in a two-dimensional convolution matrix corresponding to an 8×8 image and 3×3 point spread function. The 8×8 block structure is indicated by red lines.

the pixels as follows:

x_1	x_{K+1}						
x_2	x_{K+2}						
x_3	x_{K+3}						
.	.						
.	.						
.	.						
.	.						
x_K	x_{2K}						x_{KL}

(2.11)

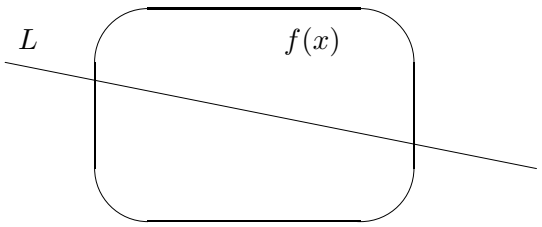
Note that this renumeration corresponds to the MATLAB operation `x = X(:)`. The $KL \times KL$ measurement matrix A can now be constructed by combining (2.9) and (2.10) and (2.11). In the case $K = 8$ and $L = 8$ the nonzero elements in A are located as shown in Figure 2.5. The exact construction is left as an exercise.

2.3 Tomography

Tomography is related to recovering a function from the knowledge of line integrals of the function over a collection of lines.

In this work tomographic problems provide examples with nonlocal merging of information (as opposed to roughly local convolution kernels) combined naturally with large-scale problems. Also, geometrical restrictions in many practical applications lead to limited angle tomography, where line integrals are available only from a restricted angle of view. The limited angle tomography is significantly more ill-posed than full-angle tomography, providing excellent test cases for inversion methods.

In X-ray tomography the line integrals of the function are based on X-ray images. X-ray imaging gives a relation between mathematics and real world via the following model. When a X-ray travels through a physical object (patient) along a straight line L, interaction between radiation and matter lowers the intensity of the ray. We think of the X-ray having initial intensity I_0 when entering the object and smaller intensity I_1 when exiting the object.



The physical object is represented by a non-negative attenuation coefficient function $f(x)$, whose value gives the relative intensity loss of the X-ray within a small distance dx:

$$\frac{dI}{I} = -f(x)dx.$$

A thick tissue such as bone has higher attenuation coefficient than, say, muscle. Integration from initial to final state gives

$$\int_0^1 \frac{I'(x)dx}{I(x)} = -\int_0^1 f(x)dx,$$

where the left-hand side equals $\log I_1 - \log I_0 = \log I_1/I_0$. Thus we get

$$\frac{I_1}{I_0} = e^{-\int_L f(x)dx}. \tag{2.12}$$

Now the left hand side of (2.12) is known from measurements (I_0 by calibration and I_1 from detector), whereas the right hand side of (2.12) consists of integrals of the unknown function f over straight lines.

We remark that in the above model we neglect scattering phenomena and the energy dependence of the attenuation function.

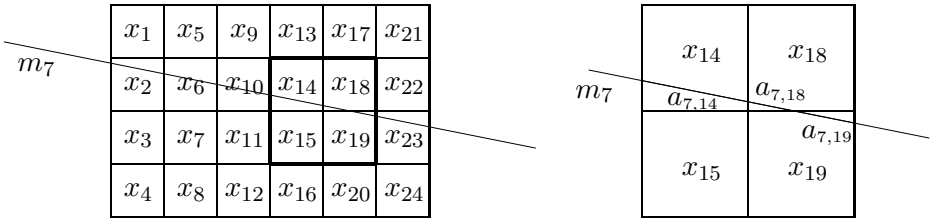


Figure 2.6: Left: discretized object and an X-ray traveling through it. Right: four pixels from the left side picture and the distances (in these pixels) traveled by the X-ray corresponding to the measurement m_7. Distance $a_{i,j}$ corresponds to the element on the ith row and jth column of matrix A.

In order to express the continuous model in the matrix form (2.1) we divide the object into pixels (or voxels in 3D case), e.g. like shown in Figure 2.6. Now each component of x represents the value of the unknown attenuation coefficient function f in the corresponding pixel. Assume we have a measurement m_i of the line integral of f over line L. Then we can approximate

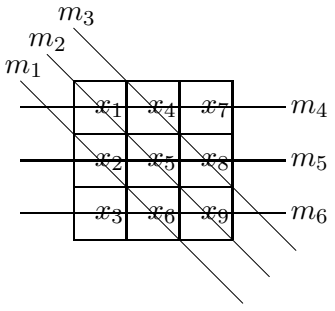
$$m_i = \int_L f(x)dx \approx \sum_{j=1}^{n} a_{i,j}x_j, \tag{2.13}$$

where $a_{i,j}$ is the distance that L "travels" in the pixel corresponding to x_j. Further, if we have k measurements in vector $m \in \mathbb{R}^k$, then (2.13) yields a matrix equation $m = Ax$, where matrix $A = (a_{i,j})$.

To illustrate how the matrix A is constructed, consider the discretization and X-ray (measurement m_7) in Figure 2.6. The equation for the measurement m_7 is

$$m_7 = a_{7,2}x_2 + a_{7,6}x_6 + a_{7,10}x_{10} + a_{7,14}x_{14} + a_{7,18}x_{18} + a_{7,19}x_{19} + a_{7,23}x_{23}.$$

In other words the ith row of A is related to the measurement m_i. Let us take another example. With the following discretization and measurements



the model can be written in the matrix form as follows:

$$\begin{bmatrix} 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 \\ \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \end{bmatrix}.$$

Tomographic problems can be classified according to the measurement data into three cases:

- full angle full data tomography,
- limited angle tomography,
- sparse data tomography.

In the first case we have a sufficient amount of measurements from all around the object in order to compute the reconstruction stably. In fact, the full angle full data tomography is not very ill-posed problem.

Instead, in the limited angle tomography the projection images are available only from a restricted angle of view and the reconstruction process is very sensitive to measurement error. In addition, due to the incompleteness of the measurement data, it is not possible to reconstruct the object perfectly, even though there were no errors in the data. Limited angle tomography occurs in technical applications, for instance, in dental imaging.

In the case of sparse data tomography we have only a few projection images but possibly from any direction. This case leads to an extremely ill-posed inverse problem and therefore some kind of prior knowledge of the solution is necessary in order to reconstruct the object properly. In medical imaging it is reasonable to minimize the patient's radiation dose, which makes the sparse data tomography practically interesting problem.

2.4 Numerical differentiation

Consider continuous functions on $[0, 1]$. Now the direct problem is: given a continuous function $x(t),\ t \in [0, 1]$, find it's antiderivative $y(t),\ t \in [0, 1]$, that satisfies

$$y(t) = \int_0^t x(s)ds,\ t \in [0, 1], \quad \text{and} \quad y(0) = 0. \tag{2.14}$$

The corresponding inverse problem is "given a continuously differentiable function $y(t),\ t \in [0, 1],\ y(0) = 0$, find its derivative $x(t),\ t \in [0, 1]$". In other words the

task is to solve (2.14) for x. Our aim is now to write this problem in the standard form (2.1).

Assume the function y is given as a measurement $m \in \mathbb{R}^k$, whose ith component m_i corresponds to the value $y(t_i)$, where $t_i = \frac{i}{k}$. With this discretization the integral in (2.14) can be approximated simply as

$$\int_0^{t_i} x(s)ds \approx \frac{1}{k}\sum_{j=1}^{i} x_j, \tag{2.15}$$

where $x_j = x(t_j)$. (Note that there are more sophisticated methods to compute integrals numerically, e.g. Simpson's rule, but we use formula (2.15) here for simplicity.) Using this approximation we get

$$m_i = \frac{1}{k}\sum_{j=1}^{i} x_j.$$

Thus the model between the measurement m and the unknown derivative x can be written in matrix form $m = Ax + \varepsilon$, where

$$A = \begin{bmatrix} \frac{1}{k} & 0 & 0 & 0 & \dots & 0 \\ \frac{1}{k} & \frac{1}{k} & 0 & 0 & \dots & 0 \\ \frac{1}{k} & \frac{1}{k} & \frac{1}{k} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{k} & \frac{1}{k} & \frac{1}{k} & \frac{1}{k} & \dots & \frac{1}{k} \end{bmatrix}. \tag{2.16}$$

2.5 Laplace transform and its inverse

Let $f : [0, \infty) \to \mathbb{R}$. The Laplace transform F of f is defined by

$$F(s) = \int_0^{\infty} e^{-st} f(t)dt, \quad s \in \mathbb{C}, \tag{2.17}$$

provided that the integral converges. The direct problem is to find the Laplace transform for a given function f according to (2.17). The opposite to this, i.e. the inverse problem, is: given a Laplace transform F, find the corresponding function f.

Assume we know the values of F in real points $0 < s_1 < s_2 < \ldots < s_n < \infty$. Then we may approximate the integral in (2.17) for example with the trapezoidal rule as

$$\begin{aligned} \int_0^{\infty} e^{-st} f(t)dt \;\approx\; \frac{t_k}{k}\Bigg(&\frac{1}{2}e^{-st_1}f(t_1) + e^{-st_2}f(t_2) + e^{-st_3}f(t_3) + \dots \\ &+ e^{-st_{k-1}}f(t_{k-1}) + \frac{1}{2}e^{-st_k}f(t_k)\Bigg), \end{aligned} \tag{2.18}$$

where vector $t = [t_1\ t_2\ \ldots\ t_k]^T \in \mathbb{R}^k$, $0 \leq t_1 < t_2 < \ldots < t_k$, contains the points, in which the unknown function f will be evaluated. By denoting $x_l = f(t_l),\ l = 1, \ldots, k$, and $m_j = F(s_j),\ j = 1, \ldots, n$, and using (2.18), we get a linear model of the form $m = Ax + \varepsilon$ with

$$A = \frac{t_k}{k} \begin{bmatrix} \frac{1}{2}e^{-s_1 t_1} & e^{-s_1 t_2} & e^{-s_1 t_3} & \ldots & e^{-s_1 t_{k-1}} & \frac{1}{2}e^{-s_1 t_k} \\ \frac{1}{2}e^{-s_2 t_1} & e^{-s_2 t_2} & e^{-s_2 t_3} & \ldots & e^{-s_2 t_{k-1}} & \frac{1}{2}e^{-s_2 t_k} \\ \vdots & & & & & \vdots \\ \frac{1}{2}e^{-s_n t_1} & e^{-s_n t_2} & e^{-s_n t_3} & \ldots & e^{-s_n t_{k-1}} & \frac{1}{2}e^{-s_n t_k} \end{bmatrix}.$$

2.6 Heat equation

Consider the temperature distribution in a one-dimensional wire with a length of π. The heat equation for the temperature distribution $u(s,t),\ s \in [0,\pi],\ t \in [0,\infty)$ is a partial differential equation of the form

$$\frac{\partial u(s,t)}{\partial t} = C\frac{\partial^2 u(s,t)}{\partial s^2}, \tag{2.19}$$

where $C \in \mathbb{R}$ is a constant called thermal diffusivity. For simplicity, we take $C = 1$. We assume that the temperature in the ends of the wire is held at zero, that is

$$u(0,t) = 0 = u(1,t), \quad \forall\, t \in [0,\infty) \tag{2.20}$$

and also, that the initial temperature distribution is

$$u(s,0) = f(s), \quad s \in [0,\pi]. \tag{2.21}$$

With this model the easier (direct) problem would be to find the temperature distribution $u(s,T)$ at certain time $T > 0$, when we know the initial temperature distribution $f(s)$. However, much more difficult problem is the inverse problem: given a temperature distribution $u(s,T)$, find the initial temperature distribution $f(s)$.

The partial differential equation (2.19) with boundary conditions (2.20) and initial conditions (2.21) can be solved for example by separation of variables, and the solution is ($C = 1$)

$$u(s,t) = \frac{2}{\pi}\int_0^\pi k(s,y)f(y)dy, \quad s \in [0,\pi] \tag{2.22}$$

where

$$k(s,y) = \sum_{i=1}^{\infty} e^{-i^2 t}\sin(is)\sin(iy). \tag{2.23}$$

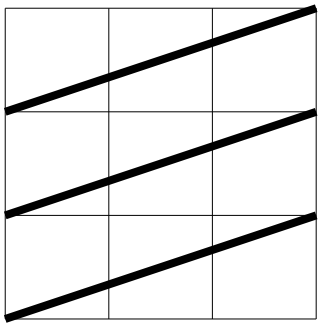
Divide the wire into n points $s_1, s_2, \ldots, s_n$, and assume the temperature distribution measured in these points at time T is given by vector $m \in \mathbb{R}^n$. Furthermore,

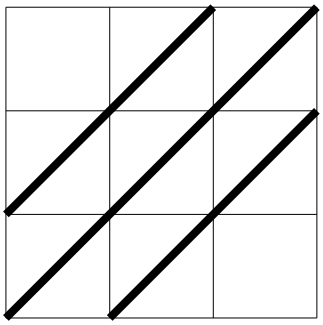
denote $x_i = f(s_i)$. Then computing the integral in (2.22) with trapezoidal rule yields a linear model of the form $m = Ax + \varepsilon$, where

$$A = \frac{2}{n}\begin{bmatrix} \frac{1}{2}k(s_1,s_1) & k(s_1,s_2) & k(s_1,s_3) & \dots & k(s_1,s_{n-1}) & \frac{1}{2}k(s_1,s_n) \\ \frac{1}{2}k(s_2,s_1) & k(s_2,s_2) & k(s_2,s_3) & \dots & k(s_2,s_{n-1}) & \frac{1}{2}k(s_2,s_n) \\ \vdots & & & & & \\ \frac{1}{2}k(s_n,s_1) & k(s_n,s_2) & k(s_n,s_3) & \dots & k(s_n,s_{n-1}) & \frac{1}{2}k(s_n,s_n) \end{bmatrix}. \quad (2.24)$$

2.7 Exercises

1. Let $x \in \mathbb{R}^8$ be a signal and $p = [p_{-1} \quad p_0 \quad p_1]^T$ a point spread function. Write down the 8×8 matrix A modeling the one-dimensional convolution $p * x$. Use the periodic boundary condition $x_j = x_{j+8}$ for all $j \in \mathbb{Z}$.





2. In the above figure, thin lines depict pixels and thick lines X-rays. Give a numbering to the nine pixels ($x \in \mathbb{R}^9$) and to the six X-rays ($m \in \mathbb{R}^6$), and construct the measurement matrix A. The length of the side of a pixel is one.

3. Let $X \in \mathbb{R}^{\nu^2}$ be an image of size $\nu \times \nu$ and $p \in \mathbb{R}^{q \times q}$ a point spread function. Denote by

$$A : \mathbb{R}^n \to \mathbb{R}^n$$

the matrix representing the linear operator $X \mapsto p * X$ (with zero extension of the image outside the boundaries) in the standard coordinates of $\mathbb{R}^n$. Here $n = \nu^2$. Construct matrix A in the case $\nu = 5$ and $q = 3$.

4. Show that the matrix A in the previous exercise is symmetric ($A^T = A$) for any $\nu > 1$ and $q > 1$.

Chapter 3

Ill-posedness and inverse crimes

We move from the direct problem to the inverse problem regarding the measurement model $m = Ax + \varepsilon$. The direct problem is to determine m when x is known, and the inverse problem is

$$\text{Given } m, \text{ reconstruct } x. \tag{3.1}$$

In this section we explore various problems related to the seemingly simple task (3.1).

3.1 Naive reconstruction attempts

Let's try to reconstruct a one-dimensional signal by deconvolution. Choose $k = n = 64$ and compute both ideal measurement $y = Ax$ and noisy measurement $m = Ax + \varepsilon$ as in Figure 2.4. The simplest thing to try is to recover x by applying the inverse of matrix A to the measurement: $x = A^{-1}m$. As we see in the rightmost plot of Figure 3.1, the result looks very bad: the measurement noise seems to get amplified in the reconstruction process. To get a more quantitative idea of the badness of the reconstruction, let us introduce a relative error formula for comparing the reconstruction to the original signal:

$$\frac{\|\text{original} - \text{reconstruction}\|}{\|\text{original}\|} \cdot 100\%. \tag{3.2}$$

We calculate 40% relative error for the reconstruction from data with noise level $\sigma = 0.02$. Let us try the ideal non-noisy case $y = Ax$. The middle plot of Figure 3.1 shows the vector $x = A^{-1}y$ that recovers the original x perfectly.

What is going on? Why is there such a huge difference between the two cases that differ only by a small additive random error component?

Perhaps we are modeling the continuum measurement too coarsely. However, this seems not to be the case since repeating the above test with $k = 128$ produces

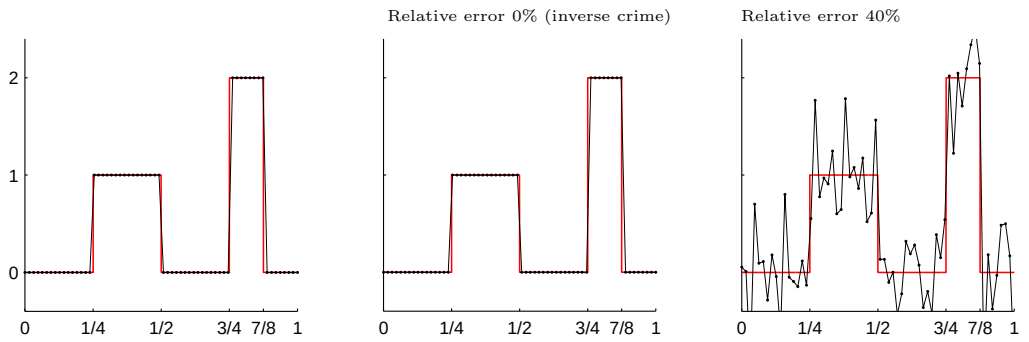


Figure 3.1: Left: the original signal $x \in \mathbb{R}^{64}$. Middle: reconstruction from ideal data $y = Ax$ by formula $x = A^{-1}y$. The surprisingly good quality is due to inverse crime. Right: reconstruction from noisy data $m = Ax + \varepsilon$ by formula $x = A^{-1}m$. The noise level is $\sigma = 0.02$. The percentages on the plots are relative errors as defined in (3.2).

no results at all: when Matlab tries to compute either $x = A^{-1}y$ or $x = A^{-1}m$, only vectors full of `NaN`s, or not-a-numbers, appear. There is clearly something strange going on.

Actually we are dealing with two separate problems above. The perfect reconstruction result in the middle plot of Figure 3.1 is just an illusion resulting from the so-called *inverse crime* where one simulates the data and implements the reconstruction using the same computational grid. The amplification of noise in the reconstruction shown in the rightmost plot of Figure 3.1 is a more profound problem coming from the ill-posedness of the continuum deconvolution task. Let us next discuss both of these problems in detail.

3.2 Inverse crime

Inverse crimes, or too-good-to-be-true reconstructions, may appear in situations when data for inverse problems is simulated using the same computational grid that is used in the inversion process. Note carefully that inverse crimes are not possible in situations where actual real-world measured data is used; it is only a problem of computational simulation studies.

Let us revisit the example of Section 3.1. Now we create the measurement data on a grid with $k = 487$ points. (The reason for the apparently strange choice 487 is simply that the fine grid used for simulation of measurement is not a multiple of the coarse grid where the inversion takes place.) Then we interpolate that measurement to the grid with $k = 64$ points using linear interpolation, and we add a noise component $\varepsilon \in \mathbb{R}^{64}$ with noise level $\sigma = 0.02$. This way the simulation of the measurement data can be thought of fine modelling of the continuum problem, and the interpolation procedure with the addition of noise models a practical measurement instrument.

Now we see a different story: compare Figures 3.1 and 3.2. The reconstruction

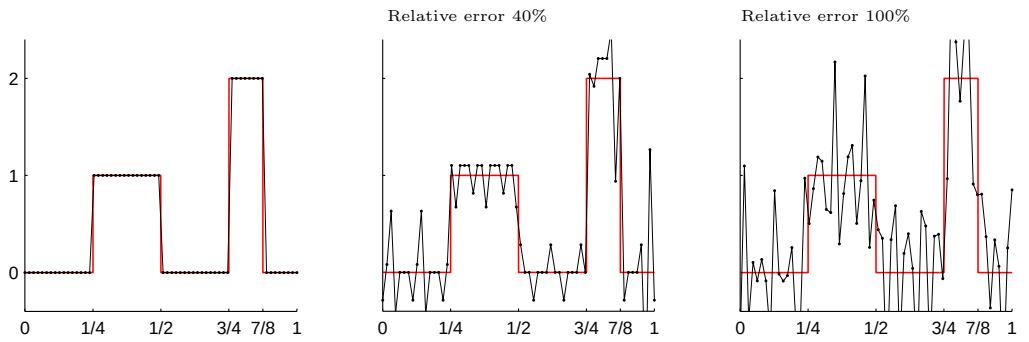


Figure 3.2: Left: the original signal $x \in \mathbb{R}^{64}$. Middle: reconstruction from ideal data $y = Ax$ by formula $x = A^{-1}y$. The inverse crime is avoided by simulating the data on a finer grid. Right: reconstruction from noisy data $m = Ax + \varepsilon$ by formula $x = A^{-1}m$. The noise level is $\sigma = 0.02$. The percentages on the plots are relative errors as defined in (3.2). Compare these plots to those in Figure 3.1.

from ideal data is now quite erroneous as well, and the relative error percentage in the reconstruction from noisy data jumped up from 40% to a whopping 100%. The conclusion is that we have successfully avoided the inverse crime but are on the other hand faced with huge instability issues. Let us next attack them.

3.3 Singular value analysis of ill-posedness

Let A be a $k \times n$ matrix and consider the measurement $m = Ax + \varepsilon$. The inverse problem "given m, find x" seems to be formally solvable by approximating x with the vector

$$A^{-1}m.$$

However, as we saw in sections 3.1 and 3.2, there are problems with this simple approach. Let us discuss such problems in detail.

Recall the definitions of the following linear subspaces related to the matrix A:

$$\begin{aligned} \mathrm{Ker}(A) &= \{x \in \mathbb{R}^n \,:\, Ax = 0\}, \\ \mathrm{Range}(A) &= \{y \in \mathbb{R}^k \,:\, \text{there exists } x \in \mathbb{R}^n \text{ such that } Ax = y\}, \\ \mathrm{Coker}(A) &= (\mathrm{Range}(A))^{\perp} \subset \mathbb{R}^k. \end{aligned}$$

See Figure 3.3 for a diagram illustrating these concepts.

Now if $k > n$ then $\dim(\mathrm{Range}(A)) < k$ and we can choose a nonzero $y_0 \in \mathrm{Coker}(A)$ as shown in Figure 3.3. Even in the case $\varepsilon = 0$ we have problems since there does not exist any $x \in \mathbb{R}^n$ satisfying $Ax = y_0$, and consequently the existence condition (1.1) fails since the output $A^{-1}y_0$ is not defined for the input y_0. In case of nonzero random noise the situation is even worse since even though $Ax \in \mathrm{Range}(A)$, it might happen that $Ax + \varepsilon \notin \mathrm{Range}(A)$.

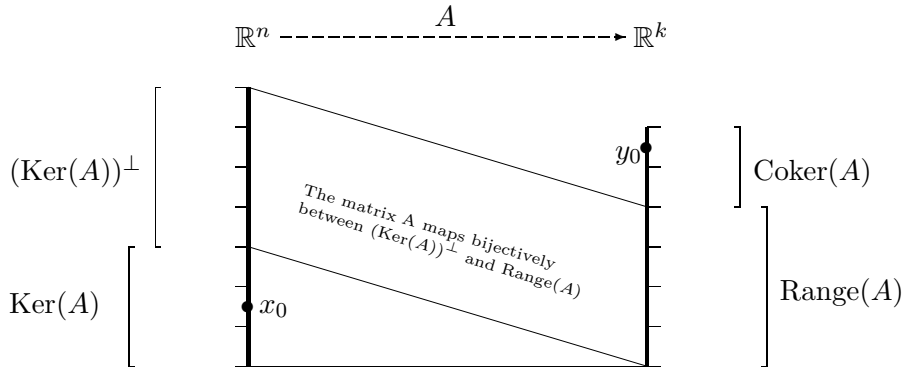


Figure 3.3: This diagram illustrates various linear subspaces related to a matrix mapping $\mathbb{R}^n$ to $\mathbb{R}^k$. The two thick vertical lines represent the linear spaces $\mathbb{R}^n$ and $\mathbb{R}^k$; in this schematic picture we have $n = 7$ and $k = 6$. Furthermore, $\dim(\mathrm{Ker}(A)) = 3$ and $\dim(\mathrm{Range}(A)) = 4$ and $\dim(\mathrm{Coker}(A)) = 2$. Note that the 4-dimensional orthogonal complement of $\mathrm{Ker}(A)$ in R^n is mapped in a bijective manner to $\mathrm{Range}(A)$. The points $x_0 \in \mathrm{Ker}(A)$ and $y_0 \in \mathrm{Coker}(A)$ are used in the text.

If $k < n$ then $\dim(\mathrm{Ker}(A)) > 0$ and we can choose a nonzero $x_0 \in \mathrm{Ker}(A)$ as shown in Figure 3.3. Then even in the case $\varepsilon = 0$ we have a problem of defining $A^{-1}m$ uniquely since both $A^{-1}m$ and $A^{-1}m + x_0$ satisfy $A(A^{-1}m) = m = A(A^{-1}m + x_0)$. Thus the uniqueness condition (1.2) fails unless we specify an explicit way of dealing with the null-space of A.

The above problems with existence and uniqueness are quite clear since they are related to integer-valued dimensions. In contrast, ill-posedness related to the continuity condition (1.3) is more tricky in our finite-dimensional context. Consider the case $n = k$ so A is a square matrix, and assume that A is invertible. In that case we can write

$$A^{-1}m = A^{-1}(Ax + \varepsilon) = x + A^{-1}\varepsilon, \tag{3.3}$$

where the error $A^{-1}\varepsilon$ can be bounded by

$$\|A^{-1}\varepsilon\| \leq \|A^{-1}\|\|\varepsilon\|.$$

Now if $\|\varepsilon\|$ is small and $\|A^{-1}\|$ has reasonable size then the error $A^{-1}\varepsilon$ is small. However, if $\|A^{-1}\|$ is large then the error $A^{-1}\varepsilon$ can be huge even when ε is small. This is the kind of amplification of noise we see in Figure 3.2.

Note that if $\varepsilon = 0$ in (3.3) then we do have $A^{-1}m = x$ even if $\|A^{-1}\|$ is large. However, in practice the measurement data always has some noise, and even computer simulated data is corrupted with round-off errors. Those inevitable perturbations prevent using $A^{-1}m$ as a reconstruction method for an ill-posed problem.

Let us now discuss a tool that allows explicit analysis of possible difficulties related to Hadamard's conditions, namely *singular value decomposition*. We know

from matrix algebra that any matrix $A \in \mathbb{R}^{k\times n}$ can be written in the form

$$A = UDV^T, \tag{3.4}$$

where $U \in \mathbb{R}^{k\times k}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, that is,

$$U^TU = UU^T = I, \quad V^TV = VV^T = I,$$

and $D \in \mathbb{R}^{k\times n}$ is a diagonal matrix. In the case $k = n$ the matrix D is square-shaped: $D = \mathrm{diag}(d_1, \ldots, d_k)$. If $k > n$ then

$$D = \begin{bmatrix} \mathrm{diag}(d_1, \ldots, d_n) \\ \mathbf{0}_{(k-n)\times n} \end{bmatrix} = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & \cdots & d_n \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}, \tag{3.5}$$

and in the case $k < n$ the matrix D takes the form

$$\begin{aligned} D &= [\mathrm{diag}(d_1, \ldots, d_k), \mathbf{0}_{k\times(n-k)}] \\ &= \begin{bmatrix} d_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & d_2 & & \vdots & \vdots & & \vdots \\ \vdots & & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & \cdots & d_k & 0 & \cdots & 0 \end{bmatrix}. \end{aligned} \tag{3.6}$$

The diagonal elements d_j are nonnegative and in decreasing order:

$$d_1 \geq d_2 \geq \ldots \geq d_{\min(k,n)} \geq 0. \tag{3.7}$$

Note that some or all of the d_j can be equal to zero.

The right side of (3.4) is called the singular value decomposition (SVD) of matrix A, and the diagonal elements d_j are the singular values of A.

Failure of Hadamard's existence and uniqueness conditions can now be read off the matrix D: if D has a column of zeroes then $\dim(\mathrm{Ker}(A)) > 0$ and uniqueness fails, and if D has a row of zeroes then $\dim(\mathrm{Coker}(A)) > 0$ and the existence fails. Note that if $d_{\min(k,n)} = 0$ then both conditions fail.

Ill-posedness related to the continuity condition (1.3) is related to sizes of singular values. Consider the case $n = k$ and $d_n > 0$, when we do not have the above problems with existence or uniqueness. It seems that nothing is wrong since we can invert the matrix A as

$$A^{-1} = VD^{-1}U^T, \qquad D^{-1} = \mathrm{diag}(\frac{1}{d_1}, \ldots, \frac{1}{d_k}),$$

and define $\mathcal{R}(y) = A^{-1}y$ for any $y \in \mathbb{R}^k$. The problem comes from the *condition number*

$$\mathrm{Cond}(A) := \frac{d_1}{d_k} \tag{3.8}$$

being large. Namely, if d_1 is several orders of magnitude greater than d_k then numerical inversion of A becomes difficult since the diagonal inverse matrix D^{-1} contains floating point numbers of hugely different size. This in turn leads to uncontrollable amplification of truncation errors.

Strictly mathematically speaking, though, A is an invertible matrix even in the case of large condition number, and one may ask how to define ill-posedness in the sense of condition (1.3) failing in a finite-dimensional measurement model? (This question has actually been asked by a student every time I have lectured this course.) The answer is related to the continuum problem approximated by the matrix model. Suppose that we model the continuum measurement by a sequence of matrices A_k having size $k \times k$ for $k = k_0, k_0+1, k_0+2, \ldots$ so that the approximation becomes better when k grows. Then we say that condition (1.3) fails if

$$\lim_{k\to\infty} \mathrm{Cond}(A_k) = \infty.$$

So the ill-posedness cannot be rigorously detected from one approximation matrix A_k but only from the sequence $\{A_k\}_{k=k_0}^{\infty}$.

We give a concrete infinite-dimensional example in Appendix A. That example is a simple model of electrocardiography. Since it is based on some results in the field of partial differential equations, we think that it is outside the main scope of this material so we put it in an appendix.

In practice we can plot the singular values on a logarithmic scale and detect ill-posedness with incomplete mathematical rigor but with sufficient computational relevance. Let us take a look at the singular values of the measurement matrices corresponding to the one-dimensional convolution measurement of Section 2.2.1. See Figure 3.4.

3.4 Exercises

1. Consider the inverse problem related to the measurement $y = Ax$ in the cases

   (a) $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $y = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, (b) $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 13 & 31 \end{bmatrix}$, $y = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

   Which of Hadamard's conditions is violated, if any?

2. Assume that the matrix $U : \mathbb{R}^n \to \mathbb{R}^n$ is orthogonal: $UU^T = I = U^TU$. Show that $\|U^Ty\| = \|y\|$ for any $y \in \mathbb{R}^n$.

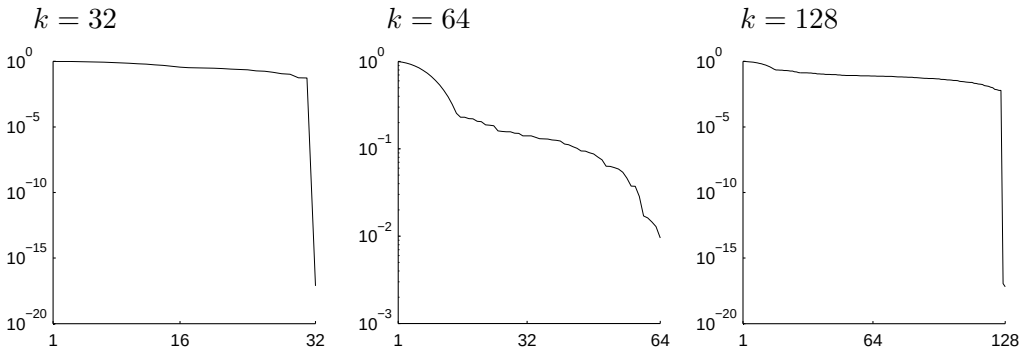
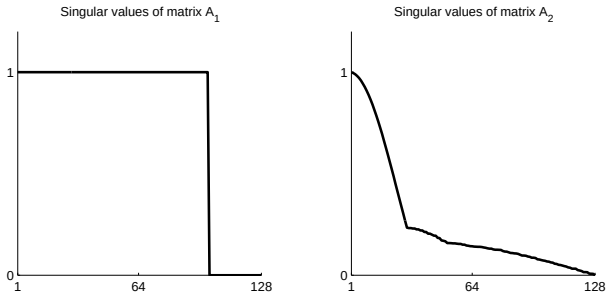


Figure 3.4: Singular values of measurement matrices corresponding to one-dimensional convolution. The last (smallest) singular value is several orders of magnitude smaller than the first (largest) singular value, so the condition number of these matrices is big.

3. Let $A, U : \mathbb{R}^n \to \mathbb{R}^n$ be matrices and assume that U is orthogonal. Show that $\|UA\| = \|A\|$.

4. If matrices A_1 and A_2 have the singular values shown below, what conditions of Hadamard do they violate, if any?



5. Download the Matlab routines `ex_conv1Ddata_comp.m` and `ex_conv1D_naive.m` from the course web page to your working directory. Create a new folder called `data`. Modify the two routines to study the following problems.

   Consider a continuous point spread function $p : \mathbb{R} \to \mathbb{R}$ given by

$$p(s) = \begin{cases} 1 - s & \text{for } 0 \le s \le 1, \\ 1 + s & \text{for } -1 \le s < 0, \\ 0 & \text{otherwise.} \end{cases}$$

   Choose any function $x : \mathbb{R} \to \mathbb{R}$ satisfying $x(s) = 0$ for $s < 0$ and $s > 10$.

   (a) Create data with `ex_conv1Ddata_comp.m` and try to reconstruct your signal with `ex_conv1D_naive.m` using the same discretization in both files. Do you see unrealistically good reconstructions? What happens when the discretization is refined?

(b) Repeat (a) using finer discretization for generating data. Can you avoid inverse crime?

Chapter 4

Regularization methods

We saw in Chapter 3 that recovering x from noisy measurement $m = Ax + \varepsilon$ is difficult for various reasons. In particular, the simple idea of approximating x by $A^{-1}m$ may fail miserably.

We would like to define a reconstruction function $\mathcal{R} : \mathbb{R}^k \to \mathbb{R}^n$ so that the problem of determining $\mathcal{R}(m)$ for a given m would be a well-posed problem in the sense of Hadamard. First of all, the value $\mathcal{R}(y)$ should be well-defined for every $y \in \mathbb{R}^k$; this is the existence requirement (1.1). Furthermore, the function $\mathcal{R} : \mathbb{R}^k \to \mathbb{R}^n$ should be single-valued and continuous, as stated in (1.2) and (1.3), respectively.

Let us adapt the notions of *regularization strategy* and *admissible choice of regularization parameter* from the book [15] by Andreas Kirsch to our finite-dimensional setting. We need to assume that $\mathrm{Ker}(A) = \{0\}$; however, this is not a serious lack of generality since we can always consider the restriction of A to $(\mathrm{Ker}(A))^\perp$ by working in the linear space of equivalence classes $[x + \mathrm{Ker}(A)]$.

Definition 4.1. *Consider the measurement $m = Ax + \varepsilon$ with A a $k \times n$ matrix with $Ker(A) = \{0\}$. A family of linear maps $\mathcal{R}_\delta : \mathbb{R}^k \to \mathbb{R}^n$ parameterized by $0 < \delta < \infty$ is called a regularization strategy if*

$$\lim_{\delta \to 0} \mathcal{R}_\delta A x = x \tag{4.1}$$

for every $x \in \mathbb{R}^n$. Further, a choice of regularization parameter $\delta = \delta(\kappa)$ as function of noise level $\kappa > 0$ is called admissible if

$$\delta(\kappa) \to 0 \text{ as } \kappa \to 0, \text{ and} \tag{4.2}$$

$$\sup_m \left\{ \|\mathcal{R}_{\delta(\kappa)} m - x\| \,:\, \|Ax - m\| \le \kappa \right\} \to 0 \text{ as } \kappa \to 0 \text{ for every } x \in \mathbb{R}^n \tag{4.3}$$

In this chapter we introduce several classes of regularization strategies.

4.1 Truncated singular value decomposition

The problems with Hadamard's existence and uniqueness conditions (1.1) and (1.2) can be dealt with using the Moore-Penrose *pseudoinverse*. Let us look

at that first before tackling problems with the continuity condition (1.3) using truncated SVD.

4.1.1 Minimum norm solution

Assume given a $k \times n$ matrix A. Using SVD write A in the form $A = UDV^T$ as explained in Section 3.3. Let r be the largest index for which the corresponding singular value is positive:

$$r = \max\{j \mid 1 \leq j \leq \min(k,n),\ d_j > 0\}. \tag{4.4}$$

Remember that the singular values are ordered from largest to smallest as shown in (3.7). As the definition of index r is essential in the sequel, let us be extra-specific:

$$d_1 > 0, \quad d_2 > 0, \quad \cdots \quad d_r > 0, \quad d_{r+1} = 0, \quad \cdots \quad d_{\min(k,n)} = 0.$$

Of course, it is also possible that all singular values are zero. Then r is not defined and A is the zero matrix.

Let us define the minimum norm solution of matrix equation $Ax = y$, where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^k$ and A has size $k \times n$. First of all, a vector $F(y) \in \mathbb{R}^n$ is called a *least squares solution* of equation $Ax = y$ if

$$\|AF(y) - y\| = \inf_{z \in \mathbb{R}^n} \|Az - y\|. \tag{4.5}$$

Furthermore, $F(y)$ is called the *minimum norm solution* if

$$\|F(y)\| = \inf\{\|z\| \,:\, z \text{ is a least squares solution of } Ax = y\}. \tag{4.6}$$

The next result gives a method to determine the minimum norm solution.

Theorem 4.1. *Let A be a $k \times n$ matrix. The minimum norm solution of equation $Ax = y$ is given by*

$$A^+ y = V D^+ U^T y,$$

where

$$D^+ = \begin{bmatrix} 1/d_1 & 0 & \cdots & 0 & & \cdots & 0 \\ 0 & 1/d_2 & & & & & \vdots \\ \vdots & & \ddots & & & & \\ & & & 1/d_r & & & \\ & & & & 0 & & \\ \vdots & & & & & \ddots & \vdots \\ 0 & \cdots & & & & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n \times k}.$$

Proof. Denote $V = [V_1 \quad V_2 \quad \cdots \quad V_n]$ and note that the column vectors $V_1, \ldots, V_n$ form an orthogonal basis for $\mathbb{R}^n$. We write $x \in \mathbb{R}^n$ as linear combination $x = \sum_{j=1}^n \alpha_j V_j = V\alpha$, and our goal is to find such coefficients $\alpha_1, \ldots, \alpha_n$ that x becomes a minimum norm solution.

Set $y' = U^T y \in \mathbb{R}^k$ and compute

$$\begin{aligned} \|Ax - y\|^2 &= \|UDV^T V\alpha - Uy'\|^2 \\ &= \|D\alpha - y'\|^2 \\ &= \sum_{j=1}^{r} (d_j\alpha_j - y'_j)^2 + \sum_{j=r+1}^{k} (y'_j)^2, \end{aligned} \tag{4.7}$$

where we used the orthogonality of U (namely, $\|Uz\| = \|z\|$ for any vector $z \in \mathbb{R}^k$). Now since d_j and y'_j are given and fixed, the expression (4.7) attains its minimum when $\alpha_j = y'_j/d_j$ for $j = 1, \ldots, r$. So any x of the form

$$x = V \begin{bmatrix} d_1^{-1} y'_1 \\ \vdots \\ d_r^{-1} y'_r \\ \alpha_{r+1} \\ \vdots \\ \alpha_n \end{bmatrix}$$

is a least squares solution. The smallest norm $\|x\|$ is clearly given by the choice $\alpha_j = 0$ for $r < j \leq n$, so the minimum norm solution is uniquely determined by the formula $\alpha = D^+ y'$. □

The matrix A^+ is called the *pseudoinverse*, or the *Moore-Penrose inverse* of A.

How does the pseudoinverse take care of Hadamard's existence and uniqueness conditions (1.1) and (1.2)? First of all, if $\mathrm{Coker}(A)$ is nontrivial, then any vector $y \in \mathbb{R}^k$ can be written as the sum $y = y_A + (y_A)^\perp$, where $y_A \in \mathrm{Range}(A)$ and $(y_A)^\perp \in \mathrm{Coker}(A)$ and $y_A \cdot (y_A)^\perp = 0$. Then A^+ simply maps $(y_A)^\perp$ to zero. Second, if $\mathrm{Ker}(A)$ is nontrivial, then we need to choose the reconstructed vector from a whole linear subspace of candidates. Using A^+ chooses the candidate with smallest norm.

4.1.2 Regularization by truncation

It remains to discuss Hadamard's continuity condition (1.3). Recall from Section 3 that we may run into problems if d_r is much smaller than d_1. In that case even the use of the pseudoinverse $F(m) = A^+ m = VD^+U^T m$ because the diagonal element d_r^{-1} appearing in D^+ is much larger than d_1^{-1}, resulting in numerical instability. We can overcome this by using truncated SVD. For any $\delta > 0$ define

$$r_\delta = \min\Big\{r,\ \max\{j \mid 1 \leq j \leq \min(k,n),\ d_j > \delta\}\Big\}. \tag{4.8}$$

Define then

$$D_\delta^+ = \begin{bmatrix} 1/d_1 & 0 & \cdots & 0 & & \cdots & 0 \\ 0 & 1/d_2 & & & & & \vdots \\ \vdots & & \ddots & & & & \\ & & & 1/d_{r_\delta} & & & \\ & & & & 0 & & \\ \vdots & & & & & \ddots & \vdots \\ 0 & \cdots & & & & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n\times k}$$

and define a reconstruction function F_δ by the formula

$$F_\delta(m) = VD_\delta^+U^Tm. \tag{4.9}$$

Then all Hadamard's conditions hold: $F_\delta : \mathbb{R}^k \to \mathbb{R}^n$ is a well-defined, single-valued linear mapping with norm

$$\|F_\delta\| = \|VD_\delta^+U^T\| \le \|V\|\|D_\delta^+\|\|U^T\| = \|D_\delta^+\| = d_{r_\delta}^{-1},$$

implying continuity. Let us be specific here. Of course the linear mapping is continuous in the mathematical sense since $\|F_\delta\| = d_{r_\delta}^{-1} < \infty$. However, equation (3.3) now takes the form

$$F_\delta(m) = VD_\delta^+U^T(Ax+\varepsilon) = VD_\delta^+DV^Tx + VD_\delta^+U^T\varepsilon, \tag{4.10}$$

where $VD_\delta^+DV^Tx$ is an approximation to x and the error term can be estimated as follows:

$$\|VD_\delta^+U^T\varepsilon\| \le \|VD_\delta^+U^T\|\|\varepsilon\| = \|D_\delta^+\|\|\varepsilon\| = d_{r_\delta}^{-1}\|\varepsilon\|. \tag{4.11}$$

By the ordering (3.7) of singular values we have

$$d_1^{-1} \le d_2^{-1} \le \cdots \le d_r^{-1},$$

and by (4.11) the noise gets amplified in the inversion less and less if we choose smaller r_δ (or, equivalently, greater δ).

We see from definition (4.9) and by denoting $\alpha := D_\delta^+U^Tm$ that the reconstruction is a linear combination of the columns $V_1, \ldots, V_n$ of matrix $V = [V_1 \quad V_2 \quad \cdots \quad V_n]$:

$$F_\delta(m) = V\alpha = \alpha_1V_1 + \cdots + \alpha_nV_n.$$

Thus the columns $V_1, \ldots, V_n$, called *singular vectors*, are the building blocks of any reconstruction using truncated SVD.

Next we show that the truncated SVD method is a regularization strategy with admissible choice $\delta(\kappa) = \kappa$ of regularization parameter in the sense of Definition 4.1.

Theorem 4.2. *Let A be a $k \times n$ matrix satisfying the assumption $Ker(A) = \{0\}$, and let $A = UDV^T$ be the singular value decomposition.*

Then the family $F_\delta : \mathbb{R}^k \to \mathbb{R}^n$ defined by the truncated SVD *method in (4.9) is a regularization strategy in the sense of (4.1). Furthermore, the choice $\delta(\kappa) = \kappa$ satisfies (4.2) and (4.3).*

Proof. Note that the assumption $\mathrm{Ker}(A) = \{0\}$ implies $n \le k$ and $d_n > 0$. In particular, $r = n$ in (4.4).

Since the map $F_\delta : \mathbb{R}^k \to \mathbb{R}^n$ is defined by matrix multiplication, it is linear. To prove (4.1), note that by (4.8) we have $r_\delta \to n$ as $\delta \to 0$ because $d_n > 0$ and $r = n$. It follows that $\lim_{\delta\to 0} D_\delta^+ = \mathrm{diag}_{n\times k}\{d_1^{-1}, \dots, d_n^{-1}\}$ with n rows and k columns. Take any $x \in \mathbb{R}^n$ and compute using SVD

$$\lim_{\delta\to 0} F_\delta A x = \lim_{\delta\to 0} V D_\delta^+ D V^T x = V I_{n\times n} V^T x = x,$$

and (4.1) follows.

Condition (4.2) is trivial. It remains to prove

$$\sup_m \big\{ \|F_\kappa m - x\| \,:\, \|Ax - m\| \le \kappa \big\} \to 0 \text{ as } \kappa \to 0$$

for every $x \in \mathbb{R}^n$. As before, let us denote $m' := U^T m$ and $x = V\alpha$. Recall the definition of operator norm for a matrix B:

$$\|B\| := \sup_z \frac{\|Bz\|}{\|z\|},$$

and recall that $\|Bz\| \le \|B\|\|z\|$. Denote

$$\begin{aligned} D_n &:= \mathrm{diag}_{n\times n}\{d_1, \dots, d_n\} \\ D_n^{-1} &:= \mathrm{diag}_{n\times n}\{d_1^{-1}, \dots, d_n^{-1}\} \end{aligned}$$

and note that $\|D_n\| = d_1$ and $\|D_n^{-1}\| = d_n^{-1}$. Estimate now

$$\begin{aligned} \|F_\kappa m - x\| &= \|V D_\kappa^+ U^T m - V\alpha\| \\ &= \|D_\kappa^+ m' - \alpha\| \\ &= \|D_n^{-1} D_n D_\kappa^+ m' - D_n^{-1} D_n \alpha\| \\ &\le d_n^{-1} \|D_n D_\kappa^+ m' - D_n \alpha\|. \end{aligned}$$

In the case $\kappa < d_n$ we have $D_n D_\kappa^+ = I_{n\times n}$ and thus

$$\begin{aligned} \|F_\kappa m - x\|^2 &\le d_n^{-2} \sum_{j=1}^n (d_j\alpha_j - m')^2 \\ &\le d_n^{-2} \left(\sum_{j=1}^n (d_j\alpha_j - m')^2 + \sum_{j=n+1}^k (m')^2 \right) \\ &= d_n^{-2} \|D\alpha - m'\|^2 \\ &= d_n^{-2} \|UDV^T V\alpha - Um'\|^2 \\ &= d_n^{-2} \|Ax - m\|^2. \end{aligned}$$

Then $\|Ax - m\| \leq \kappa$ implies $\|F_\kappa m - x\| \leq d_n^{-1}\kappa$, and the proof is complete. □

Let us return to the one-dimensional deconvolution problem. In Figure 4.1 we show how the reconstruction builds up from the singular vectors one by one.

4.2 Tikhonov regularization

4.2.1 Definition via minimization

The Tikhonov regularized solution of equation $m = Ax + \varepsilon$ is the vector $T_\delta(m) \in \mathbb{R}^n$ that minimizes the expression

$$\|AT_\delta(m) - m\|^2 + \delta\|T_\delta(m)\|^2,$$

where $\delta > 0$ is called a regularization parameter. We denote

$$T_\delta(m) = \arg\min_{z\in\mathbb{R}^n} \Big\{ \|Az - m\|^2 + \delta\|z\|^2 \Big\}. \tag{4.12}$$

Tikhonov regularization can be understood as a balance between two requirements:

(i) $T_\delta(m)$ should give a small residual $AT_\delta(m) - m$,

(ii) $T_\delta(m)$ should be small in L^2 norm.

The regularization parameter $\delta > 0$ can be used to "tune" the balance.

Note that in inverse problems there are typically infinitely many choices of $T_\delta(m)$ satisfying (i), and one of the roles of (ii) is to make the solution unique.

Theorem 4.3. *Let A be a $k \times n$ matrix. The Tikhonov regularized solution for equation $m = Ax + \varepsilon$ is given by*

$$T_\delta(m) = V\mathcal{D}_\delta^+ U^T m, \tag{4.13}$$

where $A = UDV^T$ is the singular value decomposition, and

$$\mathcal{D}_\delta^+ = \operatorname{diag}\left(\frac{d_1}{d_1^2 + \delta}, \ldots, \frac{d_{\min(k,n)}}{d_{\min(k,n)}^2 + \delta} \right) \in \mathbb{R}^{n\times k}. \tag{4.14}$$

Proof. Write $T_\delta(m) \in \mathbb{R}^n$ as linear combination of column vectors of matrix V:

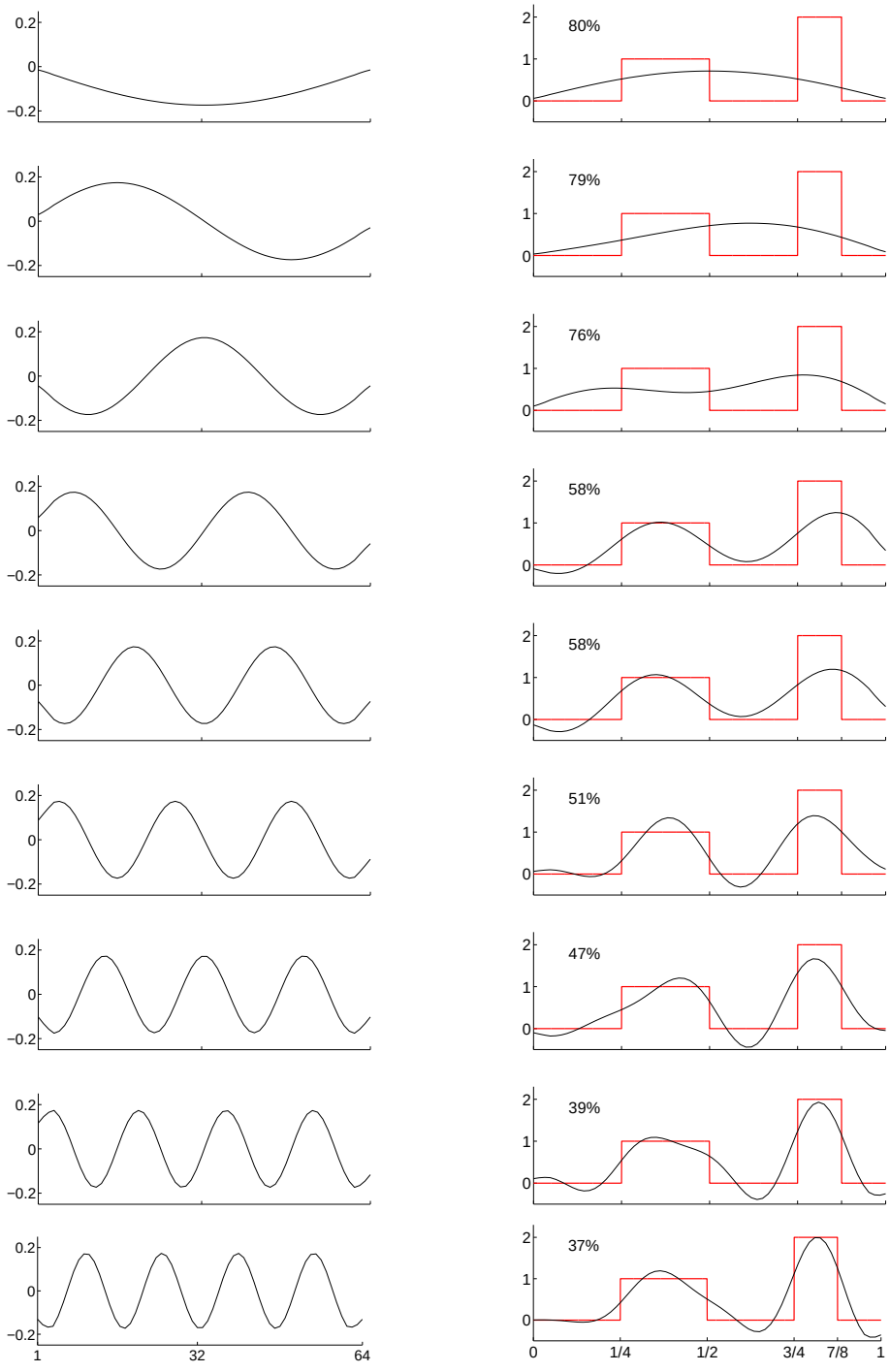


Figure 4.1: Left column: singular vectors 1–9 related to the one-dimensional convolution matrix. Right column: Reconstructions using all singular vectors up to the row number in the truncated SVD. The percentages shown are relative errors of reconstructions.

$T_\delta(m) = \sum_{j=1}^n \alpha_j V_j = V\alpha$. Set $m' = U^T m$ and compute

$$\begin{aligned}
& \|AT_\delta(m) - m\|^2 + \delta\|T_\delta(m)\|^2 \\
= \; & \|UDV^TV\alpha - UU^Tm\|^2 + \delta\|V\alpha\|^2 \\
= \; & \|D\alpha - m'\|^2 + \delta\|\alpha\|^2 \\
= \; & \sum_{j=1}^r (d_j\alpha_j - m'_j)^2 + \sum_{j=r+1}^k (m'_j)^2 + \delta\sum_{j=1}^n \alpha_j^2 \\
= \; & \sum_{j=1}^r \left(d_j^2+\delta\right)\left(\alpha_j^2 - 2\frac{d_j m'_j}{d_j^2+\delta}\alpha_j\right) + \delta\sum_{j=r+1}^n \alpha_j^2 + \sum_{j=1}^k (m'_j)^2 \qquad (4.15) \\
= \; & \sum_{j=1}^r \left(d_j^2+\delta\right)\left(\alpha_j - \frac{d_j m'_j}{d_j^2+\delta}\right)^2 + \delta\sum_{j=r+1}^n \alpha_j^2 \qquad (4.16) \\
& - \sum_{j=1}^r \frac{(d_j m'_j)^2}{d_j^2+\delta} + \sum_{j=1}^k (m'_j)^2,
\end{aligned}$$

where completing the square in the leftmost term in (4.15) yields (4.16). Our task is to choose such values for the parameters $\alpha_1, \ldots, \alpha_n$ that (4.16) attains its minimum. Clearly the correct choice is

$$\alpha_j = \begin{cases} \dfrac{d_j}{d_j^2+\delta} m'_j \;, & 1 \le j \le r, \\ 0 \quad , & r+1 \le j \le n, \end{cases}$$

or in short $\alpha = \mathcal{D}_\delta^+ m'$. □

Let us apply Tikhonov regularization to our basic test problem of one-dimensional deconvolution. In Figure 4.2 we see the Tikhonov regularized solutions corresponding to three different choices of regularization parameter, namely $\delta = 10$ and $\delta = 0.1$ and $\delta = 0.001$. Here the noise level is 1% in all three reconstructions. The result of increasing the noise level to 10% can be seen in Figure 4.3: it seems that the smaller regularization parameter δ, the more robust the Tikhonov regularized solution is with respect to measurement noise. Let us make one more test to find evidence for this statement: namely, we recompute the result shown in Figure 4.2 ten times with noise level 1%, but taking a different realization of our random error vector each time. As we see in Figure 4.4, the variance is indeed greater in reconstructions using smaller values of δ.

4.2.2 Choosing δ: the Morozov discrepancy principle

How to choose the regularization parameter $\delta > 0$ optimally? This is a difficult question and in general unsolved.

There are some methods for choosing δ, for example Morozov's discrepancy principle: If we have an estimate on the magnitude of error in the data, then

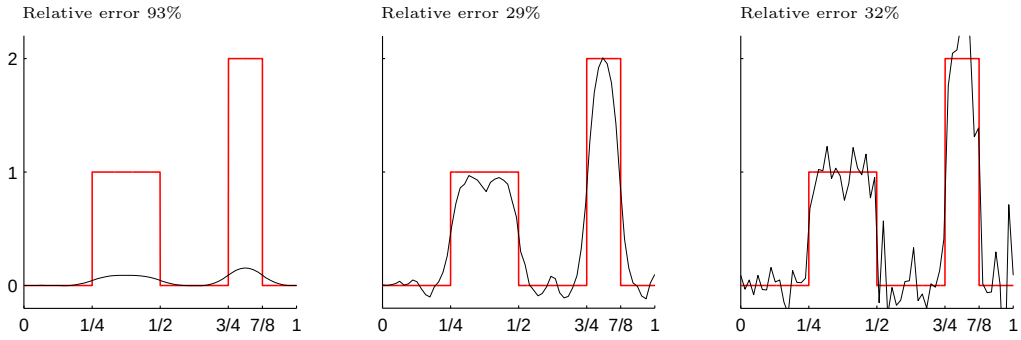


Figure 4.2: Tikhonov regularized solutions. Left: $\delta = 10$. Middle: $\delta = 0.1$. Right: $\delta = 0.001$. Here the noise level is 1% in all three reconstructions.

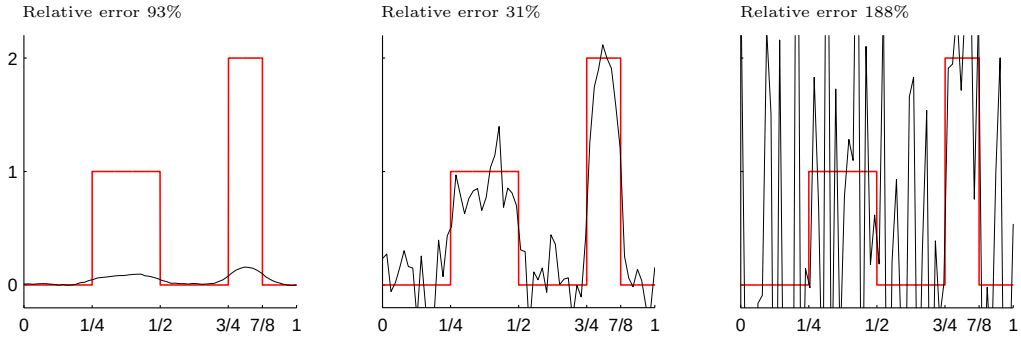


Figure 4.3: Tikhonov regularized solutions. Left: $\delta = 10$. Middle: $\delta = 0.1$. Right: $\delta = 0.001$. Here the noise level is 10% in all three reconstructions. Compare to Figure 4.2.

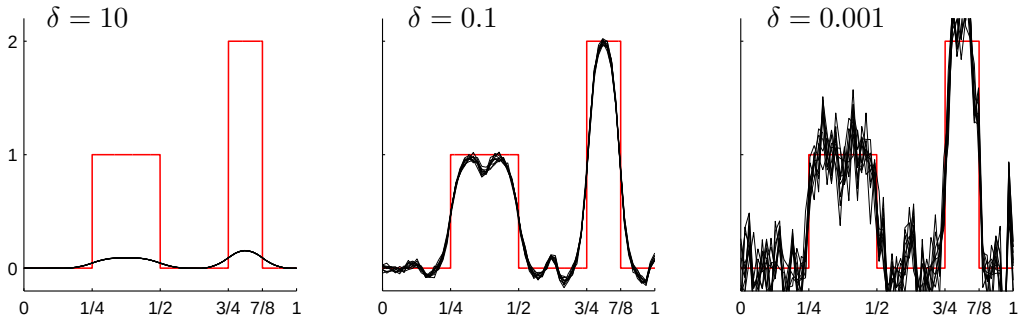


Figure 4.4: Tikhonov regularized solutions with various realizations of random noise and three different choices of the regularization parameter δ. Here the noise level is 1% in all reconstructions. Note how the noise is amplified more when δ is smaller. Compare to Figures 4.2 and 4.3.

any solution that produces a measurement with error of the same magnitude is acceptable.

For instance, assume that $m = Ax + \varepsilon$ and that we know the size of noise: $\|\varepsilon\| = \kappa > 0$. Then $T_\delta(m)$ is an acceptable reconstruction if

$$\|AT_\delta(m) - m\| \leq \kappa.$$

For example, if the elements of the noise vector $\varepsilon \in \mathbb{R}^k$ satisfy $\varepsilon_j \sim N(0, \sigma^2)$, then we can take $\kappa = \sqrt{k}\sigma$ since the expectation of the size is $\mathrm{E}(\|\varepsilon\|) = \sqrt{k}\sigma$.

The idea of Morozov discrepancy principle is to choose $\delta > 0$ such that

$$\|AT_\delta(m) - m\| = \kappa.$$

Theorem 4.4. *Morozov discrepancy principle gives a unique choice for $\delta > 0$ if and only if κ satisfies*

$$\|Pm\| \leq \kappa \leq \|m\|,$$

where P is orthogonal projection to the subspace Coker(A).

Proof. From the proof of Theorem 4.3 we find the equation

$$AT_\delta(m) = UDV^T V\mathcal{D}_\delta^+ U^T m = UD\mathcal{D}_\delta^+ m',$$

so we have

$$\begin{aligned}
\|AT_\delta(m) - m\|^2 &= \|D\mathcal{D}_\delta^+ m' - m'\|^2 \\
&= \sum_{j=1}^{\min(k,n)} \left(\frac{d_j^2}{d_j^2 + \delta} - 1 \right)^2 (m_j')^2 + \sum_{j=\min(k,n)+1}^{k} (m_j')^2 \\
&= \sum_{j=1}^{r} \left(\frac{\delta}{d_j^2 + \delta} \right)^2 (m_j')^2 + \sum_{j=r+1}^{k} (m_j')^2.
\end{aligned}$$

From this expression we see that the mapping

$$\delta \mapsto \|AT_\delta(m) - m\|^2$$

is monotonically increasing and thus, noting the formal identity $\sum_{j=r+1}^{k}(m_j')^2 = \|AT_0(m) - m\|^2$ we get

$$\sum_{j=r+1}^{k} (m_j')^2 \leq \|AT_\delta(m) - m\|^2 \leq \lim_{\delta\to\infty} \|AT_\delta(m) - m\|^2 = \sum_{j=1}^{k} (m_j')^2$$

and the claim follows from orthogonality of U. □

Numerical implementation of Morozov's method is now simple. Just find the unique zero of the function

$$f(\delta) = \sum_{j=1}^{r} \left(\frac{\delta}{d_j^2 + \delta} \right)^2 (m_j')^2 + \sum_{j=r+1}^{k} (m_j')^2 - \kappa^2. \tag{4.17}$$

Let us try Morozov's method in practice.

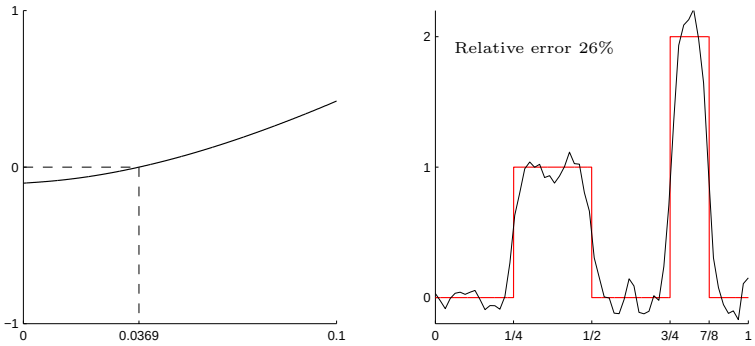


Figure 4.5: Demonstration of Morozov's discrepancy principle with noise level 1%. Left: Plot of function $f(\delta)$ defined in (4.17). Note that as the theory predicts, the function f is strictly increasing. Right: Tikhonov regularized reconstruction using $\delta = 0.0369$.

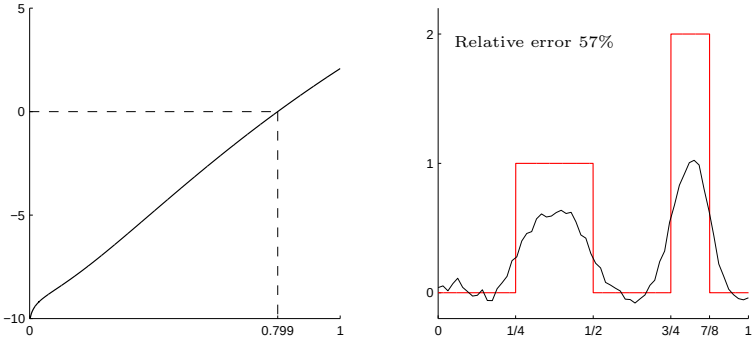


Figure 4.6: Demonstration of Morozov's discrepancy principle with noise level 10%. Left: Plot of function $f(\delta)$ defined in (4.17). Right: Tikhonov regularized reconstruction using $\delta = 0.799$.

4.2.3 Generalized Tikhonov regularization

Sometimes we have *a priori* information about the solution of the inverse problem. For example, we may know that x is close to a signal $x_* \in \mathbb{R}^n$; then we minimize

$$T_\delta(m) = \arg\min_{z\in\mathbb{R}^n} \Big\{ \|Az - m\|^2 + \delta \|z - x_*\|^2 \Big\}. \tag{4.18}$$

Another typical situation is that x is known to be smooth. Then we minimize

$$T_\delta(m) = \arg\min_{z\in\mathbb{R}^n} \Big\{ \|Az - m\|^2 + \delta \|Lz\|^2 \Big\}. \tag{4.19}$$

or

$$T_\delta(m) = \arg\min_{z\in\mathbb{R}^n} \Big\{ \|Az - m\|^2 + \delta \|L(z - x_*)\|^2 \Big\}. \tag{4.20}$$

where L is a discretized differential operator.

For example in dimension 1, we can discretize the derivative of the continuum signal by difference quotient

$$\frac{d\mathcal{X}}{ds}(s_j) \approx \frac{\mathcal{X}(s_{j+1}) - \mathcal{X}(s_j)}{\Delta s} = \frac{x_{j+1} - x_j}{\Delta s}.$$

This leads to the discrete differentiation matrix

$$L = \frac{1}{\Delta s} \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & \cdots & & 0 \\ 0 & -1 & 1 & 0 & 0 & \cdots & & 0 \\ 0 & 0 & -1 & 1 & 0 & \cdots & & 0 \\ \vdots & & & & \ddots & & & \\ \vdots & & & & & \ddots & & \\ 0 & & \cdots & & 0 & -1 & 1 & 0 \\ 0 & & \cdots & & 0 & 0 & -1 & 1 \end{bmatrix} \tag{4.21}$$

4.2.4 Normal equations and stacked form

Consider the quadratic functional $Q_\delta : \mathbb{R}^n \to \mathbb{R}$ defined by

$$Q_\delta(x) = \|Ax - m\|^2 + \delta \|x\|^2.$$

It can be proven that Q_δ has a unique minimum for any $\delta > 0$. The minimizer $T_\delta(m)$ (i.e. the Tikhonov regularized solution of $m = Ax + \varepsilon$) satisfies

$$0 = \frac{d}{dt} \left\{ \|A(T_\delta(m) + tw) - m\|^2 + \delta \|T_\delta(m) + tw\|^2 \right\} \Big|_{t=0}$$

for any $w \in \mathbb{R}^n$.

Compute

$$\begin{aligned}
&\frac{d}{dt}\|A(T_\delta(m)+tw)-m\|^2\Big|_{t=0}\\
=&\frac{d}{dt}\langle AT_\delta(m)+tAw-m, AT_\delta(m)+tAw-m\rangle\Big|_{t=0}\\
=&\frac{d}{dt}\Big\{\|AT_\delta(m)\|^2+2t\langle AT_\delta(m),Aw\rangle+t^2\|Aw\|^2\\
&\qquad -2t\langle m,Aw\rangle-2\langle AT_\delta(m),m\rangle+\|m\|^2\Big\}\Big|_{t=0}\\
=&2\langle AT_\delta(m),Aw\rangle-2\langle m,Aw\rangle,
\end{aligned}$$

and

$$\begin{aligned}
&\frac{d}{dt}\delta\langle T_\delta(m)+tw, T_\delta(m)+tw\rangle\Big|_{t=0}\\
=&\delta\frac{d}{dt}\left\{\|T_\delta(m)\|^2+2t\langle T_\delta(m),w\rangle+t^2\|w\|^2\right\}\Big|_{t=0}\\
=&2\delta\langle T_\delta(m),w\rangle.
\end{aligned}$$

So we get $\langle AT_\delta(m)-m, Aw\rangle+\delta\langle T_\delta(m),w\rangle=0$, and by taking transpose

$$\langle A^TAT_\delta(m)-A^Tm, w\rangle+\delta\langle T_\delta(m),w\rangle=0,$$

so finally we get the variational form

$$\langle (A^TA+\delta I)T_\delta(m)-A^Tm, w\rangle=0. \tag{4.22}$$

Since (4.22) holds for any nonzero $w\in\mathbb{R}^n$, we necessarily have $(A^TA+\delta I)T_\delta(m)=A^Tm$. So the Tikhonov regularized solution $T_\delta(m)$ satisfies

$$T_\delta(m)=(A^TA+\delta I)^{-1}A^Tm, \tag{4.23}$$

and actually (4.23) can be used for computing $T_\delta(m)$ defined in the basic situation (4.12).

In the generalized case of (4.19) we get by similar computation

$$T_\delta(m)=(A^TA+\delta L^TL)^{-1}A^Tm. \tag{4.24}$$

Next we will derive a computationally attractive *stacked form* version of (4.13). We rethink problem (4.13) so that we have two measurements on x that we minimize simultaneously in the least squares sense. Namely, we consider both equations $Ax=m$ and $Lx=0$ as independent measurements of the same object x, where $A\in\mathbb{R}^{k\times n}$ and $L\in\mathbb{R}^{\ell\times n}$. Now we stack the matrices and right hand sides so that the regularization parameter $\delta>0$ is involved correctly:

$$\begin{bmatrix} A \\ \sqrt{\delta}L \end{bmatrix} x=\begin{bmatrix} m \\ 0 \end{bmatrix}. \tag{4.25}$$

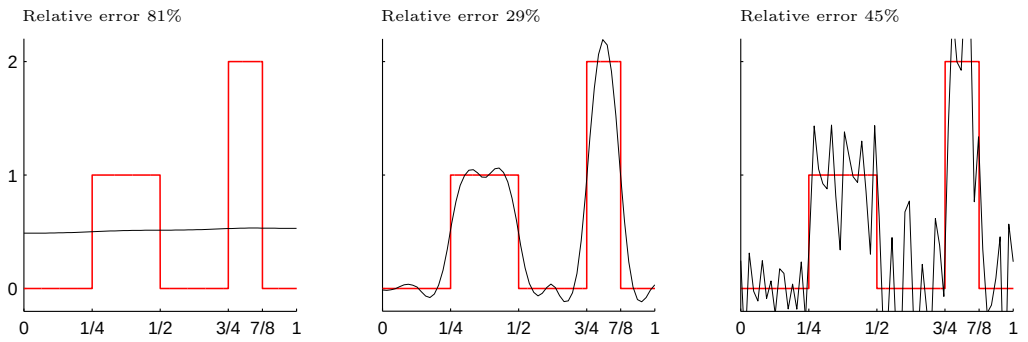


Figure 4.7: Generalized Tikhonov regularized solutions with matrix L as in (4.21). Left: $\delta = 1$. Middle: $\delta = 10^{-3}$. Right: $\delta = 10^{-6}$. Here the noise level is 1% in all three reconstructions. Compare to Figure 4.2.

We write (4.25) as $\tilde{A}x = \tilde{m}$ and solve for $T_\delta(m)$ defined in (4.24) in Matlab by

$$x = \tilde{A}\backslash\tilde{m}, \tag{4.26}$$

where \ stands for least squares solution. This is a good method for medium-dimensional inverse problems, where n and k are of the order $\sim 10^3$. Formula (4.26) is applicable to higher-dimensional problems than formula (4.13) since there is no need to compute the SVD for (4.26).

Why would (4.26) be equivalent to (4.24)? In general, a computation similar to the above shows that a vector z_0, defined as the minimizer

$$z_0 = \arg\min_z \|Bz - b\|^2,$$

satisfies the normal equations $B^TBz_0 = B^Tb$. In this case the minimizing z_0 is called the least squares solution to equation $Bz = b$. In the context of our stacked form formalism, the least squares solution of (4.25) satisfies the normal equations

$$\tilde{A}^T\tilde{A}x = \tilde{A}^T\tilde{m}.$$

But

$$\tilde{A}^T\tilde{A} = \begin{bmatrix} A^T & \sqrt{\delta}L^T \end{bmatrix} \begin{bmatrix} A \\ \sqrt{\delta}L \end{bmatrix} = A^TA + \delta L^TL$$

and

$$\tilde{A}^T\tilde{m} = \begin{bmatrix} A^T & \sqrt{\delta}L^T \end{bmatrix} \begin{bmatrix} m \\ 0 \end{bmatrix} = A^Tm,$$

so it follows that $(A^TA + \delta L^TL)x = A^Tm$.

Let us try out generalized Tikhonov regularization on our one-dimensional deconvolution problem.

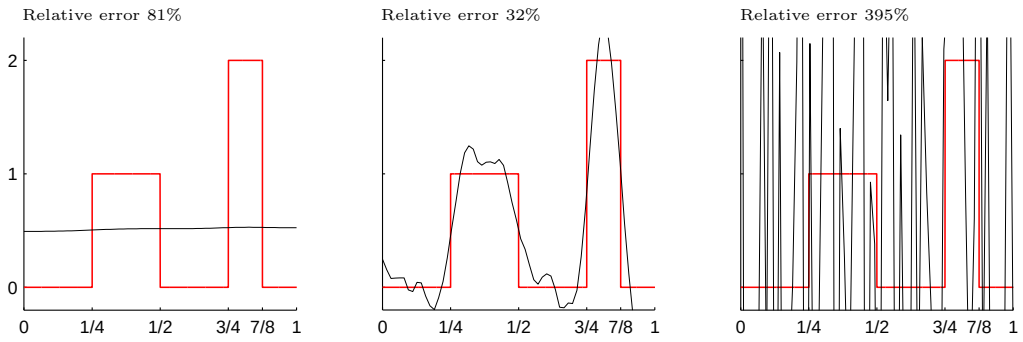


Figure 4.8: Generalized Tikhonov regularized solutions with matrix L as in (4.21). Left: $\delta = 10$. Middle: $\delta = 0.1$. Right: $\delta = 0.001$. Here the noise level is 10% in all three reconstructions. Compare to Figures 4.3 and 4.7.

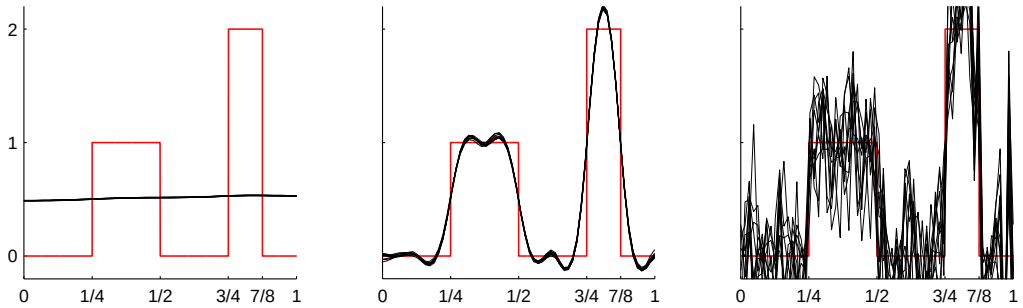


Figure 4.9: Generalized Tikhonov regularized solutions with matrix L as in (4.21) and various realizations of random noise. Left: $\delta = 10$. Middle: $\delta = 0.1$. Right: $\delta = 0.001$. Here the noise level is 1% in all reconstructions. Compare to Figures 4.4, 4.7 and 4.8.

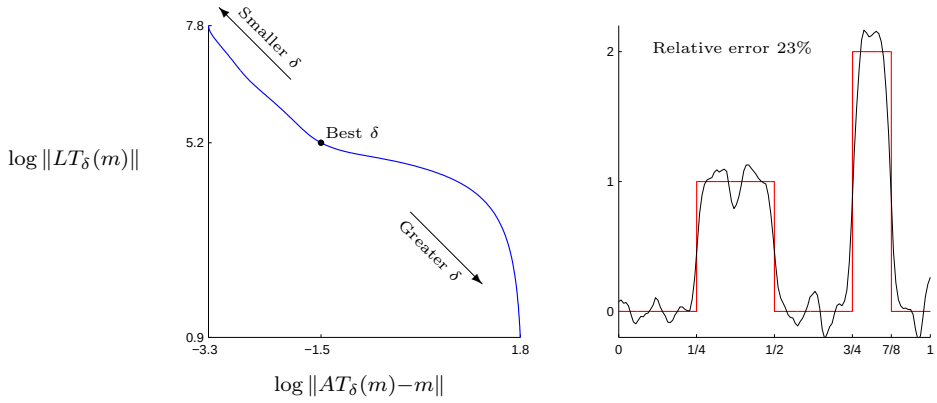


Figure 4.10: Using L-curve to find regularization parameter. Left: L-curve shown in blue. Right: reconstruction using the best value for δ.

4.2.5 Choosing δ: the L-curve method

In Section 4.2.2 we discussed Morozov's method for choosing the regularization parameter $\delta > 0$ for Tikhonov regularization. As the method of Morozov does not apply to the generalized regularization formulas (4.18)–(4.20), we need to discuss alternative approaches. One possibility is to use the so-called L-curve method.

The idea of the L-curve method is to choose a collection of candidates for regularization parameter:

$$0 < \delta_1 < \delta_2 < \cdots < \delta_M < \infty,$$

and compute $T_{\delta_j}(m)$ for each $1 \le j \le M$. Then the points

$$(\log \|AT_\delta(m) - m\|, \log \|LT_\delta(m)\|) \in \mathbb{R}^2$$

are plotted in the plane, forming approximately a smooth curve. This curve has typically the shape of the letter L with smooth corner. The optimal value of δ is thought to be found as near the corner as possible.

Let us consider the generalized Tikhonov regularization of the form (4.19) with L given by (4.21). See Figure 4.10 for the result. For more information about the L-curve method, see the book by Hansen [9] and references therein.

4.2.6 Large-scale computation: matrix-free iterative method

The formulation (4.24) of Tikhonov regularization is remarkable because it allows matrix-free implementation. Namely, assume that we have available computational routines called `Amult` and `Lmult` that take an arbitrary vector $z \in \mathbb{R}^n$ as argument and return

$$\texttt{Amult}(z) = Az \in \mathbb{R}^k, \qquad \texttt{Lmult}(z) = Lz \in \mathbb{R}^{k'},$$

respectively. Further, since the transposes $A^T : \mathbb{R}^k \to \mathbb{R}^n$ and $L^T : \mathbb{R}^{k'} \to \mathbb{R}^n$ appear in (4.24) as well, we need computational routines called `ATmult` and `LTmult` that take vectors $v \in \mathbb{R}^k$ and $w \in \mathbb{R}^{k'}$ as arguments and return

$$\texttt{ATmult}(v) = A^T v \in \mathbb{R}^n, \qquad \texttt{LTmult}(w) = L^T w \in \mathbb{R}^n.$$

Now we can solve the linear equation $(A^T A + \delta L^T L)x = A^T m$ without actually constructing any of the matrices A, A^T, L or L^T! The trick is to use an iterative solution strategy, such as the conjugate gradient method.

4.3 Total variation regularization

Rudin, Osher and Fatemi introduced in 1992 the following idea: instead of minimizing

$$\|Ax - m\|_2^2 + \delta \|Lx\|_2^2 \tag{4.27}$$

let us minimize

$$\|Ax - m\|_2^2 + \delta \|Lx\|_1. \tag{4.28}$$

Recall that $\|z\|_2^2 = |z_1|^2 + \cdots + |z_n|^2$ and $\|z\|_1 = |z_1| + \cdots + |z_n|$.

The idea is that (4.28) should allow occasional larger jumps in the reconstruction, leading to piecewise smoothness instead of overall smoothness. It turns out that (4.28) really is a powerful method, but numerical minimization is more difficult than in the case of Tikhonov regularization; this is because the function to be minimized is no more quadratic (and actually not even differentiable).

We will take a look at two different ways to compute total variation regularized solutions.

4.3.1 Quadratic programming

Consider applying total variation regularization for a discretized one-dimensional continuum inverse problem. We want to minimize

$$\|Ax - m\|_2^2 + \delta \sum_{j=0}^{n} |(Lx)_j|, \tag{4.29}$$

where $(Lx)_j = x_{j+1} - x_j$ for $j = 0, \ldots, n$ with the convention $x_0 = 0$ and $x_{n+1} = 0$. These boundary conditions lead to slightly different form for the L

matrix compared to (4.21):

$$L = \frac{1}{\Delta s}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & & & & \ddots & & \\ \vdots & & & & & \ddots & \\ 0 & & \cdots & & 0 & -1 & 1 & 0 \\ 0 & & \cdots & & 0 & 0 & -1 & 1 \\ 0 & & \cdots & & 0 & 0 & 0 & -1 \end{bmatrix}. \tag{4.30}$$

Write Lx in the form

$$V_+ - V_- = Lx,$$

where $V_\pm$ are nonnegative vectors: $V_\pm \in \mathbb{R}^{n+1}_+$, or $(V_\pm)_j \geq 0$ for all $j = 1, \ldots, n+1$. Now minimizing (4.29) is equivalent to minimizing

$$\|Ax\|_2^2 - 2m^T Ax + \delta \mathbf{1}^T V_+ + \delta \mathbf{1}^T V_-,$$

where $\mathbf{1} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T \in \mathbb{R}^{n+1}$ and the minimization is taken over vectors

$$y = \begin{bmatrix} x \\ V_+ \\ V_- \end{bmatrix}, \quad \text{where} \quad \begin{matrix} x \in \mathbb{R}^n \\ V_+ \in \mathbb{R}^{n+1}_+ \\ V_- \in \mathbb{R}^{n+1}_+ \end{matrix}.$$

Write now

$$H = \begin{bmatrix} 2A^TA & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad f = \begin{bmatrix} -2A^Tm \\ \delta\mathbf{1} \\ \delta\mathbf{1} \end{bmatrix}.$$

Then we deal with the standard form quadratic problem

$$\arg\min_y \left\{ \frac{1}{2} y^T H y + f^T y \right\} \tag{4.31}$$

with the constraints

$$L \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_{n+1} \\ \vdots \\ y_{2n+1} \end{bmatrix} - \begin{bmatrix} y_{2n+2} \\ \vdots \\ y_{3n+2} \end{bmatrix} \quad \text{and} \tag{4.32}$$

$$y_j \geq 0 \text{ for } j = n+1, \ldots, 3n+2.$$

Several software packages (such as `quadprog.m` routine in Matlab's Optimization toolbox) exist that can deal with a problem of the form (4.31) with constraints of type (4.32).

The two-dimensional case is slightly more complicated since we need to discretize $\nabla\mathcal{X}$ with respect to two directions. One possibility is to write horizontal and vertical difference quotients in the form of two matrices L_H and L_V.

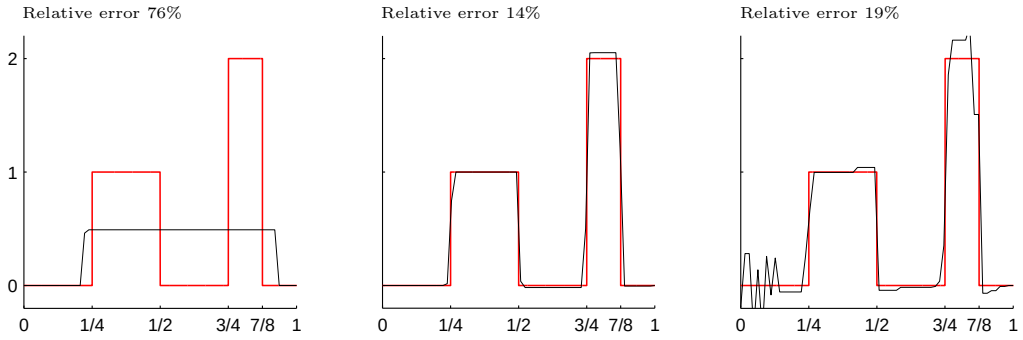


Figure 4.11: Total variation regularized solutions with matrix L as in (4.21). Left: $\delta = 10$. Middle: $\delta = 0.1$. Right: $\delta = 10^{-4}$. Here the noise level is 1% in all three reconstructions.

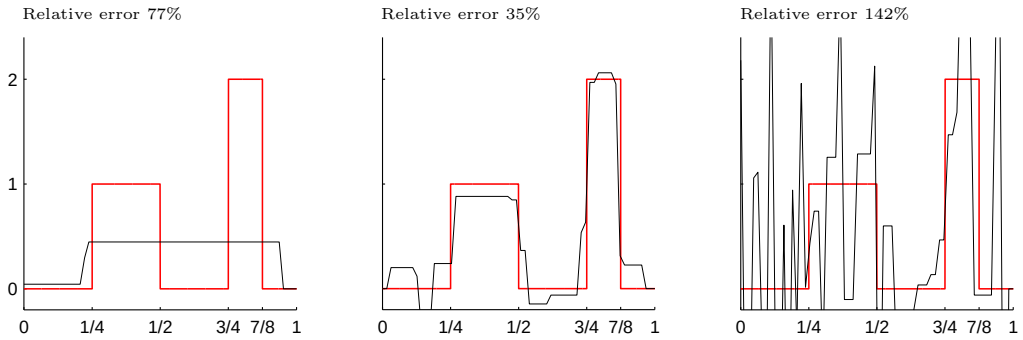


Figure 4.12: Total variation regularized solutions with matrix L as in (4.21). Left: $\delta = 10$. Middle: $\delta = 0.1$. Right: $\delta = 10^{-4}$. Here the noise level is 10% in all three reconstructions. Compare to Figure 4.11.

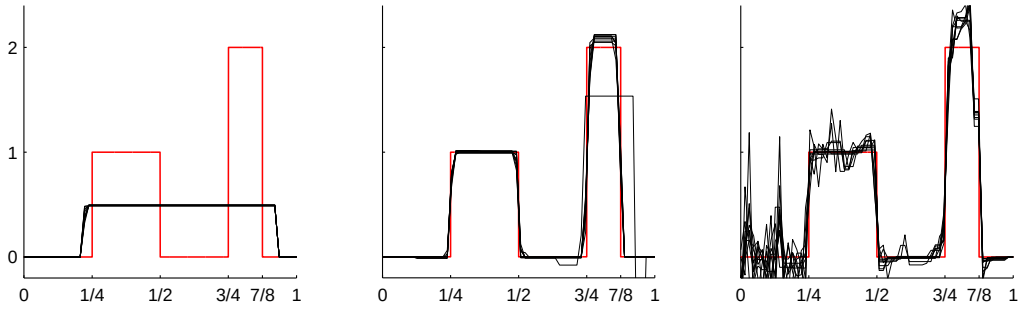


Figure 4.13: Total variation regularized solutions with matrix L as in (4.21). Left: $\delta = 10$. Middle: $\delta = 0.1$. Right: $\delta = 10^{-4}$. Here the noise level is 1% and we repeated the computation with 10 different realizations of noise. Compare to the corresponding computation using Tikhonov regularization in Figure 4.9.

4.3.2 Large-scale computation: Barzilai-Borwein method

Consider again applying total variation regularization for a discretized one-dimensional continuum inverse problem. Our aim is to minimize

$$\begin{aligned} f(x) &= \|Ax - m\|_2^2 + \delta \|Lx\|_1 \\ &= \|Ax - m\|_2^2 + \delta \sum_{j=1}^{n-1} |x_i - x_j|, \end{aligned}$$

but since f is not continuously differentiable we cannot apply any derivative-based optimization method.

Let us replace the absolute value function $|t|$ by an approximation:

$$|t|_\beta := \sqrt{t^2 + \beta},$$

where $\beta > 0$ is small. (Another possible choice is $|t|_\beta = \frac{1}{\beta} \log(\cosh(\beta t))$.)

Then the objective function

$$f_\beta(x) = \|Ax - m\|_2^2 + \delta \sum_{j=1}^{n-1} |x_i - x_j|_\beta$$

is continuously differentiable and we can apply gradient-based optimization methods.

The *steepest descent method* was introduced by Cauchy in 1847. It is an iterative method where the initial guess $x^{(1)}$ is just chosen somehow (e.g. $x^{(1)} = 0$) and the next iterates are found inductively by

$$x^{(\ell+1)} = x^{(\ell)} - \alpha_\ell \nabla f(x^{(\ell)}),$$

where the step size α_ℓ is determined from

$$\alpha_\ell = \arg\min_\alpha f(x^{(\ell)} - \alpha \nabla f(x^{(\ell)})).$$

The steepest descent method is known to be slow and badly affected by ill-conditioning.

Barzilai and Borwein introduced in 1988 the following optimization strategy which differs from the steepest descent method only by the choice of steplength:

$$x^{(\ell+1)} = x^{(\ell)} - \alpha_\ell \nabla f(x^{(\ell)}),$$

where α_ℓ is given by setting $y_\ell := x^{(\ell+1)} - x^{(\ell)}$ and $g_\ell := \nabla f(x^{(\ell+1)}) - \nabla f(x^{(\ell)})$ and

$$\alpha_\ell = \frac{y_\ell^T y_\ell}{y_\ell^T g_\ell}.$$

This method converges faster and is less affected by ill-conditioning than the steepest descent method. (Especially for quadratic f) There are some practical problems with the method of Barzilai and Borwein:

(i) How to choose the first steplength α_1?

(ii) The objective function is not guaranteed to get smaller with each step. What to do in the case it becomes bigger?

The quick-and-dirty solution to (i) is just choosing α_1 to be small, for example $\alpha_1 = \frac{1}{10\,000}$. Another practical way to choose α_1 by line minimization.

One simple way to deal with (ii) is to check if f increases, and if so, half the steplength. However, this is not the best possible way to ensure the convergence of the method, since just the increasing steps have turned out to be essential for the local convergence properties of the Barzilai-Borwein method. It is often advisable to just let the method run in spite of occasionally increasing objective function values.

Strategies to guarantee the global convergence of the Barzilai-Borwein method can be found for instance in papers of Dai & Fletcher (2003) and Raydan (1997). Constrained optimization, such as enforcing nonnegativity, using Barzilai-Borwein method is discussed in [6, 29].

Note that the storage need of the Barzilai-Borwein method method is of the order n instead of n^2 typical for many other methods. If x is a large $M \times N$ size image, then $n^2 = M^2N^2$ is too large for most computer memories!

4.4 Truncated iterative solvers

This topic is not discussed in the 2008 implementation of the course.

4.5 Exercises

1. Show that the matrix $A^TA + \delta I$ is always invertible when $\delta > 0$ and A is an arbitrary $k \times n$ matrix. Hint: use SVD.

2. Consider regularized solution $T_\delta(m)$ of equation $m = Ax+\varepsilon$ using truncated SVD with truncation index $p(\delta)$ for $\delta > 0$. Assume that the noise level is $\kappa = \|\varepsilon\|$. Show that the discrepancy condition $\|AT_\delta(m) - m\| \leq \kappa$ can be written in the form
$$\sum_{j=p(\delta)+1}^{m} (y_j')^2 \leq \kappa^2.$$
(This is the equivalent of Morozov's discrepancy condition for truncated SVD.)

3. Show that the variational form corresponding to the minimization problem
$$T_\delta(m) = \arg\min_{z\in\mathbb{R}^n} \{\|Az - m\|^2 + \delta\|Lz\|^2\}$$
is given by
$$\big\langle (A^TA + \delta L^TL)T_\delta(m) - A^Tm, w\big\rangle = 0 \qquad \text{for all } w \in \mathbb{R}^n.$$

4. Write the generalized Tikhonov problem

$$T_\delta(m) = \arg\min_{z\in\mathbb{R}^n}\{\|Az - m\|^2 + \delta\|L(z - x_\star)\|^2\}$$

in stacked form.

Chapter 5

Statistical inversion

5.1 Introduction to random variables

One way to think about a real-valued random variable X is through drawing samples. We can take a *sample* $x^{(1)} \in \mathbb{R}$ (also called *realization*) of X. The probabilistic nature of X is described by its *probability density function* $p_X : \mathbb{R} \to \mathbb{R}$ satisfying the following properties:

$$p_X(x) \geq 0 \text{ for all } x \in \mathbb{R}, \tag{5.1}$$

$$\int_{-\infty}^{\infty} p_X(x)dx = 1. \tag{5.2}$$

The probability for the sample $x^{(1)}$ to belong to the interval $[a, b] \subset \mathbb{R}$ is given by the integral

$$\Pr(a \leq x^{(1)} \leq b) = \int_a^b p_X(x)dx.$$

Note that (5.2) ensures that the probability for $x^{(1)}$ to belong to $\mathbb{R}$ is one. If we draw a series $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(N)}\}$ of samples of X, then the histogram of the samples is close to the graph of p_X when N is large. See Figure 5.1.

We only consider continuously differentiable probability density functions in this course. For more general treatment involving σ-algebras see [23].

Another useful function related to the random variable X is the *cumulative distribution function* $P_X : \mathbb{R} \to [0, 1]$ defined by

$$P_X(x) = \int_{-\infty}^{x} p_X(t)dt. \tag{5.3}$$

Note that $\lim_{x\to-\infty} P_X(x) = 0$ and $\lim_{x\to\infty} P_X(x) = 1$ and that P_X is a monotonically increasing function. The cumulative distribution function is handy when drawing samples from real-valued random variables. Namely, one can draw a sample $r^{(1)}$ from the uniform distribution on the interval $[0, 1]$ (by MATLAB function `rand`, for instance) and then set

$$x^{(1)} := P_X^{-1}(r^{(1)}). \tag{5.4}$$

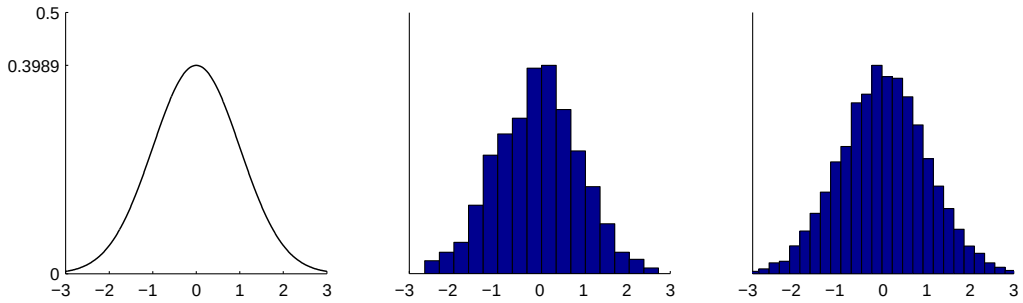


Figure 5.1: Left: Gaussian normal distribution with $\sigma = 1$, or more precisely $p_X(x) = (2\pi)^{-1/2}\exp(-\frac{1}{2}x^2)$. Middle: histogram of 1000 samples drawn using the MATLAB function `randn`. Right: histogram of 10000 samples.

A series $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(N)}\}$ of samples of X is then produced by

$$\{P_X^{-1}(r^{(1)}), P_X^{-1}(r^{(2)}), P_X^{-1}(r^{(3)}), \ldots, P_X^{-1}(r^{(N)})\},$$

where sampling $r^{(j)}$ is trivial.

The Bayesian approach to solving inverse problems is based on conditional probabilities. Let us consider a joint probability density $p_{XM} : \mathbb{R}^2 \to \mathbb{R}$ of two $\mathbb{R}$-valued random variables X and M. We must have

$$p_{XM}(x,m) \geq 0 \text{ for all } x, m \in \mathbb{R}, \tag{5.5}$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{XM}(x,m)dxdm = 1. \tag{5.6}$$

Now the probability that a sampled pair $(x^{(1)}, m^{(1)})$ belongs to the rectangle $[a,b]\times[c,d]$ is given by the integral

$$\Pr(a \leq x^{(1)} \leq b \text{ and } c \leq m^{(1)} \leq d) = \int_a^b\int_c^d p_{XM}(x,m)dxdm.$$

Now we can define the *marginal distributions* of X and M by

$$p_X(x) = \int_{-\infty}^{\infty} p_{XM}(x,m)dm, \qquad p_M(m) = \int_{-\infty}^{\infty} p_{XM}(x,m)dx,$$

respectively. Furthermore, the conditional probability of X given a fixed value of M is defined by

$$p_{X|M}(x|m) = \frac{p_{XM}(x,m)}{p_M(m)}. \tag{5.7}$$

It is easy to check that

$$\int_{-\infty}^{\infty} p_{X|M}(x|m)dx = 1.$$

Similarly we define the conditional probability of M given a fixed value of X by

$$p_{M|X}(m|x) = \frac{p_{XM}(x,m)}{p_X(x)}. \tag{5.8}$$

A combination of (5.7) and (5.8) yields the Bayes formula

$$p_{X|M}(x|m) = \frac{p_X(x)\, p_{M|X}(m|x)}{p_M(m)}. \tag{5.9}$$

5.2 Bayesian framework

In statistical inversion we consider the measurement model

$$M = AX + \varepsilon,$$

where ε is as before and now M and X are considered random variables taking values in $\mathbb{R}^k$ and $\mathbb{R}^n$, respectively. Using Bayes formula (5.9) we can define the *posterior distribution*

$$p_{X|M}(x|m) \sim p_X(x)\, p_{M|X}(m|x), \tag{5.10}$$

where $\sim$ means that we ignore normalization constants.

The density $p_{M|X}(m|x)$ in (5.10) is called *likelihood distribution* and is related to data misfit. In the case of Gaussian noise treated in this course, the likelihood distribution takes the form

$$p_{M|X}(m|x) \sim \exp(-\frac{1}{2\sigma^2}\|Ax - m\|_2^2). \tag{5.11}$$

We consider $p_{M|X}(m|x)$ as a function of both x and m. For a fixed x the density $p_{M|X}(m|x)$ specifies a high probability to a measurement m that could come from x via Ax and a low probability for measurements m far away from Ax. On the other hand, for a fixed m the density $p_{M|X}(m|x)$ assigns high probability only to vectors x for which Ax is close to m.

The role of the *prior distribution* $p_X(x)$ in (5.10) is to code all *a priori* information we have on the unknown X in the form of a probability density. The function $p_X : \mathbb{R}^n \to \mathbb{R}$ should assign high probability to vectors $x \in \mathbb{R}^n$ that are probable in light of *a priori* information, and low probability to atypical vectors x. Constructing $p_X(x)$ in a computationally effective way is often the central difficulty in statistical inversion.

The posterior distribution $p_{X|M}(x|m)$ defined in (5.10) is considered to be the complete solution of the inverse problem

Given a realization of M, find information about X.

The probabilities encoded in $p_{X|M}(x|m)$ are difficult to visualize, however, since x ranges in n-dimensional space. This is why we need to compute some point

estimate (and possibly confidence intervals) from the posterior density. Popular point estimates include the *Maximum a posteriori* estimate

$$\arg\max_{x\in\mathbb{R}^n} p_{X|M}(x|m), \tag{5.12}$$

or the vector $\mathbb{R}^n$ giving the largest probability, and the *conditional mean* estimate

$$\int_{\mathbb{R}^n} x\, p_{X|M}(x|m)\, dx. \tag{5.13}$$

Numerical computation of (5.12) is an optimization problem, while the evaluation of (5.13) requires integration in n-dimensional space. In Section 5.3 we will discuss sampling methods for the evaluation of (5.13).

Let us remark that if the matrix A is well-conditioned then one can compute the vector x that gives the maximum value for $p_{M|X}(m|x)$ with a given and fixed m. This is called the maximum likelihood (ML) method. In inverse problems the matrix A is ill-conditioned, and there is a large number (possibly even a full linear subspace) of vectors x that give essentially the same value for $p_{M|X}(m|x)$. So the prior distribution represents information that is necessary for stable solution of the inverse problem because the measurement information coded in the likelihood distribution is not enough to specify x uniquely and robustly. In ideal situations the prior distribution contains orthogonal information to the likelihood in the sense that the product $p_X(x)\, p_{M|X}(m|x)$ describes a nicely centered probability mass even though $p_{M|X}(m|x)$ does not.

5.3 Monte Carlo Markov chain methods

The idea is to compute the conditional mean estimate approximately using the formula

$$\int_{\mathbb{R}^n} x\, p_{X|M}(x|m)\, dx \approx \frac{1}{N}\sum_{\ell=1}^{N} x^{(\ell)}, \tag{5.14}$$

where the sequence $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(N)}\}$ is distributed according to the posterior density $p_{X|M}(x|m)$. Such sequences can be constructed using Monte Carlo Markov chain methods. The term *Markov chain* is related to the construction of the sequence; it means roughly that the generation of a new member $x^{(N+1)}$ for the sequence only depends on the previous member $x^{(N)}$.

The initial guess $x^{(1)}$ is often far away from the conditional mean, and some of the first sample vectors need to be discarded. This leads to choosing some $1 < N_0 \ll N$ and replacing (5.14) by

$$\int_{\mathbb{R}^n} x\, p_{X|M}(x|m)\, dx \approx \frac{1}{N-N_0}\sum_{\ell=N_0+1}^{N} x^{(\ell)}. \tag{5.15}$$

The discarded part $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(N_0)}\}$ is called the *burn-in period*.

We will discuss in Sections 5.3.2 and 5.3.3 two different MCMC methods. For more detailed information about MCMC methods and their convergence properties see e.g. [8, 11, 7, 24, 26, 21].

5.3.1 Markov chains

5.3.2 Gibbs sampler

The so-called *single component Gibbs sampler* proceeds as follows:

1. Fix the initial draw $x^{(1)} = (x_1^{(1)}, \ldots, x_n^{(1)})$ and set $\ell = 2$.
2. Generate $x^{(\ell)}$ a single variable at a time:

$$\begin{aligned}
&\text{Draw} \quad x_1^{(\ell)} \quad \text{from the density } t \mapsto p(t, x_2^{(\ell-1)}, \ldots, x_n^{(\ell-1)}|m),\\
&\text{draw} \quad x_2^{(\ell)} \quad \text{from the density } t \mapsto p(x_1^{(\ell)}, t, x_3^{(\ell-1)}, \ldots, x_n^{(\ell-1)}|m),\\
&\qquad\qquad \vdots\\
&\text{draw} \quad x_n^{(\ell)} \quad \text{from the density } t \mapsto p(x_1^{(\ell)}, \ldots, x_{n-1}^{(\ell)}, t|m).
\end{aligned}$$

3. Set $\ell \leftarrow \ell + 1$ and go to (ii).

Drawing of samples in step 2 is simple to implement via (5.4). One needs to evaluate the cumulative distribution function of the density using numerical integration (and making sure that the grid is fine enough; too coarse grid is a typical source of errors). Note that it does not matter if the density integrates to 1 or some other positive value $I > 0$. If $I \neq 1$ then (5.4) is just modified as follows:

$$x_j^{(\ell)} := P_X^{-1}(I \cdot \texttt{rand}),$$

where the MATLAB command `rand` is sampled from the uniform density in $[0, 1]$.

The determination of the so-called full conditional densities of a single component x_k while the remaining ones are fixed can in some cases be carried out analytically, at least with respect to some variables, but since they are functions of a single variable only, it is relatively straightforward to approximate the associated distribution functions non-parametrically and then employ the well know golden rule to draw the samples. Compared to the Metropolis-Hastings method, the virtue of the Gibbs sampler is the absence of the problems related to the choice of the proposal distribution as well as questions related to the acceptance rule. The significant drawback is that it becomes easily slow when the number of the components is large as it is in real X-ray tomography problems.

Let us consider a deconvolution problem in 1-D: Given a convolution operator A and data $m = Ax + \varepsilon$, the task is to find the original signal x. The noise ε is assumed to be white Gaussian noise with noise level σ. This model yields the likelihood

$$p_{M|X}(m|x) \sim \exp(-\frac{1}{2\sigma^2}\|Ax - m\|_2^2).$$

In addition, it is known that the components x_j of the original signal are confined to the interval $[0, 2]$. This gives the prior distribution

$$p_X(x) \sim \chi_{[0,2]}(x) := \prod_{j=1}^{n} \chi_{[0,2]}(x_j).$$

In consequence, the posterior distribution is

$$p_{X|M}(x|m) \sim \exp(-\frac{1}{2\sigma^2}\|Ax - m\|_2^2)\chi_{[0,2]}(x).$$

In order to use Gibbs sampling, we need to be able to sample from the distribution

$$t \mapsto p_{X|M}(\tilde{x}^{(\ell,j)}(t)), \quad j \in \{1, 2, \ldots, n\}, \tag{5.16}$$

where

$$\tilde{x}^{(\ell,j)}(t) = (x_1^{(\ell)} \ldots, x_{j-1}^{(\ell)}, t, x_{j+1}^{(\ell-1)}, \ldots, x_n^{(\ell-1)})^T.$$

Distribution (5.16) can be written as

$$p_{X|M}(\tilde{x}^{(\ell,j)}(t)) = C \exp(-\frac{1}{2\sigma^2}\|a^{(j)}t - \tilde{m}^{(\ell,j)}\|_2^2)\chi_{[0,2]}(t),$$

where

$$\tilde{m}^{(\ell,j)} = m - A\tilde{x}^{(\ell,j)}(0)$$

and $a^{(j)}$ denotes the j:th column of A. We use Householder transformation to obtain a simple formula for this distribution. Let $e^{(1)} = (1, 0, \ldots, 0)^T$ and

$$Q^{(j)} = I - 2\frac{(a^{(j)} - \|a^{(j)}\|e^{(1)})(a^{(j)} - \|a^{(j)}\|e^{(1)})^T}{(a^{(j)} - \|a^{(j)}\|e^{(1)})^T(a^{(j)} - \|a^{(j)}\|e^{(1)})},$$

i.e., the matrix $Q^{(j)}$ is chosen in such a way that $Q^{(j)}a^{(j)} = \|a^{(j)}\|_2 e^{(1)}$. Since the matrix $Q^{(j)}$ is orthogonal, we have

$$\begin{aligned} p_{X|M}(\tilde{x}^{(\ell,j)}(t)) &\sim \exp\left(-\frac{1}{2\sigma^2}\|Q^{(j)}(a^{(j)}t - \tilde{m}^{(\ell,j)})\|_2^2\right)\chi_{[0,2]}(t) \\ &\sim \exp\left(-\frac{\|a^{(j)}\|_2^2}{2\sigma^2}\|t - \mu^{(\ell,j)}\|_2^2\right)\chi_{[0,2]}(t), \end{aligned} \tag{5.17}$$

where

$$\mu^{(\ell,j)} = \frac{{e^{(1)}}^T Q^{(j)}\tilde{m}^{(\ell,j)}}{\|a^{(j)}\|}.$$

Samples from distribution (5.17) can be obtained using the inverse cdf-method. Let us denote the cumulative distribution function of the standard normal random variable by Φ. The distribution (5.17) is a truncated normal distribution, and its cumulative distribution function is

$$F_{\ell,j}(t) = \begin{cases} 0, & t < 0, \\ \frac{\Phi\left((t-\mu^{(\ell,j)})\|a^{(j)}\|_2/\sigma\right) - \Phi\left(-\mu^{(\ell,j)}\|a^{(j)}\|_2/\sigma\right)}{\Phi\left((2-\mu^{(\ell,j)})\|a^{(j)}\|_2/\sigma\right) - \Phi\left(-\mu^{(\ell,j)}\|a^{(j)}\|_2/\sigma\right)}, & 0 \le t \le 2, \\ 1, & t > 2. \end{cases}$$

The inverse of $F_{\ell,j}$ can be written as

$$F_{\ell,j}^{-1}(z) = \frac{\sigma}{\|a^{(j)}\|_2}\Phi^{-1}\left\{\left[\Phi\Big((2-\mu^{(\ell,j)})\frac{\|a^{(j)}\|_2}{\sigma}\Big) - \Phi\Big(-\mu^{(\ell,j)}\frac{\|a^{(j)}\|_2}{\sigma}\Big)\right] z \right. \\ \left. + \Phi\Big(-\mu^{(\ell,j)}\frac{\|a^{(j)}\|_2}{\sigma}\Big)\right\} + \mu^{(\ell,j)}. \quad (5.18)$$

Now, when we draw z from a uniform distribution on $[0,1]$, $F_{\ell,j}^{-1}(z)$ will be a sample from (5.16). However, (5.18) is not always applicable: The mean $\mu^{(\ell,j)}$ can be far away from the interval $[0,2]$, and, in such a case,

$$\Phi((2-\mu^{(\ell,j)})\frac{\|a^{(j)}\|_2}{\sigma}) \quad \text{and} \quad \Phi(-\mu^{(\ell,j)}\frac{\|a^{(j)}\|_2}{\sigma})$$

are numerically equal. In this example, we encounter this problem. As a result, (5.18) fails to map x to t. This saturation problem with Φ can be circumvented using the tabulated form of inverse cdf method. Even with the tabulated form, one has to take measures in order to avoid problems caused by floating point computations.

In fact, Gibbs sampler is not that effective in the framework of this example. First, we need to sample the components of x from distributions that are not trivial. Second, the components of x are not independent. As a result, the sampling distribution has to be numerically computed for each pair (ℓ, j), which requires a large number of floating point operations. Nevertheless, Gibbs sampler is well suited for some problems: For example, it is possible that only conditional probabilites are known. In such a case, Gibbs sampler is a natural choice for exploring the posterior.

5.3.3 Metropolis-Hastings method

In the Metropolis-Hastings algorithm the states of the Markov chain are generated as follows: Given the state $x^{(\ell)}$ of the chain, a candidate x_c for the next state is drawn from the proposal density $q(x_c|x^{(\ell)})$. Loosely speaking, $q(x_c|x^{(\ell)})$ is the probability of the move from $x^{(\ell)}$ to x_c. The candidate is not accepted automatically, however.

To understand the acceptance rule, assume first that the proposal density is symmetric, i.e., $q(x|y) = q(y|x)$ for all $x, y \in \mathbb{R}^n$. It can be interpreted by saying that the probability for moving from y to x equals the probability of moving from x to y; one simple symmetric choice is to set

$$y_j = x_j + \rho \cdot \texttt{randn} \qquad \text{for all } 1 \le j \le n, \quad (5.19)$$

where `randn` is a normally distributed random number with variance 1. In this particular case, the acceptance rule is simple: If the proposed state x_c has higher

probability than the previous state $x^{(\ell)}$, the candidate is automatically accepted. However, if it has a lower probability, it is accepted only by a probability that is proportional to the ratio of the probabilities. Hence, the acceptance probability γ of the candidate is simply

$$\gamma = \min\left\{1,\ \frac{p(x_{\mathrm{c}}|m)}{p(x^{(\ell)} \mid m)}\right\}. \tag{5.20}$$

If $q(x|y) \neq q(y|x)$, a modification of (5.20) is needed to compensate the asymmetry:

$$\gamma = \min\left\{1,\ \frac{p(x_{\mathrm{c}}|m)\, q(x^{(\ell)}|x_{\mathrm{c}})}{p(x^{(\ell)}|m)\, q(x_{\mathrm{c}}|x^{(\ell)})}\right\}. \tag{5.21}$$

If the candidate is accepted, the next state is $x^{(\ell+1)} = x_{\mathrm{c}}$. Otherwise, $x^{(\ell+1)} = x^{(\ell)}$.

The distribution of the samples converges asymptotically to $p(x|m)$. In practice the acceptance is carried out so that one first draws a sample from the proposal distribution and computes γ. Then a random number from the uniform distribution $\mathrm{Uni}\,(0,1)$ is drawn and compared with γ.

As it can be seen from the equations (5.20) and (5.21), the normalization constant $p(m)$ is canceled out in the computation of the acceptance probability and therefore it is sufficient to know the posterior density up to the normalization constant only. This is a very important issue since the computation of $p(m)$ is a formidable task.

The key problem in the Metropolis-Hastings method is to find effective proposal distribution. This is especially crucial in case of large dimensional problems. If the proposal distribution is not feasible, $\gamma \approx 0$ for almost all draws and very few of the candidates get accepted. On the other hand, the proposal distribution has to be one from which we can perform the draws. Let us demonstrate these properties with the one-dimensional deconvolution problem studied earlier.

Choose $n = 32$ and consider the measurement model described in Section 2.2.1. We create noisy data with noise level $\sigma = .1$; see Figure 5.2. We wish to examine the properties of the Metropolis-Hastings algorithm quantitatively; to this end we compute the Tikhonov regularized solution of the problem, see Figure 5.2. Namely, we will use a Gaussian smoothness prior

$$p_X(x) = \exp(-\alpha\|Lx\|_2^2) \tag{5.22}$$

implying that the posterior distribution is Gaussian. For Gaussian distributions the MAP estimate (5.12) and the conditional mean estimate (5.13) coincide, so it holds that

$$\arg\min_{z\in\mathbb{R}^n}\left\{\|Az - m\|^2 + \delta\|Lz\|^2\right\} = \arg\max_{x\in\mathbb{R}^n} p_{X|M}(x|m) \tag{5.23}$$

if we take the parameter α in formula (5.22) to be $\alpha = \delta/(2\sigma^2)$, where σ^2 is the noise variance appearing in the likelihood distribution (5.11). Showing that (5.23)

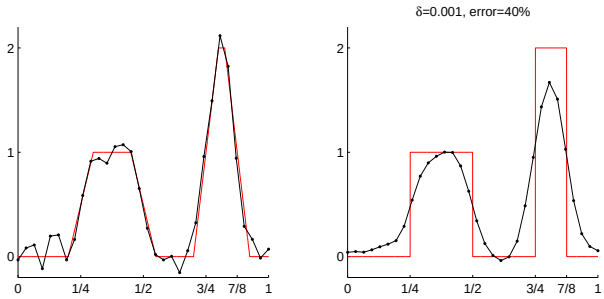


Figure 5.2: Left: ideal (red) and noisy (black) data. Right: Generalized Tikhonov regularized solution, which will be compared to the results of the Metropolis-Hastings algorithm.

holds is left as an exercise. So we have two independent methods for computing the same function, allowing quantitative measurement of error.

We remark that while formula (5.23) shows that computing the MAP estimate is sometimes equivalent to computing the Tikhonov regularized solution, the Bayesian approach is more general as it gives quantitative information about uncertainties in the estimate as well.

Let us take the simple proposal distribution (5.19), where the only degree of freedom is choosing the parameter $\rho > 0$ that determines the width of the distribution. Theoretically, every positive choice of ρ guarantees the convergence in the approximation (5.14) as the number of samples increases. However, with unfortunate choices odρ the convergence may be extremely slow and in practice leads to never-ending computation. More precisely, if ρ is too small, then most of the candidates are accepted, but they are very close to each other. In this case the average of the samples is simply close to the initial sample and approaches the actual average very slowly. On the other hand, too large ρ leads to a chain where most of the candidates are declined, and consequently same vectors are repeated extremely often. This situation also leads to very slow convergence.

Programming Metropolis-Hastings method is typically like this: it is pretty quick to get the algorithm running since it is very simple. However, tuning the proposal distribution so that the chain samples the posterior distribution effectively may take a long time. One useful practice is to calculate the *acceptance ratio*, or the percentage of accepted candidates, of the chain when performing test runs. Another helpful thing is to plot values of selected components of the vectors in the chain as function of the chain index. Let us demonstrate this numerically. We compute 3000 samples with the choices $\rho = 0.001$, $\rho = 0.025$ and $\rho = 0.2$ in formula (5.19). See Figure 5.3 for plots of components 8 and 16 of the sample vectors.

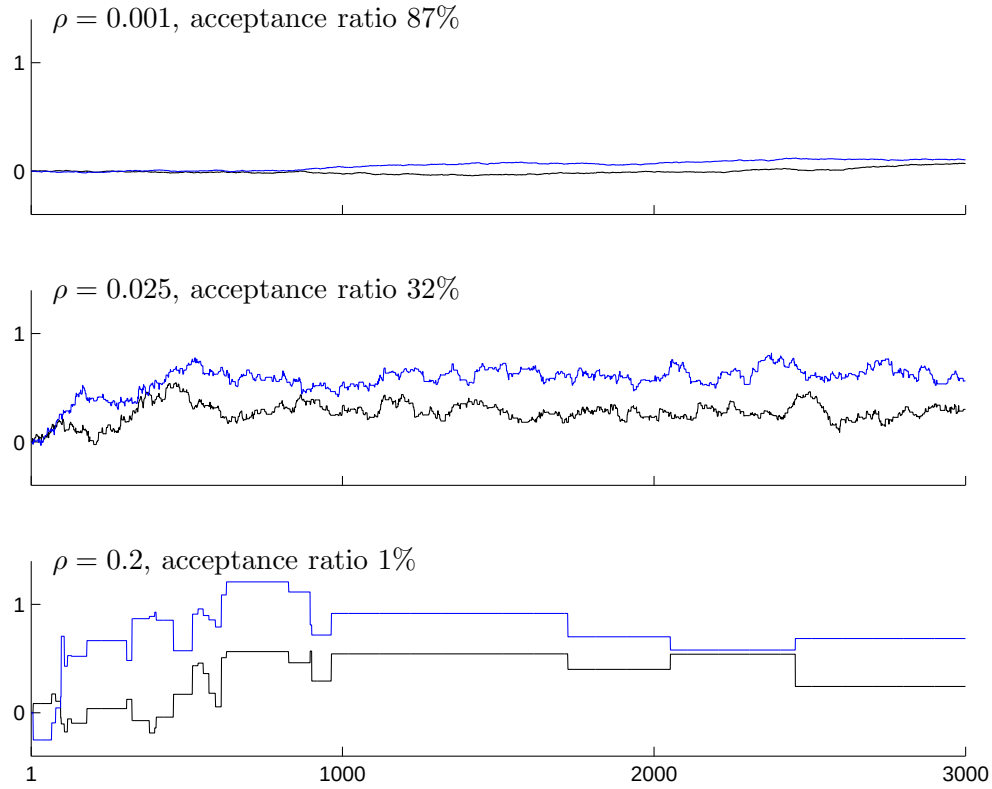


Figure 5.3: Three different choices of proposal distribution in Metropolis-Hastings algorithm: $\rho = 0.001$, $\rho = 0.025$ and $\rho = 0.2$ in formula (5.19). Plotted are components 8 (black line) and 16 (blue line) of the sample vectors. In the top plot ρ is too small and the chain moves very slowly. In the middle plot ρ is quite appropriate, whereas in the bottom plot ρ is too large, leading to the rejection of most candidates.

5.3.4 Adaptive Metropolis-Hastings method

5.4 Discretization invariance

This topic will not be discussed in the 2008 course.

5.5 Exercises

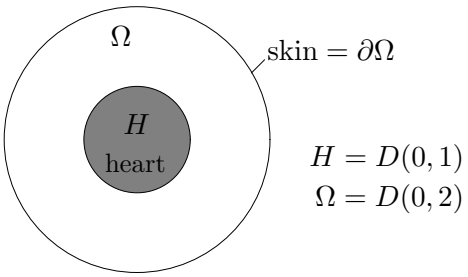
1. Prove that formula (5.23) holds if we choose $\alpha = \delta/(2\sigma^2)$.

Appendix A

Electrocardiography

This example is due to David Isaacson.

Situation:



Heart induces an electric potential on ∂H. Assume the disc $\Omega \setminus H$ is homogeneous. Then quasi-static approximation yields a boundary value problem

$$\begin{cases} \Delta u = 0 \ \text{ in } \Omega \setminus H \\ u|_{\partial H} = f \\ \frac{\partial u}{\partial \vec{n}}\Big|_{\partial\Omega} = 0 \end{cases} . \tag{A.1}$$

Assume also that the skin is totally insulated; therefore $\frac{\partial u}{\partial \vec{n}}\big|_{\partial\Omega} = 0$.

We measure on the skin the voltage $u|_{\partial\Omega}$ (in practice with electrodes; here in every point). Write

$$f = \sum_{n=-\infty}^{\infty} \langle f, \varphi_n \rangle \varphi_n, \quad \varphi_n = \frac{1}{\sqrt{2\pi}} e^{in\theta}. \quad \text{(on the edge } \partial H)$$

Let's solve (A.1) in the case $f = \varphi_n$. We know from the theory of elliptic partial differential equations, that the problem has a unique solution u_n. Write a trial solution ($n \neq 0$)

$$u_n(r, \theta) = a_n r^{|n|} e^{in\theta} + b_n r^{-|n|} e^{in\theta},$$

which is harmonic. From the boundary conditions

$$\frac{1}{\sqrt{2\pi}}e^{in\theta} = \varphi_n(\theta) = u_n(1,\theta) = (a_n + b_n)e^{in\theta}$$
$$0 = \frac{d}{dr}u_n(r,\theta)\Big|_{r=2} = \left(a_n|n|2^{|n|-1} + b_n|n|2^{-|n|-1}\right)e^{in\theta}.$$

Further

$$\begin{aligned} a_n + b_n &= \frac{1}{\sqrt{2\pi}} \\ 2^{|n|}a_n - 2^{-|n|}b_n &= 0 \\ \implies a_n = \frac{1}{\sqrt{2\pi}}\frac{1}{(1+2^{2|n|})}, &\quad b_n = \frac{1}{\sqrt{2\pi}}\frac{1}{(1+2^{-2|n|})}. \end{aligned}$$

The voltage measured on the skin is

$$u_n(2,\theta) = \left(a_n 2^{|n|} + b_n 2^{-|n|}\right)e^{in\theta}.$$

Write the functions given on $\partial\Omega$ in the Fourier basis $\psi_n = \frac{1}{\sqrt{4\pi}}e^{in\theta}$, in which case $\langle\psi_n, \psi_m\rangle = \delta_{nm}$. We obtain

$$u_n(2,\theta) = c_n\psi_n,$$

where

$$\begin{aligned} c_n &= \sqrt{4\pi}\left(a_n 2^{|n|} + b_n 2^{-|n|}\right) \\ &= \sqrt{2}\left(\frac{2^{|n|}}{1+2^{2|n|}} + \frac{2^{-|n|}}{1+2^{-2|n|}}\right) \\ &= \sqrt{2}\left(\frac{2^{|n|} + 2^{-|n|} + 2^{-|n|} + 2^{|n|}}{(1+2^{2|n|})(1+2^{-2|n|})}\right) \\ &= \sqrt{2}\left(\frac{2^{|n|+1} + 2^{-|n|+1}}{1+2^{2|n|}+2^{-2|n|}+1}\right) \\ &= \sqrt{2}\left(\frac{2^{|n|+1} + 2^{-|n|+1}}{2^{2|n|}+2^{-2|n|}+2}\right)\left(\frac{2^{-|n|-1}}{2^{-|n|-1}}\right) \\ &= \sqrt{2}\left(\frac{1+2^{-2|n|}}{2^{|n|-1}+2^{-|n|}+2^{-3|n|-1}}\right). \end{aligned}$$

The mapping from the quantity to the measurement is

$$\varphi_n \longmapsto c_n\psi_n,$$

and c_n satisfies (at least when $n \neq 0$):

$$c_n \leq \frac{2}{2^{|n|-1}} = \frac{4}{2^{|n|}}.$$

Naive reconstruction: Write the potential on the heart in the form

$$f(\theta) = \sum_{n=-\infty}^{\infty} \langle f, \varphi_n \rangle \varphi_n(\theta) = \sum_{n=-\infty}^{\infty} \hat{f}(n) \varphi_n(\theta).$$

Then the measurement is (ideally) of the form

$$u\big|_{\partial\Omega}(\theta) = \sum_{n=-\infty}^{\infty} c_n \hat{f}(n) \psi_n(\theta).$$

Now choose $N > 0$ and consider the truncated bases $\{\varphi_n\}_{n=-N}^{N}$ on ∂H and $\{\psi_n\}_{n=-N}^{N}$ on $\partial\Omega$. Define

$$x = \begin{bmatrix} \hat{f}(-N) \\ \vdots \\ \hat{f}(N) \end{bmatrix} \in \mathbb{R}^{2N+1}$$

and the measurement

$$m = \begin{bmatrix} \langle u|_{\partial\Omega}, \psi_{-N} \rangle \\ \vdots \\ \langle u|_{\partial\Omega}, \psi_N \rangle \end{bmatrix} = \begin{bmatrix} c_{-N} \hat{f}(-N) \\ \vdots \\ c_N \hat{f}(N) \end{bmatrix}.$$

Then

$$m = Ax$$

with

$$A = \begin{bmatrix} c_{-N} & & 0 \\ & \ddots & \\ 0 & & c_N \end{bmatrix}.$$

So we can make naive reconstruction:

$$x = A^{-1} m$$

and recover the voltage at the heart as

$$\sum_{n=-N}^{N} x_n \varphi_n(\theta).$$

However, this fails in the case

$$m = Ax + \varepsilon,$$

where ε_n are random numbers from $N(0, \sigma^2)$. Why?

A.1 Exercises

1. Show that the function $r^n e^{in\theta}$ is harmonic in the domain $1 \leq r \leq 2$ and $0 \leq \theta \leq 2\pi$. (Hint: use the Laplace operator in polar coordinates.)

2. The inverse problem of ECG was reduced to a simple form using Fourier series. More precisely, we study the measurement model $m = Ax + \varepsilon$ in the case when A is the diagonal matrix

$$A = \begin{bmatrix} c_{-N} & & \\ & \ddots & \\ & & c_N \end{bmatrix}, \qquad c_n = \sqrt{2}\left(\frac{1 + 2^{-2|n|}}{2^{|n|-1} + 2^{-|n|} + 2^{-3|n|-1}}\right).$$

   Why does the naive reconstruction $x \approx A^{-1}m$ fail in the presence of noise ε when N is large? (Hint: for large n we have the estimate $|c_n| \leq 2^{2-|n|}$.)

Bibliography

[1] Bal G: Lecture notes
`http://www.columbia.edu/~gb2030/COURSES/E6901/LectureNotesIP.pdf`

[2] Chadan K, Colton D, Päivärinta L and Rundell W, An introduction to inverse scattering and inverse spectral problems, SIAM 1997

[3] Colton D, Engl H W, Louis A K, McLaughlin J R and Rundell W (eds.), Surveys on solution methods for inverse problems, Springer 2000

[4] Colton D and Kress R, Integral Equation Methods in Scattering Theory, Wiley 1983.

[5] Colton D and Kress R, Inverse acoustic and electromagnetic scattering theory, Springer 1998

[6] Yu-Hong Dai and Roger Fletcher, Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming, Numer. Math. (2005) 100: 21ï£¡47

[7] Gamerman D 1997 *Markov chain Monte Carlo - Stochastic simulation for Bayesian inference*. Chapman & Hall.

[8] Gilks W R, Richardson S and Spiegelhalter D J 1996 Markov Chain Monte Carlo in Practice Chapmann & Hall

[9] Hansen P C, Rank-deficient and discrete ill-posed problems, SIAM 1998

[10] Hansen P C, Nagy J G and O'Leary D P, Deblurring images, SIAM 2006

[11] Hastings W K 1970 Monte Carlo sampling methods using Markov Chains and their applications *Biometrika* **57** 97–109

[12] Isakov V, Inverse problems for partial differential equations, Springer 1998

[13] Kaipio J and Somersalo E, Statistical and computational inverse problems, Springer 2005

[14] Katchalov A, Kurylev Y and Lassas M, Inverse boundary spectral problems, Chapman & Hall/CRC 2001

[15] Kirsch A, An introduction to the mathematical theory of inverse problems, Springer 1996

[16] Lassas M and Siltanen S, *Can one use total variation prior for edge preserving Bayesian inversion?,* Inverse Problems **20** (2004), 1537–1564.

[17] Morozov V A, Methods for solving incorrectly posed problems, Springer 1984

[18] Natterer F, The mathematics of computerized tomography, Wiley 1986

[19] Natterer F and Wübbeling F, Mathematical methods in image reconstruction, SIAM 2001

[20] Potthast R, Point sources and multipoles in inverse scattering theory, Chapman & Hall/CRC 2001

[21] Roberts G O and Smith A F M 1994 Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stoch Processes Appl*, 49:207–216.

[22] Santamarina J C and Fratta D, Discrete signals and inverse problems, Wiley 2005

[23] Shiryaev A N, Probability, Springer 1996

[24] Smith A F M and Roberts G O 1993 Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *J R Statist Soc B*, 55:3–23.

[25] Tarantola A, Inverse problem theory and methods for model parameter estimation, SIAM 2005

[26] Tierney L 1994 Markov chains for exploring posterior distributions, with discussion. *The Annals of Statistics* **22** 1701–1762

[27] Tikhonov A N and Arsenin V Y, Solutions of ill-posed problems, Winston & Sons 1977

[28] Vogel C R, Computational methods for inverse problems, SIAM 2002

[29] Wang, Y., Ma, S., Projected Barzilai-Borwein method for large-scale nonnegative image restoration, Inverse Problems in Science and Engineering, Vol. 15, No. 6, September 2007.