Inverse Problems

Rosemary Renaut

Winter 2011

Some Background on Inverse Problems
Constructing PSF Matrices
The DFT
Rosemary Renaut
October 12, 2011

- Deblurring Images Matrices Spectra and Filtering, Hansen, Nagy, and O'Leary, SIAM 2006
  http://www2.imm.dtu.dk/~pch/HNO/
- Computational Methods for Inverse Problem, Vogel, SIAM 2002.
  http://www.math.montana.edu/~vogel/Book/
- Rank Deficient and Discrete Ill-Posed Inverse Problems, Hansen, SIAM 1997
  http://www2.imm.dtu.dk/~pch/Regutools/

Illustrative Example: **Fredholm first kind integral equation**

$$g(x) = \int_0^1 h(x, x')f(x')dx', \quad 0 < x, x' < 1.$$

This is a one dimensional example of signal blur - similar to the two dimensional image blur problem.

- f is the light **source** intensity
- g is the **measured** image intensity
- h is the **kernel** characterizing blurring effects.
- When $h(x, x') = h(x - x')$. The kernel is **spatially invariant.**
- Typical choice of $h(x)$ in 1D
  - **Gaussian** $h(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(\frac{-x^2}{2\sigma^2}), \sigma > 0.$
  - **Out of focus** $h(x) = \begin{cases} C & l \leq x \leq u \\ 0 & otherwise. \end{cases}$

Forward Problem: Given f, h calculate g

- Suppose that the integral is approximated using numerical quadrature yielding a matrix A
- Let $dx = 1/n$, $x_i = idx$, $1 \leq i \leq n$. Then

$$A_{ij} = w_j h_{ij} = w_j h(x_i, x_j) \text{ or } w_j h((i-j)dx),$$

  where w_j is the weight for the quadrature, e.g. $w_j = dx$.
- f is sampled at $f(x_j)$, $g(x)$ is sampled at x_i.
- Matrix equation

$$g = Af$$

  describes the linear relationship
- Requirement $\int_0^1 h(x)dx$ (for PSF no energy is introduced, the signal is spread) leads to the normalization $\sum_j w_j h(x_j) = 1$.

Example of two Blurs, 1D

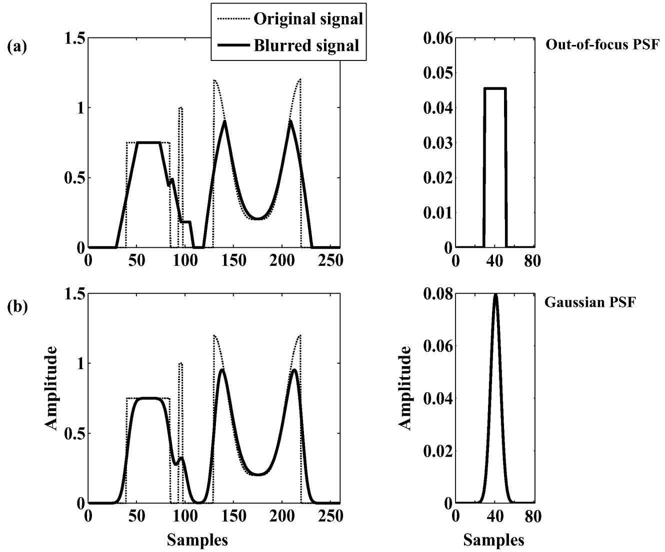
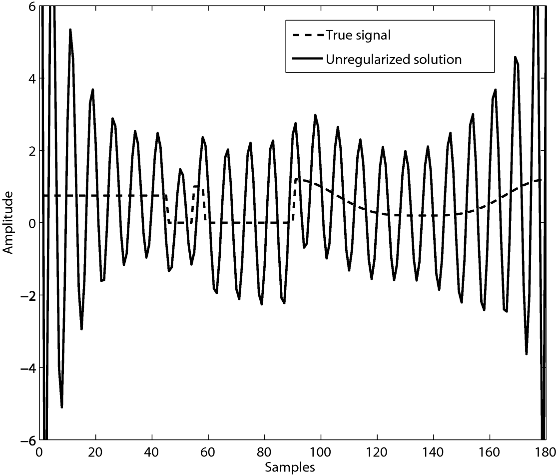


Figure: Notice that without restoration of the data it will be difficult to extract image features

Inverse Problem solve $g \approx Af$ when the problem ill-posed.

The solution depends on the conditioning of A and on the noise in measurements g. Reconstruction with noise $\mathbb{N}(0, 10^{-7})$



- The model is the same just extended to two dimensions:

$$G(x,y)_{\text{exact}} = \int\int H(x,x',y,y')X(x',y')dx'dy'$$

- $H(x,x',y,y')$ is often smooth, **Impulse Response** or **Point Spread** Function of imaging system.
- $X(x,y)$ is the original exact image, or light source.
- When H is invariant in space **isotropic**

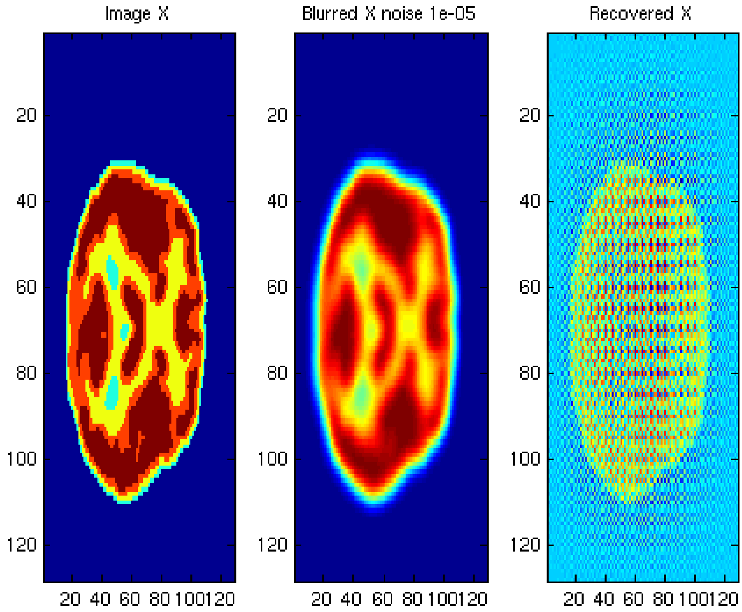
$$H(x,x',y,y') = H(x-x',y-y')$$

- H might represent a projection as for example with PET images.
- Typically a noisy image is obtained

$$G(x,y) = \int\int H(x,x',y,y')X(x',y')dx'dy' + E(x,y)$$

- Often the noise distribution $E(x,y)$ is known. Distribution of E should be used in finding a solution.

Example of restoration of 2D image with noise



Actual PET Images are Obtained from Projected Data

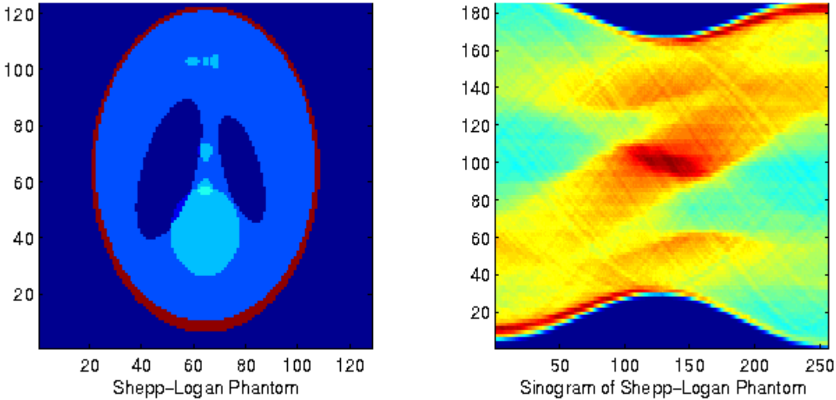
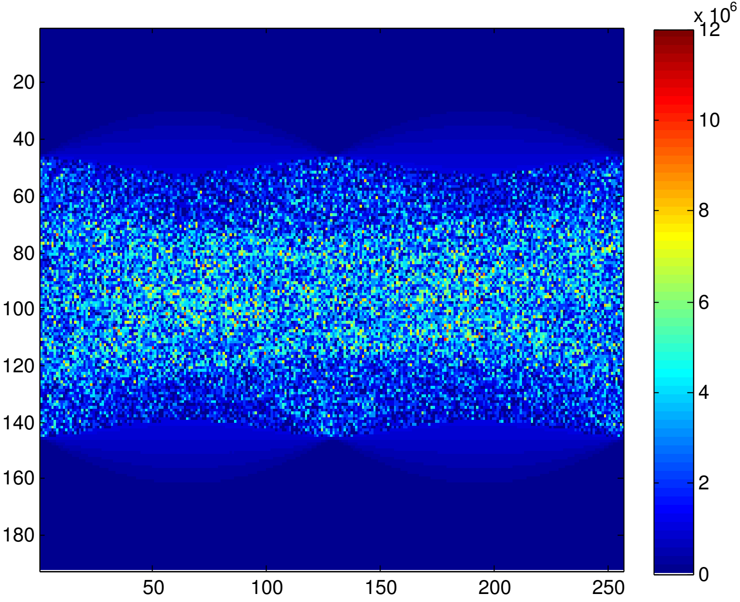
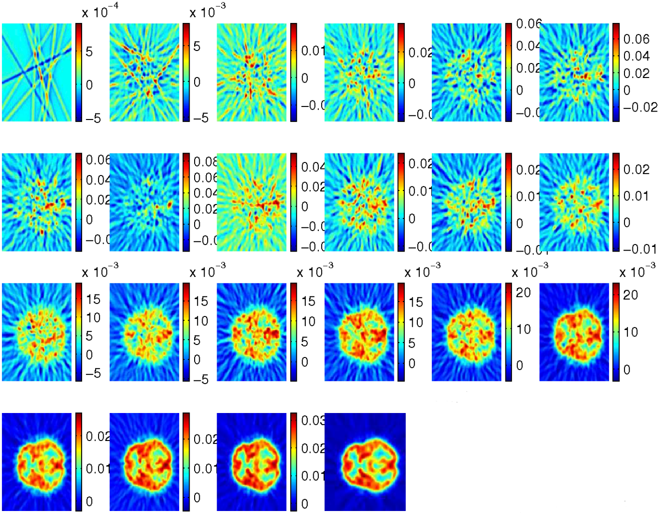


Figure: Sinogram Data

Noisy Sinogram



Typical PET Frames



1D Formulation for the Convolution

Blurred Signal

$$g(x) = \int_{-\infty}^{\infty} h(x - x')f(x')dx'$$

Limits Take $h(x) = 0$ for $|x| > r$ then integration is limited

$$g(x) = \int_{x-r}^{x+r} h(x - x')f(x')dx'$$

Description

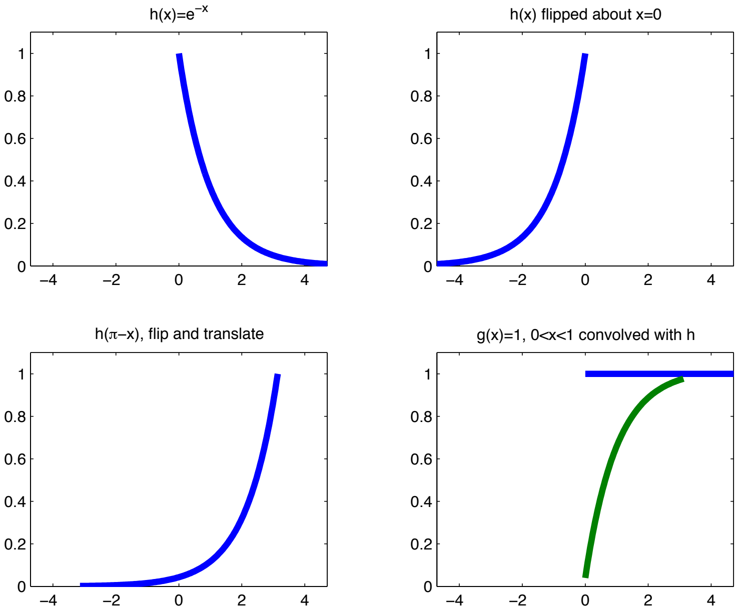
1. Flip the PSF function to obtain $h(-x')$.
2. Shift by x to give $h(x - x')$.
3. Then we see that g is a weighted local average of f with weights from h.

Discrete

$$g_i = \sum_j h_{i-j} f_j$$

assuming normalization to ignore constant weighting in PSF h

Flip and Translate



- Assume $\mathbf{f} \in \mathbb{R}^n$, $\mathbf{h} \in \mathbb{R}^m$ and $n > m$.
- Want coefficients

$$g_i = (\mathbf{f} * \mathbf{h})_i = \sum_{j \in \Omega(i)} f_j h_{i-j+1}$$

  $\Omega(i)$ is the set of integers for the sum and may depend on i, depending on how the extent of h outside the defined range of the signal is implemented.
- Equivalently we suppose that f is given only for a limited range, a *Full* convolution will require elements of f beyond the defined range.
- Elements of $\mathbf{f}$ and $\mathbf{h}$ are only defined for positive index.

The equations

$$\begin{aligned}
g_1 &= h_1 f_1 \\
g_2 &= h_2 f_1 + h_1 f_2 \\
\vdots &= \vdots \\
g_m &= h_m f_1 + h_{m-1} f_2 + \ldots h_1 f_m \\
\vdots &= \vdots \\
g_n &= \ldots\ldots\ldots h_m f_{n-m+1} + h_{m-1} f_{n-m+2} + \ldots \qquad h_1 f_n \\
\vdots &= \vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
g_{m+n-1} &= \ldots \qquad\qquad\qquad\qquad\qquad\qquad\qquad h_m f_n
\end{aligned}$$

Notice that $|\Omega(i)| = m$, $i = n : m$.

$$\Omega(i) = \{\max(1, i + 1 - m), \ldots, \min(i, n)\}$$

For other i not all values of $\mathbf{h}$ because relevant indices of $\mathbf{f}$ are not defined.

Matrix Formulation

$$
\begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \\ \hline g_m \\ g_{m+1} \\ \vdots \\ g_n \\ \hline \vdots \\ g_{m+n-1} \end{pmatrix}
=
\begin{pmatrix}
h_1 & & & & & & & & \\
h_2 & h_1 & & & & & & & \\
h_3 & h_2 & h_1 & & & & & & \\
\vdots & \vdots & \vdots & \vdots & & & & & \\ \hline
h_m & h_{m-1} & \dots & & h_1 & & & & \\
 & h_m & h_{m-1} & \dots & & h_1 & & & \\
 & & \vdots & & & & & & \\
 & & & h_m & h_{m-1} & \dots & \dots & h_1 \\ \hline
 & & & & & \vdots & & \\
 & & & & & & & h_m
\end{pmatrix}
\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_m \\ f_{m+1} \\ \vdots \\ f_n \end{pmatrix}
$$

- Notice that the middle block contains the terms which can be calculated using all values of h.
- In this formulation the lack of use of h amounts to padding f by zeros outside the valid domain.

The complete Toeplitz System

$$\begin{pmatrix} g_1 \\ \vdots \\ g_m \\ \vdots \\ g_n \\ \vdots \\ g_{m+n-1} \end{pmatrix} = \begin{pmatrix} h_m & h_{m-1} & \dots & h_1 & & & & \\ & h_m & h_{m-1} & \dots & h_1 & & & \\ & & \vdots & & & & & \\ & & & & h_m & h_{m-1} & \dots & h_1 \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ f_1 \\ \vdots \\ f_n \\ y_1 \\ \vdots \end{pmatrix}$$

- w_l, y_l correspond to padding of f_i. Total number of padded entries is $m-1$. $\mathbf{f} \in \mathbb{R}^{m+n-1}$. Pad $\mathbf{h}$ to length $m+n-1$ similarly. (Note that for only n entries for g, use the middle n entries.)
- $w_l = y_l = 0$ corresponds to using zero boundary conditions on f.

- Discrete convolution of two equal length sequences is accomplished by **point wise** multiplication in the Fourier domain.
- Suppose $\mathbf{f}$ is a sequence of length n with components f_j. The unitary DFT of $\mathbf{f}$ is $\hat{\mathbf{f}}$ with component k

$$\hat{f}_k = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} f_j e^{\frac{-2\pi i(j-1)(k-1)}{n}}$$

- The inverse DFT is given by

$$f_j = \frac{1}{\sqrt{n}} \sum_{k=1}^{n} \hat{f}_k e^{\frac{2\pi i(j-1)(k-1)}{n}}$$

- Notice that the sequences are periodic with period n: $f_{j+n} = f_j$.
- Extending to length $m + n - 1$ can be done by periodicity, or reflection at the boundaries.

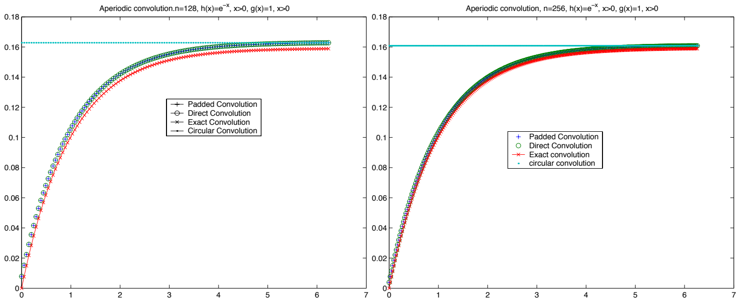


Figure: Comparison of Convolution With Different Boundary Conditions

Fourier transform as a Matrix Operator

It may be useful to note that the Fourier transform can be implemented (thought of) as a matrix operation:

$$\hat{\mathbf{f}} = F\mathbf{f}$$

$$F_{kj} = \frac{1}{\sqrt{n}} e^{\frac{-2\pi i(k-1)(j-1)}{n}}$$

$$\mathbf{f} = F^{-1}\hat{\mathbf{f}} = F^{*}\mathbf{f}$$

$$F^{*}_{kj} = \frac{1}{\sqrt{n}} e^{\frac{2\pi i(k-1)(j-1)}{n}}$$

using the fact that the matrix F is unitary, $F^{*}F = I_n$.

Summary Implementation of the 1D Convolution

$$g_i = (\mathbf{f} * \mathbf{h})_i = \sum_{j \in \Omega(i)} \mathbf{f}_j \mathbf{h}_{i-j+1}, \quad i = 1 : m+n-1$$

1. Pad sequences with zeros to length $m+n-1$. Further pad to length power of two.
2. Transform both sequences to frequency domain by DFT
3. Multiply DFTS pointwise

$$\hat{g}_i = \hat{f}_i \hat{h}_i$$

4. Inverse transform $\hat{\mathbf{g}}$ to yield $\mathbf{g}$ sampled at $x_i, i = 1 : m+n-1$.
5. For real data ignore imaginary part of FFT which occurs due to floating point accuracy.
6. Extract needed components of g.

Other Boundary Conditions : Periodic on f

Suppose that the original values of f are assumed to be periodic. Then we fill in the entries in the Toeplitz matrix:

$$
\left(\begin{array}{c}
\vdots \\
g_{m-2} \\
g_{m-1} \\ \hline
g_m \\
g_{m+1} \\
\vdots \\
g_n \\ \hline
g_{n+1} \\
g_{n+2} \\
\vdots
\end{array}\right)
=
\left(\begin{array}{cccccccc}
\vdots & \vdots & \vdots & \vdots & & & & \\
h_{m-2} & h_{m-3} & \dots & \dots & & & h_m & h_{m-1} \\
h_{m-1} & h_{m-2} & \dots & h_1 & 0 & \dots & 0 & h_m \\ \hline
h_m & h_{m-1} & \dots & & h_1 & & & \\
 & h_m & h_{m-1} & \dots & & h_1 & & \\
 & & \vdots & & & & & \\
 & & & h_m & h_{m-1} & \dots & \dots & h_1 \\ \hline
h_1 & 0 & \dots & 0 & h_m & h_{m-1} & \dots & h_2 \\
h_2 & h_1 & 0 & \dots & 0 & h_m & \dots & h_3 \\
\vdots & \vdots & \vdots & & & & & \vdots
\end{array}\right)
\left(\begin{array}{c}
f_1 \\
f_2 \\
f_3 \\
\vdots \\
f_m \\
f_{m+1} \\
\vdots \\
f_n
\end{array}\right)
$$

This is called a **CIRCULANT** matrix. Each row (column) is a periodic shift of previous row (column).

Assume that the function f is reflected at the boundary. For example look at the equations for g_{m-1} and g_{n+1}

$$\begin{aligned} g_{m-1} &= h_m f_1 + h_{m-1} f_1 + h_{m-2} f_2 + \ldots h_1 f_m \\ g_{n+1} &= h_m f_{n-m+2} + h_{m-1} f_{n-m+3} + \ldots h_2 f_n + h_1 f_n \end{aligned}$$

yielding

$$\begin{aligned} g_{m-1} &= (h_{m-1} + h_m) f_1 + h_{m-2} f_2 + \cdots + h_1 f_m \\ g_{n+1} &= h_m f_{n-m+2} + h_{m-1} f_{n-m+3} + \cdots + (h_2 + h_1) f_n \end{aligned}$$

A matrix which has constant entries on each anti diagonal is a Hankel matrix. Hence the resulting matrix is Toeplitz+Hankel.

Extension to 2D : In Matrix Notation

- Let $\mathbf{x} = \mathrm{vec}(X)$, $\mathbf{b} = \mathrm{vec}(G)$, be the matrices X, G with entries stacked as a vector. i.e. we take the entries of $X \in \mathbb{R}^{m \times n}$ column wise and set $N = mn$,

$$\mathrm{vec}(X) = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} \in \mathbb{R}^N$$

- Then the convolution operator is rewritten as

$$A\mathbf{x} = \mathbf{b}$$

  where A represents the PSF H, and entries of A depend on how this is implemented.
- Condition of the problem can still be investigated naïvely for the linear system. This requires the generation of the matrix A for a two dimensional problem.

Descriptive Version of 2D Convolution

1. Regard the PSF as a mask with elements in an array.
2. Rotate the PSF array by 180°.
3. To compute convolution for a certain pixel we have to line up the central point of the rotated mask on that pixel
4. Multiply the corresponding elements within the mask
5. Sum the elements within the mask
6. Take account of boundary conditions as for the 1D Case.

Simple Example

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \; H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \; G = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}$$

Rotated H, align with x_{22} and multiply

$$Hflip = \begin{pmatrix} h_{33} & h_{32} & h_{31} \\ h_{23} & h_{22} & h_{21} \\ h_{13} & h_{12} & h_{11} \end{pmatrix} \quad X.*Hflip = \begin{pmatrix} x_{11}h_{33} & x_{12}h_{32} & x_{13}h_{31} \\ x_{21}h_{23} & x_{22}h_{22} & x_{23}h_{21} \\ x_{31}h_{13} & x_{32}h_{12} & x_{33}h_{11} \end{pmatrix}$$

where $X.*H$ denotes element wise multiplication. We then sum all the entries. In the Vec notation we have

$$g_{22} = \mathrm{vec}(Hflip)^T \mathrm{vec}(X)$$

as an inner product. All entries are obtained as inner products.

Calculating all entries for the case of zero boundary conditions

$$\begin{pmatrix} g_{11} \\ g_{21} \\ g_{31} \\ g_{12} \\ g_{22} \\ g_{32} \\ g_{13} \\ g_{23} \\ g_{33} \end{pmatrix} = \left(\begin{array}{ccc|ccc|ccc} h_{22} & h_{12} & & h_{21} & h_{11} & & & & \\ h_{32} & h_{22} & h_{12} & h_{31} & h_{21} & h_{11} & & & \\ & h_{32} & h_{22} & & h_{31} & h_{21} & & & \\ \hline h_{23} & h_{13} & & h_{22} & h_{12} & & h_{21} & h_{11} & \\ h_{33} & h_{23} & h_{13} & h_{32} & h_{22} & h_{12} & h_{31} & h_{21} & h_{11} \\ & h_{33} & h_{23} & & h_{32} & h_{22} & & h_{31} & h_{21} \\ \hline & & & h_{23} & h_{13} & & h_{22} & h_{12} & \\ & & & h_{33} & h_{23} & h_{13} & h_{32} & h_{22} & h_{12} \\ & & & & h_{33} & h_{23} & & h_{32} & h_{22} \end{array}\right) \begin{pmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{12} \\ x_{22} \\ x_{32} \\ x_{13} \\ x_{23} \\ x_{33} \end{pmatrix}$$

Note that each block is of Toeplitz form and that as a block matrix the matrix is also Toeplitz.
Matrix is **block Toeplitz with Toeplitz Blocks** BTTB.

Constructing matrix A: Boundary Conditions

PSF is usually of finite extent, but for a finite section of an image, convolution with the PSF uses values outside the specific image. If this is not handled correctly, boundary effects will contaminate the image, leading to artifacts. This is true whether for the forward or inverse model. **Suggestions**

- Zero Padding: Embed the image in a larger image, with sufficient zeros to handle the extent of the PSF.

O	O	O
O	X	O
O	O	O

- Introduces edges, discontinuities, leads to ringing and artificial black borders.

Matrix is **block Toeplitz with Toeplitz Blocks** BTTB.

Constructing matrix A

- Periodicity: Embed the image periodically in a larger image.

X	X	X
X	X	X
X	X	X

- Introduces edges, discontinuities, leads to ringing and artificial black borders. Matrix is **block Circulant with Circulant Blocks** BCCB. Matrix is a sum of BTTB, BTHB,BHTB, BHHB
  H for a Hankel block.

- Periodicity: Embed the image periodically in a larger image.

X	X	X
X	X	X
X	X	X

- Introduces edges, discontinuities, leads to ringing and artificial black borders. Matrix is **block Circulant with Circulant Blocks** BCCB.
- Reflexive: Reflect and embed.

X_{lrud}	X_{ud}	X_{lrud}
X_{lr}	X	X_{lr}
X_{lrud}	X_{ud}	X_{lrud}

- Obvious reflections about the central block: no edges
  Matrix is a sum of BTTB, BTHB,BHTB, BHHB
  H for a Hankel block.

Separable PSF: $A = \mathbf{c}\mathbf{r}^T$

- $\mathbf{c}$ coefficients of blur across columns of the image
- $\mathbf{r}$ coefficients of blur across rows of the image
- $\mathbf{c} = (c_1, c_2, \ldots, c_m)^T$. $\mathbf{r} = (r_1, r_2, \ldots, r_n)^T$.
- The PSF matrix A is a rank one matrix. $A_{kj} = c_k r_j$.
- $$A = \begin{pmatrix} c_1 r_1 & c_1 r_2 & c_1 r_3 \\ c_2 r_1 & c_2 r_2 & c_2 r_3 \\ c_3 r_1 & c_3 r_2 & c_3 r_3 \end{pmatrix}$$
- Then let

$$A_r = \begin{pmatrix} r_2 & r_1 & \\ r_3 & r_2 & r_1 \\ & r_3 & r_2 \end{pmatrix} \quad A_c = \begin{pmatrix} c_2 & c_1 & \\ c_3 & c_2 & c_1 \\ & c_3 & c_2 \end{pmatrix}$$

- A_r and A_c inherit the structure of the 1D matrices dependent on how boundary conditions are applied.
- We obtain the overall blur matrix $A = A_r \otimes A_c$.

Kronecker Product

- To obtain the two dimensional PSF matrix A which acts on the image in vec form we map A using its special structure.

$$
\begin{aligned}
A_r \otimes A_c &= \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \cdot & \cdot & a_{mn} \end{bmatrix} \otimes A_c \\
&= \begin{bmatrix} a_{11}A_c & a_{12}A_c & \cdot & \cdot & a_{1n}A_c \\ a_{21}A_c & a_{22}A_c & \cdot & \cdot & a_{2n}A_c \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1}A_c & a_{m2}A_c & \cdot & \cdot & a_{mn}A_c \end{bmatrix}
\end{aligned}
$$

Evaluating using the Kronecker Product

- Assume $A_c \in \mathcal{R}^{m\times m}$ and $A_r \in \mathcal{R}^{n\times n}$, for an image of size $m \times n$. Then, for the exact blurred image,

$$G = A_c X A_r^T.$$

- Left multiplication by A_c applies PSF to each column of X, is vertical blur.
- Left multiplication of X^T by A_r applies PSF to each column of X^T, which are rows of X. Hence the horizontal blur is applied as $(A_r X^T)^T = X A_r^T$
- Let $A = A_r \otimes A_c$ be the Kronecker product of size $mn \times mn$, the ij block of A is $(A_r)_{ij} A_c$.
- Then $G = A_c X A_r^T$ in vector form is $\mathbf{b} = A\mathbf{x}$.

Evaluating using the Kronecker Product

- Assume $A_c \in \mathcal{R}^{m\times m}$ and $A_r \in \mathcal{R}^{n\times n}$, for an image of size $m \times n$. Then, for the exact blurred image,
$$G = A_c X A_r^T.$$
- Left multiplication by A_c applies PSF to each column of X, is vertical blur.
- Left multiplication of X^T by A_r applies PSF to each column of X^T, which are rows of X. Hence the horizontal blur is applied as $(A_r X^T)^T = X A_r^T$
- Let $A = A_r \otimes A_c$ be the Kronecker product of size $mn \times mn$, the ij block of A is $(A_r)_{ij} A_c$.
- Then $G = A_c X A_r^T$ in vector form is $\mathbf{b} = A\mathbf{x}$.

Evaluating using the Kronecker Product

- Assume $A_c \in \mathcal{R}^{m \times m}$ and $A_r \in \mathcal{R}^{n \times n}$, for an image of size $m \times n$. Then, for the exact blurred image,

$$G = A_c X A_r^T.$$

- Left multiplication by A_c applies PSF to each column of X, is vertical blur.
- Left multiplication of X^T by A_r applies PSF to each column of X^T, which are rows of X. Hence the horizontal blur is applied as $(A_r X^T)^T = X A_r^T$
- Let $A = A_r \otimes A_c$ be the Kronecker product of size $mn \times mn$, the ij block of A is $(A_r)_{ij} A_c$.
- Then $G = A_c X A_r^T$ in vector form is $\mathbf{b} = A\mathbf{x}$.

Evaluating using the Kronecker Product

- Assume $A_c \in \mathcal{R}^{m\times m}$ and $A_r \in \mathcal{R}^{n\times n}$, for an image of size $m \times n$. Then, for the exact blurred image,
$$G = A_c X A_r^T.$$
- Left multiplication by A_c applies PSF to each column of X, is vertical blur.
- Left multiplication of X^T by A_r applies PSF to each column of X^T, which are rows of X. Hence the horizontal blur is applied as $(A_r X^T)^T = X A_r^T$
- Let $A = A_r \otimes A_c$ be the Kronecker product of size $mn \times mn$, the ij block of A is $(A_r)_{ij} A_c$.
- Then $G = A_c X A_r^T$ in vector form is $\mathbf{b} = A\mathbf{x}$.

Evaluating using the Kronecker Product

- Assume $A_c \in \mathcal{R}^{m\times m}$ and $A_r \in \mathcal{R}^{n\times n}$, for an image of size $m \times n$. Then, for the exact blurred image,
$$G = A_c X A_r^T.$$
- Left multiplication by A_c applies PSF to each column of X, is vertical blur.
- Left multiplication of X^T by A_r applies PSF to each column of X^T, which are rows of X. Hence the horizontal blur is applied as $(A_r X^T)^T = XA_r^T$
- Let $A = A_r \otimes A_c$ be the Kronecker product of size $mn \times mn$, the ij block of A is $(A_r)_{ij}A_c$.
- Then $G = A_c X A_r^T$ in vector form is $\mathbf{b} = A\mathbf{x}$.

Typically we only see a finite region of an image that extends in all directions. We have to be careful in constructing the matrix A.

Typically we only see a finite region of an image that extends in all directions. We have to be careful in constructing the matrix A.

Remark

The blurring matrix A depends on two ingredients. The PSF defines the blur. But the boundary conditions imposed on the image, and hence on A, specify assumptions on the scene outside the image, which then impact the reconstruction of the image.

Typically we only see a finite region of an image that extends in all directions. We have to be careful in constructing the matrix A.

Remark

The blurring matrix A depends on two ingredients. The PSF defines the blur. But the boundary conditions imposed on the image, and hence on A, specify assumptions on the scene outside the image, which then impact the reconstruction of the image.

Remark

Practically, we look for algorithms that do not calculate A.

Constructing Ax for structured A
Spectral decompositions
Solutions in terms of spectral decompositions
Rosemary Renaut
October 19, 2011

Definition

Toeplitz A is circulant if rows / columns are circular shifts.

$$A = (a_{ij}) = (h_{i-j+1}) = \begin{pmatrix} h_1 & h_0 & \dots & h_{-(n-3)} & h_{-(n-2)} \\ h_2 & h_1 & \dots & h_{-(n-4)} & h_{-(n-3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h_{n-1} & \dots & h_2 & h_1 & h_0 \\ h_n & h_{n-1} & \dots & h_2 & h_1 \end{pmatrix}$$

- Write $A = \textbf{toeplitz}(h^{\text{ext}})$ where h^{ext} is the extension of h of length $2n-1$

$$h^{\text{ext}} = (h_{-(n-2)}, h_{-(n-3)}, \dots, h_0, h_1, h_2, \dots, h_n)$$

- For circulant it is periodic with period n and

$$A = \textbf{circulant}(h_1, h_2, \dots h_n)$$

- Important : h is the first column of A.

Relating circulant matrices and the DFT

Definition (Circulant Right Shift Matrix)

$$R = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix} \quad \text{or} \quad R_{kj} = \delta([j-k+1]_n)$$

Notation $\delta(j) = 1$ only for $j = 0$.

- Note right shift property $R(h_1, h_2, h_3, \dots, h_n)^T = (h_n, h_1, \dots, h_{n-1})^T$
- Induction: $R^j(h_1, h_2, h_3, \dots, h_n)^T = (h_{n-j+1}, h_{n-j+2}, \dots, h_{n-j})^T$

$$R^j = \left(\begin{array}{ccccccc} 0 & 0 & \dots & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \dots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & 1 \\ 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & \ddots & \dots & \dots & \dots & \dots \\ \vdots & \ddots & \ddots & \ddots & \dots & \dots & \dots \end{array}\right) \begin{array}{l} \\ \\ \\ \text{Row } j \\ \\ \\ \\ \end{array}$$

Expressing the Circulant Matrix

$$A = \begin{pmatrix} h_1 & h_n & \dots & h_3 & h_2 \\ h_2 & h_1 & \dots & h_4 & h_3 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h_{n-1} & \dots & h_2 & h_1 & h_n \\ h_n & h_{n-1} & \dots & h_2 & h_1 \end{pmatrix} = \sum_{j=1}^{n} h_{j+1} R^j \text{ (periodicity } h_{n+1} = h_1)$$

Let $\omega = \exp(-2\pi i/n)$, then $\sqrt{n}F_{kj} = \omega^{(k-1)(j-1)}$, $\sqrt{n}F^*_{kj} = \bar{\omega}^{(k-1)(j-1)}$ and use

$$\frac{1}{n}\sum_{j=1}^{n} \omega^{(j-1)(k-1)} = \delta([k-1]_n), \quad F^*F = FF^* = I_n.$$

Then it is immediate to show

$$\begin{aligned} (i) \quad & R = F^* \mathrm{diag}(1, \omega, \omega^2, \dots, \omega^{n-1})F = F^* \mathrm{diag}(\omega^{l-1})F \\ & RF^*_l = \omega^{l-1} F^*_l, \quad l = 1:n \\ (ii) \quad & R^j = F^* \mathrm{diag}(\omega^{l-1})^j F \end{aligned}$$

Here F^*_l is the l^{th} column of F^*.

The Spectral Decomposition

We apply the spectral decomposition results for R to show

$$\begin{aligned}
A &= \sum_{j=1}^{n} h_{j+1} R^{j} \\
&= F^{*}\mathrm{diag}\Big(\sum_{j=1}^{n} h_{j+1}\omega^{(l-1)j}\Big)F \\
&= \sqrt{n}F^{*}\mathrm{diag}(\hat{h})F \\
&= F^{*}\Lambda F, \quad \Lambda = \sqrt{n}\,\mathrm{diag}(\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_n)
\end{aligned}$$

$(\sqrt{n}\hat{h}_l, F_l^{*})$ are the eigenvalue - eigenvector pairs of A.

- To find eigenvalues of A observe

$$\begin{aligned}
FA &= \Lambda F \quad \text{and} \\
F_1 &= (1/\sqrt{n})(1, 1, \ldots, 1)^{T} \quad \text{hence} \\
(\sqrt{n}FA_1)_l &= \lambda_l
\end{aligned}$$

Computing $A\mathbf{x}$

- If the system is nonsingular
$$A^{-1} = F^* \Lambda^{-1} F$$
- Note also that A is normal
$AA^* = F^* \Lambda F F^* \Lambda F = F^* (\Lambda)^2 F = A^* A$.
- Ax is calculated via
$$\begin{aligned} Ax &= F^* \mathrm{diag}(n\hat{h}) F x, & \text{transform} \quad Fx = \hat{x} \\ &= \sqrt{n} F^* (\hat{h}. * \hat{x}) & \text{convolution} \quad \hat{y} = \hat{h}. * \hat{x} \\ &= \sqrt{n}\, y & \text{inverse} \quad y = F^* \hat{y} \end{aligned}$$
- Practically if Λ is known we write $Ax = y$ where $y = F^*(\hat{\lambda}. * \hat{x})$.
- To calculate the product $A\mathbf{x}$ when A is Toeplitz, embed in a larger circulant matrix (see Vogel chap 5).
- A similar spectral decomposition is obtained.

The Two Dimensional BCCB Case : Simple

Recall the earlier case with 3×3 PSF

$$\left(\begin{array}{ccc|ccc|ccc}
h_{22} & h_{12} & h_{32} & h_{21} & h_{11} & h_{31} & h_{23} & h_{13} & h_{33} \\
h_{32} & h_{22} & h_{12} & h_{31} & h_{21} & h_{11} & h_{33} & h_{23} & h_{13} \\
h_{12} & h_{32} & h_{22} & h_{11} & h_{31} & h_{21} & h_{13} & h_{33} & h_{23} \\
\hline
h_{23} & h_{13} & h_{33} & h_{22} & h_{12} & h_{32} & h_{21} & h_{11} & h_{31} \\
h_{33} & h_{23} & h_{13} & h_{32} & h_{22} & h_{12} & h_{31} & h_{21} & h_{11} \\
h_{13} & h_{33} & h_{23} & h_{12} & h_{32} & h_{22} & h_{11} & h_{31} & h_{21} \\
\hline
h_{21} & h_{11} & h_{31} & h_{23} & h_{13} & h_{33} & h_{22} & h_{12} & h_{32} \\
h_{31} & h_{21} & h_{11} & h_{33} & h_{23} & h_{13} & h_{32} & h_{22} & h_{12} \\
h_{11} & h_{31} & h_{21} & h_{13} & h_{33} & h_{23} & h_{12} & h_{32} & h_{22}
\end{array}\right) = \left(\begin{array}{c|c|c}
H_2 & H_1 & H_3 \\
\hline
H_3 & H_2 & H_1 \\
\hline
H_1 & H_3 & H_2
\end{array}\right)$$

$$\begin{aligned}
H_1 &= \textbf{circulant}(h_{11}, h_{21}, h_{31}) = \textbf{circulant}(h_{.,1}) \\
H_2 &= \textbf{circulant}(h_{12}, h_{22}, h_{32}) = \textbf{circulant}(h_{.,2}) \\
H_3 &= \textbf{circulant}(h_{13}, h_{23}, h_{33}) = \textbf{circulant}(h_{.,3})
\end{aligned}$$

and the entire PSF can be obtained from the first column of A.

The general case for BCCB

$$A(h) = \begin{pmatrix} H_2 & H_1 & \dots & H_4 & H_3 \\ H_3 & H_2 & H_1 & \dots & H_4 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ H_{n_y} & \vdots & H_3 & H_2 & H_1 \\ H_1 & H_{n_y} & \dots & H_3 & H_2 \end{pmatrix}$$

- Each block is circulant, $H_j = \textbf{circulant}(h_{\cdot,j})$, h of size $n_x \times n_y$.
- A is of size $n_x n_y \times n_x n_y$, $n = n_x n_y$. The first column defines h.
- Again A is normal and we have a spectral decomposition.

$$A = F^* \mathrm{diag}(\sqrt{n}\,\mathrm{vec}(\hat{h}))F = F^* \Lambda F$$

Here $F = F_r \otimes F_c$ and $\hat{h}$ is the two dimensional FFT of h of size $n_x \times n_y$. $\hat{h} = F(h) = (F_r \otimes F_c.)h$.
- Calculation scheme: $A\,\mathrm{vec}(X) = \mathrm{vec}(F^*(\hat{\lambda}. * \mathrm{vec}(FX)))$.

Calculating the Spectral Decomposition Λ for 2D BCCB

- As for the 1D case to find eigenvalues of A observe $FA = \Lambda F$ and $F_1 = (1/\sqrt{n})(1, 1, \ldots, 1)^T$, (look at the Kronecker product). Hence $(\sqrt{n}FA_1)_l = \lambda_l$.
- For the BCCB first column of A gives on h.
- Suppose center point of h in spatial domain is at index $\text{center} = (r_0, c_0)$
- Matlab can be used to find this column A_1 and F acting on A_1 is accomplished as the 2D transform

  $$\texttt{fft2}(\texttt{circshift}(h, 1 - \text{center})).$$

  i.e. `circshift` performs necessary circular shift of the PSF matrix.
- Note that `fft2` in Matlab implements $\sqrt{n}F$ but also `ifft2` implements $1/\sqrt{n}F^{-1}$.

Spectral Decomposition Λ for Reflexive : Hansen (2010)

- Assume underlying Toeplitz matrix is symmetric, i.e. the PSF is doubly symmetric about the central point:

$$\texttt{fliplr}(h) = \texttt{flipud}(h) = h$$

  The `flip` operators in Matlab flip columns(rows) of array.
- As 1D **circulant** use expansion: $A = \sum_{j=1}^{n} h_j S_j$ for basis matrices S_j

$$(S_j)_{lk} = 1 \quad |l-k| = j-1 \quad l+k=j \quad l+k = 2n+2-j$$

- To find the decomposition use the discrete cosine transform (DCT)
- dct vectors $\mathbf{w}_l$, $l = 1:n$ are eigenvectors of each of basis matrices S_j.

$$\mathbf{w}_l = c_l(\cos((l-1)\theta_1), \cos((l-1)\theta_2), \ldots, \cos((l-1)\theta_n))^T, \; l = 1:n$$

$$\theta_k = \frac{(2k-1)\pi}{2n}, \; k = 1:n, \quad c_l = \sqrt{\frac{2}{n}} \quad l > 1, c_1 = \sqrt{\frac{1}{n}}$$

- Matrix W_n defined with columns $\mathbf{w}_l$ is orthogonal.
- Using the eigenvector decomposition for each basis matrix, the spectrum for A can be obtained.

$$A_1 = A\mathbf{e}_1 = W_n \Lambda W_n^T \mathbf{e}_1 \leftrightarrow W_n^T A_1 = \Lambda W_n^T \mathbf{e}_1$$

- Eigenvalues are $\lambda_l = [\mathrm{dct}(A_1)]_l / [\mathrm{dct}(\mathbf{e}_1)]_l$ because $\mathrm{dct}(\mathbf{x}) = W_n^T \mathbf{x}$.

DCT analysis extends for the 2D case with reflexive BCs

$$A = W^T \Lambda W$$

where W is the orthogonal two-dimensional DCT matrix

- 2D DCT is calculated for array X by computing DCT of each column, followed by DCT of each row.
- Inverse is calculated equivalently $A = W^T \Lambda^{-1} W$
- Use Matlab functions `dct2` and `idct2` in the image processing toolbox. Alternatively, toolbox for the HNO book!
- The spectrum is obtained from $\texttt{dct2}(\texttt{dctshift}(h, \mathrm{center}))./\texttt{dct2}(\mathbf{e}_1)$
- DCT Implementation uses real arithmetic rather than complex for the FFT.
- Note cost equivalent to FFT cost.

Summary: Spectral Decomposition for (Most) PSF Matrices :BTTB, BCCB, Double Symmetric Reflexive

Given h the PSF array, $\mathrm{center} = (r_0, c_0)$ the central index of the PSF

- Eigenvalues of A are obtained by a transform applied to shifted h
- A spectral decomposition exists

  $$A = U^* \Lambda U$$

  which can be used for forward and inverse operations.
- All forward and inverse operations use transforms and convolution.
- We do not know anything about the ordering of the eigenvalues in Λ in terms of size.

The Singular Value Decomposition (SVD): General A

Suppose, here, $A \in \mathcal{R}^{N\times N}$:

- $A = U\Sigma V^T$, U and V are orthogonal $U^T U = I_N = V^T V$,
- σ_i are the singular values of A,

$$\Sigma = \mathrm{diag}(\sigma_i) \quad \sigma_1 \geq \sigma_2 \geq \ldots \sigma_N \geq 0.$$

- $\mathbf{u}_i$ (columns of U) are the left singular vectors of A
- $\mathbf{v}_i$ (columns of V) are the right singular vectors of A
- Assume $\sigma_N > 0$, A is nonsingular, then

$$A^{-1} = V\Sigma^{-1}U^T = \sum_{i=1}^{N} \frac{\mathbf{v}_i\mathbf{u}_i^T}{\sigma_i}, \quad A = \sum_{i=1}^{N} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

- Solution expressed with the SVD components is a weighted linear combination of basis vectors $\mathbf{v}_i$.

$$\mathbf{x} = \sum_{i=1}^{N} \left(\frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\right)\mathbf{v}_i$$

The Two Dimensional Result

- We note that $\mathbf{x} = \mathrm{vec}(X)$
- Likewise we can write $V_i = \mathrm{array}(\mathbf{v}_i)$, where array is the reverse of vec converting vector to array.
- Then

$$
\begin{aligned}
X &= \sum_{i=1}^{N} \left(\frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i}\right) \mathrm{array}(\mathbf{v}_i) \\
&= \sum_{i=1}^{N} \left(\frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i}\right) V_i
\end{aligned}
$$

This is a decomposition in the basis images V_i.

1.
$$(A \otimes B)\mathrm{vec}(X) = \mathrm{vec}(BXA^T)$$

2. Transpose is distributive
$$(A \otimes B)^T = (A^T \otimes B^T)$$

3. When both matrices are invertible
$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

4. Mixed product property for matrices of appropriate dimensions
$$(A \otimes B)(C \otimes D) = (AB \otimes CD)$$

5. The Kronecker product is associative, but not commutative.

$$A = F^* \Lambda F$$

- Use $F = F_r \otimes F_c$, the one dimensional FT matrices
- $$\mathbf{x} = F^* \Lambda^{-1} F \mathbf{b} = (F_r \otimes F_c)^* \Lambda^{-1} \mathrm{vec}(F_c G F_r^T)$$
- Notice $\mathrm{vec}(F_c G F_r^T)$ is the two dimensional FT of the blurred image G.
- Let $\mathrm{vec}(\tilde{G}) = \Lambda^{-1} \mathrm{vec}(F_c G F_r^T)$
- $$X = F_c^* \tilde{G} (F_r^*)^T = \mathrm{conj}(F_c) \tilde{G}\, \mathrm{conj}(F_r^T)$$
$$X = \sum_i \sum_j \tilde{G}_{ij}\, \mathrm{conj}(\mathbf{f}_{c,i} \mathbf{f}_{r,j}^T)$$

$\mathbf{f}_{c,i}$ and $\mathbf{f}_{r,j}$ are the i^{th} and j^{th} columns of F_c and F_r. This is a decomposition in the basis images $\mathbf{f}_{c,i} \mathbf{f}_{r,j}^T$.

$$A = W^T \Lambda W$$

- Use $W = W_r \otimes W_c$, the one dimensional DCT matrices

$$\mathbf{x} = W^T \Lambda^{-1} W \mathbf{b} = (W_r \otimes W_c)^T \Lambda^{-1} \mathrm{vec}(W_c G W_r^T)$$

- $\mathrm{vec}(W_c G W_r^T)$ is the two dimensional DCT of the blurred image G.
- Let $\mathrm{vec}(\tilde{G}) = \Lambda^{-1} \mathrm{vec}(W_c G W_r^T)$
-

$$X = (W_c)^T \tilde{G} (W_r)$$

$$X = \sum_i \sum_j \tilde{G}_{ij} \, (\mathbf{w}_{c,i} \mathbf{w}_{r,j}^T)$$

$\mathbf{w}_{c,i}$ and $\mathbf{w}_{r,j}$ are the i^{th} and j^{th} columns of W_c and W_r. This is a decomposition in the basis images $\mathbf{w}_{c,i} \mathbf{w}_{r,j}^T$.

The Separable PSF: $A = \mathbf{c}\mathbf{r}^T$

- Matrix A is obtained as Kronecker product $A = A_r \otimes A_c$
- $A\mathbf{x} = A_c X A_r^T$.
- SVD of each matrix

$$A_r = U_r \Sigma_r V_r^T \quad A_c = U_c \Sigma_c V_c^T$$

- SVD of Kronecker product

$$(U_r \Sigma_r V_r^T) \otimes (U_c \Sigma_c V_c^T) = (U_r \otimes U_c)(\Sigma_r \otimes \Sigma_c)(V_r \otimes V_c)^T$$

- Extend the one dimensional SVD solution for $\mathbf{b} = A\mathbf{x}$.

$$X = V_c \Sigma_c^{-1} U_c^T G U_r \Sigma_r^{-1} V_r^T$$

- $\mathrm{vec}(U_c G U_r^T) = (U_r \otimes U_c)^T \mathrm{vec}(G)$ represents the spectral transform of the blurred image G
- Let $\tilde{G}$ be the matrix $\Sigma_c^{-1} U_c^T G U_r \Sigma_r^{-1}$ then

$$X = V_c \tilde{G} V_r^T = (V_r \otimes V_c)\tilde{G} = \sum \sum \tilde{G}_{ij}(\mathbf{v}_{c,i}\mathbf{v}_{r,j}^T)$$

$\mathbf{v}_{c,i}$ and $\mathbf{v}_{r,j}$ are the i^{th} and j^{th} columns of V_c and V_r. This is a decomposition in the basis images $\mathbf{v}_{c,i}\mathbf{v}_{r,j}^T$.

- For all cases we have a decomposition of the solution as a sum of basis images.
- For the Fourier case the order of the spectral components is not defined as it is for the SVD.
- We can examine the solution by examining the spectral components in each case. $\tilde{G}_{ij}$ are equivalent to the weights $\mathbf{u}_i^T\mathbf{b}/\sigma_i$ of the SVD.
- We note that the DFT and DCT provide feasible approaches for finding the defining spectrum, the large scale SVD may be too expensive for practical problems.

The Noise

- Practically the measured data is contaminated by noise, we denote by $\mathbf{e}$ or $\mathrm{array}(\mathbf{e}) = E$.
- The spectral decomposition acts on the noise term in the same way it acts on the **exact** right hand side **b**. e.g.

$$\mathbf{x} = \sum_{i=1}^{r} (\frac{\mathbf{u}_i^T(\mathbf{b}_{\mathrm{exact}} + \mathbf{e})}{\sigma_i})\mathbf{v}_i = \mathbf{x}_{\mathrm{exact}} + \sum_{i=1}^{r} (\frac{\mathbf{u}_i^T\mathbf{e}}{\sigma_i})\mathbf{v}_i$$

  r is the rank of A
- If $\mathbf{e}$ is uniform, we expect $|\mathbf{u}_i^T\mathbf{e}|$ to be similar magnitude $\forall i$.
- Note $\mathrm{cond}(A) = \sigma_1/\sigma_r$. Typically, expect σ_i to decay to zero. Hence $\mathrm{cond}(A)$ may be large.
- σ_i small represents *high frequency* component in the sense that $\mathbf{u}_i$, $\mathbf{v}_i$ have more sign changes as i increases.
- $(\frac{\mathbf{u}_i^T\mathbf{e}}{\sigma_i})$ is the coefficient of $\mathbf{v}_i$ in the error image, the weight of $\mathbf{v}_i$ in the error image, where $\mathbf{v}_i$ is associated with a spatial frequency. When $1/\sigma_i$ is large the contribution of the high frequency error is magnified due to $(\frac{\mathbf{u}_i^T\mathbf{e}}{\sigma_i})$.

Example of singular values dependent on Width of Gaussian PSF and two noise levels

Example taken from Hansen, Nagy and O'Leary, Figure 5.5

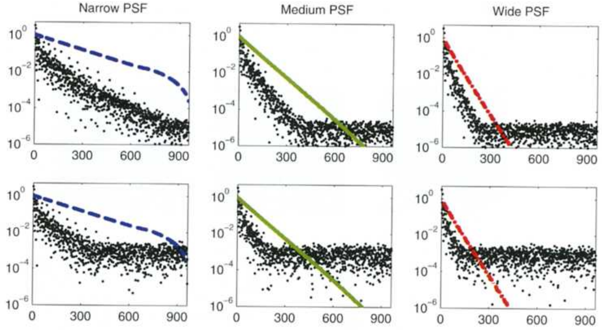


Figure 5.5. *Plots of singular values σ_i (colored lines) and coefficients $|\mathbf{u}_i^T \mathbf{b}|$ (black dots) for the three blurring matrices $\mathbf{A}$ defined by the PSFs in Figure 5.4, and two different levels of the noise $\mathbf{E}$ in the model $\mathbf{B} = \mathbf{B}_{\mathrm{exact}} + \mathbf{E}$. Top row: $\|\mathbf{E}\|_F = 3 \cdot 10^{-4}$; bottom row: $\|\mathbf{E}\|_F = 3 \cdot 10^{-2}$.*

Example of basis images for Middle PSF in 5.5, using SVD

Example taken from Hansen, Nagy and O'Leary, Figure 5.7

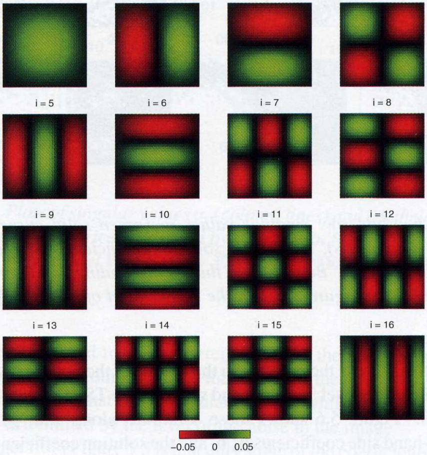


Figure 5.7. *Plots of the first 16 basis images* $\mathbf{V}_i$. *Green represents positive values in* $\mathbf{V}_i$ *while red represents negative values. These matrices satisfy* $\mathbf{v}_i = \text{vec}(\mathbf{V}_i)$, *where the singular vectors* $\mathbf{v}_i$ *are from the SVD of the matrix* $\mathbf{A}$ *for the middle PSF in Figure 5.4.*

Example of basis images : Periodic Boundary Conditions

Example taken from Hansen, Nagy and O'Leary, Figure 5.8
Notice the periodicity effect on the basis images

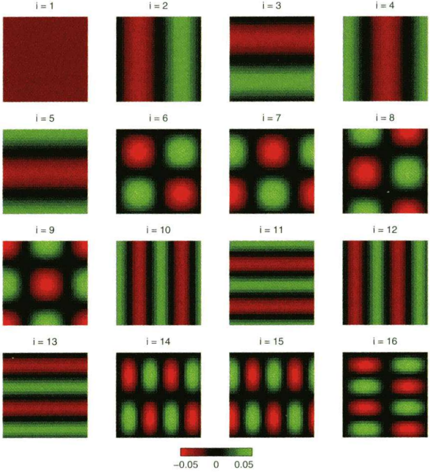


Figure 5.8. *The first* 16 *basis images* $\mathbf{V}_i$ *for the PSF in Figure* 5.7, *with* periodic *boundary conditions in the blurring model.*

Example of basis images : Reflexive Boundary Conditions

Example taken from Hansen, Nagy and O'Leary, Figure 5.9
Notice the reflexive effect on the basis images

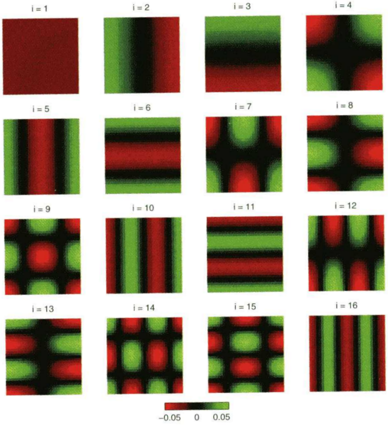


Figure 5.9. *The first* 16 *basis images* $\mathbf{V}_i$ *for the PSF in Figure* 5.7, *with* reflexive *oundary conditions in the blurring model.*

Observations

- σ_i depend on matrix A, which depends on
  - width of the PSF. σ_i decay faster for wider PSF than narrow PSF.
  - implemented boundary conditions. In particular the basis images satisfy the boundary conditions.
- Moreover $|\mathbf{u}_i^T\mathbf{b}|$ depends on the image, and on the noise level in the image.
- Thus $(\mathbf{u}_i^T\mathbf{b})/\sigma_i$ depends on A, $\mathbf{e}$ and $\mathbf{b}$.
- When σ_i is small relative $|\mathbf{u}_i^T\mathbf{e}|$ solution may be contaminated.

Example taken from Hansen, Nagy and O'Leary, Figure 5.10
Notice that DFT basis images are quite different.

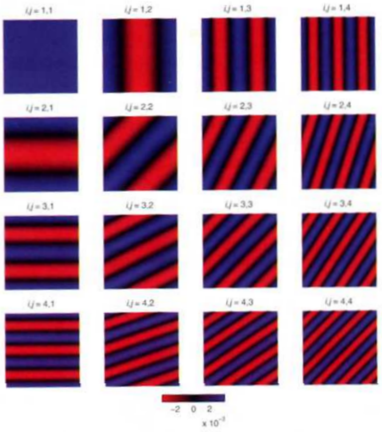


Figure 5.10. *Plots of the real parts of some of the DFT-based basis images* $\mathbf{f}_{c,i}\,\mathbf{f}_{r,j}^T$ *used when the blurring matrix* $\mathbf{A}$ *is a BCCB matrix; blue and red represent, respectively, positive and negative values. The imaginary parts of the basis images have the same appearance. We used* $m = n = 256$.

Example taken from Hansen, Nagy and O'Leary, Figure 5.11
Notice that DCT basis images are again also quite different.

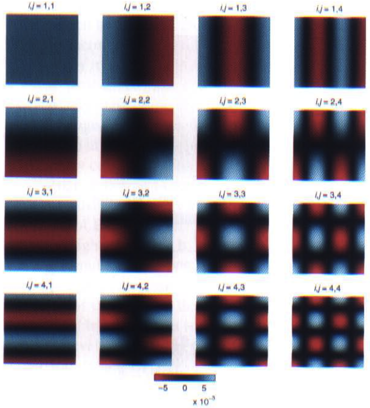


Figure 5.11. *Plots of some of the DCT-based basis images* $\mathbf{c}_{c,i}\,\mathbf{c}_{r,j}^T$ *used in the treatment of BTTB + BTHB + BHTB + BHHB matrices; cyan and red represent, respectively, positive and negative values. We used* $m = n = 256$.

Comparing magnitudes of Spectral Components in DFT, DCT and SVD Bases: separable Gaussian blur

Example taken from Hansen, Nagy and O'Leary, Figure 5.12
Ordered: we see FFT and DCT have fewer large SVs.

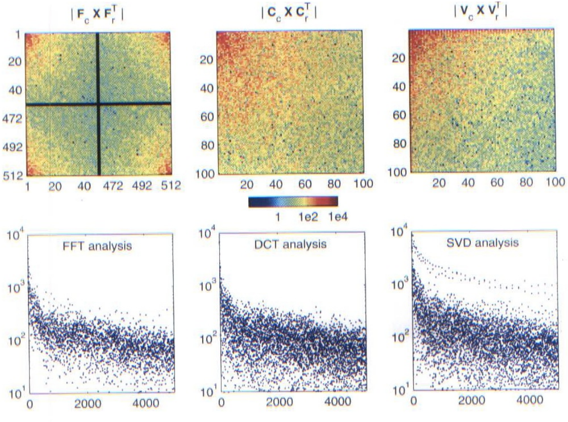


Figure 5.12. *Two representations of the magnitudes of the spectral components of the* `iogray.tif` *image in the DFT, DCT, and SVD bases for separable Gaussian blur. Top: the coefficients as they appear in the computed arrays. Bottom: the coefficients ordered according to decreasing eigenvalues or singular values.*

- SVD assists with understanding the impact of noise on the solution process
- Spectral decomposition shows how the solution is composed of series of basis images
- Practically SVD is not useful for large scale problems
- Practically when A arises from image deblurring the FFT and DCT can be used to find the spectral decomposition and to efficiently implement forward operations $A\mathbf{y}$, $A^T\mathbf{y}$ and $A^{-1}\mathbf{y}$ for any $\mathbf{y}$.
- Spectral decomposition suggests truncation to remove the components which lead to contamination of the solution.

Truncating the SVD expansion

The SVD expansion shows the impact of the noise on the calculation of $\mathbf{x} = \sum_{i=1}^{r} (\frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i}) \mathbf{v}_i$. Therefore it seems reasonable to consider the truncated solution

$$\mathbf{x}_k = \sum_{i=1}^{k} (\frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i}) \mathbf{v}_i = A_k \mathbf{b}$$

Here A_k is a rank k matrix.

- The use of the truncated expansion is feasible only if we can first calculate the SVD of A efficiently.
- The limit on the sum k is regarded as a regularization parameter. We can change k and obtain different images.
- Choice of k controls the degree of low pass filtering which is applied. i.e. controls the attenuation of the high frequency components.
- Look at the example: the image is over or under smoothed dependent on k.

Example of Truncated SVD

Example from Hansen, Nagy and O'Leary, Fig 5.1 Notice under/over smoothing dependent on choice of k in the TSVD

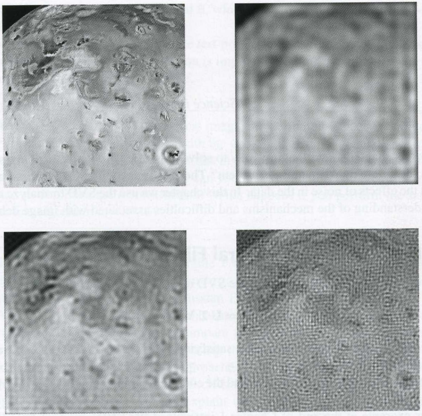


Figure 5.1. *Exact image (top left) and three TSVD solutions* $\mathbf{x}_k$ *to the image deblurring problem, computed for three different values of the truncation parameter:* $k = 658$ *(top right),* $k = 2813$ *(bottom left), and* $k = 7243$ *(bottom right). The corresponding solutions range from oversmoothed to undersmoothed, as k goes from small to large values.*

- SVD assists with understanding the impact of noise on the solution process
- Spectral decomposition shows how the solution is composed of series of basis images
- Practically SVD is not useful for large scale problems
- Practically when A arises from image deblurring the FFT and DCT can be used to find the spectral decomposition and to efficiently implement forward operations $A\mathbf{y}$, $A^T\mathbf{y}$ and $A^{-1}\mathbf{y}$ for any $\mathbf{y}$.
- Decomposition can be useful for finding a preconditioner.
- Spectral decomposition suggests truncation to remove the components which lead to contamination of the solution.

Well-Posedness
Stability
Rosemary Renaut
October 26, 2011

Acknowledgement to Hansen for figures from his slides
http://www2.imm.dtu.dk/~pch/DIPbook.html

Consider the mapping A which takes the solution f to output data g. $Af = g$. Inverse problem: find f given g and A.

Definition (Well - Posed)

The problem of finding f from g is called well-posed (by Hadamard in 1923) if all

Existence *a solution exists for any data g in data space,*

Uniqueness *the solution f is unique*

Stability *continuous dependence of f on g : the inverse mapping $g \to f$ is continuous.*

The first two conditions are equivalent to saying that the operator A has a well defined inverse A^{-1}.

Suppose there exists f_1 such that $Af_1 = 0$, then $Af = g$ and $Af(f + f_1) = g$ so f is not unique.

Moreover, we require that the domain of A^{-1} is all of data space.

Definition (Ill-Posed: according to Hadamard)

A problem is ill-posed if it does not satisfy all three conditions for well-posedness.

Alternatively an ill-posed problem is one in which

1. $g \notin \mathrm{range}(A)$
2. inverse is not unique because more than one image is mapped to the same data, or
3. an arbitrarily small change in the data can cause an arbitrarily large change in the image.

Example from Discrete problem - do we know if a solution is *good*

Consider the linear system

$$A = \begin{pmatrix} 0.16 & 0.10 \\ 0.17 & 0.11 \\ 2.02 & 1.29 \end{pmatrix}, b = \begin{pmatrix} 0.26 \\ 0.28 \\ 3.31 \end{pmatrix}, x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

The least squares solution yields

$$x_{ls} = [1, 1]^T \; \|Ax_{ls} - b\|^2 = 0, \; \|x - x_{ls}\|^2 = 0$$

Perturbing b by $\delta b = [.01, .01, .001]$ yields

$$x'_{ls} = [1.6857, -0.0718]^T, \; \|Ax'_{ls} - b\|^2 = .00018, \; \|x - x'_{ls}\|^2 = 1.6189$$

- A small residual does not imply a realistic solution
- Ill-conditioning of A leads to a poor solution
- Perturbing b leads to a larger perturbation in x.

Singular Value Expansion

Let $L_2([0,1]\times[0,1])$ be space of square integrable functions on $[0,1]\times[0,1]$, i.e. $h \in L_2([0,1]\times[0,1])$

$$\|h\|_2^2 = \int_0^1 \int_0^1 h(s,t)^2 \, ds \, dt \text{ is finite.}$$

The **Singular Value Expansion** is

$$h(s,t) = \sum_{i=1}^{\infty} \mu_i u_i(s) v_i(t)$$

- For the inner product $< \phi, \psi >= \int_0^1 \phi(t)\psi(t)dt$,

$$< u_i, u_j >=< v_i, v_j >= \delta(i-j) \quad \text{orthonormality}$$

- μ_i are the **singular values** of h, ordered from large to small

$$\mu_1 \geq \mu_2 \geq \cdots \geq 0.$$

Properties of the SVE

1. $h(s,t)$ square integrable implies $\sum_{i=1}^{\infty}\mu_i^2$ must be bounded:

$$\|h\|^2 = \int_0^1\int_0^1\left(\sum_{i,j=1}^{\infty}\mu_i\mu_j u_i(s)u_j(s)v_i(t)v_j(t)\right)ds\,dt$$
$$= \sum_{i=1}^{\infty}\mu_i^2, \text{ by orthonormality of } (v_i(t), u_i(s)).$$

   For the sum to be bounded μ_i must decay faster than $1/\sqrt{i}$.

2. Left and right **singular functions** of $h(s,t)$ are (u_i, v_i):
   - $\{\mu_i^2, u_i\}$ of $\int_0^1 h(s,x)h(t,x)dx$ and
   - $\{\mu_i^2, v_i\}$ of $\int_0^1 h(x,s)h(x,t)dx$.

$$< \int_0^1 h(s,x)h(t,x)\,dx, u_k(s) > =< \sum_{i,j}\mu_i\mu_j u_i(s)u_j(t)\int_0^1 v_i(x)v_j(x)\,dx, u_k(s) >$$
$$=< \sum_i \mu_i^2 u_i(s)u_j(t), u_k(s) >= \mu_k^2 u_k(t).$$

Properties of the Singular Functions

Basis for function Space u_i and v_i provide a basis for $L_2([0,1])$.

If $f, g \in L_2([0,1])$ then

$$f(t) = \sum_i < v_i, f > v_i(t) \text{ and } g(s) = \sum_i < u_i, g > u_i(s)$$

Fundamental Mapping v_i to u_i:

$$\int_0^1 h(s,t)v_i(t)dt = \int_0^1 \sum_{j=1}^{\infty} \left(\mu_j u_j(s) v_j(t)\right) v_i(t)dt$$

$$\text{i.e. } < h(s,t), v_i(t) > = \mu_i u_i(s), \quad i = 1, 2, \ldots$$

by the orthonormality of v_i.

Smoothing of v_i by kernel $h(s,t)$ to give $\mu_i u_i$.

Observe

$$g(s) = \int_0^1 \left(\sum_j \mu_j u_j(s) v_j(t) \right) f(t)\, dt = \int_0^1 \sum_{ij} \mu_j u_j(s) v_j(t) < v_i, f > v_i(t)\, dt$$

$$\text{i.e. } \sum_i < u_i, g > u_i(s) = \sum_i \mu_i < v_i, f > u_i(s) \text{ and } f(t) = \sum_i < v_i, f > v_i(t) \tag{1}$$

- By Fundamental relation $\mu_i u_i(s)$ is **smoothed** v_i.
- Hence comparing sums of f and g, g is smoother than f.
- Also, if $\mu_i = 0$ we can only have solution f of the integral equation if component $< u_i, g > u_i(s)$ is zero.
- For $\mu_i \neq 0, i = 1, 2, \ldots$ with $\mu_i < v_i, f >=< u_i, g >$ (taking inner product with $u_j(s)$ on each side in (**??**)) yields

$$f(t) = \sum_i \frac{< u_i, g >}{\mu_i} v_i(t)$$

Picard Condition

$f \in L_2([0,1])$ only exists if the infinite sum converges.

Definition (Picard Condition)

f is square integrable if

$$\|f\|_2^2 = \int_0^1 f(t)^2\, dt = \sum_{i=1}^{\infty} (< v_i, f >)^2 = \sum_{i=1}^{\infty} \left(\frac{< u_i, g >}{\mu_i} \right)^2 < \infty$$

- Right hand coefficients $< u_i, g >$ must decay faster than μ_i.
- It is necessary that $\exists N$ s.t. $\forall i > N$, $< u_i, g >$ decays faster than μ_i.
- g is square integrable if $< u_i, g >$ decay faster than $1/\sqrt{i}$ (look at $\|g\|^2$), but Picard condition requires faster decay than $\mu_i/\sqrt{i}$.

f is characterised by (μ_i, u_i, v_i)

$$f(t) = \sum_i \frac{< u_i, g >}{\mu_i} v_i(t)$$

- Decay rate of μ_i is fundamental to behavior of ill-posed problem.
- Calculating f from g amplifies components v_i as $\mu_i \to 0$.
- High frequency v_i are amplified by inversion.
- the smaller μ_i the greater the oscillations in u_i, v_i.
- if h has continuous derivatives of order $0, \ldots, q$ μ_i decay approximately as $O(i^{-(q+1/2)})$
- if h is infinitely differentiable μ_i decay faster, i.e. as $O(\rho^i)$ for some $0 < \rho < 1$.

More on Ill-posedness

Degree if $\exists\, \alpha > 0$ s.t. $\mu_i = O(i^{-\alpha})$ then α is the degree of ill-posedness

Mildly ill-posed if $\alpha \leq 1$.

Moderately ill-posed if $\alpha > 1$

Severely ill-posed if $\mu_i = O(e^{-\alpha i})$.

Further Details

- Note in discussion we always assume that the equality signs with infinite sums imply uniform convergence for continuous functions. i.e. for solution of the integral equation there exists N such that for all $n > N$

$$\left| f(t) - \sum_{i=1}^{n} \frac{< u_i, g >}{\mu_i} v_i(t) \right| < \epsilon, \quad \forall\, t \in [0, 1]$$

- If the kernel is discontinuous convergence is with respect to the mean square,

$$\| f(t) - \sum_{i=1}^{n} \frac{< u_i, g >}{\mu_i} v_i(t) \|_2 < \epsilon, \quad \forall\, t \in [0, 1]$$

Consider the first kind Fredholm integral equation $h(s,t)$,

$$\int_0^1 h(s,t)f(t)dt = g(s).$$

Consider functions f_p $f_p = \sin(2\pi pt), \quad p = 1, 2, \ldots$. For arbitrary square integrable h,

$$g_p = \int_0^1 h(s,t) sin(2\pi pt)dt \to 0 \,\text{as}\, p \to \infty.$$

As frequency of f increases, amplitude of g decreases due to smoothing by h

This is a statement of the Riemann Lebesque Lemma

Finding f from smoother g amplifies high frequencies in f.

Ratio $\frac{f_p}{g_p}$ can become arbitrarily large for p large enough.

Problem is ill-posed because solution f is not continuously dependent on data g.

Illustration of the Lebesgue Lemma Result

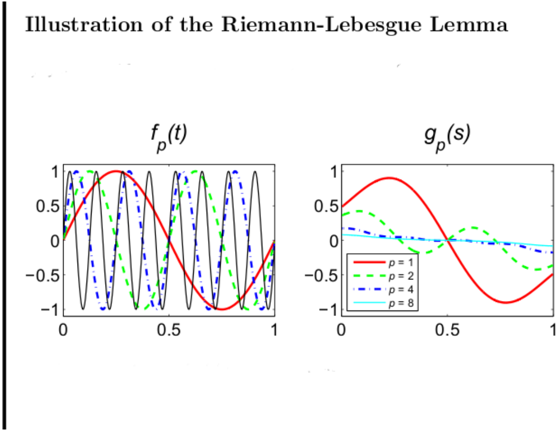


Figure: Notice the increasing oscillation in f_p with increasing p while g is more smooth

Non existence of a Solution : Ursell

$$\int_0^1 \frac{1}{s+t+1} f(t)\,dt = 1, \quad 0 \leq s \leq 1$$

Clearly $h(s,t) = \frac{1}{s+t+1}$ and $g = 1$.

$$\text{Defining} \quad g_k(s) = \sum_i^k < u_i, g > u_i(s) \quad \text{then}$$

$$\|g - g_k\|_2 \to 0 \quad \text{with} \quad k \to \infty \quad \text{but}$$

$$f_k(t) = \sum_i^k \frac{< u_i, g >}{\mu_i} v_i(t) \quad \text{satisfies} \|f_k\|_2 \to \infty.$$

f_k does not converge to a square integrable solution.

Illustration of the Ursell Solution

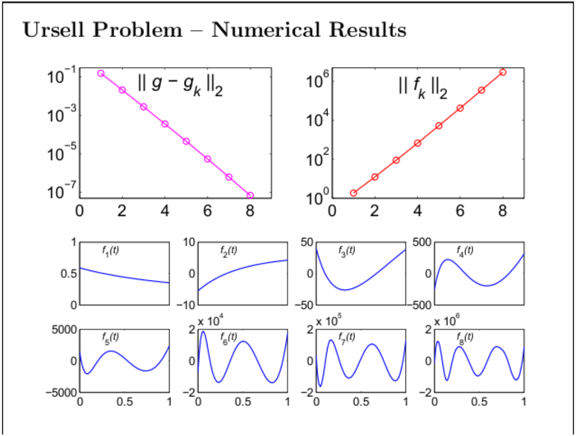


Figure: Notice the scales, the bottom panel shows solutions f_k for $k = 1 \ldots 8$

Ambiguity in Inverse Problems: non unique solution

If the integral operator has a null space

$$\int_0^1 h(s,t)f(t) = 0, \quad \text{for some } f(t)$$

then f is called an **annihilator** for h, as is αf for any scalar α. Null space of h is the space of all annihilators f. Moreover, by the fundamental relation

$$\text{null}(h) = \text{span}\{v_i | \mu_i = 0\}$$

If there are only a finite number of $\mu_i > 0$, the kernel is **degenerate**

Another example: degenerate kernel

Consider

$$\int_0^1 (s + 2t) f(t)\, dt = g(s), \quad -1 \le s \le 1.$$

It is possible to find the solution. We find

$$u_1(s) = 1/\sqrt{2} \quad u_2(s) = \sqrt{3}/\sqrt{2}s \quad v_1(t) = \sqrt{3}/\sqrt{2}t \quad v_2(t) = 1/\sqrt{2}$$
$$\mu_1 = 2/\sqrt{3} \quad \mu_2 = 4/\sqrt{3} \quad \mu_i = 0, i > 2$$

A solution exists only if

$$g \in \mathrm{range}(h) = \mathrm{span}\{u_1, u_2\} \quad \text{implies}$$
$$g = a + bs \quad f = \frac{b}{4} + \frac{3}{2}at$$

But notice that $f = 3t^2 - 1$ is an annihilator and hence f is not unique!

Spectral Properties of the Singular Functions: Brief Overview

Define the integral operator

$$[Hf](s) = \int_{-\pi}^{\pi} h(s,t)f(t)\, dt \text{ and assume}$$

1. h is real and continuous on $[-\pi, \pi] \times [-\pi, \pi]$.
2. for simplicity $\|h(\pi, t) - h(-\pi, t)\|_2 = 0$.
3. h is square integrable with singular set (μ_i, u_i, v_i) such that

$$[Hv_j](s) = \mu_j u_j(s), \quad [H^* u_j](t) = \mu_j v_j(t)$$

An analysis can be performed to demonstrate that
Singular functions are similar to Fourier functions : for small j, the large singular values and corresponding singular functions correspond to low Fourier frequencies, but for large j (small singular values) correspond to the high frequencies
i.e. the singular functions are similar to trigonometric functions which explains the increasing oscillations for the smaller singular values.

Noise in the data

Suppose that g is measured and contaminated by errors $g = g_{\text{exact}} + \eta$ where

$$g_{\text{exact}} \in \text{ range}(h) \text{ with } \|\eta\|_2 \ll \|g_{\text{exact}}\|_2$$

Immediately from the SVE

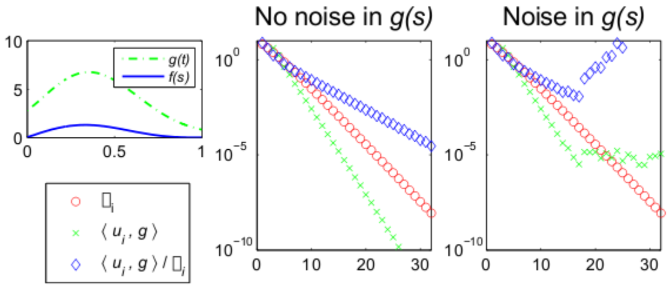
$$f = f_{\text{exact}} + \sum_{i=1}^{\infty} \frac{< u_i, \eta >}{\mu_i} v_i(t)$$

where the second term is the contribution due to the noise.

- Noise is typically high frequency and we cannot anticipate η to satisfy the Picard condition.
- Thus $g \notin \text{range}(h)$.
- Using the infinite sum for obtaining f is unlikely to yield a useful estimate of f_{exact}

Illustration of the Picard Condition for Noisy Data

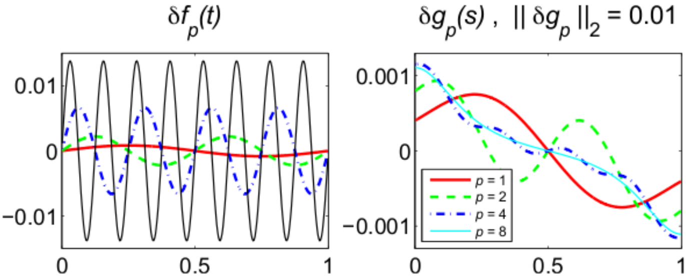
Illustration of the Picard Condition



The violation of the Picard condition is the simple explanation of the instability of linear inverse problems in the form of first-kind Fredholm integral equations.

Difficulties with High Frequencies

In this example $\delta g(s) = \int_0^1 K(s,t)\,\delta f(t)\,dt$ and $\|\delta g\|_2 = 0.01$.



Higher frequencies are amplified more in the reconstruction process.

Summary

1. First kind Fredholm integral equation provides a linear model for inverse problem analysis
2. For such models the solutions may be arbitrarily sensitive to perturbations of the data.
3. The SVE provides a means to analyse stability and existence of solutions.
4. Picard condition is necessary for existence of solution which is square integrable.
5. Right hand side g must be sufficiently smooth as measured by its SVE coefficients.
6. For more general inverse problems, e.g. Laplace transform, the operator is not compact, but a similar analysis for continuum of singular values can be applied.

1. First kind Fredholm integral equation provides a linear model for inverse problem analysis
2. The SVE provides a means to analyse stability and existence of solutions.
3. Picard condition is necessary for existence of solution which is square integrable.
4. Right hand side g must be sufficiently smooth as measured by its SVE coefficients.
5. For more general inverse problems, e.g. Laplace transform, the operator is not compact, but a similar analysis for continuum of singular values can be applied.
6. Most cases we cannot calculate the SVE.

Solution with the SVD - defined as for the SVE
Discretizing the Integral
Using the SVD for the SVE
Spectral Filtering
Errors Rosemary Renaut
November 2, 2011

Consider general overdetermined discrete problem

$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{m \times n}, \quad \mathbf{b} \in \mathbb{R}^m, \quad \mathbf{x} \in \mathbb{R}^n, \quad m \geq n.$$

Thin singular value decomposition (SVD) of rectangular A is

$$A = U\Sigma V^T = \sum_{i=1}^{n} \mathbf{u}_i \sigma_i \mathbf{v}_i^T, \quad \Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_n).$$

U of size $m \times n$, V and Σ square of size n:

$$U = [\mathbf{u}_1, \dots, \mathbf{u}_n], \quad V = [\mathbf{v}_1, \dots, \mathbf{v}_n], \sigma_1 \geq \sigma_2 \geq \sigma_n \geq 0$$

Orthonormal columns in U and V, left and right singular vectors for A

$$\mathbf{u}_i^T \mathbf{u}_j = \mathbf{v}_i^T \mathbf{v}_j = \delta(i-j) \rightarrow U^T U = V^T V = V V^T = I_n.$$

If A has full column rank $\sigma_n > 0$

$$A^\dagger = V \Sigma^{-1} U^T = A^{-1}, \quad m = n.$$

Moore Penrose Generalized Inverse

1. $AA^{\dagger}A = A$
2. $A^{\dagger}AA^{\dagger} = A^{\dagger}$
3. $(AA^{\dagger})^{*} = AA^{\dagger}$
4. $(A^{\dagger}A)^{*} = A^{\dagger}A$

Deriving the Solution: Similarly to SVE

1. We can write $\mathbf{x} = VV^T\mathbf{x} = \sum_{i=1}^{n}(\mathbf{v}_i^T\mathbf{x})\mathbf{v}_i$ and
$$A\mathbf{x} = \sum_{i=1}^{n}(\mathbf{v}_i^T\mathbf{x})A\mathbf{v}_i$$
2. But $A\mathbf{v}_i = U\Sigma V^T\mathbf{v}_i = U\Sigma\mathbf{e}_i = \sigma_i\mathbf{u}_i$. Thus
$$A\mathbf{x} = \sum_{i=1}^{n}\sigma_i(\mathbf{v}_i^T\mathbf{x})\mathbf{u}_i$$
3. Similarly, for $m = n$, $\mathbf{b} = \sum_{i=1}^{n}(\mathbf{u}_i^T\mathbf{b})\mathbf{u}_i$.
4. Immediately compare coefficients and obtain $\sigma_i(\mathbf{v}_i^T\mathbf{x}) = \mathbf{u}_i^T\mathbf{b}$, $i = 1, \ldots, n$ and
$$\mathbf{x} = \sum_{i=1}^{n}\frac{(\mathbf{u}_i^T\mathbf{b})}{\sigma_i}\mathbf{v}_i$$
5. Sensitivity of solutions depends on $\text{cond}(A) = \sigma_1/\sigma_n$

Full SVD: Rectangular A

- Let $U = [U_1, U_2]$ be square of size m, Σ rectangular of size $m \times n$:
$$A = U\Sigma V^T = [U_1, U_2] \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T$$
- The inverse is replaced by the pseudo inverse: if A has rank r
$$A^\dagger = \sum_{i=1}^{r} \frac{1}{\sigma_i} \mathbf{v}_i \mathbf{u}_i^T$$
- Solution of the LS problem is given by
$$\mathbf{x} = \sum_{i=1}^{r} \frac{(\mathbf{u}_i^T \mathbf{b})}{\sigma_i} \mathbf{v}_i$$
- Sensitivity of solution depends on the condition as measured by σ_1/σ_r.
- Recall singular values relation to eigenvalues λ_i of $A^T A$, $\sigma_i^2 = \lambda_i$

Quadrature - how do we obtain A

Need to understand how we go from integral to matrix.

$$\begin{array}{lcl} \text{Integral Equation} & < h, f > & = g \\ \text{Discrete Form} & A\mathbf{x} & = \mathbf{b} \end{array}$$

Quadrature to evaluate the integral (finite range $[a, b] \to [0, 1]$)

$$\int_0^1 p(t)dt = \sum_{j=1}^{n} \omega_j p(t_j) + E_n(p)$$

- E_n is the error which depends on n and the function p.
- t_j are the abscissae, ω_j are weights for the rule.

For $< h, f >= g$, $p(t) = h(s, t)f(t)$. Thus

$$\sum_{j=1}^{n} \omega_j h(s_i, t_j) f(t_j) = g(s_i) + E_n(s_i), \quad i = 1 \dots m.$$

Notice error depends also on the collocation point s_i.

Neglecting E and setting $\tilde{\mathbf{x}}$ as the approximation to $\mathbf{x}$ we obtain

$$\sum_{j=1}^{n} \omega_j h(s_i, t_j)\tilde{\mathbf{x}}_j = g(s_i), \quad i = 1 \dots m.$$

Thus defining $A = HD$, where

- D is a diagonal matrix $d_{jj} = \omega_j$
- $H_{ij} = h(s_i, t_j)$

$$A\tilde{\mathbf{x}} = \mathbf{b}$$

We could solve for scaled $\mathbf{x}$ say $\tilde{\mathbf{x}} = D^{-1}\bar{\mathbf{x}}$.

Given $\mathbf{b}^{(m)}$ i.e. of length m we obtain $\mathbf{x}^{(n)}$ of length n.

What do we use for the weights ω_j and abscissae t_j?

Trapezium rule etc, are collocation based methods. Give values of f at discrete t_j.

Expansion methods provide an expression for $f(t)$.

Galerkin Approach

SVE expands f and g in terms of basis functions and coefficients g_i, f_j.

$$g^{(m)} \in \mathrm{span}\{\psi_1(s), \psi_2(s), \ldots, \psi_m(s)\} \quad f^{(n)} \in \mathrm{span}\{\phi_1(s), \phi_2(s), \ldots, \phi_n(s)\}$$

$$g^{(m)}(s) = \sum_{i=1}^{m} g_i \psi_i(s), \quad f^{(n)}(t) = \sum_{j=1}^{n} f_j \phi_j(t) \quad \text{integrate}$$

$$g(s) = \int_0^1 h(s,t) f(t) dt \approx \int_0^1 h(s,t) \sum_{j=1}^{n} f_j \phi_j(t) dt := \theta(s)$$

$g(s) - \theta(s)$ is the residual. Galerkin approach require $\theta(s) - g(s)$ orthogonal to $\mathrm{span}\{\psi_1(s), \ldots, \psi_m(s)\}$

$$< \psi_i(s), \theta(s) - g(s) >= 0 \quad i = 1 \ldots m$$

Hence $< \psi_i, \theta >=< \psi_i, g > \quad i = 1 \ldots m$ gives

$$< \psi_i, g >=< \psi_i, \theta > = \sum_{j=1}^{n} f_j < \psi_i(s), \int_0^1 h(s,t) \phi_j(t) dt >$$

$$= \sum_{j=1}^{n} \left(\int_0^1 \int_0^1 h(s,t) \psi_i(s) \phi_j(t) ds dt \right) f_j$$

The integration defines A and right hand side b by

$$A_{ij} = \int_0^1 \int_0^1 h(s,t)\psi_i(s)\phi_j(t)dsdt, \quad b_i = <\psi_i, g>$$

Requires numerical quadrature for

$$\int_0^1 \int_0^1 h(s,t)\psi_i(s)\phi_j(t)dsdt \ \forall (i,j), \quad b_i = \int_0^1 \psi_i(s)g(s)ds, \ \forall i$$

If $h(s,t)$ is symmetric $h(s,t) = h(t,s)$; use $\phi_i(s) = \psi_i(s)$. Then A is symmetric.

Consider the case $\phi_i = \psi_i = \rho_i$ where ρ_i is the top hat

$$\rho_i(t) = \begin{cases} \frac{1}{\sqrt{h}} & t \in [(i-1)h, ih] \\ 0 & \text{otherwise} \end{cases}$$

$$A_{ij} = \frac{1}{h}\int_{(i-1)h}^{ih}\int_{(j-1)h}^{jh} h(s,t)dsdt \quad b_i = \frac{1}{\sqrt{h}}\int_{(i-1)h}^{ih} g(s)ds$$

Sampling

g is sampled at s_i, thus

$$g(s_i) = \int_0^1 \delta(s - s_i) g(s) ds \quad \text{suggests } \psi_i(s) = \delta(s - s_i) \text{ and}$$

$$A_{ij} = \int_0^1 \int_0^1 h(s,t)\delta(s - s_i)\phi_j(t) ds dt = \int_0^1 h(s_i, t)\phi_j(t) dt$$

so that the quadrature is reduced to one dimensional.

Sampling can also be implemented with the top hat and then

$$A_{ij} = \frac{1}{\sqrt{h}} \int_0^1 \phi_j(t) \left(\int_{(i-1)h}^{ih} h(s,t) ds \right) dt, \quad b_i = \sqrt{h} g(s_i)$$

Idea : calculate an approximate SVE numerically via the SVD.

Given the SVD how are the relevant components $\mathbf{u}_j$, $\mathbf{v}_j$ (columns of U and V)) and σ_j related to SVE basis functions $u_i(s)$, $v_i(t)$, singular values μ_i.

Discrete matrix A depends on $\psi_i, i = 1, \ldots m$ and $\phi_j, j = 1 \ldots n$.

Continuous kernel h depends on ψ_i, ϕ_j, $(i, j) = 1 \ldots \infty$.

Consider the approximate kernel $\tilde{h}$ which is obtained by using the discrete set $\psi_i, i = 1 \ldots n$ and $\phi_j, j = 1 \ldots n$.

The result relates SVD of A to SVE of $\tilde{h}$

Suppose that matrix A is calculated using the expansion method with functions ψ_i, ϕ_j, $i, j = 1, \dots n$.

Calculate its SVD: $\Sigma = \mathrm{diag}(\sigma_i)$, $U = (u_{ij})$, $V = (v_{ij})$

Let $\tilde{\mathbf{u}}_j^{(n)}(s) := \sum_{i=1}^n u_{ij}\psi_i(s)$, $\tilde{\mathbf{v}}_j^{(n)}(t) := \sum_{i=1}^n v_{ij}\phi_i(t)$, $j = 1 : n$.

Theorem $\sigma_j^{(n)}$, $\tilde{\mathbf{u}}_j^{(n)}$, $\tilde{\mathbf{v}}_j^{(n)}$ are exact singular values and functions of degenerate kernel

$$\tilde{h}(s,t) := \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}\psi_i(s)\phi_j(t)$$

i.e. we have SVE for an approximate kernel - how does that relate to the exact kernel?

$$h(s,t) = \sum_{I} \mu_i u_i(s) v_i(t)$$

Limits with n, $\sigma_j^{(n)}$

SVD of $A^{(n)} = U^{(n)}\Sigma^{(n)}(V^{(n)})^T$

Error of the kernel: $\delta_n^2 := \|h - \tilde{h}\|^2 = \|h\|^2 - \|A\|_F^2$

note $\|\cdot\|_F^2$ is the Frobenious norm $\|A\|_F^2 = \sum_{i,j=1}^n a_{ij}^2$

Singular values converge $\sigma_i^{(n)} \leq \sigma_i^{(n+1)} \leq \mu_i$, $i = 1, \dots, n$.

Errors are bounded $0 \leq \mu_i - \sigma_i^{(n)} \leq \delta_n$, $i = 1, \dots, n$.

Hence if $\delta_n \to 0$ with n increasing, approximate singular values converge uniformly to true singular values.

SSE $\sum_{i=1}^n [\mu_i - \sigma_i^{(n)}]^2 \leq \delta_n^2$.

Estimation of δ_n from $\|h\|^2$

Orthonormality $\tilde{\mathbf{u}}_i^{(n)}, \tilde{\mathbf{v}}_i^{(n)}$ are orthonormal. Convergence

$$\max\{\|u_i - \tilde{\mathbf{u}}_i^{(n)}\|, \|v_i - \tilde{\mathbf{v}}_i^{(n)}\|\} \leq \left(\frac{2\delta_n}{\mu_i - \mu_{i+1}}\right)^{1/2}$$

Practically observe that approximate singular values are more accurate than approximate singular functions.

$< \tilde{\mathbf{u}}_j^{(n)}, g^{(n)} >$ is important in the Picard condition.

$$< \tilde{\mathbf{u}}_j^{(n)}, g^{(n)} > = \int_0^1 (\sum_{i=1}^{n} u_{ij}^{(n)} \psi_i(s))(\sum_{k=1}^{n} b_k \psi_k(s)) ds$$
$$= \sum_{i,k} u_{ij}^{(n)} b_k < \psi_i, \psi_k > = \sum_i u_{ij}^{(n)} b_i = \mathbf{u}_j^T \mathbf{b}$$

SVD and approximate inner products are related.
i.e. the exact inner products $< u_j, g >, i = 1, \dots,$ are approximated by $< \mathbf{u}_j^{(n)}, g^{(n)} >$ which is immediately obtained from the SVD for A.

Discrete Picard Condition Let τ denote the level such that $\forall j > r$, $\sigma_j \approx O(\tau)$, due to noise and rounding . The discrete Picard condition is satisfied if for $j \leq r$ the coefficients $|(\mathbf{u}_j^{(n)})^T \mathbf{b}|$ decay faster than σ_j.

Picard condition applies only for $\sigma_j > O(\tau)$. It is a condition on the size of the inner products $(\mathbf{u}_j^{(n)})^T \mathbf{b}$ for $j \leq r$.

SVE Solution	SVD Solution
$f(t) = \sum_j \frac{<u_j,g>}{\mu_j} v_j(t)$	$\tilde{\mathbf{x}} = \sum_{j=1}^{n} \frac{<\mathbf{u}_j^{(n)},\mathbf{b}>}{\sigma_j} \mathbf{v}_j^{(n)}$

But $< \mathbf{u}_j^{(n)}, \mathbf{b} >=< \tilde{\mathbf{u}}_j^{(n)}, g^{(n)} >$ where $\tilde{\mathbf{u}}_j^{(n)}$ tends to u_j with increasing n, while $\sigma_j^{(n)}$ converges to μ_j with n.

Equivalently, if the discretization with increasing n is sufficiently good, the approximate solution obtained from the SVD is essentially independent of the discretization.

For solving the first kind Fredholm integral equation numerically, the coefficients $(\mathbf{u}_j^{(n)})^T\mathbf{b}$ and singular values σ_j reveal important information about the true quantities $< u_j, g >$ and μ_j.

For increasing n until converged

1. Choose the orthonormal basis functions $\psi_i(s)$ and $\phi_i(t)$.
2. Calculate matrix A with entries $a_{ij} = <\psi_i, h\phi_j>$, $i, j = 1, \ldots, n$.
3. Compute SVD of A
4. Estimate the singular functions $\tilde{\mathbf{u}}_j(s)$ and $\tilde{\mathbf{v}}_j(t)$

Test Convergence of set of singular values.

End For

Is square integrable required for the theory?

Consider solving for f from the Laplace transform

$$g(s) = \int_0^{\infty} e^{-st} f(t) dt$$

Kernel e^{-st} is not square integrable:

$$\int_0^a (e^{(-st)})^2 ds = \int_0^a e^{-2st} ds = \frac{1 - e^{-2ta}}{2t} \to \frac{1}{2t} \text{ for } a \to \infty$$

But $\int_0^\infty t^{-1}$ is infinite, $\int_0^\infty \int_0^\infty (e^{(-st)})^2 ds dt$ is infinite. No SVE
Now $f(t)$ bounded for $t \to \infty$ implies $g(s)$ is bounded $\forall s \geq 0$.
Truncation for large a in Laplace transform introduces small error in g, and g decays with s. We obtain integral equation

$$\int_0^a e^{-st} f(t) dt = g(s), \quad 0 \leq s \leq a.$$

Now the kernel is square integrable.
Pick a and increase n, the SVD converges.
Pick n and increase a, the SVD does not converge.
Demonstrates the lack of SVE for the Laplace Transform.

Solution for Noisy Data

- Denote noise by $\mathbf{e}$ or $\mathrm{array}(\mathbf{e}) = E$.
- Spectral decomposition acts on the noise term in the same way it acts on the **exact** right hand side $\mathbf{b}$. e.g.

$$\mathbf{x} = \sum_{i=1}^{r} \left(\frac{\mathbf{u}_i^T(\mathbf{b}_{\mathrm{exact}} + \mathbf{e})}{\sigma_i}\right)\mathbf{v}_i = \mathbf{x}_{\mathrm{exact}} + \sum_{i=1}^{r} \left(\frac{\mathbf{u}_i^T \mathbf{e}}{\sigma_i}\right)\mathbf{v}_i$$

- If $\mathbf{e}$ is uniform, anticipate $|\mathbf{u}_i^T \mathbf{e}|$ of similar magnitude $\forall i$.
- Can only recover components that arise from $|\mathbf{u}_i^T \mathbf{b}|$ greater than the noise level.
- But anticipate $\sigma_i \to 0$. σ_i small represents *high frequency* component in the sense that $\mathbf{u}_i$, $\mathbf{v}_i$ have more sign changes as i increases.
- $(\frac{\mathbf{u}_i^T \mathbf{e}}{\sigma_i})$ is the coefficient of $\mathbf{v}_i$ in the error image.
- If $1/\sigma_i$ large the contribution of the high frequency error is magnified due to $(\frac{\mathbf{u}_i^T \mathbf{e}}{\sigma_i})$.

Truncating the SVD expansion

The SVD expansion shows the impact of the noise on the calculation of $\mathbf{x} = \sum_{i=1}^{r}(\frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i})\mathbf{v}_i$. Therefore it seems reasonable to consider the truncated solution

$$\mathbf{x}_k = \sum_{i=1}^{k}(\frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i})\mathbf{v}_i = A_k\mathbf{b}$$

Here A_k is a rank k matrix.

- The use of the truncated expansion is feasible only if we can first calculate the SVD of A efficiently.
- The limit on the sum k is regarded as a regularization parameter. We can change k and obtain different solutions.
- Choice of k controls the degree of low pass filtering which is applied. i.e. controls the attenuation of the high frequency components.
- Look at the example: the image is over or under smoothed dependent on k.

Example of Truncated SVD

Example from Hansen, Nagy and O'Leary, Fig 5.1

Notice under or over smoothing is dependent on choice of k in the TSVD

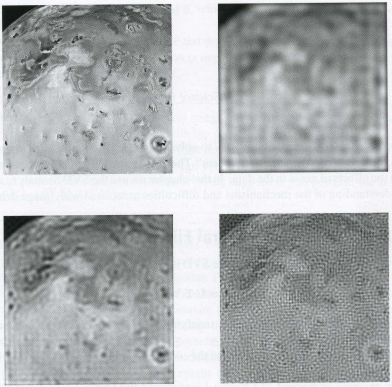


Figure 5.1. *Exact image (top left) and three TSVD solutions* $\mathbf{x}_k$ *to the image deblurring problem, computed for three different values of the truncation parameter:* $k = 658$ *(top right),* $k = 2813$ *(bottom left), and* $k = 7243$ *(bottom right). The corresponding solutions range from oversmoothed to undersmoothed, as k goes from small to large values.*

Estimating rank of A for implementing the TSVD

- Results suggest that we need information on SVD of A
- Also need information on the spread of the singular values.
- Ideally information on the noise level in the data is available.
- Practically we need the **numerical** rank of A.
- Practically it is not always viable to find the effective numerical rank
- We turn to other methods to find acceptable solutions.

The Filtered SVD - more general than truncation

The truncated SVD is a special case of spectral filtering

Recall $\mathbf{x} = A^\dagger \mathbf{b} = V\Sigma^\dagger U^T \mathbf{b}$.

The filtered solution is given by

$$\mathbf{x}_{\mathrm{filt}} = \sum_{i=1}^{r} \gamma_i \left(\frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i}\right)\mathbf{v}_i = V\Sigma_{\mathrm{filt}}^\dagger U^T \mathbf{b}, \quad \Sigma_{\mathrm{filt}}^\dagger := \mathrm{diag}\left(\frac{\gamma_i}{\sigma_i}, 0_{m-r}\right)$$

i.e

$$\mathbf{x} = V\Gamma\Sigma^\dagger U^T \mathbf{b},$$

where Γ is the diagonal matrix with entries γ_i.

Notice again the relationship with the SVE - filter out the terms which are noise contaminated.

$\gamma_i \approx 1$ for large σ_i, $\gamma_i \approx 0$ for small σ_i

Spectral filtering is used to filter the components in the spectral basis, such that noise in signal is damped.

How to chose filter factors γ_i?

Truncated SVD takes $\gamma_i = 1$, $1 \le i \le k$ and 0 otherwise to obtain solution $\mathbf{x}_k$.

- **Tikhonov** $\gamma_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2}$, $i = 1 \dots r$, λ is the regularization parameter, and solution is

$$\mathbf{x}_\lambda = \arg\min_{\mathbf{x}} \{ \|\mathbf{b} - A\mathbf{x}\|^2 + \lambda^2 \|\mathbf{x}\|^2 \}$$

- Regularized solution trades of $\|\mathbf{x}\|^2$ against $\|\mathbf{b} - A\mathbf{x}\|^2$.
- Notice

$$\gamma_i = \begin{cases} 1 - (\frac{\lambda}{\sigma_i})^2 + O(|\frac{\lambda}{\sigma_i}|^4) & \sigma_i \gg \lambda \\ (\frac{\sigma_i}{\lambda})^2 + O(|\frac{\sigma_i}{\lambda}|^4) & \sigma_i \ll \lambda \end{cases}$$

- If $\lambda \in [\sigma_r, \sigma_1]$, $\gamma_i \approx 1$ for small i, and $\gamma \approx (\sigma_i/\lambda)^2$ for large i (small σ_i)
- **Conclude** Parameter λ controls the filtering. If $\lambda \approx \gamma_k$, then filtered solution does not include components related to $\sigma_{k+1} \dots \sigma_r$.
- **Moreover** it is sensible to keep $\lambda \in [\sigma_r, \sigma_1]$.

- Again, now noting error in $\mathbf{b} = \mathbf{b}_{\text{exact}} + \mathbf{e}$

$$\begin{aligned} \mathbf{x}_{\text{filt}} &= V\Sigma_{\text{filt}}^{\dagger}U^{T}(\mathbf{b}_{\text{exact}} + \mathbf{e}) \\ &= V\Sigma_{\text{filt}}^{\dagger}U^{T}(U\Sigma V^{T}\mathbf{x}) + V\Sigma_{\text{filt}}^{\dagger}U^{T}\mathbf{e} \\ &= V\Gamma V^{T}\mathbf{x} + V\Gamma\Sigma^{\dagger}U^{T}\mathbf{e} \\ \text{implies} \quad \mathbf{x} - \mathbf{x}_{\text{filt}} &= (I_n - V\Gamma V^{T})\mathbf{x} - V\Gamma\Sigma^{\dagger}U^{T}\mathbf{e} \\ &= \text{Regularization Error} \qquad \text{Perturbation Error} \end{aligned}$$

- **Regularization Error** due to using Σ_{filt} in place of Σ.
- **Perturbation Error** the inverted and filtered noise, consistently zero if $\Gamma = 0$.

Size of the Regularization Error

- Notice

$$
\begin{aligned}
\|(I_n - V\Gamma V^T)\mathbf{x}\|_2^2 &= \|(I_n - \Gamma)V^T\mathbf{x}\|_2^2, \quad V \quad \text{orthogonal} \\
&= \|(I_n - \Gamma)\Sigma^\dagger U^T\mathbf{b}\|_2^2 \\
&= \sum_{i=1}^{n}((1-\gamma_i)\frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i})^2
\end{aligned}
$$

- $|\frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}|$ decays on average by Picard condition
- For small i (σ_i big) $\gamma_i \approx 1$, and $(1-\gamma_i) \approx 0$. Little error from large $|\frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}|$
- For large i $(1-\gamma_i) \approx 1$ provides little damping of smaller $|\frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}|$
- Choice of Γ controls size of regularization error.

Summary: Nov 2

1. Discretization of the Integral Equation
2. Galerkin approach for discretization leads to relationship of SVD and SVE
3. Discrete Picard Condition
4. Truncated SVD and filter factors
5. Basic Tikhonov Solution
6. Parameter determines the filtering

Review the Tikhonov Regularization
L-Curve for Parameter Estimation
Definition of Resolution/Influence Matrix
Review of Statistical Results
Unbiased Predictive Risk
Generalized Cross Validation
Rosemary Renaut
November 9, 2011

Consider general overdetermined discrete problem

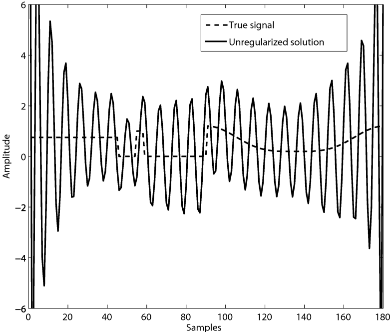
$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{m\times n}, \quad \mathbf{b} \in \mathbb{R}^m, \quad \mathbf{x} \in \mathbb{R}^n, \quad m \geq n.$$

Fit to data functional of the least squares problem is $\|A\mathbf{x} - \mathbf{b}\|_2^2$

Define $\mathbf{x}_{\mathrm{LS}} = \arg\min_{\mathbf{x}} \|A\mathbf{x}-\mathbf{b}\|^2$ and $\hat{\mathbf{x}}$ by $A\hat{\mathbf{x}} = \hat{\mathbf{b}}$, *i.e.* $\hat{\mathbf{b}} \in \mathrm{Range}(A)$

We know that $\mathbf{x}_{\mathrm{LS}}$ is noise contaminated if A is ill-conditioned.

Example: noise in $\mathbf{b}$ is $\eta \sim \mathrm{N}(0, 10^{-7})$ (normally distributed, mean 0 and variance 10^{-7})



Tikhonov Regularization : add a penalty term with regularization parameter $\lambda > 0$

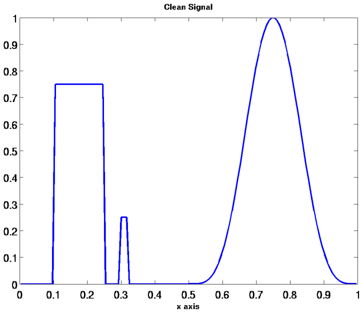
$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}}\{\|A\mathbf{x} - \mathbf{b}\|^2 + \lambda^2\|\mathbf{x}\|^2\}$$

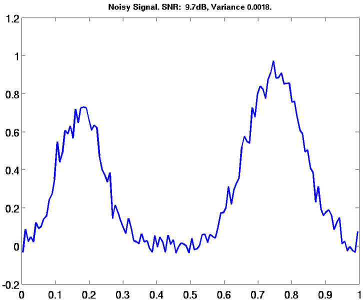
Regularized solution trades of $\|\mathbf{x}(\lambda)\|^2$ against $\|\mathbf{b} - A\mathbf{x}(\lambda)\|^2$.

For small λ the fit to data term is more closely enforced and the solution may be noisy

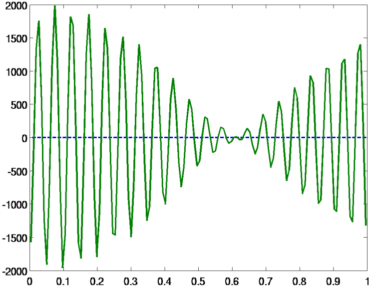
For larger λ the regularization term is enforced and so the solution is smoothed.

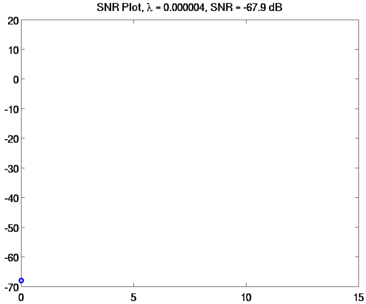
1-D Original and Noisy Signal





Solution for Different Choices of λ

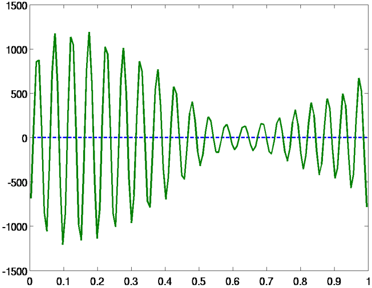


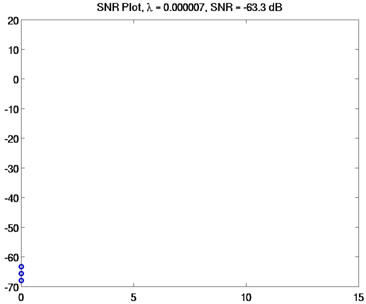


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}}-\mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Solution for Different Choices of λ

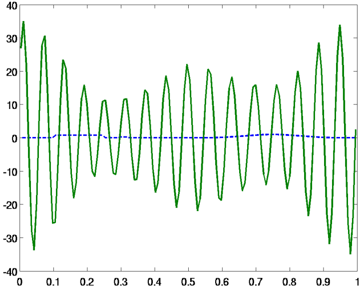


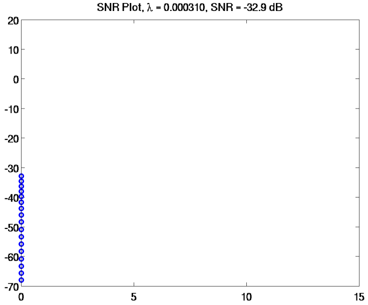


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}}-\mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Solution for Different Choices of λ

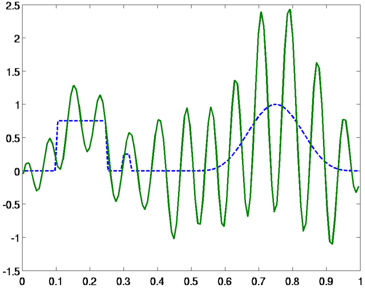


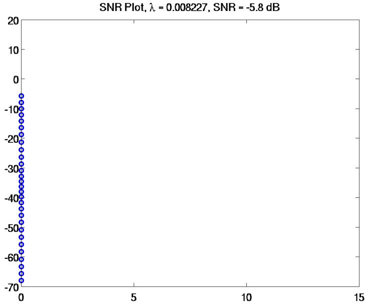


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}}-\mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

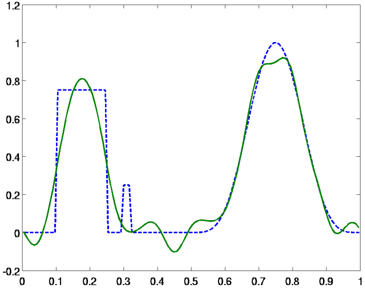
Solution for Different Choices of λ

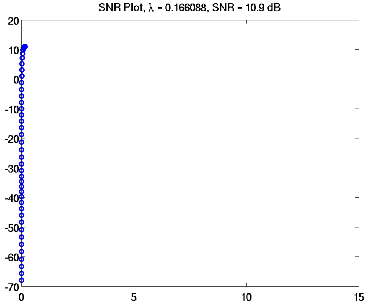




Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}}-\mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

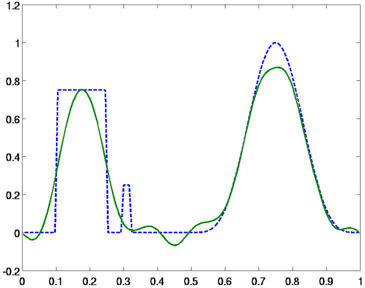


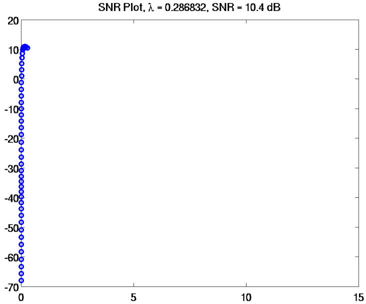


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}} - \mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Solution for Different Choices of λ

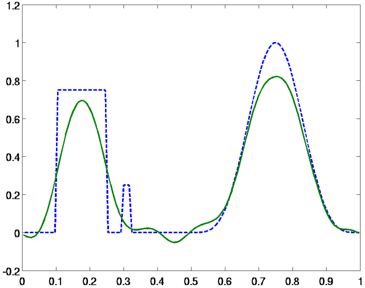


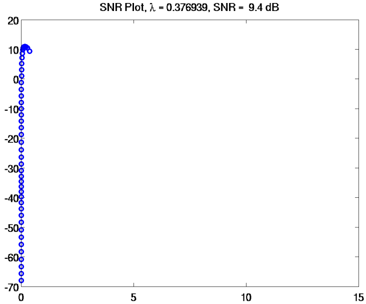


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}}-\mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Solution for Different Choices of λ

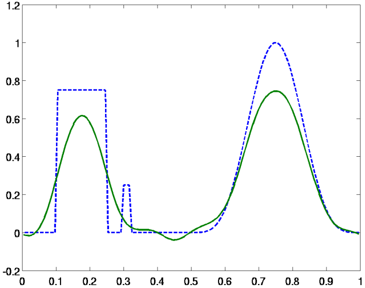


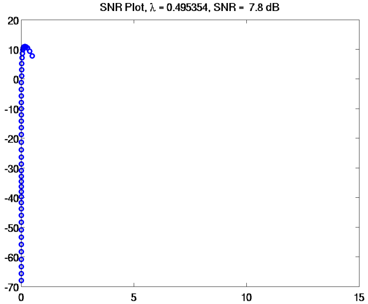


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}} - \mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Solution for Different Choices of λ

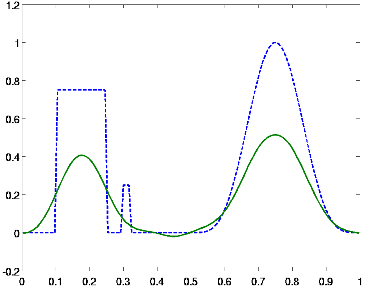


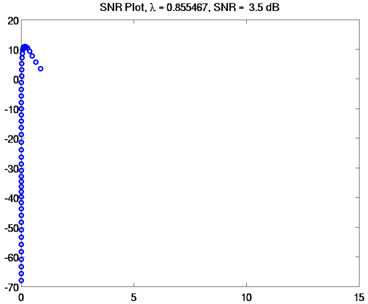


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}} - \mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Solution for Different Choices of λ

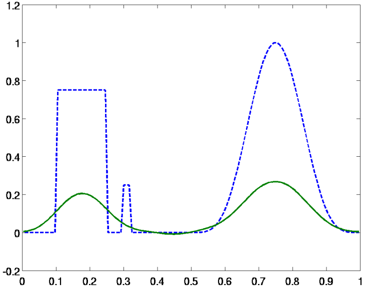


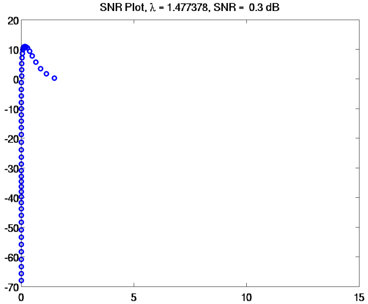


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10 \log_{10} \|\hat{\mathbf{x}}\| / \|\hat{\mathbf{x}} - \mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Solution for Different Choices of λ

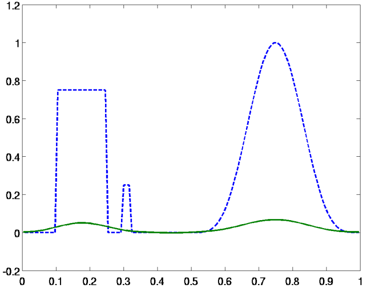


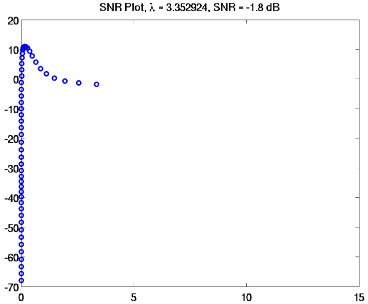


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10 \log_{10} \|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}} - \mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Solution for Different Choices of λ

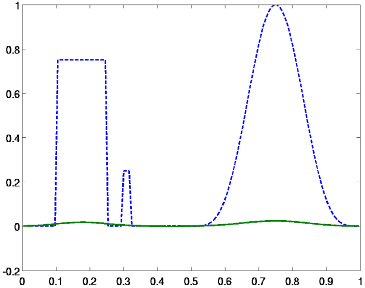


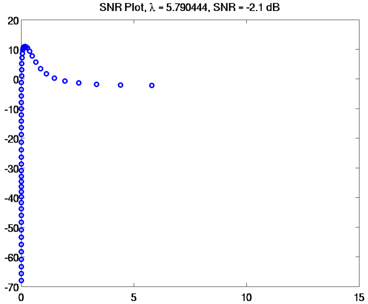


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}}-\mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Solution for Different Choices of λ

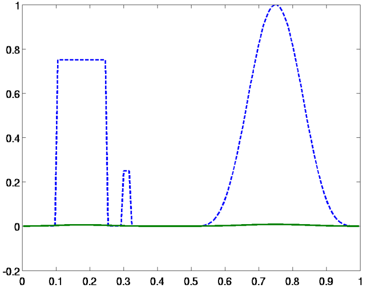


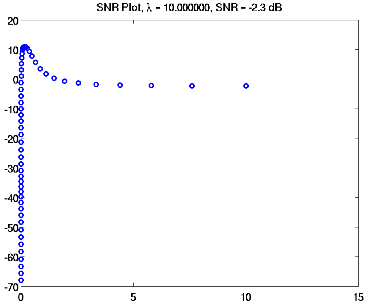


Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}}-\mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Solution for Different Choices of λ





Solutions $\mathbf{x}(\lambda)$

The Signal to Noise Ratio of the solution $10\log_{10}\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}} - \mathbf{x}(\lambda)\|$, true solution $\hat{\mathbf{x}}$

Investigating the Regularized solution

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}}\{\|A\mathbf{x} - \mathbf{b}\|^2 + \lambda^2\|\mathbf{x}\|^2\}$$

has equivalent formulation

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}} \left\| \begin{pmatrix} A \\ \lambda I_n \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b} \\ 0_n \end{pmatrix} \right\|^2$$

In practice we use iterative methods for the second formulation.

For theoretical discussion we solve the normal equations:

Regularization matrix

$$\mathbf{x}(\lambda) = R(\lambda)\mathbf{b} \text{ where } R(\lambda) = (A^T A + \lambda^2 I)^{-1} A^T$$

Matrix R is also sometimes referred to as the generalized inverse. Denoted by $A^\sharp$ in Vogel.

Bias and Variance in the Solution: exact values $\hat{\mathbf{x}}$ and $\hat{\mathbf{b}}$ not available

Model error (not computable)

$$\begin{aligned}
\mathbf{e}(\lambda) = \mathbf{x}(\lambda) - \hat{\mathbf{x}} &= R(\lambda)(\mathbf{b}) - \hat{\mathbf{x}} \\
&= R(\lambda)(\hat{\mathbf{b}} + \boldsymbol{\eta}) - \hat{\mathbf{x}} = R(\lambda)A\hat{\mathbf{x}} - \hat{\mathbf{x}} + R(\lambda)\boldsymbol{\eta} \\
&= \underbrace{(R(\lambda)A - I_n)\hat{\mathbf{x}}}_{\text{bias}} + \underbrace{R(\lambda)\boldsymbol{\eta}}_{\text{variance}} \\
&= (V\Gamma V^T - I_n)\hat{\mathbf{x}} + V\Gamma\Sigma^{\dagger}U^T\boldsymbol{\eta}
\end{aligned}$$

Bias the loss of information introduced by regularization

Express $\hat{\mathbf{x}} = VV^T\hat{\mathbf{x}}$ and rewrite $(V\Gamma V^T - I_n)\hat{\mathbf{x}}$ then we can filter so that the bias tends to zero:

$$\begin{array}{ll}
\Gamma_{\text{TSVD}} = \operatorname{diag}(I_k, 0_{n-k}) & (\sum_{i=k+1}^{n}(\mathbf{v}_i^T\hat{\mathbf{x}})\mathbf{v}_i) \to 0 \text{ as } k \to n \\
\Gamma_{\text{TIK}} = \operatorname{diag}(\gamma_i) & \sum_{i=1}^{n}(1-\gamma_i)(\mathbf{v}_i^T\hat{\mathbf{x}})\mathbf{v}_i = (\sum_{i=1}^{n}\frac{\lambda^2}{\sigma_i^2+\lambda^2}(\mathbf{v}_i^T\hat{\mathbf{x}})\mathbf{v}_i) \to 0 \text{ as } \lambda \to 0
\end{array}$$

Variance is the amplification of the noise in the error - the perturbation error also tends to zero by appropriate choice of λ

Estimating the model error

First express the variance as a sum

$$V\Gamma\Sigma^\dagger U^T\boldsymbol{\eta} = \sum_{i=1}^{n}(\mathbf{u}_i^T\boldsymbol{\eta})\frac{\gamma_i}{\sigma_i}\mathbf{v}_i = \sum_{i=1}^{n}(\mathbf{u}_i^T\boldsymbol{\eta})\frac{\sigma_i^2}{(\lambda^2+\sigma_i^2)\sigma_i}\mathbf{v}_i$$

Suppose in TSVD k is chosen by a threshold:

$$\gamma_k = 1 \text{ for } \sigma_k^2 > \lambda^2, \text{ and } \gamma_i = 0,\ i > k.$$

Consider $i \le k$ then $\gamma_i = 1 \le \sigma_i/\lambda$. Thus $\gamma_i \le \sigma_i/\lambda$ for all i.

Now for the Tikhonov filter observe $(\sigma_i^2/(\sigma_i^2+\lambda^2))/\sigma_i = (\sigma_i + \lambda^2/\sigma_i)^{-1}$

- For $\sigma_i^2 > \lambda^2$, $\sigma_i + \lambda^2/\sigma_i > \lambda$
- For $\sigma_i^2 \le \lambda^2$, $1/\sigma_i > 1/\lambda$, so $\lambda^2/\sigma_i > \lambda$ and $\sigma_i + \lambda^2/\sigma_i > \lambda$.

Hence in each case $\gamma_i/\sigma_i = (\sigma_i + \lambda^2/\sigma_i)^{-1} \le 1/\lambda$ and $\gamma_i \le \sigma_i/\lambda$ for all i

Suppose that $\|\boldsymbol{\eta}\|_2 < \delta^2$ then

$$\|\sum_{i=1}^{n}(\mathbf{u}_i^T\boldsymbol{\eta})\gamma_i/\sigma_i\mathbf{v}_i\|^2 = \sum_{i=1}^{n}|(\mathbf{u}_i^T\boldsymbol{\eta})|^2(\gamma_i/\sigma_i)^2 < (\delta/\lambda)^2.$$

If $\lambda = \delta^p$ where $p < 1$ then as $\delta \to 0$ variance error goes to zero. Notice that we use $\|U^T\boldsymbol{\eta}\|^2 = \|\boldsymbol{\eta}\|^2$ for U orthogonal,

If in addition $p > 0$ then the bias also goes to zero.

We conclude that λ can be chosen so that **e** goes to zero for both TSVD and Tikhonov

Note first that the model error cannot be computed.

Predictive error (not computable) requires $\mathbf{e}$

$$\mathbf{p}(\lambda) = A\mathbf{e}(\lambda) = A\mathbf{x}(\lambda) - A\hat{\mathbf{x}} = AR(\lambda)\mathbf{b} - \hat{\mathbf{b}}$$

Define Influence Matrix or Resolution Matrix $A(\lambda) = AR(\lambda)$.

$$A(\lambda) = AA^\sharp = A(A^TA + \lambda^2 I)^{-1}A^T$$

Regularized Residual or Predictive Risk is computable

$$\mathbf{p}(\lambda) = A\mathbf{x}(\lambda) - A\hat{\mathbf{x}} \approx (A\mathbf{x}(\lambda) - \mathbf{b}) = (A(\lambda) - I_m)\mathbf{b} := \mathbf{r}(\lambda)$$

It can be used to estimate the error.

How do we find λ? - L-curve

Plot regularization term against the fidelity term for λ

$$\log(\|\mathbf{x}(\lambda)\|), \log(\|A\mathbf{x}(\lambda) - \mathbf{b}\|)$$

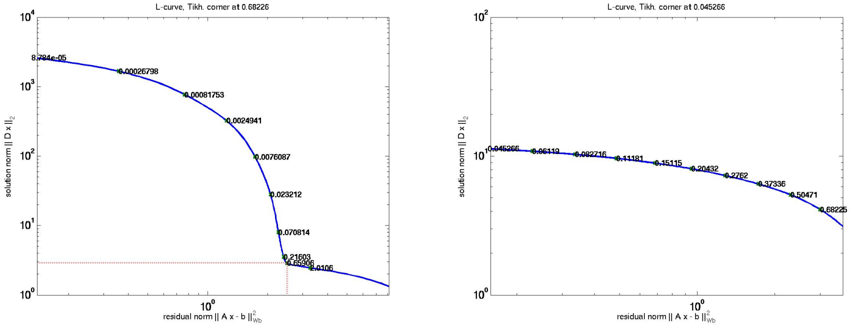


Figure: On the left a corner and on the right no corner. The L-curve is expensive for general matrices A, but is very general and straightforward. It does not consider any information on the noise structure

Derivation of the Unbiased Predictive Risk for Estimating λ

Consider the predictive error - cannot be calculated

$$
\begin{aligned}
\mathbf{p}(\lambda) &= A\mathbb{R}\mathbf{b} - \hat{\mathbf{b}} = A(\lambda)(\hat{\mathbf{b}} + \boldsymbol{\eta}) - \hat{\mathbf{b}} \\
&= \underbrace{(A(\lambda) - I_m)\hat{\mathbf{b}}} + \underbrace{A(\lambda)\boldsymbol{\eta}} \\
&= \text{deterministic} \quad \text{stochastic}
\end{aligned}
$$

Consider the predictive risk - can be calculated

$$
\begin{aligned}
\mathbf{r}(\lambda) &= (A(\lambda) - I_m)\mathbf{b} \\
&= \underbrace{(A(\lambda) - I_m)\hat{\mathbf{b}}} + \underbrace{(A(\lambda) - I_m)\boldsymbol{\eta}} \\
&= \text{deterministic} \qquad \text{stochastic}
\end{aligned}
$$

Both expressions use the noise $\boldsymbol{\eta}$.

We need to some statistical results.

Mean-Variance Suppose random vector $\mathbf{x}$ has mean $\mathbf{x}_0$, covariance-variance matrix Σ.

- we say $\mathbf{x} \sim (\mathbf{x}_0, \Sigma)$
- Then $\mathbf{b} \sim (A\mathbf{x}_0, A\Sigma A^T)$

Trace Operator is linear. trace$(A+B)$ =trace(A)+trace(B) and trace(A^T) =trace(A). We note also the cyclic property trace(ABC) =trace(CAB), provided that dimensions are consistent.

Definition (Discrete White Noise Vector)

A random vector $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_n)$ *is a discrete white noise vector provided that* $E(\boldsymbol{\eta}) = 0$ *and* $\mathrm{cov}(\boldsymbol{\eta}) = \sigma^2 I_n$*. i.e.*

$$\mathrm{E}(\eta_i) = 0, \quad \mathrm{E}(\eta_i \eta_j) = \sigma^2 \delta_{ij}^2$$

σ^2 *is the variance of the white noise*

Lemma (Trace Lemma)

Let $\mathbf{y}$ *be deterministic and* $\boldsymbol{\eta}$ *a discrete white noise vector. Then*

$$\mathrm{E}(\|\mathbf{y} + A\boldsymbol{\eta}\|^2) = \|\mathbf{y}\|^2 + \sigma^2 \mathrm{trace}(A^T A)$$

Obtaining the Estimate

Use the Trace lemma and assume that the noise vector $\boldsymbol{\eta}$ is a discrete white noise vector. Estimate mean predictive error from
$\mathrm{E}(\|\mathbf{p}(\lambda)\|^2)$ and $\mathrm{E}(\|\mathbf{r}(\lambda\|^2)$

$$\begin{aligned}
\mathrm{E}(\|\mathbf{r}(\lambda)\|^2) &= \mathrm{E}(\|(A(\lambda) - I_m)\hat{\mathbf{b}} + (A(\lambda) - I_m)\boldsymbol{\eta}\|^2)) \\
&= \|(A(\lambda) - I_m)\hat{\mathbf{b}}\|^2 + \sigma^2 \mathrm{trace}((A(\lambda) - I_m)^T(A(\lambda) - I_m)) \text{ and} \\
\mathrm{E}(\|\mathbf{p}(\lambda)\|^2) &= \mathrm{E}(\|(A(\lambda) - I_m)\hat{\mathbf{b}} + A(\lambda)\boldsymbol{\eta}\|^2)) \\
&= \|(A(\lambda) - I_m)\hat{\mathbf{b}}\|^2 + \sigma^2 \mathrm{trace}(A(\lambda)^T A(\lambda)) \\
&= \mathrm{E}(\|\mathbf{r}(\lambda)\|^2) + \sigma^2 \mathrm{trace}(A(\lambda)^T A(\lambda)) - \sigma^2 \mathrm{trace}((A(\lambda) - I_m)^T(A(\lambda) - I_m)) \\
&= \mathrm{E}(\|\mathbf{r}(\lambda)\|^2) + \sigma^2(2\,\mathrm{trace}(A(\lambda)) - m) \text{ by linearity of trace} \\
&\approx \|\mathbf{r}(\lambda)\|^2 + \sigma^2(2\,\mathrm{trace}(A(\lambda)) - m) := U(\lambda)
\end{aligned}$$

Notice that $\mathrm{E}(U(\lambda)) = \mathrm{E}(\|\mathbf{p}(\lambda)\|^2)$ so that U is an unbiased estimator.

Thus seek λ such that U is minimum :

$$\lambda = \arg\min_{\lambda}(U(\lambda)).$$

An Example with UPRE

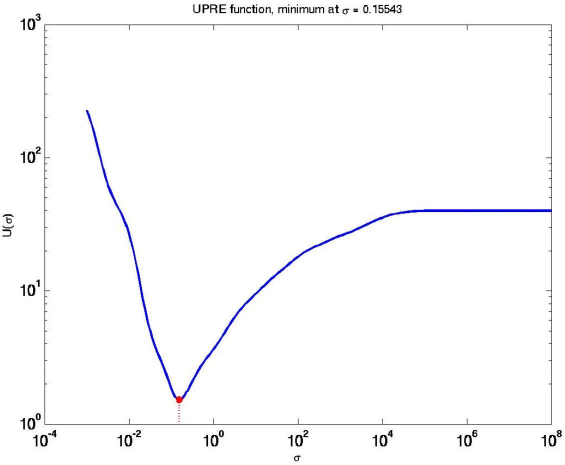


Figure: Notice the well-defined minimum. But requires the calculation of the trace

How do we use UPRE: First with the SVD rewrite $A(\lambda) - I_m$

$$
\begin{aligned}
A(\lambda) - I_m &= A(A^TA + \lambda^2 I_n)^{-1}A^T - I_m \\
&= U\Sigma V^T (V\Sigma^T U^T U \Sigma V^T + \lambda^2 VV^T)^{-1} V\Sigma^T U - UU^T \\
&= U(\Sigma(\Sigma^T\Sigma + \lambda^2 I_n)^{-1}\Sigma^T - I_m)U^T \quad \text{Hence with } \sigma_i = 0, i > n \\
\text{trace}(A(\lambda)) &= \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \text{ and } (A(\lambda) - I_m)\mathbf{b} = \sum_{i=1}^{m} (\mathbf{u}_i^T\mathbf{b}) \frac{-\lambda^2}{\sigma_i^2 + \lambda^2}\mathbf{u}_i. \quad \text{Thus} \\
\|\mathbf{r}(\lambda)\|^2 &= \sum_{i=1}^{m} |\mathbf{u}_i^T\mathbf{b}|^2 \left(\frac{\lambda^2}{\sigma_i^2 + \lambda^2}\right)^2 \text{ and} \\
U(\lambda) &= \sum_{i=1}^{m} |\mathbf{u}_i^T\mathbf{b}|^2 \left(\frac{\lambda^2}{\sigma_i^2 + \lambda^2}\right)^2 + \sigma^2 (2\sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} - m).
\end{aligned}
$$

Entries $|\mathbf{u}_i^T\mathbf{b}|$ and σ_i are calculated once. Hence given the SVD the cost is basically $O(n)$ for each λ. Basic idea is to bracket the minimum and then minimize within the bracket.

For the Fourier expansion a similar result can be obtained. (Suggest you try to derive this)

Otherwise the trace is estimated, say using randomization. (Reading material?)

Generalized Cross Validation: GCV

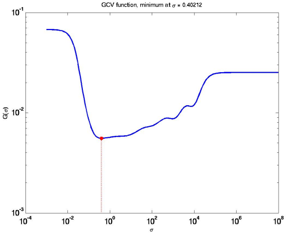
Based on omitting a data value and testing predictability of solution for this missing value: λ is chosen to minimize

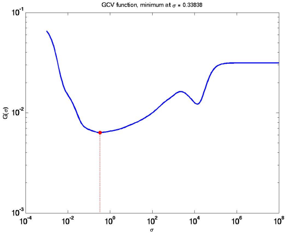
$$G(\lambda) = \frac{\|\mathbf{r}(\lambda)\|_2^2}{(\mathrm{trace}(I_m - A(\lambda)))^2}.$$

GCV provides another estimate of the predictive risk: (see e.g. Vogel)

There are the same difficulties of using the trace for large scale problems.

Often G is relatively flat near the minimum, or has multiple minima and thus difficult to apply.





Summary: Nov 9

1. Regularization is needed
2. Bias and Variance in the solution
3. The L-curve
4. Definitions and review of essential statistical properties
5. Derivation of the Unbiased Predictive Risk Estimate

Recall Tikhonov Regularization
Properties of the residual - using the discrepancy principle and NCP
Weighting by the noise to apply techniques for non white data
χ^2 property of the global residual - discrepancy and NCP
Shifted solutions
Alternative Operators for Regularization
Generalized Singular Value Decomposition
Basic Iterative Techniques: LSQR
Rosemary Renaut
November 16, 2011

Why not just use $\mathbf{r}(\lambda)$ - rather than deal with the UPRE or GCV: The discrepancy principle

Calculate $\mathbf{r}(\lambda) = A\mathbf{x}(\lambda) - \mathbf{b} = \sum_{i=1}^{m} (\lambda^2)/(\sigma_i^2 + \lambda^2)(\mathbf{u}_i^T \mathbf{b})\mathbf{u}_i$
Suppose that $\mathbf{x}(\lambda) \approx \hat{\mathbf{x}}$ then $\mathbf{r}(\lambda) \approx A\hat{\mathbf{x}} - \mathbf{b} = \boldsymbol{\eta}$.
Hence $\mathrm{E}(\frac{1}{m}\|A\mathbf{x}(\lambda) - \mathbf{b}\|^2) \approx \mathrm{E}(\frac{1}{m}\|\boldsymbol{\eta}\|^2) = \sigma^2$, where $\boldsymbol{\eta} \sim (0, \sigma^2 I)$.

Discrepancy Principle $\boxed{\text{Find } \lambda \text{ such that } \|\mathbf{r}(\lambda)\|^2 \approx m\sigma^2.}$

Is $\mathbf{x}(\lambda) \approx \hat{\mathbf{x}}$ reasonable? Consider

$$F(\lambda) = \|\mathbf{r}(\lambda)\|^2 = \sum_{i=1}^{m} \left(\lambda^2/(\sigma_i^2 + \lambda^2)\right)^2 |\mathbf{u}_i^T \mathbf{b}|^2$$

Clearly $F(\lambda)$ is continuous, $F(0) = 0$ and $F(\lambda) \to \|\mathbf{b}\|^2$ for $\lambda \to \infty$. Hence there exists unique λ such that $F(\lambda) = \delta^2$, where $\delta = \|\boldsymbol{\eta}\|$, provided $\delta < \|\mathbf{b}\|$.

For this λ,

$$\begin{aligned}\delta^2 + \lambda^2\|\mathbf{x}(\lambda)\|^2 &= \|A\mathbf{x}(\lambda) - \mathbf{b}\|^2 + \lambda^2\|\mathbf{x}(\lambda)\|^2 \leq \|A\hat{\mathbf{x}} - \mathbf{b}\|^2 + \lambda^2\|\hat{\mathbf{x}}\|^2 \\ &= \|A\hat{\mathbf{x}} - \hat{\mathbf{b}} - \boldsymbol{\eta}\|^2 + \lambda^2\|\hat{\mathbf{x}}\|^2 = \|\boldsymbol{\eta}\|^2 + \lambda^2\|\hat{\mathbf{x}}\|^2 = \delta^2 + \lambda^2\|\hat{\mathbf{x}}\|^2.\end{aligned}$$

where the first inequality arises from the definition of $\mathbf{x}(\lambda)$ as the argmin of the regularized functional.

Thus $\|\mathbf{x}(\lambda)\|^2 \leq \|\hat{\mathbf{x}}\|^2$. While it seems that the assumption is reasonable, we anticipate that smoothing may occur to achieve $\|\mathbf{x}(\lambda)\|^2 \leq \|\hat{\mathbf{x}}\|^2$.

Some observations: Parameter Choice is Difficult

L-curve	Discrepancy	UPRE	GCV
▶ No unique solution (depends on finding the L) ▶ Requires multiple solves to find appropriate range for the corner ▶ Does not use noise level ▶ Easily justified.	▶ Unique solution ▶ Direct solve using Newton method ▶ Requires noise level ▶ Leads to a smoothed solution	▶ No unique solution ▶ Gives good estimates ▶ Requires noise level ▶ Minimization required ▶ Needs trace estimator	▶ Multiple minima or flat ▶ Minimization required ▶ Needs trace estimator

We again look at the residual

$$\begin{aligned}\mathbf{r}(\lambda) = A\mathbf{x}(\lambda) - \mathbf{b} &= \sum_{i=1}^{m}(\gamma_i - 1)\frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{u}_i \\ &= \sum_{i=1}^{m}(\gamma_i - 1)\frac{\mathbf{u}_i^T\hat{\mathbf{b}} + \mathbf{u}_i^T\boldsymbol{\eta}}{\sigma_i}\mathbf{u}_i\end{aligned}$$

The filter needs to be chosen so that the contributions from the reliable coefficients $|\mathbf{u}_i^T\hat{\mathbf{b}}| > |\mathbf{u}_i^T\boldsymbol{\eta}|$ dominate in $\mathbf{x}(\lambda)$.

Equivalently, we want to pick λ so that $\mathbf{r}(\lambda)$ is dominated by contributions from $|\mathbf{u}_i^T\boldsymbol{\eta}|$

We need to find λ so that $\mathbf{r}(\lambda)$ is *noise like*

i.e. we need to test whether $\mathbf{r}(\lambda) \in \mathbb{R}^m$ is a vector of standard normal entries.

Use time series analysis.

Diagnostic 1 Morozov's discrepancy principle $1/\sigma^2\|\mathbf{r}(\lambda)\|^2 \approx m$? Precisely is

$$m - 2\sqrt{2m} \leq \frac{1}{\sigma^2}\|\mathbf{r}(\lambda)\|^2 \leq m + 2\sqrt{2m}.$$

Is $1/\sigma^2\|\mathbf{r}(\lambda)\|^2$ within two standard deviations of $1/\sigma^2 E(\|\mathbf{r}(\lambda)\|^2)$?

Diagnostic 2 Is histogram of $\mathbf{r}(\lambda)_i$ consistent with $\mathbf{r}(\lambda) \sim \mathcal{N}_m(0, \sigma)$?

Diagnostic 3 Do entries $\mathbf{r}(\lambda)_i$ match the expectation that they are selected from a time series indexed by i for time?

We use the *periodogram* or *power spectrum* for $\mathbf{r}(\lambda)$.

Let dft be the discrete Fourier transform and $\mathbf{c}$ be the vector of entries

$$\mathbf{c}_j = (\mathrm{dft}(\mathbf{r}(\lambda)))_j, \quad j = 1, \dots, \tilde{m} = \lfloor m/2 \rfloor + 1.$$

$\mathbf{c} = F\mathbf{r}(\lambda)$ for F the Fourier transform matrix with normalization $1/\sqrt{m}$.

$\mathbf{c}_1$ is the 0 (DC) constant frequency component.

$\mathbf{c}_{\tilde{m}}$ is the highest frequency in the signal

For $\mathbf{r}(\lambda) \sim \mathcal{N}_m(0, \sigma_r^2 I_m)$, and row f_i^T of F

$$E(|\mathbf{c}_i|^2) = E(|f_i^T\mathbf{r}(\lambda)|^2) = \mathrm{Var}(f_i^T\mathbf{r}(\lambda)) = \sigma_r^2(f_i^T(f_i^T)^*), \rightarrow E(|\mathbf{c}_i|^2) \propto \sigma_r^2$$

where the proportionality depends on the normalization in F.

Normalized Cumulative Periodogram

Diagnostic 1 Parseval's relation: $\|\mathrm{dft}(\mathbf{r}(\lambda))\|^2 = \|\mathbf{r}(\lambda)\|^2$

Diagnostic 2 & 3 **Normalized Cumulative Periodogram**(NCP):

Calculate the ratio of the cumulative sum of entries

$$\mathbf{w}_j = \frac{\sum_{i=1}^{j} |\mathbf{c}_j|^2}{\sum_{i=1}^{\tilde{m}} |\mathbf{c}_i|^2}, \quad j = 1, \dots, \tilde{m},$$

Because $E(|\mathbf{c}_i|^2) \propto \sigma_r^2$, for all i, the power spectrum is flat: $\mathbf{w}_j \approx 2j/m$.

Define $\tilde{j} = (j-1)/(m-1)$, $0 \leq \tilde{j} \leq .5$.

Then $(\tilde{j}, \mathbf{w}_j)$ is a straight line $\mathbf{w}_j \approx 2\tilde{j}$ of length $\sqrt{5}/2$.

Ideally $\mathbf{w} = \tilde{\mathbf{w}}$, $\tilde{\mathbf{w}}_j = 2j$

For 5% significance level the NCP is within Kolmogorov Smirnoff limits $\pm 1.35\tilde{m}^{-1/2}$

How to test automatically: various options - Hansen's NCP test

Minimize the deviation from straight line $\lambda_{\mathrm{NCP}} = \arg\min_\lambda \|\mathbf{w}(\lambda) - \tilde{\mathbf{w}}\|_2$

Example NCP

Figure: Deviation from Straight line with λ

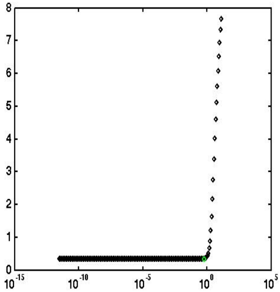
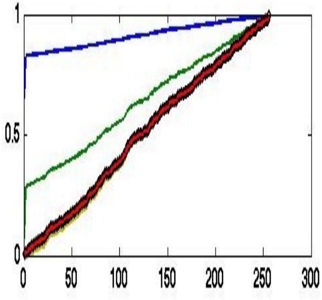


Figure: NCP for some different λ



Notice the reverse L for the deviation: hence the corner can be found.

Comparing the Residual Periodogram and the Cumulative Residual

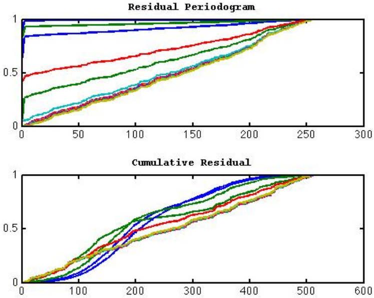


Figure: Comparing Residual and Periodogram: Various λ

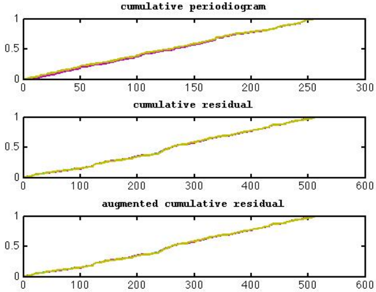


Figure: Comparing Residual, augmented residual and periodogram: Various ideal λ

Summary: Nov 16

1. Tikhonov Regularization
2. Residual Properties
3. Discrepancy Principle
4. Residual Periodogram

Weighting by the noise to apply techniques for non white data
χ^2 property of the global residual - discrepancy and NCP
Shifted solutions with prior information
Alternative Operators for Regularization
Generalized Singular Value Decomposition
Basic Iterative Techniques: LSQR

Rosemary Renaut

November 30, 2011

Weighting for the noise - results assumed white noise

Suppose that $\boldsymbol{\eta} \sim (0, C_{\mathrm{b}})$, i.e. C_{b} is symmetric positive definite covariance of the noise in $\boldsymbol{\eta}$.

C_{b} is SPD, there exists a factorization $C_{\mathrm{b}} = D^2$ and D is invertible. ($C_{\mathrm{b}}^{1/2} = D$)

To whiten the noise we multiply by D^{-1} in the equation $A\hat{\mathbf{x}} = \hat{\mathbf{b}} = \mathbf{b} + \boldsymbol{\eta}$

$$D^{-1}(A\hat{\mathbf{x}} - \mathbf{b}) = D^{-1}\boldsymbol{\eta} = \bar{\boldsymbol{\eta}}, \text{ where } \bar{\boldsymbol{\eta}} \sim (0, D^{-1}C_{\mathrm{b}}(D^{-1})^T) = (0, I_m)$$

Hence rather than solving for an unweighted fidelity term we solve the weighted problem, $W = C_{\mathrm{b}}^{-1}$

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}}\{\|A\mathbf{x} - \mathbf{b}\|_W^2 + \lambda^2\|\mathbf{x}\|^2\}$$

using the standard notation $\|A\|_W^2 = A^T W A$.

This may be immediately rewritten as before by defining $\tilde{A} = W^{1/2}A$ and $\tilde{\mathbf{b}} = W^{1/2}\mathbf{b}$

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}}\{\|\tilde{A}\mathbf{x} - \tilde{\mathbf{b}}\|^2 + \lambda^2\|\mathbf{x}\|^2\}$$

Does the weighting change the solution we obtain?

Use SVD for the matrix pair $\tilde{A}$ instead of A, and apply all operations for the weighted fidelity term

Statistical Properties of the Augmented Regularized Residual

Consider weighted regularized residual $\mathbf{r}(\lambda)_{\mathrm{A}}$, W_{x} is a SPD weighting on $\mathbf{x}$

$$\mathbf{x}(W_{\mathrm{x}}) = \arg\min_{\mathbf{x}} \left\| \begin{pmatrix} \tilde{A} \\ W_{\mathrm{x}}^{1/2} \end{pmatrix} \mathbf{x} - \begin{pmatrix} \tilde{\mathbf{b}} \\ 0_n \end{pmatrix} \right\|^2 := \arg\min_{\mathbf{x}} \|\mathbf{r}(\lambda)_{\mathrm{A}}\|^2$$

For a given solution $\mathbf{x}(W_{\mathrm{x}})$ we can calculate the cost functional

$$J(W_{\mathrm{x}}) = \mathbf{b}^T(A^T W_{\mathrm{x}}^{-1} A + W^{-1})^{-1}\mathbf{b}, \quad \mathbf{x}(W_{\mathrm{x}}) = W_{\mathrm{x}}^{-1}A^T(A^T W_{\mathrm{x}}^{-1} A + W^{-1})^{-1}\mathbf{b}$$

Using

$$(A^T BA + C)^{-1}A^T B = C^{-1}A^T(AC^{-1}A^T + B^{-1})^{-1}$$

with $B = W_{\mathrm{x}}^{-1}$ and $C = W^{-1}$.

Using the factorization $W_{\mathrm{x}}^{-1} = W_{\mathrm{x}}^{-1/2}W_{\mathrm{x}}^{-1/2}$, and the SVD for $\tilde{A}$ we can obtain

$$J(W_{\mathrm{x}}) = \mathbf{s}^T P\mathbf{s}, \quad \mathbf{s} = U^T W^{1/2}\mathbf{b}, \quad P = \Sigma V^T W_{\mathrm{x}} V \Sigma^T + I_m$$

Distribution of the Cost Functional If W and W_{x} have been chosen *appropriately functional J is a random variable which follows a χ^2 distribution with m degrees of freedom:*

$$J(W_{\mathrm{x}}) \sim \chi^2(m)$$

Appropriate weighting makes noise on $\mathbf{b}$ and on model $\mathbf{x}$ white

Of course noise on $\mathbf{x}$ is unknown, but this determines a parameter choice rule for $W_{\mathrm{x}} = \lambda^2 I$ using the *augmented discrepancy*.

χ^2 method to find the parameter (Mead and Renaut)

Interval Find $W_{\mathrm{x}} = \lambda^2 I$ such that

$$m - \sqrt{2m}z_{\alpha/2} < \mathbf{b}^T(A^T W_{\mathrm{x}}^{-1} A + W^{-1})^{-1}\mathbf{b} < m + \sqrt{2m}z_{\alpha/2}.$$

i.e. $E(J(\mathbf{x}(W_{\mathrm{x}}))) = m$ and $\mathrm{Var}(J) = 2m$.

Posterior Covariance on $\mathbf{x}$ Having found W_{x} the posterior inverse covariance matrix is

$$\tilde{W}_{\mathrm{x}} = A^T W A + W_{\mathrm{x}}$$

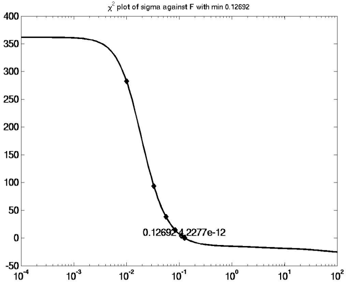
Root finding Find $\sigma_{\mathbf{x}}^2 = \lambda^{-2}$ such that

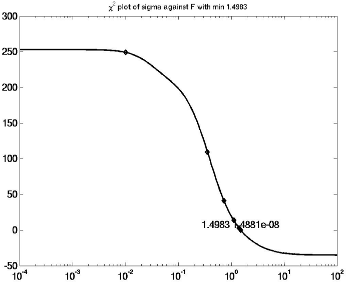
$$F(\sigma_{\mathbf{x}}) = \mathbf{s}^T \mathrm{diag}(\frac{1}{1 + \sigma_{\mathbf{x}}^2 \sigma_i^2})\mathbf{s} - m = 0.$$

Discrepancy Principle note the similarity

$$F(\sigma_{\mathbf{x}}) = \mathbf{s}^T \mathrm{diag}(\frac{1}{(1 + \sigma_{\mathbf{x}}^2 \sigma_i^2)^2})\mathbf{s} - m = 0.$$

Typical F: Two examples





Newton's method yields a unique solution (when one exists)

F is monotonically decreasing but possible $F > 0$ as $\sigma \to \infty$ which implies no regularization is needed.

$F < 0$ for all σ_x implies that the degrees of freedom is wrongly given, the noise on **b** was not correctly identified.

NCP for $\mathbf{r}(\lambda)_{\mathrm{A}}$

The overall residual $\mathbf{r}(\lambda)_{\mathrm{A}}$ is also white noise like provided *W* is chosen appropriately.

We can apply exactly the same NCP idea to this residual

Figure: Deviation from Straight line with λ

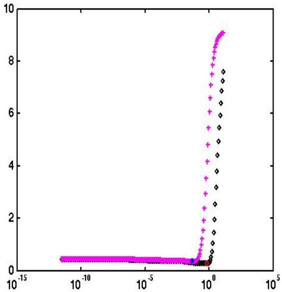
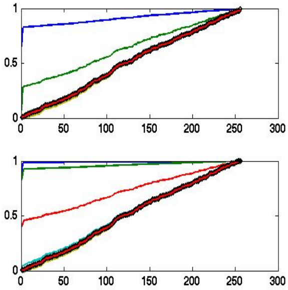


Figure: NCP for some different λ



In the comparison with the standard residual $\mathbf{r}$ is in black and $\mathbf{r}(\lambda)_{\mathrm{A}}$ in magenta. Optimal λ for the augmented residual is smaller than for the discrepancy, so the solution is smoothed less.

Extending the Regularization: Reference Solution

Background or reference solution may be known

$$\mathbf{y}(\lambda) = \arg\min_{\mathbf{y}}\{\|A\mathbf{y} - \mathbf{d}\|^2 + \lambda^2\|\mathbf{y} - \mathbf{y}_0\|^2\}$$

Solution (written as solution of the normal equations)

$$\begin{aligned}\mathbf{y}(\lambda) &= (A^TA + \lambda^2 I)^{-1}(A^T\mathbf{d} + \lambda^2\mathbf{y}) \\ &= (A^TA + \lambda^2 I)^{-1}(A^T(\mathbf{d} - A\mathbf{y}_0)) + \mathbf{y}_0\end{aligned}$$

Shifted problem with $\mathbf{b} = \mathbf{d} - A\mathbf{y}_0$, $A\mathbf{y}_0 = \mathbf{d} - \mathbf{b}$ and $\mathbf{y} - \mathbf{x} = \mathbf{y}_0$.

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}}\{\|A\mathbf{x} - \mathbf{b}\|^2 + \lambda^2\|\mathbf{x}\|^2\}$$

Then $\mathbf{y}(\lambda) = \mathbf{x}(\lambda) + \mathbf{y}_0$.

Extending the Regularization - Apply a different operator

Imposing the regularization for the norm of $\mathbf{x}$ is not necessarily appropriate, dependent on what we anticipate for the solution

Instead we consider the more general weighting $\|L\mathbf{x}\|^2$

This leads to the general problem (assuming also the weighting for the noise)

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}}\{\|\tilde{A}\mathbf{x} - \tilde{\mathbf{b}}\|^2 + \lambda^2\|L\mathbf{x}\|^2\}$$

Suppose that L is invertible then we can solve for $\mathbf{y} = L\mathbf{x}$ noting for the normal equations

$$(A^TA + L^TL)\mathbf{x} = A^T\mathbf{b} \rightarrow L^T(A^TAL^{-1} + I_n)L\mathbf{x} = L^TA^T\mathbf{b}$$

Typical L: L approximates the first or second order derivative

$$L_1 = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \quad L_2 = \begin{pmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{pmatrix}$$

$L_1 \in \mathbb{R}^{(n-1)\times n}$ and $L_2 \in \mathbb{R}^{(n-2)\times n}$. Note that neither L_1 nor L_2 are invertible.

Operators L_1 and L_2 provide approximations to derivatives

$$D_x(u_i) \approx u_{i+1} - u_i \quad D_{xx}(u_i) \approx u_{i+1} - 2u_i + u_{i-1}$$

Boundary is at u_1 and u_n.
Suppose zero outside the domain $u_0 = u_{n+1} = 0$

$$\begin{aligned} D_x(u_n) &= u_{n+1} - u_n = -u_n \\ D_{xx}(u_1) &= u_2 - 2u_1 + u_0 = u_2 - 2u_1 \\ D_{xx}(u_n) &= u_{n+1} - 2u_n + u_{n-1} = -2u_n + u_{n-1} \end{aligned}$$

$$L_1^0 = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & & -1 \end{pmatrix} \quad L_2^0 = \begin{pmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 \end{pmatrix}$$

$L_1^0, L_2^0 \in \mathbb{R}^{n\times n}$. Both L_1, L_2 are invertible.

Boundary Conditions: Reflexive

Suppose reflexive outside the domain $u_0 = u_2$, $u_{n+1} = u_{n-1}$

$$
\begin{aligned}
D_x(u_n) &= u_{n+1} - u_n = u_{n-1} - u_n \\
D_{xx}(u_1) &= u_2 - 2u_1 + u_0 = 2u_2 - 2u_1 \\
D_{xx}(u_n) &= u_{n+1} - 2u_n + u_{n-1} = -2u_n + 2u_{n-1}
\end{aligned}
$$

$$
L_1^R = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & 1 & -1 \end{pmatrix} \quad L_2^R = \begin{pmatrix} -2 & 2 & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 2 & -2 \end{pmatrix}
$$

L_1^R, $L_2^R \in \mathbb{R}^{n\times n}$. Neither L_1, L_2 invertible.

Suppose periodic outside the domain $u_0 = u_n$, $u_{n+1} = u_1$

$$\begin{aligned} D_x(u_n) &= u_{n+1} - u_n = u_1 - u_n \\ D_{xx}(u_1) &= u_2 - 2u_1 + u_0 = u_n + 2u_2 - u_1 \\ D_{xx}(u_n) &= u_{n+1} - 2u_n + u_{n-1} = u_1 - 2u_n + u_{n-1} \end{aligned}$$

$$L_1^P = \begin{pmatrix} -1 & 1 & & 0 \\ & \ddots & \ddots & \\ & & -1 & 1 \\ 1 & & & -1 \end{pmatrix} \quad L_2^P = \begin{pmatrix} -2 & 1 & & 1 \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ 1 & & 1 & -2 \end{pmatrix}$$

L_1^P, $L_2^P \in \mathbb{R}^{n \times n}$. Neither L_1, L_2 invertible.
Both are circulant banded.

Image X : rows in $x-$ direction and columns in $y-$ direction.

Let L be a derivative operator. Derivatives in the y direction are achieved by LX, and in the $x-$ direction by $(LX^T)^T = XL^T$.

If $\mathbf{x} = \mathrm{vec}(X)$, $LX = LX\,I = (I \otimes L)\mathbf{x}$ and $XL^T = IXL^T = (L \otimes I)X$.

Regularization terms $\|(I \otimes L)\mathbf{x}\|^2$ and $\|(L \otimes I)\mathbf{x}\|^2$

Note, say for L a first derivative operator,

$$\|(I \otimes L)\mathbf{x}\|^2 + \|(L \otimes I)\mathbf{x}\|^2 = \left\| \begin{pmatrix} I \otimes L \\ L \otimes I \end{pmatrix} \mathbf{x} \right\|_2^2$$

is not equivalent to

$$\|((I \otimes L) + (L \otimes I))\mathbf{x}\|^2$$

which is appropriate when we need for example the second order Laplacian.

Two Dimensions: Assume image is periodic apply Tikhonov Regularization

When A is the PSF matrix with periodic boundary conditions, and implementable using DFT matrices $F = F_r \otimes F_c$, the regularized Tikhonov solution can be written as

$$\mathbf{x}_{\lambda,L} = F^*(|\Lambda_A|^2(|\Lambda_A|^2 + \lambda^2\Delta)^{-1}\Lambda_A^{-1})F\mathbf{b}$$

Δ is the identity

Λ_A is matrix of eigenvalues of A.

$|\Lambda_A|^2(|\Lambda_A|^2 + \lambda^2\Delta)^{-1}\Lambda_A$ is diagonal and $|\Lambda_A|^2(|\Lambda_A|^2 + \lambda^2\Delta)^{-1}$ is the filtering matrix.

This corresponds to **Wiener filtering**.

Again suppose A implemented using $F = F_r \otimes F_c$, then derivatives are mapped to Fourier domain

$$(I \otimes L) \rightarrow F^*(I \otimes \Lambda_L)F \quad (L \otimes I) \rightarrow F^*(\Lambda_L \otimes I)F$$

$$\begin{pmatrix} I \otimes L \\ L \otimes I \end{pmatrix} = \begin{pmatrix} F^* & 0 \\ 0 & F^* \end{pmatrix} \begin{pmatrix} I \otimes \Lambda_L \\ \Lambda_L \otimes I \end{pmatrix} F$$

and thus regularization terms can all be expressed using the factorization, yielding

$$\mathbf{x}_{\lambda,L} = F^*(|\Lambda_A|^2|(\Lambda_A|^2 + \lambda^2\Delta)^{-1}\Lambda_A^{-1})F\mathbf{b}$$

where now Δ depends on the regularizer.

An Observation on shifting and smoothing

Find $\mathbf{y}(\lambda) = \arg\min_{\mathbf{y}}\{\|A\mathbf{y}-\mathbf{d}\|^2\}$ s.t. $\|L(\mathbf{y}-\mathbf{y}_0)\|^2$ is minimum.

TSVD solution: Let $V = [V_1, V_2] = [\mathbf{v}_1, \dots, \mathbf{v}_k, \mathbf{v}_{k+1}, \dots, \mathbf{v}_n]$, i.e. columns of V_1, V_2 span the effective range, null space, of A.

Consider the solution $\mathbf{y} = \mathbf{y}_k + \bar{\mathbf{y}} = \sum_{i=1}^k \frac{(\mathbf{u}_i^T\mathbf{d})}{\sigma_i}\mathbf{v}_i + V_2\mathbf{c}$

We want $\mathbf{y}$ such that $\|L(\mathbf{y}-\mathbf{y}_0)\|$ is minimum. Equivalently ($A^\dagger = (A^TA)^{-1}A^T$)

$$\|L(\mathbf{y}-\mathbf{y}_0)\| = \|L(\mathbf{y}_k-\mathbf{y}_0) + L\bar{\mathbf{y}}\| \quad \text{is minimum for } \mathbf{c} = -(LV_2)^\dagger L(\mathbf{y}_k-\mathbf{y}_0).$$

Thus the truncated solution is

$$\mathbf{y} = \mathbf{y}_k - V_2(LV_2)^\dagger L(\mathbf{y}_k-\mathbf{y}_0) = (I - V_2(LV_2)^\dagger L)\mathbf{y}_k + V_2(LV_2)^\dagger L\mathbf{y}_0$$

Solution of the shifted system is $\mathbf{x} = (I - V_2(LV_2)^\dagger L)\mathbf{x}_k$

We can do some algebra to show

$$\mathbf{y} - \mathbf{x} = \mathbf{y}_0 + (V_2(LV_2)^\dagger L - I_n)\sum_{i=k+1}^{n}(\mathbf{v}_i^T\mathbf{y}_0)\mathbf{v}_i \neq \mathbf{y}_0, \quad L \neq I$$

Note that $L = I$ has $\mathbf{y} = \mathbf{x}$.

It is important to be careful when shifting in the regularization.

Introduce generalization of the SVD to obtain a expansion for

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}}\{\|\tilde{A}\mathbf{x} - \tilde{\mathbf{b}}\|^2 + \lambda^2\|L(\mathbf{x} - \mathbf{x}_0)\|^2\}$$

Lemma (GSVD)

Assume invertibility and $m \geq n \geq p$. There exist unitary matrices $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{p\times p}$, and a nonsingular matrix $X \in \mathbb{R}^{n\times n}$ such that

$$A = U \begin{bmatrix} \Upsilon \\ 0_{(m-n)\times n} \end{bmatrix} X^T, \quad L = V[M, 0_{p\times(n-p)}]X^T,$$
$$\Upsilon = \mathrm{diag}(\upsilon_1, \dots, \upsilon_p, 1, \dots, 1) \in \mathbb{R}^{n\times n}, \; M = \mathrm{diag}(\mu_1, \dots, \mu_p) \in \mathbb{R}^{p\times p},$$

with

$$0 \leq \upsilon_1 \leq \cdots \leq \upsilon_p \leq 1, \; 1 \geq \mu_1 \geq \cdots \geq \mu_p > 0, \; \upsilon_i^2 + \mu_i^2 = 1, \; i = 1, \dots p.$$

Use $\tilde{\Upsilon}$ and $\tilde{M}$ to denote the rectangular matrices containing Υ and M.

Solution of the Generalized Problem using the GSVD

As for the SVD we need the expression for the regularization matrix

$$R(\lambda) = (A^TA + \lambda^2 L^TL)^{-1}A^T = (X^T)^{-1}(\tilde{\Upsilon}^T\tilde{\Upsilon} + \lambda^2\tilde{M}^T\tilde{M})^{-1}\tilde{\Upsilon}^TU^T$$

Notice

$$(\tilde{\Upsilon}^T\tilde{\Upsilon} + \lambda^2\tilde{M}^T\tilde{M})^{-1}\tilde{\Upsilon}^T = \mathrm{diag}(\mathrm{diag}(\frac{\nu_i}{\nu_i^2 + \lambda^2\mu_i^2}), 1, \dots, 1)$$

Thus

$$\mathbf{x}(\lambda) = \sum_{i=1}^{p} \frac{\nu_i}{\nu_i^2 + \lambda^2\mu_i^2}(\mathbf{u}_i^T\mathbf{b})\tilde{\mathbf{x}}_i + \sum_{i=p+1}^{n}(\mathbf{u}_i^T\mathbf{b})\tilde{\mathbf{x}}_i$$

where $\tilde{\mathbf{x}}_i$ is the i^{th} column of $(X^T)^{-1}$ With $\rho_i = \nu_i/\mu_i$ we have

$$\mathbf{x}(\lambda) = \sum_{i=1}^{p} \frac{\rho_i^2}{\nu_i(\rho_i^2 + \lambda^2)}(\mathbf{u}_i^T\mathbf{b})\tilde{\mathbf{x}}_i + \sum_{i=p+1}^{n}(\mathbf{u}_i^T\mathbf{b})\tilde{\mathbf{x}}_i$$

$$= \sum_{i=1}^{p} \gamma_i \frac{\mathbf{u}_i^T\mathbf{b}}{\nu_i}\tilde{\mathbf{x}}_i + \sum_{i=p+1}^{n}(\mathbf{u}_i^T\mathbf{b})\tilde{\mathbf{x}}_i, \quad \gamma_i = \frac{\rho_i^2}{\rho_i^2 + \lambda^2}, \; i = 1, \dots, p.$$

Notice the similarity with the filtered SVD solution

$$\mathbf{x}(\lambda) = \sum_{i=1}^{r} \gamma_i \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i, \quad \gamma_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2}.$$

For the generalized Tikhonov regularization the degrees of freedom for the cost functional are $m - n + p$, L of size $p \times n$.

Basic Newton iteration to solve $F(\sigma) = 0$,

$\mathbf{y}(\sigma^{(k)})$ is the current solution for which

$$\mathbf{x}(\sigma^{(k)}) = \mathbf{y}(\sigma^{(k)}) + \mathbf{x}_0$$

Use the derivative

$$J'(\sigma) = -\frac{2}{\sigma^3}\|L\mathbf{x}(\sigma)\|^2 < 0. \tag{2}$$

and line search parameter $\alpha^{(k)}$ to give

$$\sigma^{(k+1)} = \sigma^{(k)}(1+\alpha^{(k)}\frac{1}{2}(\frac{\sigma^{(k)}}{\|L\mathbf{x}(\sigma^{(k)})\|})^2(J(\sigma^{(k)})-(m-n+p))). \tag{3}$$

More on the Generalized Tikhonov Solutions

- The parameter estimation techniques extend for the solutions using the GSVD.
- As with the χ^2 method, other regularization techniques can be extended without the GSVD
- Observation on the shifted solution: for the result with the χ^2 we require that $E(\mathbf{x}) = \mathbf{x}_0$. i.e we know the expected average solution. When this is not available we can modify the approach for a non central χ^2 distribution.
- For missing data Hansen notes that solution in the $2-$ norm leads to solutions which have $\mathbf{x} = 0$ if there are missing data but replacing operator by L the first or second derivative operator fills in a solution which more appropriately fills in a realistic solution.

- For practical problems, i.e. of reasonable size, we do not use SVD or GSVD. Rather we use iterative methods - LSQR
- Typically use iterative Krylov method with preconditioning
- Goal to utilize forward operations for A and A^T
- We will not discuss regularizing properties of LSQR but give the algorithm for completeness

LSQR Algorithm (Paige and Saunders - Golub-Kahan bidiagonalization)

- Initial vectors $\mathbf{g}_0 \equiv 0$, $\mathbf{h}_1 \equiv \tilde{\mathbf{b}}/\beta_1$. where $\beta_1 \equiv \|\mathbf{b}\| \neq 0$.
- Orthonormal vectors $\mathbf{g}_i, \mathbf{h}_i$, $i = 1, 2, \ldots$ are columns of matrices $H_j \equiv [\,\mathbf{h}_1, \ldots, \mathbf{h}_j\,] \in \mathbb{R}^{m \times j}$, $G_j \equiv [\,\mathbf{g}_1, \ldots, \mathbf{g}_j\,] \in \mathbb{R}^{n \times j}$.
- Recurrences can be written in the matrix notation

$$A^T H_j = G_j B_j^T, \qquad A G_j = [\,H_j, \mathbf{h}_{j+1}\,] B_{j+}. \tag{4}$$

$$B_j \equiv \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_j & \alpha_j \end{bmatrix}, \quad B_{j+} \equiv \begin{bmatrix} B_j \\ \beta_{j+1}\mathbf{e}_j^T \end{bmatrix}, \qquad B_{j+} \in \mathbb{R}^{(j+1) \times j}.$$

- The j steps of the bidiagonalization yield a subproblem

$$B_{j+}\,\mathbf{y}_j \approx \mathbf{e}_1\,\beta_1 \text{ and } \mathbf{x}_j(\sigma) \equiv G_j \mathbf{y}_j(\sigma).$$

Regularizing the subproblem

- With regularization when $D = I$, $\mathbf{y}_j(\sigma)$ solves regularized problem
$$\|B_{j+}\mathbf{y}_j - \mathbf{e}_1\,\beta_1\|^2 + 1/\sigma^2\|\mathbf{y}_j\|^2$$
- Note that $\|\mathbf{x}_j(\sigma)\|_2^2 = \|G_j\mathbf{y}_j(\sigma)\|^2 = \|\mathbf{y}_j(\sigma)\|_2^2$.
- The regularized problem is of size $j+1 \times j$. $j \ll m$
- Standard Methods can be applied to find σ efficiently.
- Nagy uses a weighted GCV - with a *fudge* factor
- χ^2 principle can be extended and applied to the subproblem.
- Noise revealing properties can be utilized.

Summary: Nov 30

1. Weighting by the noise to apply techniques for non white data
2. χ^2 property of the global residual - discrepancy and NCP
3. Shifted solutions with prior information
4. Alternative Operators for Regularization: Boundary Conditions
5. Generalized Singular Value Decomposition
6. Basic Iterative Techniques: LSQR

MAP and the Tikhonov Regularization
Total Variation Regularization
Newton's Method
Iteratively reweighted Norms
Split Bregman Iteration
Rosemary Renaut
December 14, 2011

The relation of the Tikhonov and the MAP estimator

Statistical interpretation of $\mathbf{y}_0$: Let $p(\mathbf{x})$ be the probability for $\mathbf{x}$ and $p(\mathbf{y}|\mathbf{x})$ be the conditional probability of $\mathbf{y}$ given $\mathbf{x}$

Maximum Likelihood Estimator (MLE) for obtaining parameter $\mathbf{y}$ given $\mathbf{d}$ is the parameter $\hat{\mathbf{y}}$ which maximizes the likelihood function $L(\mathbf{y}) = p(\mathbf{d}; \mathbf{y})$.

MLE maximizes the log likelihood $l(\mathbf{y}) = \log p(\mathbf{d}; \mathbf{y})$.

Example Suppose $\mathbf{d}$ is a realization of vector $\mathbf{y} \sim \mathcal{N}(\mathbf{y}_0, C)$.

Assume $\mathbf{y}_0$ is unknown. The probability distribution function is

$$p(\mathbf{y}; \mathbf{y}_0, C) \propto \exp(-\frac{1}{2}(\mathbf{y}-\mathbf{y}_0)^T C^{-1}(\mathbf{y}-\mathbf{y}_0)) \quad \text{and}$$

$$l(\mathbf{y}) = -\frac{1}{2}(\mathbf{y}-\mathbf{y}_0)^T C^{-1}(\mathbf{y}-\mathbf{y}_0) + c,$$

where c is independent of $\mathbf{y}_0$, has maximizer $\mathbf{y} = \mathbf{y}_0$.

Bayes Law If $\mathbf{y}$ and $\mathbf{d}$ are jointly distributed random vectors.

$$p(\mathbf{y}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{d})}$$

MAP Maximum A Posteriori Estimator is the maximizer of $p(\mathbf{y}|\mathbf{d})$ with respect to $\mathbf{y}$.

Linear Model

Suppose $\mathbf{y} \sim \mathcal{N}(\mathbf{y}_0, C_\mathrm{y})$, $\boldsymbol{\eta} \sim \mathcal{N}(0, C)$ and $\mathbf{d}$ is a random vector defined by the linear model $A\mathbf{y} + \boldsymbol{\eta} = \mathbf{d}$

$\mathbf{d} \sim \mathcal{N}(A\mathbf{y}, C)$ implies

$$p(\mathbf{d}|\mathbf{y}) \propto \exp(-\frac{1}{2}(\mathbf{d} - A\mathbf{y})^T C^{-1}(\mathbf{d} - A\mathbf{y})) \quad \text{and the prior is}$$

$$p(\mathbf{y}) \propto \exp(-\frac{1}{2}(\mathbf{y} - \mathbf{y}_0)^T C_\mathrm{y}^{-1}(\mathbf{y} - \mathbf{y}_0))$$

Combining terms and using Bayes Law the a posteriori log likelihood function is

$$l(\mathbf{y}|\mathbf{d}) = -\frac{1}{2}(\mathbf{d} - A\mathbf{y})^T C^{-1}(\mathbf{d} - A\mathbf{y}) - \frac{1}{2}(\mathbf{y} - \mathbf{y}_0)^T C_\mathrm{y}^{-1}(\mathbf{y} - \mathbf{y}_0) + c$$

where c is independent of $\mathbf{y}$.

The MAP estimator which maximizes $l(\mathbf{y}|\mathbf{d})$ is equivalent to the minimizer of the Tikhonov regularization

$$\|\mathbf{d} - A\mathbf{y}\|^2_{C^{-1}} + \|\mathbf{y} - \mathbf{y}_0\|^2_{C_\mathrm{y}^{-1}}$$

In this case $\mathbf{y}_0$ is the expected value of $\mathbf{y}$.

A more general regularization term $R(\mathbf{x})$ may be considered to better preserve properties of the solution $\mathbf{x}$:

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}}\{\|A\mathbf{x} - \mathbf{b}\|_W^2 + \lambda^2 R(\mathbf{x})\}$$

Suppose R is total variation of $\mathbf{x}$ (general options are possible)
The total variation for a function f defined on a discrete grid is

$$\mathrm{TV}(\mathbf{x}(a)) = \sum_i |\mathbf{x}(a_i) - \mathbf{x}(a_{i-1})| \approx \Delta \sum_i |d\mathbf{x}(a_i)/da|$$

TV approximates a scaled sum of the magnitude of jumps in $\mathbf{x}$.
Δ is a scale factor dependent on the grid size.
Two dimensional version

$$\begin{aligned}\mathrm{TV}(\mathbf{x}(a,b)) &= \sum_{i,j}\sqrt{|\mathbf{x}(a_i,b_j) - \mathbf{x}(a_{i-1},b_j)|^2 + |\mathbf{x}(a_i,b_j) - \mathbf{x}(a_i,b_{j-1})|^2} \\ &\approx \Delta\sum_{i,j}\sqrt{|\partial\mathbf{x}(a_i,b_j)/\partial a|^2 + |\partial\mathbf{x}(a_i,b_j)/\partial b|^2} \\ &\approx \Delta\sum_{i,j}\sqrt{|\nabla\mathbf{x}(a_i,b_j)|^2}, \quad \nabla\mathbf{x} = [(\partial\mathbf{x}(a,b)/\partial a)^T, (\partial\mathbf{x}(a,b)/\partial b)^T]^T\end{aligned}$$

Notice $\mathrm{TV}(\mathbf{x}(a)) = \sum_i |\mathbf{x}(a_i) - \mathbf{x}(a_{i-1})| = \|L_{a,1}\mathbf{x}\|_1$ where $L_{a,1}$ is the first order derivative matrix operator.

For regularized problem we may want to solve

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}}\{\|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda^2\|L_{a,1}\mathbf{x}\|_1\}$$

Software: **Disciplined Convex Programming: an introduction** `http://cvxr.com/dcp/`

`http://cvxr.com/cvx/` describes the general formulation

Simple Matlab

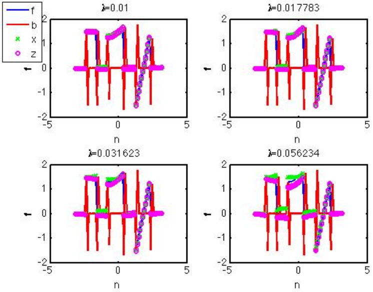
```
m = 20; n = 10; p = 4;
A = randn(m,n); b = randn(m,1);
C = randn(p,n); d = randn(p,1); e = rand;
cvx_begin
    variable x(n)
    minimize( norm( A * x - b, 2 ) )
    subject to
        C * x == d
        norm( x, Inf ) <= e
cvx_end
```

Using CVX for 1d Problem

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}} \{ \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|L_{a,1}\mathbf{x}\|_1 \}$$

In this case use A as the second order derivative smoothing operator. Noise is added to $A\mathbf{x}$

```
cvx_precision('medium');
cvx_quiet(false);
cvx_begin
    variable x(n);
    r = A*x-b;
    minimize(lambda*
     norm(L*x,1) +r'*r);
cvx_end
```



Practicalities

Viable implementation using nonlinear minimization requires

$$|\mathbf{x}| \approx \sqrt{|\mathbf{x}|^2 + \beta^2}, \quad 1 \gg \beta > 0.$$

to avoid difficulties with singularity for $|\mathbf{x}| \approx 0$. Then

$$\mathrm{TV}(\mathbf{x}(a,b)) \approx \Delta \sum_{i,j} \sqrt{|\nabla \mathbf{x}(a_i, b_j)|^2 + \beta^2} = \sqrt{\|\nabla \mathbf{x}\|^2 + \beta^2}$$

i.e. $\nabla \mathbf{x} \approx [(L_{a,1}\mathbf{x})^T, (L_{b,1}\mathbf{x})^T]^T$ where $L_{a,1}$, $L_{b,1}$ denote first order derivatives in a and b.

Define $\mathbf{l}_i^T$ to be i^{th} row of L_1

$$\mathbf{l}_i = [0, \dots, 0, -1, 1, 0, \dots, 0], \; \mathbf{x}(a_i) - \mathbf{x}(a_{i-1}) = \mathbf{l}_i^T \mathbf{x}$$

where the pair $[-1, 1]$ is in the $(i-1, i)^{th}$ position. Then

$$TV(\mathbf{x}) = \frac{1}{2} \sum_{i=2}^{n} \psi(|\mathbf{l}_i^T \mathbf{x}|^2), \; \psi(t) = \sqrt{t + \beta^2}, \psi'(t) = \frac{1}{2} \frac{1}{\sqrt{t + \beta^2}} > 0, \; t > 0$$

Another choice is the Huber function given by

$$\psi(t) = \begin{cases} t/\epsilon & t \leq \epsilon^2 \\ 2\sqrt{t} - \epsilon & t > \epsilon^2 \end{cases}$$

These functions are quite close. Use $\psi(t) = \sqrt{t + \beta^2}$

Nonlinear Minimization by Newton's method (see Vogel Chapter 8)

To use $R(\mathbf{x}) = TV(\mathbf{x})$ in a nonlinear minimization we need the gradient (and Hessian) of $R(\mathbf{x})$.

Differentiating

$$\nabla R(\mathbf{x}) = L^T \mathrm{diag}(\psi'(\mathbf{x}))L\mathbf{x} = \Psi(\mathbf{x})\mathbf{x}, \quad \Psi(\mathbf{x}) = L^T \mathrm{diag}(\psi'(\mathbf{x}))L$$

$\mathrm{diag}(\psi'(\mathbf{x}))$ denotes diagonal matrix with i^{th} entry $\psi'((\mathbf{l}_i^T\mathbf{x})^2)$

L is SPD provided $\psi'(t) > 0$ whenever $t > 0$.

Differentiating again

$$\nabla^2 R(\mathbf{x}) = \Psi(\mathbf{x}) + \Psi'(\mathbf{x})\mathbf{x}, \quad \Psi'(\mathbf{x})\mathbf{x} = L^T \mathrm{diag}(2(L\mathbf{x})^2\psi''(\mathbf{x}))L.$$

$\mathrm{diag}(2(L\mathbf{x})^2\psi''(\mathbf{x}))$ denotes diagonal matrix with i^{th} entry $2(\mathbf{l}_i^T\mathbf{x})^2\psi''((\mathbf{l}_i^T\mathbf{x})^2)$

for the functional $J(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|^2 + \lambda^2 R(\mathbf{x})$ we obtain

$$\nabla J(\mathbf{x}) = A^T(A\mathbf{x} - \mathbf{b}) + \lambda^2\Psi(\mathbf{x})\mathbf{x}$$
$$\nabla^2 J(\mathbf{x}) = A^T A + \lambda^2\Psi(\mathbf{x}) + \lambda^2\Psi'(\mathbf{x})\mathbf{x}$$

For equivalent formulations for $2D$ see the results in Vogel page 152-153.

Iteratively Reweighted Norm (Rodriguez and Wohlberg 2007 and 2009)

Goal: replace l_1 norm by quadratic l_2 and iterate to avoid Newton-type approaches

Generally in 2D consider derivative approximations $L_{1,a}\mathbf{x} = \mathbf{u}$ and $L_{1,b}\mathbf{x} = \mathbf{v}$.

$$R(\mathbf{x}) = || \ \sqrt{\|\nabla \mathbf{x}\|^2} \ ||_q^q = \sum_l (\sqrt{u_l^2 + v_l^2})^q = \sum_l (u_l^2 + v_l^2)^{q/2} \quad 0 < q \leq 2$$

Suppose that W is a diagonal matrix with entries $z_l = (u_l^2 + v_l^2)^{(q-2)/2}$, then

$$\|\mathrm{diag}(W^{1/2}, W^{1/2})[\mathbf{u}^T, \mathbf{v}^T]^T\|_2^2 = \sum_l (z_l(u_l^2 + v_l^2)) = \sum_l (u_l^2 + v_l^2)^{q/2}. \quad \text{Hence}$$

$$R(\mathbf{x}) = || \begin{pmatrix} W^{1/2} & 0 \\ 0 & W^{1/2} \end{pmatrix} \begin{pmatrix} L_{1,a} \\ L_{1,b} \end{pmatrix} \mathbf{x}||_2^2 = \|W^{1/2} L\mathbf{x}||_2^2,$$

$$W = \mathrm{diag}(W, W) \quad L = \mathrm{diag}(L_{1,a}, L_{1,b})$$

Replacing W by $W^{(l)}$ for iteration l, indicating that the elements are calculated using $L_{1,a}\mathbf{x}^{(k)} = \mathbf{u}^{(k)}$ and $L_{1,b}\mathbf{x}^{(k)} = \mathbf{v}^{(k)}$, $z_l^{(k)} = ((u_l^{(k)})^2 + (v_l^{(k)})^2)^{(q-2)/2}$ yields

$$R^{(k)}(\mathbf{x}) = ||L\mathbf{x}||_{W^{(k)}}^2$$

To avoid singularity again introduce scaling parameter:

$$W_{ll} = \tau(u_l^2 + v_l^2)(z_l), \quad \tau(z) = 1, \quad z \geq \epsilon \quad \tau(z) = 0, \quad z < \epsilon.$$

IRN Algorithm

Require: λ, ϵ, $\mathbf{b}$, A, L, TOL

Initialize: $k = 0$, $\mathbf{x}^{(k)} = (A^T A + \lambda^2 L^T L)^{-1} A^T \mathbf{b}$

while $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 > \mathrm{TOL}$ **do**

  Calculate $(L\mathbf{x}^{(k)})^T = [\mathbf{u}, \mathbf{v}]$. $z_l = (u_l^2 + v_l^2)^{(q-2)/2}$

  Update the weighting $W_{ll} = \tau(u_l^2 + v_l^2)(z_l)$

  Update solution $\mathbf{x}^{(k+1)} = (A^T A + \lambda^2 L^T W L)^{-1} A^T \mathbf{b}$

  $k = k + 1$,

end while

- Algorithm was introduced and analysed in Rodriguez and Wohlberg, IEEE Trans Signal Proc, 18, 2, 2009, and Signal Processing Letters, 2007.
- Choice of λ using UPRE and trace estimation discussed in Signal Processing, 90, 2010, 2546-2551, Lin, Wohlberg and Guo.
- Higher order approach: Yue and Jacob,Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium, pp.1154-1157, March 30 2011-April 2 2011

The main reference is here with software:
T. GOLDSTEIN, Split Bregman Software Page, `http://www.stanford.edu/~tagoldst/Tom_Goldstein/Split_Bregman.html`.

T. GOLDSTEIN AND S. OSHER, The Split Bregman Method for L1-Regularized Problems. SIAM J. Imaging Sci. **2**, 2, (2009), 323-343.

Many developments have been made since that time:
S. SETZER, Operator Splittings, Bregman Methods and Frame Shrinkage in Image Processing, International Journal of Computer Vision, in press, 2011.

Above reference discusses relationship of SB to Augmented Lagrangian and Peaceman-Rachford alternating direction

For $R(\mathbf{x}) = \Phi\mathbf{x}$ for $\Phi \in \mathcal{R}^{q \times n}$: Introduce $\mathbf{d} = \Phi\mathbf{x}$

Rewrite $R(\mathbf{x}) = \frac{\lambda}{2}\|\mathbf{d} - \Phi\mathbf{x}\|_2^2 + \mu\|\mathbf{d}\|_1$

Anisotropic Formulation:

$$(\mathbf{x}, \mathbf{d}) = \arg\min_{\mathbf{x},\mathbf{d}}\{\frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda}{2}\|\mathbf{d} - \Phi\mathbf{x}\|_2^2 + \mu\|\mathbf{d}\|_1\}$$

Solve using an alternating minimization which separates minimization for $\mathbf{d}$ from $\mathbf{x}$

Derivation uses the Bregman distance and so called Bregman *proximal* point iteration.

Various versions of the iteration can be defined. Focus on basic formulation here. Iterate over

$$S1 : \mathbf{x}^{(k+1,l)} = \arg\min_{\mathbf{x}}\{\frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda}{2}\|\Phi\mathbf{x} - (\mathbf{d}^{(k+1,l-1)} - \mathbf{g}^{(k)})\|_2^2\} \quad (5)$$

$$S2 : \mathbf{d}^{(k+1,l)} = \arg\min_{\mathbf{d}}\{\frac{\lambda}{2}\|\mathbf{d} - (\Phi\mathbf{x}^{(k+1,l)} + \mathbf{g}^{(k)})\|_2^2 + \mu\|\mathbf{d}\|_1\} \quad (6)$$

$$S3 : \mathbf{g}^{(k+1)} = \mathbf{g}^{(k)} + \Phi\mathbf{x}^{(k+1)} - \mathbf{d}^{(k+1)}. \quad (7)$$

Update for $\mathbf{x}$:

$$\mathbf{x} = \arg\min_{\mathbf{x}}\{\frac{1}{2}\|A\mathbf{x}-\mathbf{b}\|_2^2 + \frac{\lambda}{2}\|\Phi\mathbf{x}-\mathbf{h}\|_2^2\}, \quad \mathbf{h} = \mathbf{d}-\mathbf{g}$$
$$= \arg\min_{\mathbf{x}}\{\|\tilde{A}\mathbf{x}-\tilde{\mathbf{b}}\|_2^2\}, \quad \tilde{A} = \left(A^T, \sqrt{\lambda}\Phi^T\right)^T, \quad \tilde{\mathbf{b}} = \left(\mathbf{b}^T, \sqrt{\lambda}\mathbf{h}^T\right)^T.$$

Standard least squares update using a Tikhonov regularizer.

Update for $\mathbf{d}$:

$$\mathbf{d} = \arg\min_{\mathbf{d}}\{\mu\|\mathbf{d}\|_1 + \frac{\lambda}{2}\|\mathbf{d}-\mathbf{c}\|_2^2\}, \quad \mathbf{c} = \Phi\mathbf{x}+\mathbf{g}$$
$$= \arg\min_{\mathbf{d}}\{\|\mathbf{d}\|_1 + \frac{\gamma}{2}\|\mathbf{d}-\mathbf{c}\|_2^2\}, \quad \gamma = \frac{\lambda}{\mu}.$$

This is achieved using *soft* thresholding.

Thresholding for $\mathbf{d}$

If $\mathbf{d}$ has q components $(\mathbf{d})_i$ componentwise solution:

$$(\mathbf{d})_i = \frac{c_i}{|c_i|}\max(|c_i| - \frac{1}{\gamma}, 0) \quad i = 1 : q$$

If $\mathbf{d}$ is two dimensional it contains components $\mathbf{d}_a$ and $\mathbf{d}_b$.
Φ is defined for two dimensions. $\mathbf{d}_a = \Phi_a \mathbf{x}$, $\mathbf{d}_b = \Phi_b \mathbf{x}$
Threshold is applied for each component of $(\mathbf{d}_a^T, \mathbf{d}_b^T)^T$: we use

$$\|\mathbf{d}\|_1 = \|\mathbf{d}_a\|_1 + \|\mathbf{d}_b\|_1$$

The TV norm for the two dimensional case can be written

$$\|\mathbf{d}\|_{\mathrm{TV}} = (\sum_{i=1}^{n} \|\mathbf{d}_i\|_2) \quad q = 2n$$

We evaluate for example $\nabla = [\nabla_a^T, \nabla_b^T]^T$ at each point i in space, there are two components at each point.
Intrinsically TV is still local.

$$(\mathbf{d}_{\mathrm{TV}})_i = \frac{\mathbf{c}_i}{\|\mathbf{c}_i\|_2}\max(\|\mathbf{c}_i\|_2 - \frac{1}{\gamma}, 0) \quad i = 1 : n$$

The algorithm

Require: λ, τ, , A, Φ, σ^2
Ensure: $\mathbf{x}$, $\mathbf{d}$.

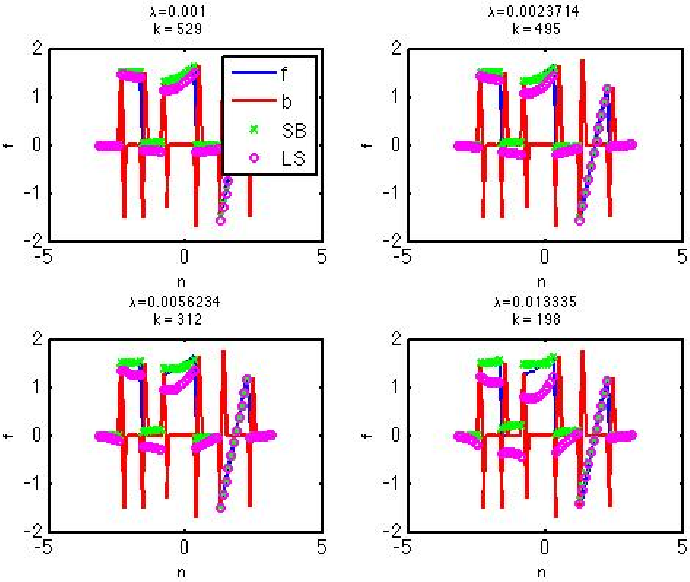
```
Initialize: k = 0, d^(k) = g^(k) = x^(k) = 0, b^(0) = b.
while ||Ax − b||_2 > σ^2 (DISCREPANCY) do
    while ||x^(k) − x^(k−1)||_2 > τ  do
        Update x : solve equation (??) where b = b^(k)
        Update d : solve equation (??)
        Update g: equation (??)
    end while
    Update b: equation b^(k+1) = b + b^(k) − Ad^(k+1)
end while
```

In the unconstrained case the outer loop is removed. No update for $\mathbf{b}$ are used.
The costly step is that for $\mathbf{x}$. Overall cost is basically repeats of quadratic Tikhonov

Matlab for SB in One D

```
function [xk,k]=SB(A, L, b,lambda,mu,tol,itk)
update_x=inline('(At*A)\At*b','A','At','b');
update_d=inline('c./abs(c).*max(abs(c)-1/gamma,0)','gamma','c');
gamma=lambda/mu;
[n,p]=size(L);
gk=zeros(n,1); dk=gk; bk=b;hk=gk;xold=bk;testx=xold;
Atilde=[A; sqrt(lambda)*L];k=0;
while ((norm(testx,2)>tol) & (k<itk)),
    btilde=[b;sqrt(lambda)*hk];
    xk=update_x(Atilde,Atilde',btilde);
    c=L*xk+gk;
    dk=update_d(gamma, c);
    gk=c-dk;hk=dk-gk;testx=xk-xold;xold=xk;
    k=k+1;
end
```

Solutions with SB



Summary: Dec 14

1. Total Variation Regularization
2. Newton's Method - see book of Vogel
3. Iteratively reweighted Norms
4. Split Bregman Iteration

Handling Spectral Data
Detecting edges in data
With Blur
With Noise
With Undersampling
Rosemary Renaut
Jan 4, 2012

So far considered the regularization problem in spatial domain

$$\|A\mathbf{x} - \mathbf{b}\|_2^2 + R(\mathbf{x})$$

for some regularizer $R(\mathbf{x})$

What if we have information on $\mathbf{x}$ the image in some other domain?

For example we may want to use $\mathbf{x}$ as an expansion in wavelet domain where we can resolve features of $\mathbf{x}$

The regularizer should be posed appropriately

Perhaps the data fidelity term can be posed in an alternative setting

Main idea : one should step back and consider the origin of the data

Magnetic Resonance Imaging

- Data are collected in Fourier space i.e. we do not obtain the data for the image but the Spectral representation of the image
- Reconstructing Fourier data comes with a number of issues - not least that the images are smooth
- In general images are actually pixel values so an image is a set of piecewise constant values
- But, around edges (discontinuities) we obtain characteristic *ringing* effect - oscillations
- Oscillations around the edges are intrinsic, taking more points will not remove the ringing
- One also needs to take care of aliasing effects

Engineers have a good *bag of tricks* for handling difficulties with Fourier data.
What can mathematicians contribute?

Spectral Data for function f

Assume:

Function f is s 2π-periodic and piecewise-smooth function in $[-\pi, \pi)$.

It has Fourier series coefficients

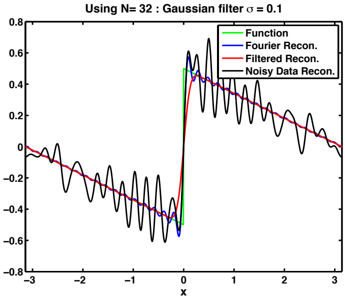
$$\hat{f}_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-ikx} dx \,, \quad k \in [-N, N]$$

$\{\hat{f}_k\}$ is a global representation; i.e., $\hat{f}_k$ are obtained using values of f over the entire domain $[-\pi, \pi)$.

Goal: Given $\hat{f}_k$ only can we construct f reliably

1. With blur
2. With noise
3. With under sampling or irregular sampling.

Figure: Reconstructed from 32 samples in Fourier Space to domain length 1024, with noise $.1 * randn$ in Fourier coefficients, or blur with Gaussian



Given f can we accurately locate the position of the discontinuity in f?

Definition: The Jump Function - understanding the data

Assume: f is piecewise smooth then

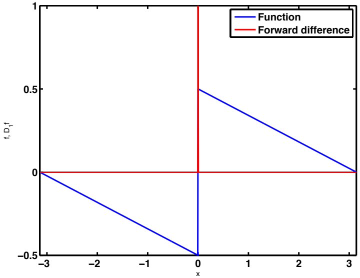
- Its *jump function* is defined by

$$[f](x) := f(x^+) - f(x^-)$$

- i.e. this is the jump in approaching from either side of a discontinuity at x
- A jump discontinuity is a local feature; i.e., the jump function at any point x only depends on the values of f at x^+ and x^-.
- Jump function is nonzero only at a jump discontinuity. Moreover, it takes the value of the jump at the discontinuity.
- We can *estimate* the jump function using a forward difference. $f(x_{j+1}) - f(x_j)$, if difference $x_{j+1} - x_j$ is small enough.

Illustration Jump Detection by First Order Forward Differences

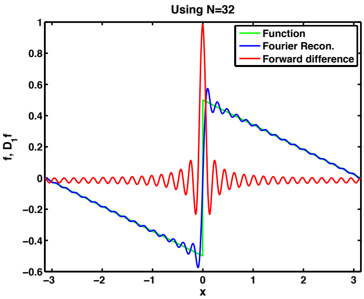
Figure: Physical space jump detection



Jump Detection of the unit ramp function with original data on 1025 grid and using first order forward difference
For this data $f_{j+1} - f_j = .001$ away from the jump on grid with grid size $2\pi/1024$.

Illustration Jump Detection by First Order Forward Differences

Figure: Fourier space using $N = 32$ modes

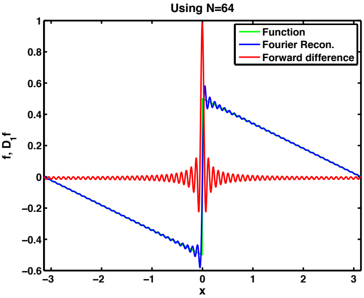


.

Suppose f is given by its exact Fourier coefficients: and found by inversion. Detection has to be applied to the reconstructed data

Illustration Jump Detection by First Order Forward Differences

Figure: Fourier space using $N = 64$ modes



.

Increasing the sampling does not improve the edge detection because of the intrinsic ringing in the reconstruction

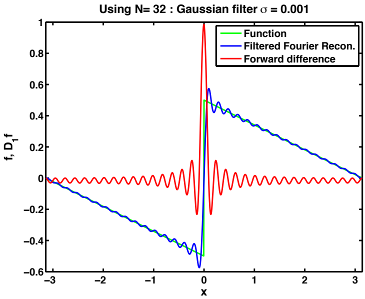
Observations

- Fourier reconstruction of data with edges exhibits Gibbs ringing
- Actually estimation of edges from data that is reconstructed from Fourier samples is challenging.
- Can we filter to remove the Gibbs oscillations - ringing around the edges? This is the standard approach
- But filtering necessarily removes some of the intrinsic features of the data.
- Edges that are close may be totally lost.

Jump Detection for Filtered Reconstruction

Gaussian filter $h(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2})$

Figure: Fourier space using $N = 32$ modes, $\sigma = .001$

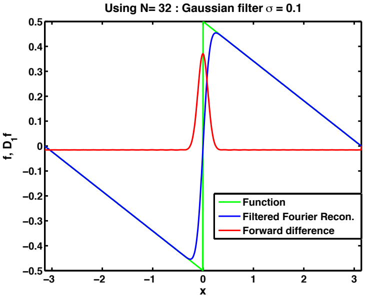


$N = 32$ nodes and narrow filter- too narrow insufficient filtering

Jump Detection for Filtered Reconstruction

Gaussian filter $h(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2})$

Figure: Fourier space using $N = 32$ modes, $\sigma = .1$

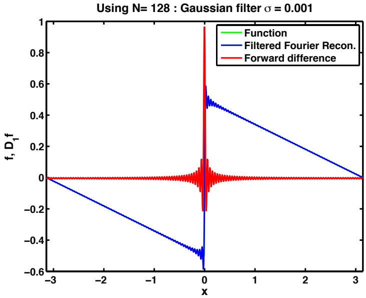


$N = 32$ nodes and wide filter - too smooth

Jump Detection for Filtered Reconstruction

Gaussian filter $h(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2})$

Figure: Fourier space using $N = 128$ modes, $\sigma = .001$

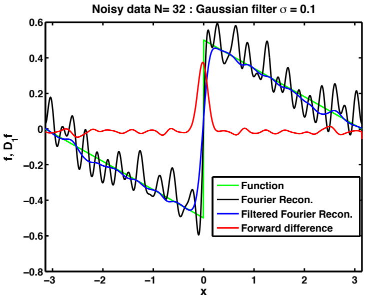


$N = 128$ nodes and narrow filter - too narrow also for more points

Jump Detection for Filtered Reconstruction

Gaussian filter $h(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2})$

Figure: Fourier Space using $N = 32$ modes, $\sigma = .1$, noise .05



Jump Detection Filtered Noisy Signal

Concentration Factor Edge Detection Method (Gelb, Tadmor)

Exploit Fourier domain information to approximate $[f](x)$

Consider the generalized conjugate partial Fourier sum depending on a *concentration* factor σ

$$S_N^{\sigma}[f](x) = i \sum_{k=-N}^{N} \hat{f}_k \,\mathrm{sgn}(k)\, \sigma\left(\frac{|k|}{N}\right) e^{ikx} = \sum_{k=-N}^{N} (\widehat{S_N^{\sigma}[f]})_k e^{ikx} \tag{8}$$

Convergence

$$S_N^{\sigma}[f](x) = [f](x) + \mathcal{O}(\epsilon), \epsilon = \epsilon(N) > 0 \text{ small}$$

depends on σ and distance between x and a discontinuity of f.

In fact we can see that this is a convolution with the kernel C_N^{σ}

$$(\widehat{S_N^{\sigma}[f]})_k = (\widehat{f * C_N^{\sigma}})_k, \quad (\widehat{C_N^{\sigma}})_k = i\mathrm{sgn}(k)\, \sigma\left(\frac{|k|}{N}\right) \tag{9}$$

Concentration Factors are filters

Factor	Expression : $\eta = \lvert k \rvert / N$
Trigonometric	$\sigma_T(\eta) = \dfrac{\pi \sin(\alpha\,\eta)}{Si(\alpha)}$ $Si(\alpha) = \displaystyle\int_0^{\alpha} \frac{\sin(x)}{x}\,dx$
Polynomial	$\sigma_P(\eta) = p\,\pi\,\eta^p$ p is the order of the factor
Exponential	$\sigma_{\exp}(\eta) = C\,\eta\,\exp\left(\dfrac{1}{\alpha\,\eta\,(\eta-1)}\right)$ C - normalizing constant; $\alpha > 0$ - $C = \dfrac{\pi}{\int_{\frac{1}{N}}^{1-\frac{1}{N}} \exp\left(\frac{1}{\alpha\,\tau\,(\tau-1)}\right) d\tau}$

Table: Examples of concentration factors

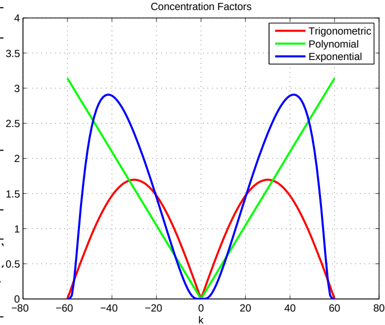


Figure: Envelopes of the Concentration Factors in Fourier Space

Approximate $[f](x)$ using convolution with

$$C_N^{\sigma}(x) = \sum_{k=-N}^{N} \operatorname{sgn}(k)\sigma(\frac{k}{N})\exp(ikx)$$

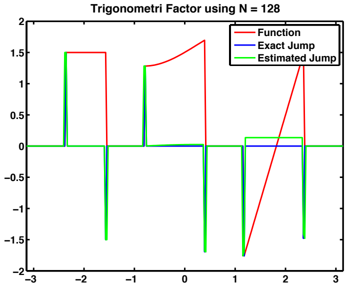
$$S_N^{\sigma}[f](x) = i \sum_{k=-N}^{N} \hat{f}_k \operatorname{sgn}(k)\, \sigma\left(\frac{|k|}{N}\right) e^{ikx} = (f * C_N^{\sigma})(x) \qquad (10)$$

For convergence of (**??**) concentration factors have to satisfy *admissibility properties*:

1. $\sum_{k=1}^{N} \sigma\left(\frac{k}{N}\right) \sin(kx)$ is odd
2. $\frac{\sigma(\eta)}{\eta} \in C^2(0,1)$
3. $\int_{\epsilon}^{1} \frac{\sigma(\eta)}{\eta} \to \pi$, $\epsilon = \epsilon(N) > 0$ is small

Estimation for signal with No Noise, No Blur, Cartesian Grid

Figure: Trigonometric Factor: Jump Response. Notice that the heights and locations are quite well-estimated

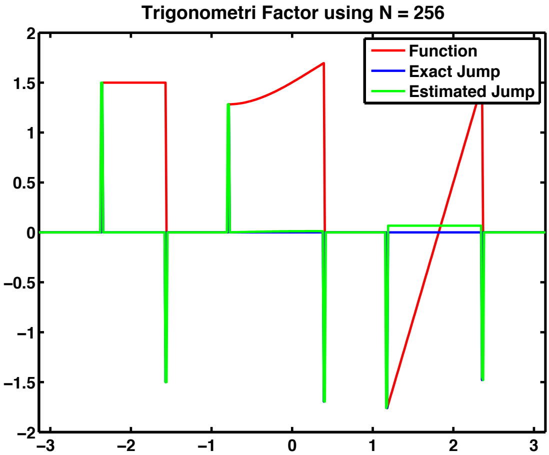


Here f is given by

$$f = \begin{cases} 1.5 & -3\pi/4 \leq x < -\pi/2 \\ 1.75 - x/2.. & \\ + \sin(x - .25) & \pi/4 \leq x < \pi/8 \\ 2.75x - 5 & 3\pi/8 \leq x < -3\pi/4 \\ 0 & \text{otherwise} \end{cases}$$

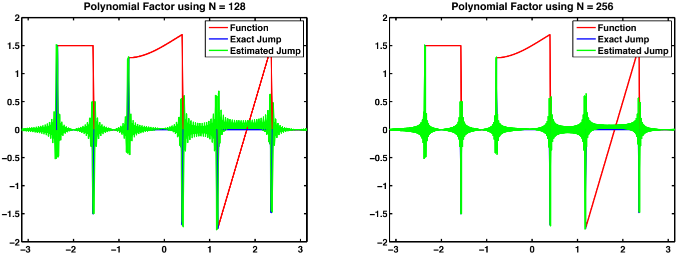
Estimation for signal with No Noise, No Blur, Cartesian Grid

Figure: Trigonometric Factor: Notice that the heights and locations are quite well-estimated:Increasing the sampling improves the detection



Estimation for signal with No Noise, No Blur, Cartesian Grid

Figure: Polynomial Factor: Results are less satisfying



Results depend on the concentration factor

Observations

This is not perfect:

- Dependent on the CF the edge function changes
- Uncertainty in identifying a jump depends on a threshold and the CF
- The height of the jump is correctly identified if we have normalized σ correctly.

Solution

- Use multiple CFs and only accept jumps identified by all CFs

Reduce uncertainty by only accepting matching results: MINMOD (Gelb and Tadmor (2006))

Use the **minmod** function over different concentration functions

$$\textbf{minmod}\{a_1, \dots, a_n\} := \begin{cases} s \min(|a_1|, |a_2|, \dots, |a_n|) & s := \mathrm{sgn}(a_i), \forall i \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

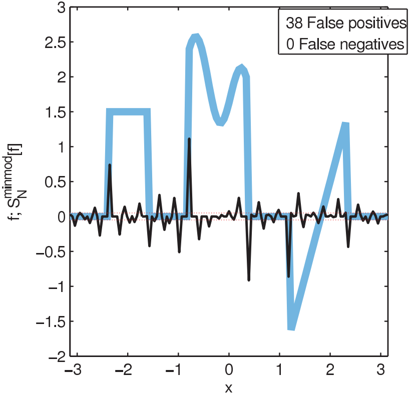
yielding the approximation obtained by finding the jump approximation with multiple σ

$$S_N^{MM}[f](x) = \textbf{minmod}\{S_N^{\sigma_1}[f](x), S_N^{\sigma_2}[f](x), \dots, S_N^{\sigma_n}[f](x)\}. \tag{12}$$

But this does not work unless the data are exact

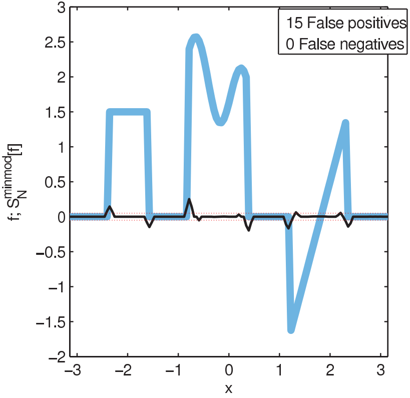
Minmod CF edge detection for noisy and blurred functions: 2% threshold

Under sampling: 10% missing Fourier Coefficients.



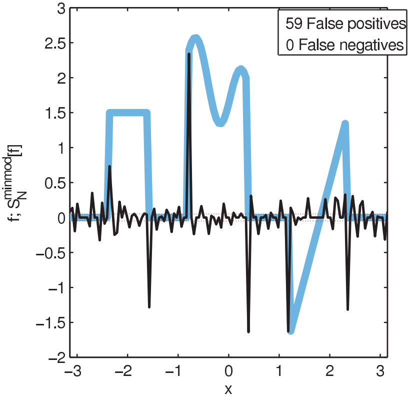
Minmod CF edge detection for noisy and blurred functions: 2% threshold

Blurring by a Gaussian blur of variance $\tau = 0.05$ for point spread function coefficients $\hat{h}_k = e^{-\frac{k^2\tau^2}{2}}$



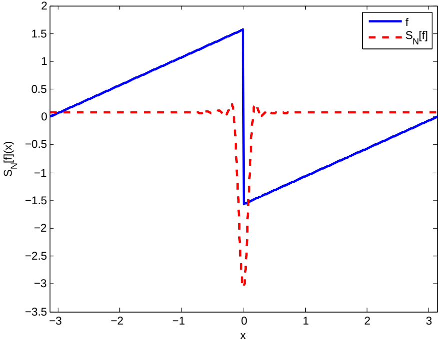
Minmod CF edge detection for noisy and blurred functions: 2% threshold

Noise variance .015 applied to Fourier Coefficients.



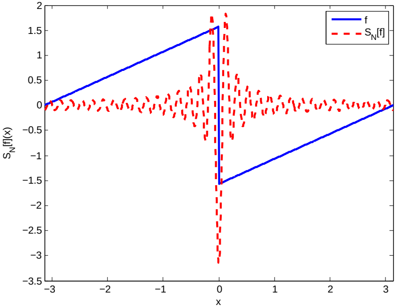
Let $r(x)$ denote the unit ramp function. Notice that this defines a periodic function with one jump

$$r(x) = \begin{cases} \frac{\pi+x}{2} & x < 0 \\ \frac{x-\pi}{2} & x > 0 \end{cases}, \qquad [r](x) = \begin{cases} -\pi & x = 0 \\ 0 & \text{else} \end{cases}$$

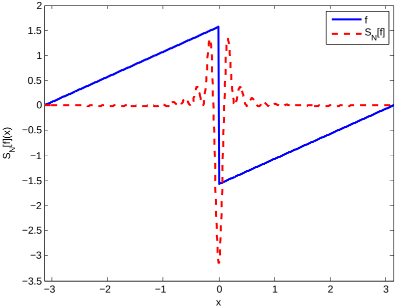


Trigonometric Factor

Sample Jump Responses to a Ramp Function



Polynomial Factor



Exponential Factor

Definition
The jump response, denoted by $W_N^\sigma(x)$, is defined as the jump function approximation of the unit ramp with positive jump 1 as generated by the concentration sum, i.e.,

$$W_N^\sigma(x) := S_N^\sigma[r](x) = i \sum_{|k|\leq N} \hat{r}(k)\operatorname{sgn}(k)\sigma\left(\frac{|k|}{N}\right) e^{ikx}$$
$$= \frac{1}{2\pi} \sum_{0<|k|\leq N} \frac{\sigma\left(\frac{|k|}{N}\right)}{|k|} e^{ikx} \qquad (13)$$

The jump response describes the unique oscillatory pattern of the jump function approximation in the immediate vicinity of jumps and away from jumps.

If jumps are sufficiently far apart one should see this pattern at each jump.

Independence of jump approximation: height and location but not f

In particular we can show that

- jump approximation at $x = \xi$ depends on size $[f]$, location ξ, and choice σ, but not f:

$$\begin{aligned} S_N^{\sigma}[f](x) &= \frac{[f](\xi)}{\pi} \sum_{k=1}^{N} \sigma\left(\frac{|k|}{N}\right) \frac{\cos k(x-\xi)}{k} + \mathcal{O}\left(\frac{\log N}{N}\right) \\ &= [f](\xi) W_N^{\sigma}(x-\xi) + \mathcal{O}\left(\frac{\log N}{N}\right) \end{aligned}$$

- This result characterizes the shape that the concentration method should produce for a single jump discontinuity at $x = \xi$.
- Convolving $S_N^{\sigma}[f](x)$ with the signature W_N^{σ} generates the locations of best match for edges in the signal.

Matching Waveform Concentration Factor (A. Gelb and D. Cates, 2008)

- Correlation gives the matching waveform

$$S_N^{\sigma_{mw}}[f](x) = \frac{1}{\gamma_{mw}}(S_N^{\sigma}[f] * W_N^{\sigma})(x), \qquad \gamma_{mw} = \frac{1}{\pi}\sum_{k=1}^{N}\left(\frac{\sigma(\frac{|k|}{N})}{k}\right)^2 \tag{14}$$

- Gives admissible *matching waveform concentration factor* (MWCF)

$$\sigma_{mw}\left(\frac{|k|}{N}\right) := \frac{1}{\gamma_{mw}}\sigma\left(\frac{|k|}{N}\right)\int_{-\pi}^{\pi} W_N^{\sigma}(\rho)\exp(-ik\rho)\, d\rho. \tag{15}$$

- MWCF performs better in the presence of noise, does not remove oscillations.
- Performance deteriorates for nearby jumps.

Combining the knowledge for segmenting images from Fourier Data

Summary of The Approach

1. The concentration function applied in Fourier domain enhances scales at edges
2. We can find the Fourier expansion approximating jumps using concentration at the edges
3. Correlate the obtained jump approximation to a matching waveform. (use the MWCF) to improve jump identification.
4. Can we use this to improve the segmentation inverse problem?

Estimate Jump Function $[f]$ given $\hat{g}_k$ of Noisy Blurred f

Given $\hat{g}_k$ for blur function h and noise n, $\hat{g}_k = \hat{h}_k \hat{f}_k + \hat{n}_k$

Approximate $[f]$ from $[g]$, given $\hat{g}_k$: $(\widehat{S_N^\sigma[g]})_k = \left(i\sigma(\frac{|k|}{N})\mathrm{sgn}(k)\right)\hat{g}_k$

Observe $\hat{g}_k \approx \hat{h}_k \hat{f}_k$. Thus
$\left(i\sigma(\frac{|k|}{N})\mathrm{sgn}(k)\right)\hat{g}_k \approx \left(i\sigma(\frac{|k|}{N})\mathrm{sgn}(k)\right)\hat{f}_k\hat{h}_k \approx \hat{h}_k(\widehat{S_N^\sigma[f]})_k$

Seek sparse $y(x)$ which approximates the jump function $[f]$

Convolve $y(x)$ with $W_N^\sigma(x)$ to approximate jump $S_N^\sigma[f](x)$

$$(\widehat{S_N^\sigma[f]})_k \approx (\widehat{W_N^\sigma * y})_k = (\widehat{W_N^\sigma})_k \hat{y}_k, \tag{16}$$

Obtain for $(\widehat{W_N^\sigma})_k = \frac{1}{2\pi|k|}\sigma(\frac{|k|}{N})$, $|k| \leq N, k \neq 0$

$$\hat{h}_k(\widehat{W_N^\sigma})_k\hat{y}_k \approx i\sigma(\frac{|k|}{N})\mathrm{sgn}(k)\hat{g}_k$$

Notice we cannot divide out to find $\hat{y}_k$ because deblurring with $1/\hat{h}_k$ is ill-posed.

Regularization is needed

A Discrete Inverse Formulation: for blur, noise and Cartesian

- Introduce matrices describing components of the approximate equation

$$\Sigma = \mathrm{diag}\left(\sigma\left(\frac{|-N|}{N}\right), \cdots, 0, \cdots, \sigma\left(\frac{|N-1|}{N}\right)\right)$$

$$H = \frac{1}{2\pi}\mathrm{diag}\left(\frac{1}{|-N|}\hat{h}_{-N}, \cdots, 0, \cdots, \frac{1}{|N-1|}\hat{h}_{N-1}\right) \quad \text{and}$$

$$F_{kj} = \frac{1}{2N}\exp\left(\frac{-i\pi jk}{N}\right) \quad \text{where} \quad \hat{\mathbf{y}} = F\mathbf{y}((x)).$$

- Find discrete approximation $\mathbf{y}$ to $y(x)$ which solves a constrained problem (SOCP)

$$\mathbf{y} = \arg\min_{\mathbf{u}} \|\mathbf{u}\|_1 \qquad \text{subject to} \qquad \|\Sigma(HF\mathbf{u} - \mathbf{b})\|_2^2 \leq \delta, \qquad (17)$$

$\mathbf{b} = (-i\hat{g}_{-N}, \cdots, 0, \cdots, i\hat{g}_{N-1})$. Σ weights the data fit term.

- Introduce λ and solve an l_1 minimization

$$\mathbf{y} = \arg\min_{\mathbf{u}}\{\lambda\|\mathbf{u}\|_1 + \frac{1}{2}\|\Sigma(HF\mathbf{u} - \mathbf{b})\|_2^2\}$$

Recast the problem dependent on the data

We obtain a new inverse problem replacing a total variation for spatial data

New formulation still uses total variation but on spectral data

Can now use any standard minimization technique

i.e. Split Bregman can be employed.

See results in presentation from November! `http://math.la.asu.edu/~rosie/mypresentations/PragueNov11.pdf`