Structured Matrices, Multigrid, and Image Processing

Marco Donatelli

Dept. of Science and High Tecnology – U. Insubria (Italy)

TUM 2016

Outline

1. Convolution and Structured Matrices
2. Ill-posed problems and regularization
3. Multigrid Methods for Structured Matrices

Outline

Convolution

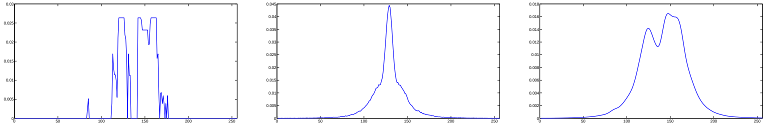
Definition

The *convolution* of two functions (signals) f and k is

$$g(x) = \int_{\mathbb{R}} k(x-s)f(s)\mathrm{d}s$$

In the applications usually f is the original signal, k is the convolution kernel and g is the observed signal.

Example



f k g

Shift-invariant: every point is subject to the same phenomenon.

Assumption

$f(x) = 0$ *for* $x \notin [a, b]$.

- The convolution becomes

$$g(x) = \int_a^b k(x-s)f(s)\mathrm{d}s$$

- Discretize the integral using n rectangles defining the grid points

$$x_j = a + jh, \qquad h = \frac{b-a}{n}, \qquad j = 0, \ldots, n-1.$$

- Approximate g at the grid points x_i, for $i = 0, \ldots, n-1$, by

$$\begin{aligned} g(x_i) &= \int_a^b k(x_i - s)f(s)\mathrm{d}s \\ &\approx h \sum_{j=0}^{n-1} k(x_i - x_j)f(x_j) \end{aligned}$$

Linear system

- Defining
$$K_{i,j} = hk(x_i - x_j) = hk\left((i-j)h\right)$$
we have that
$$g(x_i) \approx \sum_{j=0}^{n-1} K_{i,j} f(x_j), \qquad i = 0, \ldots, n-1,$$
which is the linear system
$$\mathbf{g} = K\mathbf{f}, \tag{1}$$
where $g_i = g(x_i)$ and $f_i = f(x_i)$, for $i = 0, \ldots, n-1$.
- Note that
$$k_{i-j} := hk\left((i-j)h\right) = K_{i,j} \tag{2}$$
shift-invariant property.

- Thanks to (2), the matrix

$$K = \begin{bmatrix} k_0 & k_{-1} & \dots & k_{-(n-1)} \\ k_1 & k_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & k_{-1} \\ k_{n-1} & \dots & k_1 & k_0 \end{bmatrix} \qquad (3)$$

has constant elements along the diagonals and it is called Toeplitz matrix.

- The matrix K depends on only $2n-1$ parameters

$$\mathbf{k} = [k_{-n+1}, \dots, k_{-1}, k_0, k_1, \dots, k_{n-1}]^T .$$

How to work with K

- Memorize only $\mathbf{k} \in \mathbb{R}^{2n-1}$.
- Is it possible to save CPU time for the computations (matrix-vector product, inversion, etc.)?

Discrete Convolution

- Let $\mathbf{f} = \mathbf{e}_i$ the i-th vector of the canonical base, then

$$\mathbf{g} = K\mathbf{e_i} = [\dots, k_{-1}, k_0, k_1, \dots]^T,$$

  hence, if $k_i = 0$ for $|i| > n/2$ then $\mathbf{k}$ can be obtained observing a point in the middle of the interval ... next lesson on inverse problems.

- The linear system (1) is the discrete convolution with zero-Dirichlets boundary conditions

$$g_i = \sum_{j=0}^{n-1} K_{i,j} f_j = \sum_{j=0}^{n-1} k_{i-j} f_j, \qquad i = 0, \dots, n-1$$

- Rotate the vector $\mathbf{k}$, shift, multiply component wise with $\mathbf{f}$ and then sum:

$$g_j = \sum \begin{array}{ccccccc} k_{n-1} & \cdots & k_1 & k_0 & k_{-1} & \cdots & k_{-n+1} \\ * & \cdots & * & * & * & \cdots & * \\ \tilde{f}_{j-(n-1)} & \cdots & \tilde{f}_{j-1} & \tilde{f}_j & \tilde{f}_{j+1} & \cdots & \tilde{f}_{j+n-1} \end{array}$$

$$= k_{n-1} \cdot \tilde{f}_{j-n+1)} + \cdots + k_1 \cdot \tilde{f}_{j-1} + k_0 \cdot \tilde{f}_j + k_{-1} \cdot \tilde{f}_{j+1} + \cdots + k_{-n+1} \cdot \tilde{f}_{j+n-1}$$

  where

$$\tilde{f}_i = \begin{cases} f_i & \text{if } i = 0, 1, \cdots, n-1, \\ 0 & \text{otherwise.} \end{cases}$$

Full Convolution

- Let $k_i = 0$ for $|i| > m$, $m < n-1$, then removing Assumption 1 we have the full discrete convolution

$$\mathbf{g} = K_{\text{full}} \tilde{\mathbf{f}}$$

$$= \left[\begin{array}{ccc|ccccc|ccc}
k_m & \dots & k_1 & k_0 & k_{-1} & \dots & k_{-m} & 0 & & & \\
 & k_m & \dots & k_1 & k_0 & k_{-1} & \dots & \ddots & & & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
 & & & \ddots & \dots & k_1 & k_0 & k_{-1} & \dots & k_{-m} & \\
 & & & 0 & k_m & \dots & k_1 & k_0 & k_{-1} & \dots & k_{-m}
\end{array}\right] \tilde{\mathbf{f}}$$

where $\tilde{\mathbf{f}} = \left[f_{-m} \dots f_{-1} \mid \mathbf{f}^T \mid f_n, \dots, f_{n+m-1} \right]^T \in \mathbb{R}^{n+2m}$.

- No assumptions on the boundary conditions

$$g_i = \sum_{j \in \mathbb{Z}} k_{i-j} f_j, \qquad i = 0, \dots, n-1$$

Assumption

Assume that the function f is periodic with period $b-a$. Then

$$f_{-i} = f_{n-i}, \qquad f_{n+i-1} = f_{i-1}, \qquad i = 1, 2, \ldots, m.$$

- Let $m < n/2$, i.e., $supp(k) \subset [a, b]$ as in the example, then

$$\mathbf{g} = K_{\text{full}}\tilde{\mathbf{f}} = K_{\text{circ}}\mathbf{f}$$

where

$$K_{\text{circ}} = \begin{bmatrix} k_0 & \ldots & k_{-m} & 0 & k_m & \ldots & k_1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ k_m & \ddots & \ddots & \ddots & \ddots & \ddots & k_m \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ k_{-m} & \ddots & \ddots & \ddots & \ddots & \ddots & k_{-m} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ k_{-1} & \ldots & k_{-m} & 0 & k_m & \ldots & k_0 \end{bmatrix}$$

Circulant matrix 2

- The matrix K_{circ} depends on only n parameters in the first column

$$\mathbf{k} = \Big[k_0, \dots, k_m, 0, k_{-m}, \dots, k_{-1}\Big]^T \in \mathbb{R}^n$$

- Given the observation of $\mathbf{e}_{\frac{n+1}{2}}$ (n odd for simplicity)

$$\breve{\mathbf{k}} = \Big[0, k_{-m}, \dots k_{-1}, k_0, k_1, \dots, k_m, 0\Big]^T \in \mathbb{R}^n$$

we have that

$$\mathbf{k} = \mathrm{circshift}(\breve{\mathbf{k}}, \mathrm{n} - \mathrm{i}_0),$$

where $\breve{k}_{i_0} = k_0$ (indices start from zero).

Circular discrete convolution

- Using the congruence relation index $\mathbf{k}$ in the standard way

$$\mathbf{k} \quad = \quad \begin{bmatrix} k_0 \\ \vdots \\ k_m \\ 0 \\ k_{-m} \\ \vdots \\ k_{-1} \end{bmatrix} \quad \xrightarrow{\bmod n} \quad \begin{bmatrix} k_0 \\ k_1 \\ \vdots \\ k_{n-2} \\ k_{n-1} \end{bmatrix}$$

- Circular discrete convolution

$$\mathbf{g} = K_{\mathrm{circ}}\mathbf{f} = \mathbf{f} * \mathbf{k} \tag{4}$$

where

$$g_i = \sum_{j=0}^{n-1} k_{(i-j) \bmod n} f_j, \qquad i = 0, \dots, n-1$$

Definition

Let $\mathbf{f} \in \mathbb{C}^n$ the Discrete Fourier Transform (DFT) of $\mathbf{f}$ is

$$\hat{f}_k := \sum_{j=0}^{n-1} f_j \mathrm{e}^{-\frac{\mathrm{i}2\pi jk}{n}}, \qquad k = 0, \dots, n-1.$$

To simplify the notation define

$$\omega_n := \mathrm{e}^{-\frac{\mathrm{i}2\pi}{n}}$$

(note that ω_n^k is the k-th root of the unity, for $k = 0, \dots, n-1$), thus

$$\hat{f}_k := \sum_{j=0}^{n-1} \omega_n^{jk} f_j, \qquad k = 0, \dots, n-1.$$

In matrix form

$$\hat{\mathbf{f}} = F_n \mathbf{f},$$

where $[F_n]_{k,j} = \omega_n^{jk}$, for $k, j = 0, \dots, n-1$.

Properties of the DFT

Proposition

$$\sum_{j=0}^{n-1} \omega_n^{jk} = \begin{cases} n & \textit{if } k = sn,\ s \in \mathbb{Z}, \\ 0 & \textit{otherwise}. \end{cases}$$

Properties of F_n

1. $F_n = F_n^T$.
2. $F_n^{-1} = \frac{1}{n} F_n^H$.
3. $F_n^H = \mathcal{J} F_n = F_n \mathcal{J}$ where $\mathcal{J}$ is the permutation matrix

$$\mathcal{J} = \left[\begin{array}{c|ccc} 1 & & & \\ \hline & & & 1 \\ & & \iddots & \\ & 1 & & \end{array}\right]$$

Corollary

$$F_n^{-1} = \frac{1}{n} \mathcal{J} F_n \qquad (5)$$

Remark

Thanks to (5), $F_n^{-1}\mathbf{x}$ can be computed using the same algorithm implemented for the direct product $F_n\mathbf{x}$.

Fast Fourier Transform (FFT)

- The matrix-vector requires $O(n^2)$ arithmetic operations but when the matrix is F_n it can be computed in $O(n\log(n))$ by FFT for $n = 2^\alpha$.
- FFT was included in the Top 10 Algorithms of 20^{th} Century.
- Different algorithms (decimation in time or decimation in space) can be used and several implementation details can be found in
  `C. Van Loan, ‘‘Computational Frameworks for the Fast Fourier Transform’’, Frontiers in Applied Mathematics, SIAM, 1992.`

Convolution Theorem

Theorem

Let $\mathbf{f}, \mathbf{k} \in \mathbb{C}^n$, *then*

$$F_n(\mathbf{f} * \mathbf{k}) = (F_n\mathbf{f}) \circ (F_n\mathbf{k}),$$

where $*$ *is defined in* (4) *and* $\circ$ *is the Hadamard (entrywise) product.*

Spectral decomposition of $K_{\rm circ}$

From the previous theorem and Property 2, it holds

$$K_{\rm circ}\mathbf{f} = \mathbf{f} * \mathbf{k} = \frac{1}{n}F_n^H(F_n\mathbf{f} \circ F_n\mathbf{k}) = \frac{1}{n}F_n^H{\rm diag}(F_n\mathbf{k})F_n\mathbf{f} \tag{6}$$

Since (6) has to hold for all $\mathbf{f} \in \mathbb{C}^n$, it must be

$$K_{\rm circ} = \frac{1}{n}F_n^H{\rm diag}(F_n\mathbf{k})F_n \tag{7}$$

Toeplitz matrices vs circulants

- Goal: compute the product $K\mathbf{x}$ with K Toeplitz matrix in (3).
- Construct the circulant matrix

$$C = \begin{bmatrix} K & M_1 \\ M_2 & M_3 \end{bmatrix} \in \mathbb{C}^{m \times m}$$

with $m \geq 2n - 1$ and $\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix}$.

- Compute $\mathbf{z} = C\mathbf{y}$, thus

$$K\mathbf{x} = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}$$

- Choosing m as the smallest power of 2 such that $m \geq 2n - 1$, pad with zeros if necessary, we can use FFT with a cost of $O(n \log n)$.

- The matrix-vector product of Toeplitz matrices of arbitrary size n can be computed by immersion into a circulant of size $m = 2^\alpha$ and then applying the FFT.
- How compute FFT of arbitrary size? By Toeplitz matrices!
- Use the relation $-jk = ((k-j)^2 - k^2 - j^2)/2$.
- It holds

$$F_n = \left[\mathrm{e}^{-\frac{\mathrm{i}2\pi jk}{n}}\right]_{j,k=0}^{n-1} = \left[\mathrm{e}^{\frac{\mathrm{i}\pi(((k-j)^2-k^2-j^2))}{n}}\right]_{j,k=0}^{n-1} = DTD$$

where

$$D = \mathrm{diag}_{k=0,\ldots,n-1}\left(\mathrm{e}^{-\frac{\mathrm{i}\pi k^2}{n}}\right), \qquad T = \left[\mathrm{e}^{\frac{\mathrm{i}\pi(k-j)^2}{n}}\right]_{k,j=0}^{n-1}.$$

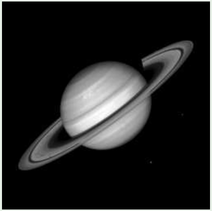
2D Convolution

Definition

The *convolution* of two functions (signals) f and k is

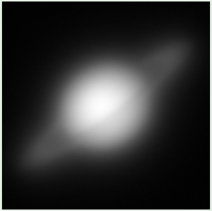
$$g(x_1, x_2) = \int_{a_1}^{b_1} \int_{a_2}^{b_2} k(x_1 - s_1, x_2 - s_2) f(s_1, s_2) \mathrm{d}s_1 \mathrm{d}s_2$$

Example



f

k



g

Discretization

- Discretize on a uniform grid on $[a_1, b_1] \times [a_2, b_2]$.
- Resizing $n \times m$ images in vectors of length nm concatenating the columns, we obtain the linear system

$$\mathbf{g} = K\mathbf{f}$$

where K is the block-Toeplitz-Toeplitz-block (BTTB) matrix

$$K = \begin{bmatrix} K_0 & K_{-1} & \dots & K_{-(n-1)} \\ K_1 & K_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & K_{-1} \\ K_{n-1} & \dots & K_1 & K_0 \end{bmatrix},$$

$$K_j = \begin{bmatrix} k_{j,0} & k_{j,-1} & \dots & k_{j,-m+1} \\ k_{j,1} & k_{j,0} & \ddots & \vdots \\ \vdots & \ddots & \ddots & k_{j,-1} \\ k_{j,m-1} & \dots & k_{j,1} & k_{j,0} \end{bmatrix}, \qquad k_{j,s} = h_x h_y k(jh_x, sh_j).$$

2D DFT

- Circulant matrices have a similar block circulant circulant block (BCCB) structure.
- 2D DFT by tensor product

$$F_n^{2D} = F_n \otimes F_n.$$

- Since $(A \otimes B)\mathrm{vec}(X) = \mathrm{vec}(BXA^T)$ it holds

$$F_n^{2D}\mathrm{vec}(X) = \mathrm{vec}(F_n X F_n),$$

which is the application of the 1D DFT to each row and column of X.

Linear Algebra approach for circulant matrices

- Exercise: Prove that the set of circulant matrices

$$\mathcal{C}_n = \left\{ A \in \mathbb{C}^{n\times n} \;:\; A = F_n^H D F_n \text{ with } D \text{ diagonal matrix} \right\}$$

is closed for sum, product and inversion (Hint: Caley-Hamilton theorem).

- Denote by $Circ(\mathbf{a})$ the circulant matrix defined by $\mathbf{a}$, e.g., $Circ(\mathbf{k}) = K_{circ}$, namely

$$Circ(\mathbf{a}) = \begin{bmatrix} a_0 & a_{n-1} & \dots & a_1 \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1} \\ a_{n-1} & \dots & a_1 & a_0 \end{bmatrix} = p_{n-1}(Z) \in \mathbb{P}_{n-1}, \qquad (8)$$

where

$$p_{n-1}(x) = \sum_{j=0}^{n-1} a_j x^j \qquad \text{and} \qquad Z := \left[\begin{array}{ccc|c} & & & 1 \\ \hline 1 & & & \\ & \ddots & & \\ & & 1 & \end{array}\right].$$

Spectral decomposition of Circulant matrices

- Define $\mathbf{y} \in \mathbb{R}^n$ by uniform sampling in $[0, 2\pi]$:

$$y_s = \frac{2\pi s}{n}, \qquad s = 0, \dots, n-1.$$

- The spectral decomposition of Z is

$$Z = F_n \Lambda F_n^{-1}, \qquad \Lambda = \mathrm{diag}(\mathrm{e}^{\mathrm{i}\mathbf{y}}) \tag{9}$$

- Combining (9) with (8), the spectral decomposition of $Circ(\mathbf{a})$ is

$$Circ(\mathbf{a}) = \frac{1}{n} F_n \mathrm{diag}(F_n^H \mathbf{a}) F_n^H \tag{10}$$

  1. the eigenvectors are the column of F_n, i.e, $\mathrm{e}^{-\mathrm{i}j\mathbf{y}}$ the j-th frequency.
  2. the eigenvalues of $Circ(\mathbf{a})$ are

$$\lambda_j = [F_n^H \mathbf{a}]_j = \sum_{s=0}^{n-1} a_s \mathrm{e}^{\mathrm{i}j y_s}, \qquad j = 0, \dots, n-1.$$

- Which is the difference between (10) and (7)?

For our convolution matrix $K_{\mathrm{circ}} = Circ(\mathbf{k})$ it holds

$$\lambda_j = \sum_{s=0}^{n-1} k_s \mathrm{e}^{\frac{\mathrm{i}2\pi js}{n}} = \sum_{s=0}^{m} k_s \mathrm{e}^{\frac{\mathrm{i}2\pi js}{n}} + \sum_{s=-m}^{-1} k_s \mathrm{e}^{\frac{\mathrm{i}2\pi js}{n}}$$

$$= \sum_{s=-m}^{m} k_s \mathrm{e}^{\frac{\mathrm{i}2\pi js}{n}} = S_m[k](y_j), \qquad j = 0, \ldots, n-1.$$

which is the *m-th partial sum of the Fourier series* of the function k assuming that $k \in L^1_{[0,2\pi]}$ is 2π-periodic and its Fourier coefficients are

$$k_j = \frac{1}{2\pi} \int_0^{2\pi} k(x) \mathrm{e}^{-\mathrm{i}jx} \mathrm{d}x, \qquad j \in \mathbb{Z}, \qquad k(x) = \sum_{j\in\mathbb{Z}} k_j \mathrm{e}^{\mathrm{i}jx}.$$

Remark

We can construct a sequence of circulant matrices associated to k with increasing size $2m+1$ using $S_m[k](x)$.

The operator of Toeplitz matrices

Definition

Given a function $f : [0, 2\pi] \to \mathbb{C}$, 2π-periodic, $f \in L^1_{[0,2\pi]}$ and with Fourier coefficients

$$a_j = \frac{1}{2\pi} \int_0^{2\pi} f(x) \mathrm{e}^{-\mathrm{i}jx} \mathrm{d}x, \qquad j \in \mathbb{Z},$$

the associated Toeplitz matrix of order n is $T_n = T_n(f) = [a_{i-j}]_{i,j=0}^{n-1}$, namely

$$T_n = \begin{bmatrix} a_0 & a_{-1} & \dots & a_{-(n-1)} \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{-1} \\ a_{n-1} & \dots & a_1 & a_0 \end{bmatrix}$$

Example

$$\begin{cases} u''(x) = g(x) & x \in (0,1) \\ u(0) = u(1) = 0 \end{cases}$$

Finite differences discretization of order 2 $\Rightarrow f(x) = 2 - 2\cos(x)$

Property of $T_n(f)$

Definition

$$T_n(\cdot) : L^1_{[0,2\pi]} \to \mathbb{C}^{n\times n}$$

Lemma

1. $T_n(\alpha f + \beta g) = \alpha T_n(f) + \beta T_n(g)$
2. f *real* $\Longrightarrow T_n(f)$ *is a Hermitian matrix,*
3. $f \geq 0 \Longrightarrow T_n(f)$ *is positive semidefinite,*
4. $f \geq 0$ *and* $\sup f > 0 \Longrightarrow T_n(f)$ *is positive definite.*

Lemma

Let f *be real such that* $m_f \leq f \leq M_f$ *with* $m_f \neq M_f$*, then* $\sigma(T_n(f)) \subset (m_f, M_f)$.

Eigenvalues distribution

Definition

Let $f : [0, 2\pi] \to \mathbb{C}$ be $L^1_{[0,2\pi]}$. Let $\{A_n\}$ be a sequence of matrices of size n with eigenvalues $\lambda_j(A_n)$, $j = 1, \ldots, n$.
$\{A_n\}$ is distributed as the pair $(f, [0, 2\pi])$ in the sense of the eigenvalues:

$$\{A_n\} \sim_\lambda (f, [0, 2\pi]),$$

if for all continuous functions F

$$\lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} F(\lambda_j(A_n)) = \frac{1}{2\pi} \int_0^{2\pi} F(f(t))\mathrm{d}t.$$

Theorem

Let f be a real and 2π-periodic function. Then

$$\begin{aligned} \{T_n(f)\} &\sim_\lambda (f, [0, 2\pi]), && \text{if } f \text{ is continuous,} \\ \{C_n(f)\} &\sim_\lambda (f, [0, 2\pi]), && \text{if } f \text{ is Lipschitz.} \end{aligned}$$

Ill-posed problems and regularization

Marco Donatelli

Dept. of Science and High Tecnology – U. Insubria (Italy)

TUM 2016

Outline

Inverse problems

From the observation of a phenomenon we would obtain its birth.

Example

- A classical example is the Fredholm integral equation of the first kind

$$g(x) = \int_{\mathbb{R}} k(x, s) f(s) \mathrm{d}s \tag{1}$$

- Discrete example

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

  - The matrix-vector product is the direct problem.
  - The solution of the linear system, i.e., $\mathbf{x} = A^{-1}\mathbf{y}$ is the inverse problem.

Remark

Continuous inverse problems are often ill-posed.

Ill-posed Problems

A good book: H.W. Engl, M. Hanke, A. Neubauer, ''Regularization of Inverse Problems'', Kluwer Academic Publishers, 1996.

Definition

We say that a mathematical problem is well-posed if

1. a solution exists;
2. the solution is unique;
3. the solution depends continuously on the data.

We say that a mathematical problem is ill-posed if one of the conditions above does not hold.

- The Riemann-Lebesgue lemma states that the integral will approach zero as the number of oscillations increases $\Rightarrow$ (1) is ill-posed.
- The discretization of an ill-posed problem is severely ill-conditioned $\Rightarrow$ discrete ill-posed problems, see P. C. Hansen, ''Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion'', Mathematical Modeling and Computation, SIAM, 1998.

Outline

Discrete least squares

Given $A \in \mathbb{C}^{m\times n}$ and $\mathbf{b} \in \mathbb{C}^m$, instead of to solve $A\mathbf{x} = \mathbf{b}$ compute

$$\operatorname*{argmin}_{\mathbf{x}\in\mathbb{C}^n} \|A\mathbf{x} - \mathbf{b}\|_2^2. \tag{2}$$

Definition

Let $A \in \mathbb{C}^{m\times n}$, then exist U and V unitary matrices such that the singular values decomposition (SVD) of A is

$$A = U\Sigma V^H,$$

with $\Sigma = \mathrm{diag}_{i=1,\dots,t}(\sigma_i) \in \mathbb{R}^{m\times n}$, $t = \min(m,n)$ and $\sigma_1 \geq \sigma_2 \geq \sigma_t \geq 0$.

- Let $r = \mathrm{rank}(A)$, then

$$A = U\Sigma V^H = U_r\Sigma_r V_r^H = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^H.$$

- The minimum solution of (2) is

$$\mathbf{x}^\dagger = A^\dagger \mathbf{b}.$$

Numerical rank

Definition

The condition number of a matrix is

$$\nu_2(A) = \|A\|_2 \|A^\dagger\|_2 = \frac{\sigma_1}{\sigma_r}.$$

If $\sigma_r \approx 0$ then $\nu_2(A) >> 1$ and it may be that in exact arithmetic $\sigma_r = 0$.

Definition

Given a matrix A, its truncated singular values decomposition of order $s \leq r$ is

$$A_s = \sum_{i=1}^{s} \sigma_i \mathbf{u}_i \mathbf{v}_i^H.$$

Lemma

$$\|A - A_s\|_2 = \min_{\substack{B \in \mathbb{C}^{m\times n} \\ \mathrm{rank}(B) = s}} \|A - B\|_2 = \sigma_{s+1}.$$

Sensitivity analysis

- The observed object is usually affected by noise:

$$\mathbf{b}^\delta = \mathbf{b} + \boldsymbol{\xi}, \qquad \mathbf{b} = A\mathbf{x}^\dagger,$$

where $\delta = \|\boldsymbol{\xi}\|_2$ is the noise level.

- The computed solution becomes

$$\tilde{\mathbf{x}} = A^\dagger \mathbf{b}^\delta = A^\dagger(\mathbf{b} + \boldsymbol{\xi}) = \mathbf{x}^\dagger + \mathbf{e},$$

where

$$\mathbf{e} = A^\dagger \boldsymbol{\xi} = \sum_{i=1}^{r} \frac{\mathbf{u_i}^H \boldsymbol{\xi}}{\sigma_i} \mathbf{v}_i .$$

If $\sigma_i << 1$ and $\mathbf{u_i}^H \boldsymbol{\xi} \neq 0$, then $\mathbf{e}$ can be large even if δ is small.

Outline

Discrete ill-posed problems

- The singular values decays exponentially at zero without a significant gap.
- The singular vectors $\mathbf{v}_j$ and $\mathbf{u}_j$ are the j-th frequency.
- The noise $\boldsymbol{\xi}$ has nonzero components also in the high frequencies.

$$\Longrightarrow \quad \|\mathbf{e}\|_2 >> 1 \quad \Longrightarrow$$

Regularization

Change a little bit the problem obtaining a new nearby problem well-posed. There is always a parameter which balances:

1. how the new problem is far from the original one (approximation error)
2. how much the new problem is sensible to noise (stability).

Truncated SVD (TSVD)

- Instead of $A^\dagger \mathbf{b}^\delta$ take $A_s^\dagger \mathbf{b}^\delta$, $0 < s < r$, as approximation of $\mathbf{x}^\dagger$:

$$\begin{aligned}\mathbf{x}_s^\delta = A_s^\dagger \mathbf{b}^\delta &= \sum_{i=1}^{s} \frac{\mathbf{u_i}^H \mathbf{b}}{\sigma_i} \mathbf{v}_i + \sum_{i=1}^{s} \frac{\mathbf{u_i}^H \boldsymbol{\xi}}{\sigma_i} \mathbf{v}_i \\ &= \mathbf{x}^\dagger - \sum_{i=s+1}^{r} \frac{\mathbf{u_i}^H \mathbf{b}}{\sigma_i} \mathbf{v}_i + \sum_{i=1}^{s} \frac{\mathbf{u_i}^H \boldsymbol{\xi}}{\sigma_i} \mathbf{v}_i\end{aligned}$$

  where the first term is the truncation (approximation) error and the second term is the noise amplification (stability).
- s is the regularization parameter.
- The computed solution can be written as

$$\mathbf{x}_s^\delta = V \Phi_s \Sigma^\dagger U^H \mathbf{b}^\delta, \qquad \Phi_s = \begin{bmatrix} I_s & \\ & 0 \end{bmatrix}_{n \times n}$$

- A different choice of Φ_s can be applied, but must be a low-pass filter.

- Tikhonov method is

$$\operatorname*{argmin}_{\mathbf{x}\in\mathbb{C}^n}\left\{\|A\mathbf{x}-\mathbf{b}^\delta\|_2^2+\alpha\|\mathbf{x}\|_2^2\right\}, \tag{3}$$

where α balances the data fitting and the noise explosion.

- The solution of (3) is equivalent to the linear system

$$(A^HA+\alpha I)\mathbf{x}=A^H\mathbf{b}^\delta,$$

thus

$$\begin{aligned}\mathbf{x}&=V(\Sigma^T\Sigma+\alpha I)^{-1}\Sigma^T U^H\mathbf{b}^\delta\\&=V\Phi_{Tik}\Sigma^\dagger U^H\mathbf{b}^\delta\end{aligned}$$

where $\Phi_{Tik}=(\Sigma^T\Sigma+\alpha I)^{-1}\Sigma^T\Sigma=\mathrm{diag}_{i=1,\dots,t}(\phi_i)$, such that

$$\phi_i=\frac{\sigma_i^2}{\sigma_i^2+\alpha}\approx\begin{cases}1 & i \text{ small},\\ 0 & i \text{ large},\end{cases}\qquad i=1,\dots,t.$$

Outline

- The relative error begins to decrease until a certain "optimal" iteration is reached and then begins to increase because of the presence of noise, which starts to dominate the restoration process (semi-convergence).
- By stopping the iterations when the error is low, we obtain a regularized approximation of the solution.
- Landweber method (gradient descent method for (2))

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau A^H(\mathbf{b}^\delta - A\mathbf{x}_k), \qquad (4)$$

which is convergent if $0 < \tau < \frac{2}{\rho(A^H A)}$.

Convergence in the noise free case

Lemma

If $\mathbf{x}_0 = \mathbf{0}$ then Landweber method in (4) converges to $\mathbf{x}^\dagger = A^\dagger \mathbf{b}^\delta$.

Filter factor of Landweber method

- From the proof of the previous Lemma we obtain the filter factors $\theta_{k,i}$ s.t.

$$\mathbf{x}_k = V\Phi_{L,k}\Sigma^\dagger U^H\mathbf{b}^\delta, \qquad \Phi_{L,k} = \mathrm{diag}_{i=1,\dots,t}(\theta_{k,i}), \qquad \theta_{k,i} = 1-(1-\tau\sigma_i^2)^{k+1}$$

- Fix k, it holds

$$\theta_{k,i} = \begin{cases} 1 & i \text{ small}, \\ 0 & i \text{ large}, \end{cases} \qquad i = 1,\dots,t.$$

- Fix i and let $s > k$, then it holds

$$\theta_{s,i} > \theta_{k,i}$$

Landweber methods starts reconstructing the low frequencies and then passes at the medium and high frequencies.

Iterated Tikhonov method

- Iterated Tikhonov method is obtained refining a given approximation $\mathbf{x}_k$ by solving the error equation using Tikhonov method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (A^H A + \alpha I)^{-1} A^H (\mathbf{b}^\delta - A\mathbf{x}_k). \tag{5}$$

- It is convergent for $\alpha > 0$ and $\mathbf{x} \to \mathbf{x}^\dagger$ whenever $\mathbf{x}_0 = \mathbf{0}$.
- The iteration (5) can interpreted as a prenditioned Landweber method, where $\tau = 1$ and the preconditioner is $(A^H A + \alpha I)^{-1}$.
- Further regularization parameter α, which balances the convergence speed and how much is steep the semiconvergence:
  - small $\alpha \Longrightarrow$ fast convergence but unstable convergence,
  - large α slow convergence like Landweber.
- $(A^H A + \alpha I)^{-1}$ could be computationally expensive. Hence it should be approximated, but what happens at the convergence?

Let β be the regularization parameter.

- Discrepancy principle, largely used with iterative methods, requires to know δ. Compute the approximation corresponding to the smallest β that satisfies the condition

$$\|A\mathbf{x}_\beta - \mathbf{b}^\delta\|_2 < \nu\delta, \qquad \nu > 1.$$

- L-curve: Compute $\mathbf{x}_\beta$ for several values of β and plot in log-scale $\|\mathbf{x}_\beta\|_2$ and $\|A\mathbf{x}_\beta - \mathbf{b}^\delta\|_2$, then the best value of β is in the corner of the L-shape curve balancing data fitting and explosion of noise.
- Generalized cross-validation (GCV), etc.

Outline

ℓ_1 regularization

- The ℓ_2-norm leads to over-smoothed restorations.
- In some applications other regularization terms could be useful, like the ℓ_1-norm if the solution is sparse (e.g. images in the wavelets domain).
- Let W^H be a wavelet or tight-frame synthesis operator $(W^H W = I)$ and $\mathbf{y}$ the frame coefficients such that

$$\mathbf{x} = W^H \mathbf{y}.$$

- Let $\mathbf{x}$ be an image, then $\mathbf{y}$ is sparse (wavelet coefficients).
- The $\ell_2 - \ell_1$ minimum problem is

$$\underset{\mathbf{y}\in\mathbb{C}^n}{\operatorname{argmin}} \left\{ \frac{1}{2} \|B\mathbf{y} - \mathbf{b}^\delta\|_2^2 + \alpha \|\mathbf{y}\|_1 \right\}, \tag{6}$$

where $B = AW^H$.

Iterative soft-thresholding algorithm (ISTA)

- Let the nonlinear soft-thresholding operator $\mathbf{S}_\mu$ be defined component-wise as

$$[\mathbf{S}_\mu(\mathbf{y})]_i = S_\mu(y_i),$$

with S_μ the soft-thresholding function

$$S_\mu(y_i) = \operatorname{sgn}(y_i) \max\left\{|y_i| - \mu,\, 0\right\}.$$

- Combining Landweber and soft-thresholding we obtain the ISTA

$$\mathbf{x}_{k+1} = \mathbf{S}_\mu(\mathbf{x}_k + \tau A^H(\mathbf{b}^\delta - A\mathbf{x}_k)), \tag{7}$$

which converges to the solution of (6)

I. Daubechies, M. Defrise, and C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Comm. Pure Appl. Math., 57--11 (2004), pp. 1413--1457.