Preconditioning

Noisy, Ill-Conditioned Linear Systems

James G. Nagy
Emory University
Atlanta, GA

Outline

1. The Basic Problem
2. Regularization / Iterative Methods
3. Preconditioning
4. Example: Image Restoration
5. Summary
Basic Problem

Linear system of equations

\[ b = Ax + e \]

where

- \( A, b \) are known
- \( A \) is large, structured, \textit{ill-conditioned}
- Goal: Compute an approximation of \( x \)

\textbf{Applications:} Ill-posed inverse problems.

- Geomagnetic Prospecting
- Tomography
- \textbf{Image Restoration}
  - \( b = \) observed image
  - \( A = \) blurring matrix (structured)
  - \( e = \) noise
  - \( x = \) true image
A = matrix

b = blurred, noisy image

x = true image

Vision is the art of seeing what is invisible to others.

Jonathan Swift
Computational difficulties revealed through SVD:

Let \( A = U\Sigma V^T \) where

- \( \Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_N) \), \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N \geq 0 \)
- \( U^T U = I \), \( V^T V = I \)

For ill-posed inverse problems,

- \( \sigma_1 \approx 1 \), small singular values cluster at 0
- small singular values \( \Rightarrow \) oscillating singular vectors
Inverse solution for noisy, ill-posed problems:

If \( A = U \Sigma V^T \), then

\[
\hat{x} = A^{-1}(b + e) \\
= V \Sigma^{-1} U^T (b + e) \\
= \sum_{i=1}^{n} \frac{u_i^T (b + e)}{\sigma_i} v_i \\
= \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i + \sum_{i=1}^{n} \frac{u_i^T e}{\sigma_i} v_i \\
= x + \text{error}
\]
b = blurred, noisy image

x = true image

Vision is the art of seeing what is invisible to others.

x = inverse solution
Regularization

Basic Idea: Filter out effects of small singular values.

\[ x_{\text{reg}} = \sum_{i=1}^{n} \phi_i \frac{u_i^T b}{\sigma_i} v_i \]

where the "filter factors" satisfy

\[ \phi_i \approx \begin{cases} 
1 & \text{if } \sigma_i \text{ is large} \\
0 & \text{if } \sigma_i \text{ is small} 
\end{cases} \]
Regularization

Some regularization methods:

1. Truncated SVD

\[ x_{\text{tsvd}} = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} v_i \]

2. Tikhonov

\[ x_{\text{tik}} = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2} \frac{u_i^T b}{\sigma_i} v_i \]

3. Wiener

\[ x_{\text{wien}} = \sum_{i=1}^{n} \frac{\delta_i \sigma_i^2}{\delta_i \sigma_i^2 + \gamma_i^2} \frac{u_i^T b}{\sigma_i} v_i \]
Iterative Regularization

Basic idea:
- Use an iterative method (e.g., conjugate gradients)
- Terminate iteration before theoretical convergence:
  - Early iterations reconstruct solution.
  - Later iterations reconstruct noise.

Some important methods:
- CGLS, LSQR, GMRES
- MR2 (Hanke, ’95)
- MRNSD (Kaufman, ’93; N., Strakos, ’00)
Iterative Regularization

Efficient for large problems, provided

1. Multiplication with $A$ is not expensive.

   Image restoration ⇔ Use FFTs

2. Convergence is rapid enough.

   - CGLS, LSQR, GMRES, MR2 often fast, especially for severely ill-posed, noisy problems.

   - MRNSD based on steepest descent, typically converges very slowly.
inverse solution

relative error

CGLS
MRNSD

659 iterations

Vision is the art of seeing what is invisible to others.
The image shows a graph plotting relative error against iteration for two methods, CGLS and MRNSD, in the context of image processing. The graph indicates that CGLS converges faster than MRNSD, as shown by the sharper decrease in relative error over iterations. The inset images represent blurred images before and after processing, with the latter showing clarity improved after 41 iterations.
Preconditioning

Purposes of preconditioning:

1. Accelerate convergence.
   - Apply iterative method to $P^{-1}Ax = P^{-1}b$.
   - In this case we minimize $||x||_2$.

2. Enforce regularization constraint on solution. (Hanke, '92; Hansen, '98)
   - Apply iterative method to $AL^{-1}Lx = b$.
   - In this case, we minimize $||Lx||_2$. 

Basic idea:

- Find a matrix $L$ to enforce smoothness constraint
  \[
  \min \|Lx\|_2
  \]

- Typically $L$ approximates a derivative operator.
Preconditioning for Speed

Typical approach for $Ax = b$

- Find matrix $P$ so that $P^{-1}A \approx I$.

- "Ideal" choice: $P = A$
  
  In this case, converge in one iteration to $x = A^{-1}b$

For ill-conditioned, noisy problems

- Inverse solution is corrupted with noise

- "Ideal" regularized preconditioner: If $A = U\Sigma V^T$
  (Hanke, N., Plemmons, ’93)

  $$P = U\Sigma_k V^T = U \text{diag}(\sigma_1, \ldots, \sigma_k, 1, \ldots, 1)V^T$$
Preconditioning for Speed

Notice that the preconditioned system is:

\[ P^{-1}A = (U \Sigma_k V^T)^{-1}(U \Sigma V^T) \]

\[ = V \Sigma_k^{-1} \Sigma V^T \]

\[ = V \Delta V^T \]

where \( \Delta = \text{diag}(1, \ldots, 1, \sigma_{k+1}, \ldots, \sigma_n) \)

That is,

- Large (good) singular values clustered at 1.
- Small (bad) singular values not clustered.
Preconditioning for Speed

Remaining questions:

1. How to choose truncation index, $k$?

   Use regularization parameter choice methods, e.g., GCV, L-curve, Picard condition

2. We can’t compute SVD, so now what?

   Use SVD approximation.
An SVD Approximation:

- Decompose $A$ as: (Van Loan and Pitsianis, ’93)
  \[
  A = C_1 \otimes D_1 + C_2 \otimes D_2 + \cdots + C_k \otimes D_k
  \]
  where $C_1 \otimes D_1 = \arg\min ||A - C \otimes D||_F$.

- Choose a “structured” (or sparse) $U$ and $V$.

- Let $\Sigma = \arg\min ||A - U\Sigma V^T||_F$.

That is,

\[
\Sigma = \text{diag} \left( U^T A V \right) \\
= \text{diag} \left( U^T \left( \sum_{i=1}^{k} C_i \otimes D_i \right) V \right)
\]
Choices for $U$ and $V$ depend on problem (application).

- Since
  \[ A = C_1 \otimes D_1 + C_2 \otimes D_2 + \cdots + C_k \otimes D_k \]
  and
  \[ C_1 \otimes D_1 = \arg\min \| A - C \otimes D \|_F \]
  we might use singular vectors of $C_1 \otimes D_1$.

- For image restoration, we also use

  Fourier Transforms (FFT)ns

  Discrete Cosine Transforms (DCT)ns
Example: Image Restoration

1. Matrix Structure

2. Efficiently computing SVD approximation
First, how do we get the matrix, $A$?

- Using linear algebra notation, the $i$-th column of $A$ can be written as:

$$Ae_i = \begin{bmatrix} a_1 & \cdots & a_i & \cdots & a_n \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = a_i$$

- In an imaging system,

$$e_i = \text{point source}$$
$$Ae_i = \text{point spread function (PSF)}$$
Matrix Structure in Image Restoration

point source

PSF
Matrix Structure in Image Restoration

Spatially invariant PSF implies:

$e_i \quad e_j \quad e_k$

$Ae_i \quad Ae_j \quad Ae_k$
Matrix Structure in Image Restoration

That is, spatially invariant implies

- Each column of $A$ is identical, modulo shift.
- One point PSF is enough to fully describe $A$.
- $A$ has Toeplitz structure.
Matrix Structure in Image Restoration

\[ \mathbf{e}_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \text{blur} \rightarrow \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \rightarrow A\mathbf{e}_5 = \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{31} \\ p_{32} \\ p_{33} \end{bmatrix} \]

\[ A = \begin{bmatrix} p_{22} & p_{21} & p_{12} & p_{11} \\ p_{23} & p_{22} & p_{13} & p_{12} \\ p_{23} & p_{23} & p_{13} & p_{12} \\ p_{32} & p_{31} & p_{22} & p_{21} \\ p_{33} & p_{32} & p_{23} & p_{21} \\ p_{33} & p_{33} & p_{23} & p_{22} \end{bmatrix} \]
## Matrix Summary

<table>
<thead>
<tr>
<th>Boundary Condition</th>
<th>Matrix Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>zero</td>
<td>BTTB</td>
</tr>
<tr>
<td>periodic</td>
<td>BCCB (use FFT)</td>
</tr>
<tr>
<td>reflexive (strongly symmetric)</td>
<td>BTTB+BTHB+BHTB+BHHB (use DCT)</td>
</tr>
</tbody>
</table>

B = block  
T = Toeplitz  
C = circulant  
H = Hankel
**Matrix Structure in Image Restoration**

For a separable PSF, we get:

\[
\begin{bmatrix}
p_{11} & p_{12} & p_{13} \\
p_{21} & p_{22} & p_{23} \\
p_{31} & p_{32} & p_{33}
\end{bmatrix} = cd^T = \begin{bmatrix}
c_1d_1 & c_1d_2 & c_1d_3 \\
c_2d_1 & c_2d_2 & c_2d_3 \\
c_3d_1 & c_3d_2 & c_3d_3
\end{bmatrix} \rightarrow Ae_5 = \begin{bmatrix}
c_1 \\
c_2 \\
c_3
\end{bmatrix} \begin{bmatrix}
d_1 \\
d_2 \\
d_3
\end{bmatrix} = \mathbf{C} \otimes \mathbf{D}
\]
If the PSF is not separable, we can still compute:

\[ P = \sum_{i=1}^{r} c_i d_i^T \quad \text{(sum of rank-1 matrices)} \]

and therefore, get

\[ A = \sum_{i=1}^{r} C_i \otimes D_i \quad \text{(sum of Kron. products)} \]

In fact, we can get “optimal” decompositions.

(Kamm, N, ’00; N., Ng, Perrone, 03)
Use $A \approx U \Sigma V^T$, where

- If $\mathcal{F} \left( \sum C_i \otimes D_i \right) \mathcal{F}^*$ is best,
  $U = V = \mathcal{F}^*$, $\Sigma = \text{diag} \left( \mathcal{F} \left( \sum C_i \otimes D_i \right) \mathcal{F}^* \right)$

- If $\mathcal{C} \left( \sum C_i \otimes D_i \right) \mathcal{C}^T$ is best,
  $U = V = \mathcal{C}^T$, $\Sigma = \text{diag} \left( \mathcal{C} \left( \sum C_i \otimes D_i \right) \mathcal{C}^T \right)$

- If $\left( U_c \otimes U_d \right)^T \left( \sum C_i \otimes D_i \right) \left( V_c \otimes V_d \right)$ is best,
  $U = U_c \otimes U_d$, $V = V_c \otimes V_d$,
  $\Sigma = \text{diag} \left( \left( U_c \otimes U_d \right)^T \left( \sum C_i \otimes D_i \right) \left( V_c \otimes V_d \right) \right)$
659 CGLS iterations

Relative error vs iteration

- CGLS
- MRNSD

Vision is the art of seeing what is invisible to others.
• Preconditioning ill-posed problems is difficult, but possible.

• Can build approximate SVD from Kronecker product approximations.

• Can implement efficiently for image restoration.

• Matlab software: RestoreTools (Lee, N., Perrone, ’02)

  Object oriented approach for image restoration.
  http://www.mathcs.emory.edu/~nagy/RestoreTools/

  Related software for ill-posed problems
  (Hansen, Jacobsen)
  http://www.imm.dtu.dk/~pch/Regutools/