SIGNAL AND IMAGE RESTORATION: SOLVING ILL-POSED INVERSE PROBLEMS - ESTIMATING PARAMETERS

Rosemary Renaut
http://math.asu.edu/~rosie

WORKSHOP ON INVERSE PROBLEMS IN SCATTERING AND IMAGING
APRIL 13, 2013
Outline

Background
  Simple 1D Example Signal Restoration
  Parameter Estimation
  Feature Extraction - Gradients/Edges

Tutorial: The Least Squares Solution
  Standard Analysis by the SVD
  Picard Condition for Ill-Posed Problems
  Importance of the Basis and Noise

Generalized regularization
  GSVD for examining the solution
  Revealing the Noise in the GSVD Basis
  Stabilizing the GSVD Solution

Applying to TV and the SB Algorithm
  Parameter Estimation for the TV

Conclusions and Future
Ill-conditioned Least Squares: Tikhonov Regularization

Solve ill-conditioned

\[ A\mathbf{x} \approx \mathbf{b} \]

Standard Tikhonov, \( L \) approximates a derivative operator

\[
x(\lambda) = \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \| A\mathbf{x} - \mathbf{b} \|_2^2 + \frac{\lambda^2}{2} \| L\mathbf{x} \|_2^2 \right\}
\]

\( x(\lambda) \) solves normal equations provided \( \text{null}(L) \cap \text{null}(A) = \{0\} \)

\[
(A^T A + \lambda^2 L^T L) x(\lambda) = A^T \mathbf{b}
\]

This is not good for preserving edges in solutions. Not good for extracting features in images. But multiple approaches exist for estimating the parameter \( \lambda \).
An Example: Problem Phillips 10% Noise
Problem Phillips 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $L_x$ for optimal $\lambda$

**Figure:** Generalized Cross Validation
Problem Phillips 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $Lx$ for optimal $\lambda$

Figure: L-curve
Problem Phillips 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $Lx$ for *optimal* $\lambda$

**Figure:** Unbiased Predictive Risk
Problem Phillips 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $Lx$ for optimal $\lambda$

Figure: Optimal $\chi^2$ - no parameter estimation!
An Example: Problem Shaw

The Data Noise Level 0.1

- Solution x
- RHS
- Noisy RHS
- Lx
Problem Shaw 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $L_x$ for *optimal* $\lambda$

**Figure:** Generalized Cross Validation
Problem Shaw 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $L_x$ for \textit{optimal} $\lambda$

\begin{figure}
\centering
\includegraphics[width=\textwidth]{lcurve.png}
\caption{L-curve}
\end{figure}
Problem Shaw 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $L_x$ for optimal $\lambda$

Figure: Unbiased Predictive Risk
Problem Shaw \(10\%\) Noise: Tikhonov Regularized Solutions \(x(\lambda)\) and derivative \(Lx\) for \textit{optimal} \(\lambda\)

![Regularized Solution and \(Lx\) by Optimal GSVD](image)

**Figure:** Optimal \(\chi^2\) - no parameter estimation!
Simple Example: Blurred Signal Restoration

\[ b(t) = \int_{-\pi}^{\pi} h(t, s)x(s)\,ds \]

\[ h(s) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(\frac{-s^2}{2\sigma^2}\right), \quad \sigma > 0. \]
Example: Sample Blurred Signal 10% Noise
Sampled Signal 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $Lx$ for **optimal** $\lambda$

**Figure:** Generalized Cross Validation
Sampled Signal 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $Lx$ for optimal $\lambda$

**Figure:** L-curve
Sampled Signal 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $Lx$ for optimal $\lambda$

Figure: Unbiased Predictive Risk
Sampled Signal 10% Noise: Tikhonov Regularized Solutions $x(\lambda)$ and derivative $Lx$ for \textit{optimal} $\lambda$

Figure: Optimal $\chi^2$- no parameter estimation!
Observations

- Solutions clearly parameter dependent
- For smooth signals we may be able to recover reasonable solutions
- For non smooth signals we have a problem
- Parameter estimation is more difficult for non smooth case
- What is best? What adapts for other situations?
Alternative Regularization: Total Variation

Consider general regularization $R(x)$ suited to properties of $x$:

$$
x(\lambda) = \arg \min_x \left\{ \frac{1}{2} \|Ax - b\|_2^2 + \frac{\lambda^2}{2} R(x) \right\}
$$

Suppose $R$ is total variation of $x$ (general options are possible)
For example $p$-norm regularization $p < 2$

$$
x(\lambda) = \arg \min_x \left\{ \|Ax - b\|_2^2 + \lambda^2 \|Lx\|_p \right\}
$$

$p = 1$ approximates the total variation in the solution.

How to solve the optimization problem for large scale?
One approach - iteratively reweighted norms (Rodriguez and Wohlberg) - suitable for $0 < p < 2$ - depends on two parameters.
Introduce $d \approx Lx$ and let $R(x) = \frac{\lambda^2}{2} \|d - Lx\|^2_2 + \mu \|d\|_1$

$$(x, d)(\lambda, \mu) = \arg\min_{x, d} \left\{ \frac{1}{2} \|Ax - b\|^2_2 + \frac{\lambda^2}{2} \|d - Lx\|^2_2 + \mu \|d\|_1 \right\}$$

Alternating minimization separates steps for $d$ from $x$

Various versions of the iteration can be defined. Fundamentally:

$S1: x^{(k+1)} = \arg\min_x \left\{ \frac{1}{2} \|Ax - b\|^2_2 + \frac{\lambda^2}{2} \|Lx - (d^{(k+1)} - g^{(k)})\|^2_2 \right\}$

$S2: d^{(k+1)} = \arg\min_d \left\{ \frac{\lambda^2}{2} \|d - (Lx^{(k+1)} + g^{(k)})\|^2_2 + \mu \|d\|_1 \right\}$

$S3: g^{(k+1)} = g^{(k)} + Lx^{(k+1)} - d^{(k+1)}$.

Notice dimension increase of the problem
Advantages of the formulation

Update for $g$: updates the Lagrange multiplier $g$

$$S3: \quad g^{(k+1)} = g^{(k)} + Lx^{(k+1)} - d^{(k+1)}.$$  

This is just - a vector update

Update for $d$:

$$S2: \quad d = \arg\min_d \left\{ \mu \|d\|_1 + \frac{\lambda^2}{2} \|d - c\|_2^2 \right\}, \quad c = Lx + g$$

$$= \arg\min_d \left\{ \|d\|_1 + \frac{\gamma}{2} \|d - c\|_2^2 \right\}, \quad \gamma = \frac{\lambda^2}{\mu}.$$  

This is achieved using soft thresholding.
Focus: Tikhonov Step of the Algorithm

$$S1: \mathbf{x}^{(k+1)} = \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \| A\mathbf{x} - \mathbf{b} \|_2^2 + \frac{\lambda^2}{2} \| L\mathbf{x} - (\mathbf{d}^{(k)} - \mathbf{g}^{(k)}) \|_2^2 \right\}$$

Update for $\mathbf{x}$: Introduce

$$\mathbf{h}^{(k)} = \mathbf{d}^{(k)} - \mathbf{g}^{(k)}.$$ 

Then

$$\mathbf{x}^{(k+1)} = \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \| A\mathbf{x} - \mathbf{b} \|_2^2 + \frac{\lambda^2}{2} \| L\mathbf{x} - \mathbf{h}^{(k)} \|_2^2 \right\}.$$ 

Standard least squares update using a Tikhonov regularizer. 

Depends on changing right hand side 

Depends on parameter $\lambda$. 

Disadvantages of the formulation

update for $x$: A Tikhonov LS update each step
Changing right hand side.
Regularization parameter $\lambda$ - dependent on $k$?
Threshold parameter $\mu$ - dependent on $k$?

Advantages of the formulation

Extensive literature on Tikhonov LS problems
To determine stability analyze Tikhonov LS step of algorithm
- Understand the impact of the basis on the solution
- Picard condition
- Use Generalized Singular Value Decomposition for analysis
Tutorial: LS Solutions

Regularization of solution $x$
- SVD for the LS solution
- Basis for the LS solution
- The Picard condition

Modified Regularization
- GSVD for the regularized solution
- Changes the basis
- Generalized Picard condition
Consider general overdetermined discrete problem

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

Singular value decomposition (SVD) of \( A \) (full column rank)

\[ A = U \Sigma V^T = \sum_{i=1}^{n} u_i \sigma_i v_i^T, \quad \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n). \]

gives expansion for the solution

\[ \mathbf{x} = \sum_{i=1}^{n} \frac{u_i^T \mathbf{b}}{\sigma_i} v_i \]

\( u_i, v_i \) are left and right singular vectors for \( A \)
Solution is a \textit{weighted} linear combination of the basis vectors \( v_i \)
The Solutions with truncated SVD- problem shaw

**Figure:** Truncated SVD Solutions: data enters through coefficients $|u_i^T b|$. On the left no noise (inverse crime) in $b$ and on the right with tiny noise $10^{-4}$

How is this impacted by the basis?
Some basis vectors for shaw

Figure: The first few left singular vectors $v_i$ - are they accurate?
Are the basis vectors accurate?
Use Matlab High Precision to examine the SVD

- Matlab `digits` allows high precision. Standard is 32.
- Symbolic toolbox allows operations on high precision variables with `vpa`.
- SVD for `vpa` variables calculates the singular values symbolically, but not the singular vectors.
- Higher accuracy for the SVs generates higher accuracy singular vectors.
- Solutions with high precision can take advantage of Matlab’s symbolic toolbox.
Left Singular Vectors and Basis Calculated in High Precision

Figure: The first few left singular vectors $u_i$ and basis vectors $v_i$. Higher precision preserves the frequency content of the basis.

How many can we use in the solution for $x$?
How does inaccurate basis impact regularized solutions?
The Truncated Solutions (Noise free data $b$) - inverse crime

Figure: Truncated SVD Solutions: Standard precision $|u_i^T b|$. Error in the basis contaminates the solution
The Truncated Solutions (Noise free data $\mathbf{b}$) - inverse crime

**Figure:** Truncated SVD Solutions: VPA calculation $|\mathbf{u}_i^T \mathbf{b}|$. Larger number of accurate terms
Technique to detect true basis: the Power Spectrum for detecting white noise: a time series analysis technique

Suppose for a given vector $y$ that it is a time series indexed by position, i.e. index $i$.

**Diagnostic 1** Does the histogram of entries of $y$ generate histogram consistent with $y \sim \mathbb{N}(0, 1)$? (i.e. independent normally distributed with mean 0 and variance 1) Not practical to automatically look at a histogram and make an assessment

**Diagnostic 2** Test the expectation that $y_i$ are selected from a white noise time series. Take the Fourier transform of $y$ and form **cumulative periodogram** $z$ from power spectrum $c$

$$c_j = |(\text{dft}(y)_j|^2, \quad z_j = \frac{\sum_{i=1}^j c_j}{\sum_{i=1}^q c_i}, \quad j = 1, \ldots, q,$$

**Automatic**: Test is the line $(z_j, j/q)$ close to a straight line with slope 1 and length $\sqrt{5}/2$?
Cumulative Periodogram for the left singular vectors

**Figure:** Standard precision on the left and high precision on the right: On left more vectors are close to white than on the right - vectors are white if the CP is close to the diagonal on the plot: Low frequency vectors lie above the diagonal and high frequency below the diagonal. White noise follows the diagonal.
Figure: Standard precision on the left and high precision on the right: On left more vectors are close to white than on the right - if you count there are 9 vectors with true frequency content on the left.
Measure Deviation from Straight Line: Basis Vectors

Figure: Testing for white noise for the standard precision vectors: Calculate the cumulative periodogram and measure the deviation from the “white noise” line or assess proportion of the vector outside the Kolmogorov Smirnov test at a 5% confidence level for white noise lines. In this case it suggests that about 9 vectors are noise free.

Cannot expect to use more than 9 vectors in the expansion for $x$. Additional terms are contaminated by noise - independent of noise in $b$. 
Standard Analysis Discrete Picard condition: assesses impact of noise in the data

Recall

\[ x = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i \]

Here

\[ |u_i^T b|/\sigma_i = \mathcal{O}(1) \]

Ratios are not large but are the values correct? Considering only the discrete Picard condition does not tell us whether the expansion for the solution is correct.
Observations

- Even when committing the inverse crime we will not achieve the solution if we cannot approximate the basis correctly.
- We need all basis vectors which contain the high frequency terms in order to approximate a solution with high frequency components - e.g. edges.
- Reminder - this is independent of the data.
- But is an indication of an ill-posed problem. In this case the data that is modified exhibits in the matrix $A$ decomposition.
- To do any uncertainty estimation one must understand noise throughout model and data.
- Why is this relevant to TV regularization?
Tikhonov Regularization is Spectral Filtering

\[ x_{\text{Tik}} = \sum_{i=1}^{n} \phi_i \left( \frac{u_i^T b}{\sigma_i} \right) v_i \]

- **Tikhonov Regularization** \( \phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \), \( i = 1 \ldots n \), \( \lambda \) is the regularization parameter, and solution is

\[ x_{\text{Tik}}(\lambda) = \arg \min_{x} \{ \| b - Ax \|^2 + \lambda^2 \| x \|^2 \} \]

- More general formulation

\[ x_{\text{Tik}}(\lambda) = \arg \min_{x} \{ \| b - Ax \|^2_W + \lambda^2 \| Lx \|^2 \} = \arg \min_{x} (J) \]

- \( \chi^2 \) distribution: find \( \lambda \) such that

\[ E(J(x(\lambda))) = m + p - n \]

- Yields an *unregularized estimate* for \( x \)

- It is exactly Tikhonov regularization that is used in the SB algorithm. But for a change of basis determined by \( L \).
Introduce generalization of the SVD to obtain expansion for

\[ x(\lambda) = \arg \min_x \{ \| Ax - b \|^2 + \lambda^2 \| L(x - x_0) \|^2 \} \]

**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 \( Z \in \mathbb{R}^{n \times n} \) such that

\[
A = U \begin{bmatrix} \Upsilon & \ 0_{(m-n) \times n} \end{bmatrix} Z^T, \quad L = V [ M, 0_{p \times (n-p)} ] Z^T,
\]

\( \Upsilon = \text{diag}(\upsilon_1, \ldots, \upsilon_p, 1, \ldots, 1) \in \mathbb{R}^{n \times n}, \ M = \text{diag}(\mu_1, \ldots, \mu_p) \in \mathbb{R}^{p \times p}, \)

with

\[ 0 \leq \upsilon_1 \leq \cdots \leq \upsilon_p \leq 1, \ 1 \geq \mu_1 \geq \cdots \geq \mu_p > 0, \ \upsilon_i^2 + \mu_i^2 = 1, \ i = 1, \ldots, p. \]

Use \( \tilde{\Upsilon} \) and \( \tilde{M} \) to denote the rectangular matrices containing \( \Upsilon \) and \( M \), and note **generalized singular values**: \( \rho_i = \frac{\upsilon_i}{\mu_i} \).
Solution of the Generalized Problem using the GSVD \( h = 0 \)

\[
x(\lambda) = \sum_{i=1}^{p} \frac{\nu_i}{\nu_i^2 + \lambda^2 \mu_i^2} (u_i^T b) \tilde{z}_i + \sum_{i=p+1}^{n} (u_i^T b) \tilde{z}_i
\]

\( \tilde{z}_i \) is the \( i^{th} \) column of \( (Z^T)^{-1} \).

Equivalently with filter factor \( \phi_i \)

\[
x(\lambda) = \sum_{i=1}^{p} \phi_i \frac{u_i^T b}{\nu_i} \tilde{z}_i + \sum_{i=p+1}^{n} (u_i^T b) \tilde{z}_i,
\]

\[
Lx(\lambda) = \sum_{i=1}^{p} \phi_i \frac{u_i^T b}{\rho_i v_i}, \quad \phi_i = \frac{\rho_i^2}{\rho_i^2 + \lambda^2},
\]
Does noise enter the GSVD basis?

Form the truncated matrix $A_k$ using only $k$ basis vectors $v_i$

Figure: Contrast GSVD basis $u$ (left) $z$ (right). Trade off $U$ and $Z$
The TIK update of the TV algorithm

Update for $x$: using the GSVD : basis $Z$

$$x^{(k+1)} = \sum_{i=1}^{p} \left( \frac{\nu_i u_i^T b}{\nu_i^2 + \lambda^2 \mu_i^2} + \frac{\lambda^2 \mu_i v_i^T h^{(k)}}{\nu_i^2 + \lambda^2 \mu_i^2} \right) z_i + \sum_{i=p+1}^{n} (u_i^T b) z_i$$

A weighted combination of basis vectors $z_i$: weights

(i) \( \frac{\nu_i^2}{\nu_i^2 + \lambda^2 \mu_i^2} \frac{u_i^T b}{\nu_i} \)

(ii) \( \frac{\lambda^2 \mu_i^2}{\nu_i^2 + \lambda^2 \mu_i^2} \frac{v_i^T h^{(k)}}{\mu_i} \)

Notice (i) is fixed by $b$, but (ii) depends on the updates $h^{(k)}$

i.e. (i) is iteration independent

If (i) dominates (ii) solution will converge slowly or not at all

$\lambda$ impacts the solution and must not over damp (ii)
Update of the mapped data

\[
\mathbf{x}^{(k+1)} = \sum_{i=1}^{p} \left( \phi_i \frac{\mathbf{u}_i^T \mathbf{b}}{\nu_i} + (1 - \phi_i) \frac{\mathbf{v}_i^T \mathbf{h}^{(k)}}{\mu_i} \right) \mathbf{z}_i + \sum_{i=p+1}^{n} (\mathbf{u}_i^T \mathbf{b}) \mathbf{z}_i
\]

\[
L \mathbf{x}^{(k+1)} = \sum_{i=1}^{p} \left( \phi_i \frac{\mathbf{u}_i^T \mathbf{b}}{\rho_i} + (1 - \phi_i) \mathbf{v}_i^T \mathbf{h}^{(k)} \right) \mathbf{v}_i.
\]

Now the stability depends on coefficients

\[
\frac{\phi_i}{\rho_i} \quad 1 - \phi_i \quad \frac{\phi_i}{\nu_i} \quad \frac{1 - \phi_i}{\mu_i}
\]

with respect to the inner products \(\mathbf{u}_i^T \mathbf{b}, \mathbf{v}_i^T \mathbf{h}^{(k)}\)
Examining the weights with $\lambda$ increasing

Coefficients $|u_i^Tb|$ and $|v_i^T h|$, $\lambda = .001, .1$ and $10$, $k = 1, 10$ and Final
It is important to analyze the components of the solution. Leads to a generalized Picard condition analysis - more than I can give here. The basis is important. Regularization parameters should be adjusted dynamically. Theoretical results apply concerning parameter estimation. We can use $\chi^2$, UPRE etc. Important to have information on the noise levels.
Theoretical results: using Unbiased Predictive Risk for the SB Tik

Lemma
Suppose noise in $h^{(k)}$ is stochastic and both data fit $Ax \approx b$ and derivative data fit $Lx \approx h$ are weighted by their inverse covariance matrices for normally distributed noise in $b$ and $h$; then optimal choice for $\lambda$ at all steps is $\lambda = 1$.

Remark
*Can we expect $h^{(k)}$ is stochastic?*

Lemma
Suppose $h^{(k+1)}$ is regarded as deterministic, then UPRE applied to find the the optimal choice for $\lambda$ at each step leads to a different optimum at each step, namely it depends on $h$.

Remark
*Because $h$ changes each step the optimal choice for $\lambda$ using UPRE will change with each iteration, $\lambda$ varies over all steps.*
Example Solution: 2D - various examples (SNR in title)

Noise level 0.0015536

Data 4.804

LS 19.664

SB UPRE $\lambda$ 20.2618

SB UPRE $\lambda=1$ 20.3218
Example Solution: 2D - various examples (SNR in title)

Noise level 0.0062143

Data 4.8021

LS 13.806

SB UPRE ≥ 15.9083

SB UPRE ≥1 15.4281
Example Solution: 2D - various examples (SNR in title)

Noise level 0.01

Data 1

LS 9.4918

SB UPRE $\lambda$ 16.7354

SB UPRE $\lambda$=1 15.2695
Example Solution: 2D - various examples (SNR in title)

Noise level 0.64973

Data 1

LS 8.082

SB UPRE 8.7654

SB UPRE λ=1 9.5021
Further Observations and Future Work

**Results**  demonstrate basic analysis of problem is worthwhile

**Extend** standard parameter estimation from LS

**Overhead** of optimal $\lambda$ for the first step - reasonable

**Stochastic** interpretation - use fixed $\lambda = 1$ after the first iteration is found.

**Practically** use LSQR algorithms rather than GSVD - changes the basis to the Krylov basis. Or use randomized SVD to generate randomized GSVD

**Noise** in the basis is relevant for LSQR / CGLS iterative algorithms

**Convergence** testing is based on $h$.  

Initial work - much to do