Lake Como School for Advanced Studies

Computational Methods for Inverse Problems and Applications in Image Processing

SVD Filters

James Nagy

Emory University
Atlanta, GA USA

Review: TSVD Regularization

Suppose $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, and $\mathbf{b} = \mathbf{A}\mathbf{x}_{\rm true} + \boldsymbol{\eta} = \mathbf{b}_{\rm true} + \boldsymbol{\eta}$.

Naive inverse solution, $\mathbf{x}_{\rm inv} = \mathbf{A}^{-1}\mathbf{b} = \mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^T\mathbf{b}$

$$\mathbf{x}_{\rm inv} = \sum_{i=1}^{n} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i = \underbrace{\sum_{i=1}^{n} \frac{\mathbf{u}_i^T\mathbf{b}_{\rm true}}{\sigma_i}\mathbf{v}_i}_{\mathbf{x}_{\rm true}} + \underbrace{\sum_{i=1}^{n} \frac{\mathbf{u}_i^T\boldsymbol{\eta}}{\sigma_i}\mathbf{v}_i}_{\text{error}}$$

The goal is to balance:

- reconstructing "good" SVD components: $\dfrac{\mathbf{u}_i^T\mathbf{b}_{\rm true}}{\sigma_i}$ (large σ_i)
- avoid reconstructing "bad" SVD components $\dfrac{\mathbf{u}_i^T\boldsymbol{\eta}}{\sigma_i}$ (small σ_i)

One approach: TSVD:

$$\mathbf{x}_{\rm tsvd} = \sum_{i=1}^{\color{red}{k}} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i$$

SVD Filter-based Regularization

The TSVD idea can be generalized:

$$\mathbf{x}_{\text{filt}} \sum_{i=1}^{n} \phi_i \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i, \quad \text{where} \quad \phi_i \approx \begin{cases} 1 & \text{for "large" } \sigma_i \\ 0 & \text{for "small" } \sigma_i \end{cases}$$

Examples:

- TSVD: $\phi_i = \begin{cases} 1 & i = 1, 2, \ldots, k \\ 0 & i = k+1, \ldots, n \end{cases}$
  We must choose regularization parameter k.
- Tikhonov: $\phi_i = \dfrac{\sigma_i^2}{\sigma_i^2 + \alpha^2}$
  We must choose regularization parameter α
- Exponential: $\phi_i = 1 - e^{-\sigma_i^2/\alpha^2}$
  We must choose regularization parameter α

Tikhonov Regularization in Variational Form

Tikhonov regularization is often formulated as:

$$\min_{\mathbf{x}} \left\{ \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \alpha^2 \|\mathbf{x}\|_2^2 \right\} \quad \Leftrightarrow \quad \min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A} \\ \alpha \mathbf{I} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \qquad (1)$$

Replacing $\mathbf{A}$ with its SVD, it is easy to show that the solution of the second minimization problem in (1) is

$$\mathbf{x}_{\text{tik}} = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$$

We will return to variational forms of regularization, including generalizations, later.

SVD Filtering in Matrix Form

We can write the filtered solution as:

$$\mathbf{x}_{\text{filt}} = \mathbf{A}_{\text{filt}}^{\dagger}\mathbf{b} = \mathbf{V}\boldsymbol{\Sigma}_{\text{filt}}^{\dagger}\mathbf{U}^T\mathbf{b}$$

where

- $\boldsymbol{\Sigma}_{\text{tsvd}}^{\dagger} = \text{diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_k}, 0, \ldots, 0\right)$
- $\boldsymbol{\Sigma}_{\text{tik}}^{\dagger} = \text{diag}\left(\frac{\sigma_i}{\sigma_i^2 + \alpha^2}\right)$
- $\boldsymbol{\Sigma}_{\text{exp}}^{\dagger} = \text{diag}\left(\frac{1 - e^{-\sigma_i^2/\alpha^2}}{\sigma_i}\right)$

Note: With this notation, we can replace singular value decomposition with unitary spectral decompositions.

FFT-based Filtering

In some cases, $\mathbf{A} = \mathcal{F}^* \boldsymbol{\Lambda} \mathcal{F}$, where $\mathcal{F}$ is unitary Fourier transform. In this case,

$$\mathbf{x}_{\text{filt}} = \mathbf{A}_{\text{filt}}^{\dagger} \mathbf{b} = \mathcal{F}^* \boldsymbol{\Lambda}_{\text{filt}}^{\dagger} \mathcal{F} \mathbf{b} \quad = \text{ifft2}(\boldsymbol{\Lambda}^{\dagger} \text{fft2}(\mathbf{b}))$$

In other cases, can use cosine transform: $\mathbf{A} = \mathbf{C}^T \boldsymbol{\Lambda} \mathbf{C}$, and

$$\mathbf{x}_{\text{filt}} = \mathbf{A}_{\text{filt}}^{\dagger} \mathbf{b} = \mathbf{C}^T \boldsymbol{\Lambda}_{\text{filt}}^{\dagger} \mathbf{C} \mathbf{b} \quad = \text{idct2}(\boldsymbol{\Lambda}^{\dagger} \text{dct2}(\mathbf{b}))$$

where, for example

- $\boldsymbol{\Lambda}_{\text{tsvd}}^{\dagger} = \text{diag}\left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_k}, 0, \ldots, 0 \right)$
- $\boldsymbol{\Lambda}_{\text{tik}}^{\dagger} = \text{diag}\left(\frac{\bar{\lambda}_i}{|\lambda_i|^2 + \alpha^2} \right)$

Finding "Optimal" Regularization Parameters

Here we assume $\mathbf{x}_{\rm true}$ is known, and we want to find regularization parameters to minimize

$$\|\mathbf{x}_{\rm filt} - \mathbf{x}_{\rm true}\|_2^2$$

where $\quad \mathbf{x}_{\rm filt} = \mathbf{A}_{\rm filt}^{\dagger}\mathbf{b} = \mathbf{V}\boldsymbol{\Sigma}_{\rm filt}^{\dagger}\mathbf{U}^T\mathbf{b}$

$$\begin{aligned}
\|\mathbf{x}_{\rm filt} - \mathbf{x}_{\rm true}\|_2^2 &= \|\mathbf{V}^T\mathbf{x}_{\rm filt} - \mathbf{V}^T\mathbf{x}_{\rm true}\|_2^2 \\
&= \|\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}_{\rm filt}^{\dagger}\mathbf{U}^T\mathbf{b} - \mathbf{V}^T\mathbf{x}_{\rm true}\|_2^2 \\
&= \|\boldsymbol{\Sigma}_{\rm filt}^{\dagger}\hat{\mathbf{b}} - \hat{\mathbf{x}}_{\rm true}\|_2^2
\end{aligned}$$

where $\hat{\mathbf{b}} = \mathbf{U}^T\mathbf{b}$ and $\hat{\mathbf{x}}_{\rm true} = \mathbf{V}^T\mathbf{x}_{\rm true}$

"Optimal" TSVD Regularization Parameter

- Observe that

$$\begin{aligned} e_k = \|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2^2 &= \|\boldsymbol{\Sigma}^{\dagger}_{\text{filt}}\hat{\mathbf{b}} - \hat{\mathbf{x}}_{\text{true}}\|_2^2 \\ &= \sum_{i=1}^{k}\left(\frac{\hat{b}_i}{\sigma_i} - \hat{x}_i\right)^2 + \sum_{i=k+1}^{n}\hat{x}_i^2 \\ &= \sum_{i=1}^{k} y_i^2 + \sum_{i=k+1}^{n}\hat{x}_i^2 \end{aligned}$$

- Then observe that

$$e_k = e_{k-1} - \hat{x}_k^2 + y_k^2$$

- Recursively compute e_k, and use MATLAB's `min` function to find minimum value, and corresponding index.

"Optimal" Tikhonov Regularization Parameter

- Observe that

$$
\begin{aligned}
e_k = \|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2^2 &= \|\boldsymbol{\Sigma}_{\text{filt}}^{\dagger}\hat{\mathbf{b}} - \hat{\mathbf{x}}_{\text{true}}\|_2^2 \\
&= \sum_{i=1}^{k} \left(\frac{\sigma_i \hat{b}_i}{\sigma_i^2 + \alpha^2} - \hat{x}_i \right)^2
\end{aligned}
$$

- Find α to minimize the function

$$
E(\alpha) = \sum_{i=1}^{k} \left(\frac{\sigma_i \hat{b}_i}{\sigma_i^2 + \alpha^2} - \hat{x}_i \right)^2
$$

- Use, for example, MATLAB's `fminbnd` function to find minimum of $E(\alpha)$. Note that we can assume $\sigma_n \leq \alpha \leq \sigma_1$.

Guides to Choosing Regularization Parameters

In each filtering example, we must choose a regularization parameter.

Do we need, or can we use additional information to help?

- Often we need to make assumptions about the noise, e.g., Gaussian white noise.
- It can be helpful to know prior information, such as $\|\boldsymbol{\eta}\|_2$.
- Some approaches attempt to extract noise information from the data.

Methods we will discuss:

1. Discrete Picard Condition
2. Discrepancy Principle
3. Generalized Cross Validation
4. L-Curve

Using DPC to Choose TSVD Cutoff

If we assume $\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\eta}$,

- Let k denote the TSVD truncation index. That is,

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

- Let k_{DPC} = the index where the coefficients $|\mathbf{u}_i^T \mathbf{b}|$ level off.
- Then,

$$\mathbf{x}_{\text{DPC}} = \sum_{i=1}^{k_{\text{DPC}}} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$$

Discrepancy Principle

We assume $\mathbf{b} = \mathbf{A}\mathbf{x}_{\rm true} + \boldsymbol{\eta}$.

Find a filtered solution, $\mathbf{x}_{\rm filt}$, so that

$$\|\mathbf{A}\mathbf{x}_{\rm filt} - \mathbf{b}\|_2 \approx \|\boldsymbol{\eta}\|_2$$

Remark: It is often recommended to use

$$\|\mathbf{A}\mathbf{x}_{\rm filt} - \mathbf{b}\|_2 = \delta\|\boldsymbol{\eta}\|_2$$

where $\delta > 1$ (e.g., $\delta = 1.01$).

Discrepancy Principle: Implementations

To implement, first recall that if $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, then we can write

$$\mathbf{x}_{\text{filt}} = \mathbf{A}_{\text{filt}}^{\dagger}\mathbf{b}, \qquad \mathbf{A}_{\text{filt}}^{\dagger} = \mathbf{V}\boldsymbol{\Sigma}_{\text{filt}}^{\dagger}\mathbf{U}^T\mathbf{b}$$

Therefore,

$$\|\mathbf{A}\mathbf{x}_{\text{filt}} - \mathbf{b}\|_2^2 = \|\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}_{\text{filt}}^{\dagger}\mathbf{U}^T\mathbf{b} - \mathbf{b}\|_2^2 = \|(\boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\text{filt}}^{\dagger} - \mathbf{I})\mathbf{U}^T\mathbf{b}\|_2^2$$

For each filtering method, we need to find regularization parameter so that

$$\|(\boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\text{filt}}^{\dagger} - \mathbf{I})\mathbf{U}^T\mathbf{b}\|_2^2 \approx \|\boldsymbol{\eta}\|_2^2$$

Discrepancy Principle: Implementations

- TSVD

$$\|(\boldsymbol{\Sigma}\boldsymbol{\Sigma}_k^\dagger - \mathbf{I})\mathbf{U}^T\mathbf{b}\|_2 = \sum_{i=k+1}^{n} (\mathbf{u}_i^T\mathbf{b})^2 \approx \|\boldsymbol{\eta}\|_2^2$$

Find smallest k so that $\displaystyle\sum_{i=k+1}^{n} (\mathbf{u}_i^T\mathbf{b})^2 \leq \|\boldsymbol{\eta}\|_2^2$

- Tikhonov

$$\|(\boldsymbol{\Sigma}\boldsymbol{\Sigma}_k^\dagger - \mathbf{I})\mathbf{U}^T\mathbf{b}\|_2 = \sum_{i=1}^{n} \left(\frac{\alpha^2\mathbf{u}_i^T\mathbf{b}}{\sigma_i^2 + \alpha^2}\right)^2$$

Let $\varepsilon = \|\boldsymbol{\eta}\|^2$, and define $D(\alpha) = \displaystyle\sum_{i=1}^{n} \left(\frac{\alpha^2\mathbf{u}_i^T\mathbf{b}}{\sigma_i^2 + \alpha^2}\right)^2 - \varepsilon^2$

Find α so that $D(\alpha) = 0$ (use, e.g., MATAB's `fzero` function).

Generalized Cross Validation

- Statistically based approach that tries to find the the regularization parameter to minimize

$$G(\cdot) = \frac{n\|\mathbf{b} - \mathbf{A}\mathbf{x}_{\text{filt}}\|_2^2}{\left(\text{trace}(\mathbf{I} - \mathbf{A}\mathbf{A}_{\text{filt}}^{\dagger})\right)^2}$$

- To implement, we need:
  - $$\begin{aligned}\|\mathbf{b} - \mathbf{A}\mathbf{x}_{\text{filt}}\|_2^2 &= \|\mathbf{b} - \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}_{\text{filt}}^{\dagger}\mathbf{U}^T\mathbf{b}\|_2^2 \\ &= \|(\mathbf{I} - \boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\text{filt}}^{\dagger})\mathbf{U}^T\mathbf{b}\|_2^2\end{aligned}$$
  - $$\begin{aligned}\text{trace}(\mathbf{I} - \mathbf{A}\mathbf{A}_{\text{filt}}^{\dagger}) &= \text{trace}(\mathbf{I} - \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}_{\text{filt}}^{\dagger}\mathbf{U}^T) \\ &= \text{trace}(\mathbf{U}(\mathbf{I} - \boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\text{filt}}^{\dagger})\mathbf{U}^T) \\ &= \text{trace}(\mathbf{I} - \boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\text{filt}}^{\dagger})\end{aligned}$$

GCV for TSVD

Suppose $\mathbf{A}$ is $n \times n$. Then

- $\|\mathbf{b} - \mathbf{A}\mathbf{x}_{\mathrm{filt}}\|_2^2 = \|(\mathbf{I} - \mathbf{\Sigma}\mathbf{\Sigma}_{\mathrm{filt}}^{\dagger})\mathbf{U}^T\mathbf{b}\|_2^2 = \displaystyle\sum_{i=k+1}^{n} (\mathbf{u}_i^T\mathbf{b})^2$
- $\mathrm{trace}(\mathbf{I} - \mathbf{A}\mathbf{A}_{\mathrm{filt}}^{\dagger}) = \mathrm{trace}(\mathbf{I} - \mathbf{\Sigma}\mathbf{\Sigma}_{\mathrm{filt}}^{\dagger}) = n - k$

Therefore, we need to find k to minimize

$$G(k) = \frac{n \displaystyle\sum_{i=k+1}^{n} (\mathbf{u}_i^T\mathbf{b})^2}{(n-k)^2}$$

Since this is a discrete function, compute all $G(k)$ values, and use MATLAB `min` function to find minimum.

Question: How does this change if $\mathbf{A}$ is $m \times n$, $m > n$?

GCV for Tikhonov

Suppose $\mathbf{A}$ is $n \times n$. Then

- $\|\mathbf{b} - \mathbf{A}\mathbf{x}_{\mathrm{filt}}\|_2^2 = \|(\mathbf{I} - \boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\mathrm{filt}}^{\dagger})\mathbf{U}^T\mathbf{b}\|_2^2 = \sum_{i=1}^{n} \left(\frac{\alpha^2 \mathbf{u}_i^T \mathbf{b}}{\sigma_i^2 + \alpha^2} \right)^2$

- $\mathrm{trace}(\mathbf{I} - \mathbf{A}\mathbf{A}_{\mathrm{filt}}^{\dagger}) = \mathrm{trace}(\mathbf{I} - \boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\mathrm{filt}}^{\dagger}) = \sum_{i=1}^{n} \frac{\alpha^2}{\sigma_i^2 + \alpha^2}$

Therefore, we need to find α to minimize

$$G(\alpha) = \frac{n \sum_{i=1}^{n} \left(\frac{\alpha^2 \mathbf{u}_i^T \mathbf{b}}{\sigma_i^2 + \alpha^2} \right)^2}{\left(\sum_{i=1}^{n} \frac{\alpha^2}{\sigma_i^2 + \alpha^2} \right)^2} = \frac{n \sum_{i=1}^{n} \left(\frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i^2 + \alpha^2} \right)^2}{\left(\sum_{i=1}^{n} \frac{1}{\sigma_i^2 + \alpha^2} \right)^2}$$

Here, $G(\alpha)$ is a continuous function; use, e.g., MATLAB's `fminbnd` function, with $\sigma_n \leq \alpha \leq \sigma_1$.

Question: How does this change if $\mathbf{A}$ is $m \times n$, $m > n$?

MATLAB Demos

Can use tomography and deblurring test problems from IR Tools.

- One approach to testing SVD filtering:
  - Create small test problem, and use `full` to construct full matrix explicitly.
  - Then use MATLAB's `svd`.

  For example,

  ```
  [A, b_true, x_true] = PRtomo(32);
  A = full(A);
  [U, S, V] = svd(A);
  ```

  Another example,

  ```
  [A, b_true, x_true] = PRblurspeckle(64);
  A = full(A);
  [U, S, V] = svd(A);
  ```

- Larger problems are more challenging; for invariant deblurring, use wrappers for codes from Hansen, N., O'Leary (HNO) book:

  ```
  IRtik_dct.m, IRtik_fft, IRtik_sep
  IRtsvd_dct, IRtsvd_fft, IRtsvd_sep
  ```

Code for Demo_SVD1

```
n = 32;
PRoptions = PRset('CTtype', 'fancurved');
[A, b_true, x_true, ProbInfo] = PRtomo(n, PRoptions);
s = svd(full(A));
figure(1), clf, semilogy(s, 'o', 'LineWidth', 2)

PRoptions90 = PRset(PRoptions, 'angles', 0:2:90);
[A90, b90, x90, ProbInfo90] = PRtomo(n, PRoptions90);
s90 = svd(full(A90));
figure(2), clf, semilogy(s90, 'o', 'LineWidth', 2)

PRoptions45 = PRset(PRoptions, 'angles', 0:2:45);
[A45, b45, x45, ProbInfo45] = PRtomo(n, PRoptions45);
s45 = svd(full(A45));
figure(3), clf, semilogy(s45, 'o', 'LineWidth', 2)
```

Code for Demo_SVD2

```
PRoptions1 = PRset('BlurLevel', 'mild');
[A1, b1, x1, ProbInfo1] = PRblur(n, PRoptions1);
s1 = svd(full(A1));
figure(1), clf, semilogy(s1, 'o', 'LineWidth', 2)

PRoptions2 = PRset('BlurLevel', 'medium');
[A2, b2, x2, ProbInfo2] = PRblur(n, PRoptions2);
s2 = svd(full(A2));
figure(2), clf, semilogy(s2, 'o', 'LineWidth', 2)

PRoptions3 = PRset('BlurLevel', 'severe');
[A3, b3, x3, ProbInfo3] = PRblur(n, PRoptions3);
s3 = svd(full(A3));
figure(3), clf, semilogy(s3, 'o', 'LineWidth', 2)
```

Code for Demo_SVD3

Test spectral filtering codes from Hansen, Nagy, O'Leary (HNO) book, which by default use GCV to choose regularization parameters:

```
[A, b_true, x_true, ProbInfo] = PRblurgauss;
b = PRnoise(b_true);
figure(1), clf, PRshowb(b, ProbInfo)

[x_dct, alpha_dct] = IRtik_dct(A, b);
figure(2), clf, PRshowx(x_dct, ProbInfo)

[x_fft, alpha_fft] = IRtik_fft(A, b);
figure(3), clf, PRshowx(x_fft, ProbInfo)

[x_sep, alpha_sep] = IRtik_sep(A, b);
figure(3), clf, PRshowx(x_sep, ProbInfo)
```

Now repeat the above experiments, but replace `PRblurgauss` with `PRblurspeckle` and `PRblurshake`.