Advanced Course of Inverse Problem

NGUYEN T. Hoang - SID: 15M54097
(ホアン)

Fall 2015, W832 Fri. Period 1-2
Due date: 2016/01/20

Problem

- Given a image of a barcode:

Figure 1: A blurred image of a barcode. File name: `barcode_b.pgm`

- A matlab program `decom.m` is available to visualize the intensity distribution of the image.
- Explain the observation system $y = Ax$ for this problem.
- Build the system matrix A.
- Reconstruct with LSM.
- Reconstruct with Tikhonov Regulation method.
- Reconstruct with TSVD.
- Find the best regularization parameter with L-Curve method.

This assignment and the source code is available at:
`https://github.com/gear/Assignments/ip_final.pdf`
`https://github.com/gear/InverseProblem`

Q1: Explain the system and build system matrix

Explain the observation system $y = Ax$ and build the system matrix A.

Answer: The problem in this assignment is to recover barcode data from a given blurred barcode image. Barcode reader can be a phone camera or a specialized device with photosensor and laser light. In this probelm, we only care about data on a perpendicular line with the barcode. Furthermore, the observation ($g(s)$ and true value ($f(t)$) depends only on the displacement $s - t$. Therefore, this problem is a 1-D deconvolution problem:

$$Ax = b$$

where A is a system matrix describing the convolution, x is a 1-D vector storing the *true* value of the barcode, and b is a 1-D vector storing sensor/camera observation.

The system formula can be re-written as follow:

$$\sum_{j=1}^{n-1} h_{i-j} x_j = b_i$$

where h_{i-j} is a function describing relationship with x_j and b_i. Our model for the blur is the Gaussian kernel:

$$h_{i-j} = \frac{1}{n} \exp\left(-\frac{(i-j)^2}{\sigma^2 n^2}\right)$$

Choose $\sigma = 0.01$, we can construct the convolutional system matrix A as follow:

Listing 1: MATLAB code for system matrix A.

```
% Get length of the observation vector from given c in decom.m
n = length(c);
sigma = 0.01;
A = (1/n) .* toeplitz(exp(-((0:n-1)/(sigma*n)).^2));
% A is a 380x380 parse matrix.
```

Q2: Reconstruct with LSM.

Reconstruct the barcode with Least Square Method.

Answer: The Least Square Solution is given by this following formula:

$$x_{ls} = (A^T A)^{-1} A^T y$$

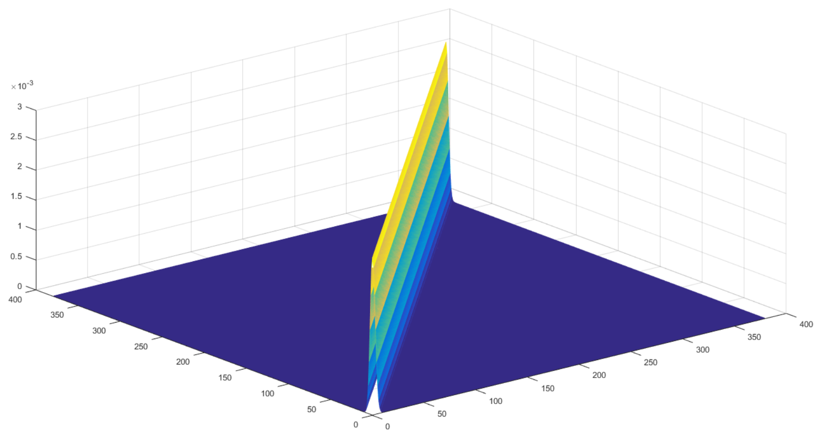

Figure 2: Mesh plot of system matrix A

This is the unique $x \in \mathcal{R}^n$ that minimizes $||Ax - y||$.
In MATLAB, we can also compute x_{ls} with the *backslash operator*. However, I will use aforementioned approximation formula.

Listing 2: MATLAB code for Least Square Method.

```
% Approximate Least Square Solution
x_ls = pinv(A' * A) * (A' * c);
figure; plot(x_ls); title('LSM Reconstruction');
```

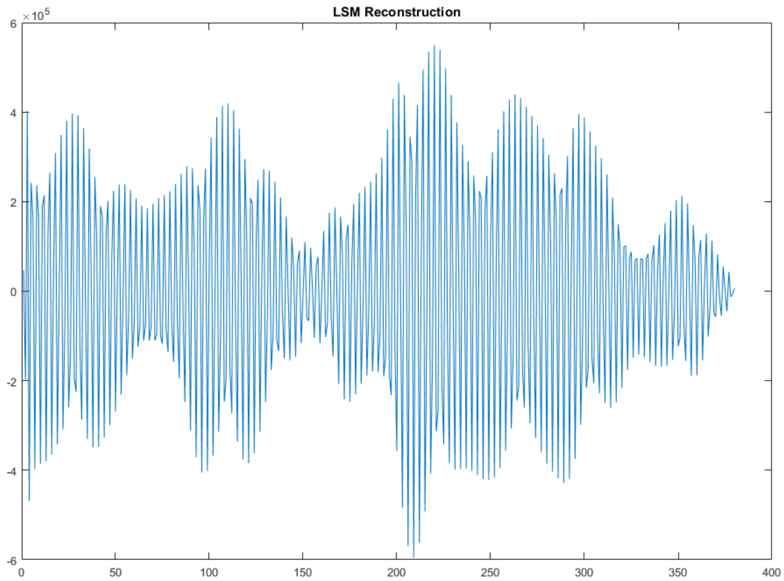

Figure 3: LSM Reconstruction result.

Q3: Reconstruct with Tikhonov Regulation Method.

Reconstruct the barcode with Tikhonov Regulation Method.

Answer: The Tikhonov Regulation is given by this following formula:

$$x_{tk} = \underset{x}{\operatorname{argmin}}(||Ax - b||_2^2 + \lambda^2 ||x||_2^2)$$

or x_λ, a solution corresponds with a real number λ, is given by:

$$x_\lambda = (A^T A + \lambda^2 I)^{-1} A^T b$$

where A is the system matrix and b is our observation. In MATLAB, I choose $\lambda = 3.5$, we can compute the value x_λ as follow:

Listing 3: MATLAB code for Tikhonov Regulation with $\lambda = 3.5$.

```
% Tikhonov Regulation solution
lambda = 3.5;
x_tk = pinv(A' * A + lambda^2 * eye(n)) * (A' * c);
figure; plot(x_tk); title('Tikhonov Regulation Reconstruction');
```

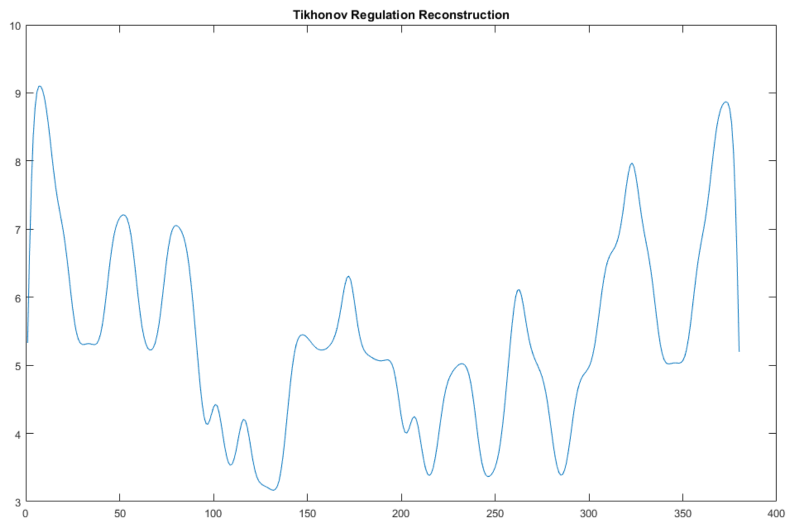

Figure 4: Tikhonov Regulation Reconstruction result.

Q4: Reconstruct with TSVD.

Reconstruct the barcode with TSVD.

Answer: The system matrix can be decomposed using Single Value Decomposition:

$$A = UDV^T,$$

where:

$$D = (diag)(d_1, ..., d_n).$$

The $\hat{x}^{(k)}$ estimation with a parameter k is given by:

$$\hat{x}^{(k)} = \Sigma_{j=1}^{k} \frac{1}{d_j}(u_j^T b)v_j$$

where d_j is elements of the diagnal, u_j is row j of matrix U, v_j is row j of matrix V, and b is our observation. In MATLAB, I choose $k = 40$, we can estimate the solution as follow:

Listing 4: MATLAB code for TSVD with $k = 40$.

```
% TSVD
[U,D,V] = svd(A);
d = diag(D);
r = find(d > eps, 1, 'last');
x_tsvd = zeros(n,r);
normX = zeros(r,1);
disp = zeros(r,1);
for k = 1:r
    x_tsvd(:,k) = V(:,1:k)*diag(1./d(1:k))*U(:,1:k)'*c;
    normX(k) = norm(x_tsvd(:,k));
    disp(k) = norm(c-A*x_tsvd(:,k));
end
figure; plot(x_tsvd(:,40)); title('TSVD Reconstruction, k = ↩
   40');
```

The TSVD by far is the best construction that I have obtained. By choosing $k = 40$, the result shows intensity for each bar clearly. In my observation, we can use this result and the JAN-8 barcode standard to derive the item this barcode refers to.

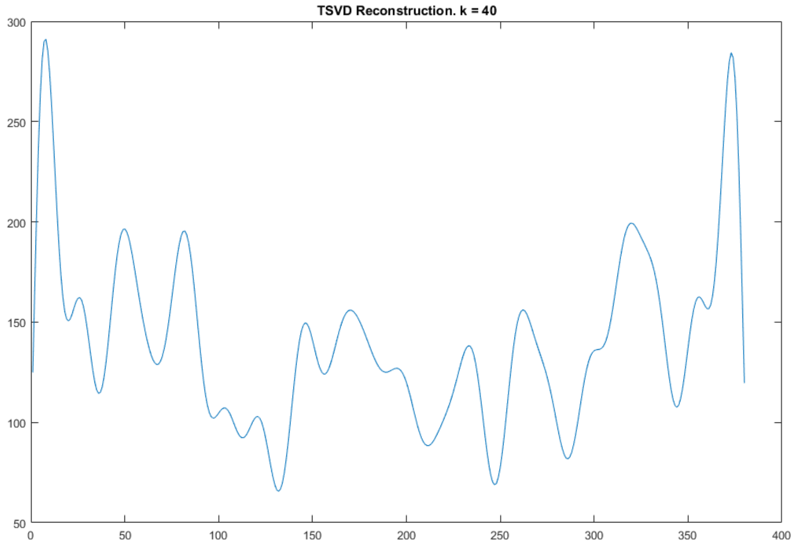

Figure 5: TSVD Reconstruction result.

Q5: Best regularization parameter with L-Curve.

Find the best regularization parameter of TR and TSVD with L-Curve.

Answer: $\lambda = 0.6$ and $k = 178$.

Listing 5: MATLAB code for ploting Tikhonov Regularization L-Curve.

```
% L-Curve Tikhonov
x_tk_l = zeros(n,100);
x_tk_norm = zeros(100,1);
x_tk_disp = zeros(100,1);
it = 1;
for i = 0.1:0.1:10
    x_tk_l(:,it) = pinv(A' * A + i^2 * eye(n))*A'*c;
    x_tk_norm(it) = norm(x_tk_l(:,it));
    x_tk_disp(it) = norm(c-A*x_tk_l(:,it));
    it = it + 1;
end
figure;
loglog(x_tk_norm,x_tk_disp,'c.','MarkerSize', 20,'MarkerEdgeColor',[0 .5 .5]);
hold on
loglog(x_tk_norm,x_tk_disp,'k-','LineWidth', 2)
hold off
text(1650.463759, 865.547447,'  \lambda = 0.6');
title('L-Curve Tikhonov Regulation');
```

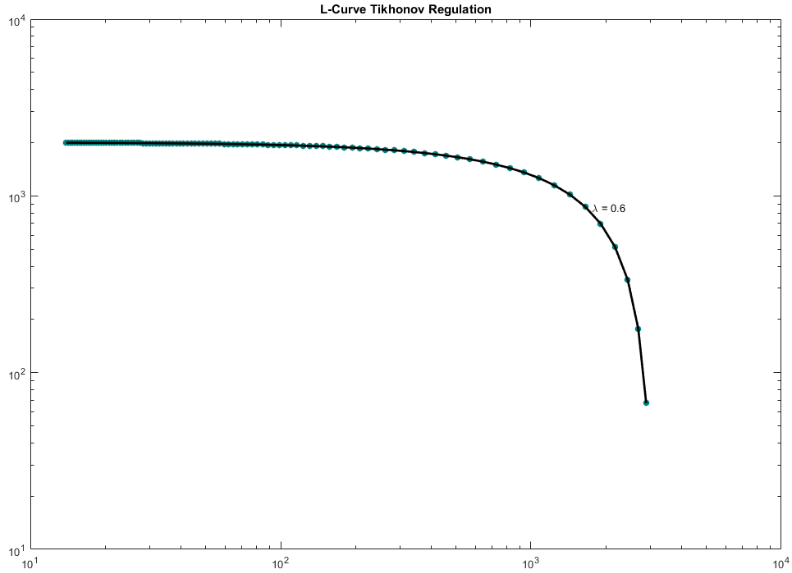

Figure 6: L-Curve result Tikhonov Regulation.

Listing 6: MATLAB code for ploting Tikhonov Regularization L-Curve.

```
figure;
loglog(normX,discr,'c.','MarkerSize', 10,'MarkerEdgeColor'↩
    ,[.5 0 0]);
hold on
loglog(normX,discr,'k-','LineWidth', 2)
hold off
text(11503.53877, 4.551855,'      k = 178');
title('L-Curve TSVDS Regulation');
```

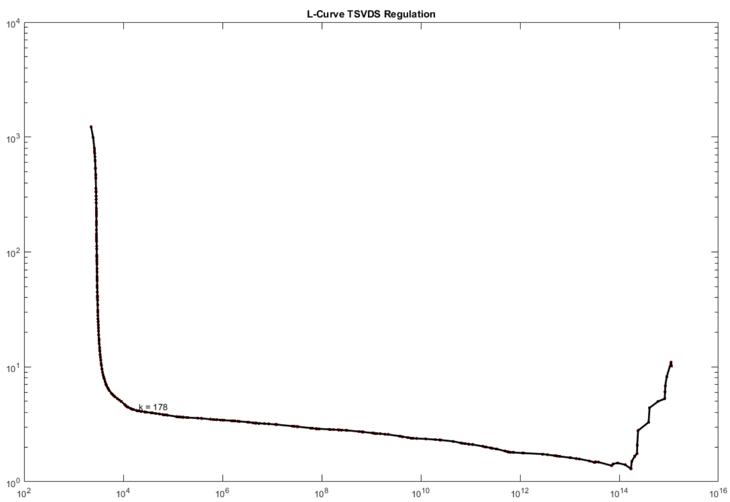

Figure 7: L-Curve result TSVD.