ILL-POSED LEAST SQUARES PROBLEMS

1. Techniques for ill-posed least squares problems. We write throughout this lecture $\|\cdot\| = \|\cdot\|_2$. Consider the following least squares problems:

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|, \qquad \min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}_\varepsilon \mathbf{x}\|,$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{A}_\varepsilon = \begin{bmatrix} 1 & 0 \\ 0 & \varepsilon \\ 0 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Their minimal norm solutions are given by

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}_\varepsilon = \begin{bmatrix} 1 \\ 1/\varepsilon \end{bmatrix}.$$

Obviously, they are arbitrarily different as ε approaches to 0. Given this sensitivity, it is natural to ask what sense it can make to solve a nearly rank-deficient least squares problem.

1.1. Truncated SVD solutions. Let

$$\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^*, \quad k \le r = \operatorname{rank}(\mathbf{A}).$$

Consider the least squares problem

$$\min_{\mathbf{x} \in \mathbb{C}^n} \|\mathbf{b} - \mathbf{A}_k \mathbf{x}\|.$$

The minimum norm least squares solution is given by

$$\mathbf{x}_k = \mathbf{A}_k^\dagger \mathbf{b} = \sum_{i=1}^{k} \frac{1}{\sigma_i} \mathbf{v}_i \mathbf{u}_i^* \mathbf{b}.$$

We have

$$\|\mathbf{x}_k\|^2 = \sum_{i=1}^{k} \frac{1}{\sigma_i^2} |\mathbf{u}_i^* \mathbf{b}|^2, \quad \mathbf{x}_r = \mathbf{A}^\dagger \mathbf{b}, \quad \|\mathbf{x}_k\|^2 = \|\mathbf{x}_r\|^2 - \sum_{i=k+1}^{r} \frac{1}{\sigma_i^2} |\mathbf{u}_i^* \mathbf{b}|^2,$$

and

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|^2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}_r\|^2 + \sum_{i=k+1}^{r} |\mathbf{u}_i^* \mathbf{b}|^2.$$

REMARK 1.1. *Using the truncated* SVD *can greatly reduce* $\|\mathbf{x}_k\|$ *by omitting the* $\mathbf{u}_i^* \mathbf{b}/\sigma_i$ *terms for small* σ_i*, but increase in the norm of the residual,* $\|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|$ *is comparatively modest, only adding terms like* $\mathbf{u}_i^* \mathbf{b}$*.*

REMARK 1.2. *In applications where* $\sigma_{k+1}, \cdots, \sigma_r$ *are small, the reduction in the norm of the solution can yield much more physically realistic answer by removing the artificial but overwhelming effects of noisy data, while the modest increase in the residual is not such a big concern.*

1.2. Regularized solutions. Consider the *penalized* problem

$$\min_{\mathbf{x}\in\mathbb{C}^n} \left\{\|\mathbf{b}-\mathbf{A}\mathbf{x}\|^2 + \lambda^2\|\mathbf{x}\|^2\right\},$$

where $\lambda > 0$ is called the *regularization parameter.*

1.2.1. For a given $\lambda > 0$, what's the optimal solution?.

THEOREM 1.3. *The unique solution of the least squares problem*

$$\min_{\mathbf{x}\in\mathbb{C}^n} \left\| \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{A} \\ \lambda\mathbf{I} \end{bmatrix} \mathbf{x} \right\|$$

is the unique optimal solution of the penalized problem.

Proof. Note that

$$\min_{\mathbf{x}\in\mathbb{C}^n} \left\{\|\mathbf{b}-\mathbf{A}\mathbf{x}\|^2 + \lambda^2\|\mathbf{x}\|^2\right\} = \min_{\mathbf{x}\in\mathbb{C}^n} \left\| \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{A} \\ \lambda\mathbf{I} \end{bmatrix} \mathbf{x} \right\|^2.$$

The statement then follows.

COROLLARY 1.4. *The unique optimal solution of the penalized problem is*

$$\mathbf{x}_\lambda = (\mathbf{A}^*\mathbf{A} + \lambda^2\mathbf{I})^{-1}\mathbf{A}^*\mathbf{b} = \sum_{i=1}^{r} \frac{\sigma_i}{\sigma_i^2+\lambda^2}\mathbf{v}_i\mathbf{u}_i^*\mathbf{b}.$$

1.2.2. How to choose λ?.

'L'-curve criterion. One way to select λ is to create a plot with $\log\|\mathbf{b}-\mathbf{A}\mathbf{x}_\lambda\|$ on the horizontal axis, and $\log\|\mathbf{x}_\lambda\|$ on the vertical axis, sampled over a wide range of λ values (varying over order of magnitude). For many applications, the best choice for λ will yield values of $\|\mathbf{b}-\mathbf{A}\mathbf{x}_\lambda\|$ and $\|\mathbf{x}_\lambda\|$ that land at the sharp bend in this 'L'-curve of points $(\log\|\mathbf{b}-\mathbf{A}\mathbf{x}_\lambda\|, \log\|\mathbf{x}_\lambda\|)$.

Exercise. In this exercise, you will use regularization to decode some barcodes, i.e., UPC or EAN. First, we explain how UPC/EAN barcodes work (see Wiki for more details). The UPC/EAN system encodes 12 digits through a series of 95 alternating black and white bars of varying widths. Each bar, be it black (`b`) or white (`w`), has one of four widths: either 1, 2, 3, or 4 units wide. Here is how they encode information. See Table 1.

Each of the 12 digits is encoded with four bars (two white, two black); the widths of these bars will differ, but the sum of the width of all four bars is always 7, so that all UPC/EAN codes have the same width (95 units wide). Each digit (0-9) corresponds to a particular pattern of bar widths, according to Table 2. (For reasons beyond our interest, two different patterns can be used for each digit.)

For example, if the four bars (read left to right) have the patter 3211 or 1123, the corresponding UPC/EAN digit is 0. To check that you understand the system, try decoding the UPC below (Figure 1 for a can of Coke). To start you out, we give the code for the first three digits. See Table 3. It can be tricky to judge the widths—you can appreciate the accuracy of optical scanners!

We want to simulate the reading of this UPC/EAN code by an optical scanner, e.g., in a supermarket checkout line. The barcode is represented mathematically as a function $f(t)$ that takes the values zero and one: zero corresponds to white bars, one corresponds to black bars. The function corresponding to the Coke bar code is shown below. See Figure 2.

Table 1
Encode information.

colors	number of bars	description
bwb	three bars of width 1	start code
wbwb	four bars of total width 7	first digit
wbwb	four bars of total width 7	second digit
wbwb	four bars of total width 7	third digit
wbwb	four bars of total width 7	fourth digit
wbwb	four bars of total width 7	fifth digit
wbwb	four bars of total width 7	sixth digit
wbwbw	five bars of width 1	middle code
bwbw	four bars of total width 7	seventh digit
bwbw	four bars of total width 7	eighth digit
bwbw	four bars of total width 7	ninth digit
bwbw	four bars of total width 7	tenth digit
bwbw	four bars of total width 7	eleventh digit
bwbw	four bars of total width 7	twelfth digit
bwb	three bars of width 1	end code

Table 2
Two patterns for digits 0–9.

digit	0	1	2	3	4	5	6	7	8	9
P1	3211	2221	2122	1411	1132	1231	1114	1312	1213	3112
P2	1123	1222	2212	1141	2311	1321	4111	2131	3121	2113

Fig. 1. *Coke* UPC *barcode.*

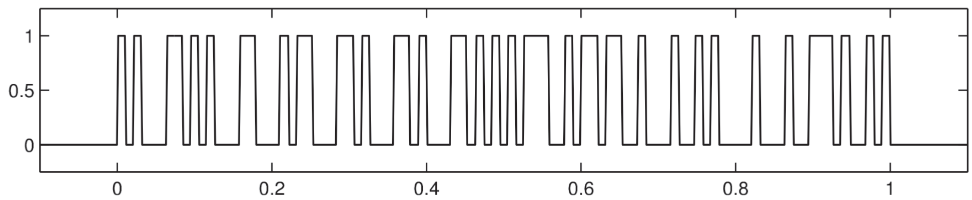

Fig. 2. *The function* $f(t)$ *for Coke* UPC *barcode.*

The optical scanner can only acquire a function b (sampled at discrete points in the vector, $\mathbf{b} \in \mathbb{C}^n$), a blurred version of f, where we assume the blurring kernel is $h(s,t) = \mathrm{e}^{-(t-s)^2/z^2}$ for $z = 0.01$:

$$\int_0^1 h(s,t) f(t) \mathrm{d}t = b(s).$$

The Coke UPC function, blurred by this kernel, is shown below. See Figure 3. From this blurred function, it would be difficult to determine the widths of the bars, and

TABLE 3
Coke UPC *barcode encoding.*

color	b w b	w b w b	w b w b	w b w b
width	1 1 1	3 2 1 1	1 1 3 2	3 1 1 2
meaning	start code	0	4	9

hence to interpret the barcode. We shall try to improve the situation by solving the inverse problem $\mathbf{Af} = \mathbf{b}$ for the vector $\mathbf{f}$ that samples the function $f(t)$ at the points $t_k = (k - 1/2)/n$.

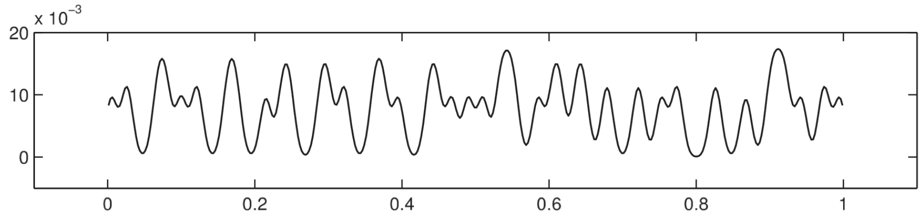

FIG. 3. *The blurred Coke* UPC *function.*

The function `[A,b,beps,fe]=coke_upc` generates, for $n = 500$: the blurring matrix for the specified kernel $\mathbf{A} \in \mathbb{C}^{500\times 500}$; the blurred function sampled at 500 points, $\mathbf{b} \in \mathbb{C}^{500}$; the blurred vector with 5% random noise, $\mathbf{b}_\varepsilon \in \mathbb{C}^{500}$; the exact bar code solution $\mathbf{f}_\mathrm{e}$. (We generateed $\mathbf{b}$ as $\mathbf{b} = \mathbf{Af}_\mathrm{e}$.)

Some of these questions are intentionally open-ended. You will be graded primarily on the thoroughness of your experiments, rather than for recovering a particular value for the barcodes. Include plenty of plots and label what they show, and describe what you learn from them.

(a) Produce a plot showing the vector $\mathbf{f}_\mathrm{i} = \mathbf{A}^{-1}\mathbf{b}$ one obtains by directly inverting (`fi = A\b`). To plot with the same aspect ratio shown above, you can use the following commands:

```
t=([1:n]-1/2)/n;
figure, axes('position',[.075 .1 .85 .2]);
plot(t,fe,'k-','linewidth',1); hold on
plot(t,fi,'r-','linewidth',1)
```

(b) Repeat part (a), but now using the noisy, blurred vector: $\mathbf{f}_\varepsilon = \mathbf{A}^{-1}\mathbf{b}_\varepsilon$. Plot your recovered barcode $\mathbf{f}_\varepsilon$ along with the true value $\mathbf{f}_\mathrm{e}$.

(c) Now see if you can do any better using the truncated singular value decomposition. If $\mathbf{A} = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^*$, then the truncated SVD solution from the leading rank-k part of $\mathbf{A}$ is:

$$\mathbf{f}_k = \sum_{i=1}^{k} \frac{1}{\sigma_i} \mathbf{v}_i (\mathbf{u}_i^* \mathbf{b}).$$

Experiment with different several values k. (Start with $k = 150$, say.) Describe how varying k affects the quality of the solution. Is there any difference if you use the exact blurred vector $\mathbf{b}$, versus the noisy blurred vector $\mathbf{b}_\varepsilon$? Illustrate your experiments by producing plots like the ones you constructed in parts (a) and (b). Are any of these recoveries good enough that you can estimate the value of the barcode?

(d) As an alternative to the truncated SVD, we shall explore the solutions obtained from the regularized least squares problem

$$\min_{\mathbf{f}\in\mathbb{C}^n} \left\{\|\mathbf{b}-\mathbf{A}\mathbf{f}\|^2 + \lambda^2\|\mathbf{f}\|^2\right\}$$

for various λ. Recall that you can find the optimal value $\mathbf{f}_\lambda$ by solving the least squares problem

$$\min_{\mathbf{f}\in\mathbb{C}^n} \|\widehat{\mathbf{b}} - \mathbf{A}_\lambda \mathbf{f}\|,$$

where

$$\mathbf{A}_\lambda = \begin{bmatrix} \mathbf{A} \\ \lambda\mathbf{I} \end{bmatrix} \in \mathbb{C}^{2n\times n}, \qquad \widehat{\mathbf{b}} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{2n}.$$

(i) Using the value $\mathbf{b}_\varepsilon$, create the 'L'-curve plot described in the previous exercise using parameter values $\lambda = 10^{-8}, \cdots, 10^0$. (Use `logspace(-8,0,100)` to generate 100 logarithmically equally-spaced values of λ in this range.) Recall that the 'L'-curve is a `loglog` plot with $\|\mathbf{b}_\varepsilon - \mathbf{A}\mathbf{f}_\lambda\|$ on the horizontal axis and $\|\mathbf{f}_\lambda\|$ on the vertical axis.
(ii) Pick a λ value corresponding to the right-angle in the 'L'-curve, and plot the recovered solution $\mathbf{f}_\lambda$ as done in earlier parts of the problem. Feel free to show plots for several different λ values that vary over one or two orders of magnitude. Also show the $\mathbf{f}_\lambda$ you get for the same value of λ if you instead use the true date $\mathbf{b}$ instead of $\mathbf{b}_\varepsilon$.

(e) Using your recovered $\mathbf{f}_k$ and/or $\mathbf{f}_\lambda$ from parts (c) and (d), attempt to reconstruct the barcode for the can of Coke. This is not trivial–just do your best, using insight from the structure of the UPC code to help.

(f) The function `[b,beps]=mystery_ean` generates the blurred $\mathbf{b}$ and blurred, noisy $\mathbf{b}_\varepsilon$ EAN barcodes for a mystery product (again with $n = 500$). Use the techniques in parts (c) and (d) to attempt to recover the mystery barcode. Show samples of the results from your various attempts.

(g) What is the product described by the mystery EAN barcode? (The first digit is 4, which is not expressed by the white and black bars. Once you have found the correct EAN, search it on Taobao or Baidu and find what it is.)