

University of Toronto
Department of Electrical and Computer Engineering
ECE367 MATRIX ALGEBRA AND OPTIMIZATION

Problem Set #4
Fall 2023

Prof. Wei Yu

Due: 11:59pm (Toronto time) November 25, 2023

Homework policy: Problem sets must be turned by the due date and time. Late problem sets will not be accepted. See the information sheet for further details. The course text “Optimization Models” is abbreviated as “OptM” and “Introduction to Applied Linear Algebra” as “IALA”.

Problems are categorized as:

- **“Theory” problems:** These are mostly mathematical questions designed to give you deeper insight into the fundamentals of the ideas introduced in this class.
- **“Application” problems:** These questions are designed to expose you to the breadth of application of the ideas developed in class and to introduce you to useful numerical toolboxes. Problems of this sort often ask you to produce plots and discuss your results; said plots and discussions should be included in and form part of your submission – think of your submitted solution like a lab book. Your attached code simply provides back-up evidence.
- **“Optional” problems:** Optional problems provide extra practice or introduce interesting connections or extensions. They need not be turned in. I will assume you have reviewed and understood the solutions to the optional problems when designing the exams.

Hand-in procedure:

- **Initial submission:** Your initial submission of the “Theory” and “Application” questions must be submitted via Quercus upload by the due date.
- **Self-assessment:** After the problem set is due we will post solutions. You will have one week from the initial due date to submit a commented version of your assignment in which, using as a reference the posted solutions, you highlight your errors or omissions in red. Please annotate the PDF you initially submitted. If you have not submitted a solution you cannot submit the self-assessment.
- **Late problem sets are not accepted**
- **Grading:** Per the course handout problem sets are graded for completion only. Points are assigned to (i) Initial submission of theory part, (ii) Initial submission of application part, (iii) Self-assessment.

These problem sets were initially developed by Prof. Stark Draper.

Points allocation

- Theory parts (initial submission): 1 pt
 - Application parts (initial submission): 1 pt
 - Self-assessment: 1 pt
-

Problem categorization and main concepts covered**Theory**

- Least Squares: Problems 4.1, 4.2, 4.3, 4.4

Application

- Optimal Control: Problem 4.5

Optional

- CAT scan: 4.6

THEORY PROBLEMS

Problem 4.1 (Least squares and total least squares)

OptM Problem 6.1, the part on finding the least-squares line. We did not cover total least-square in class, so you do not need to do the part on total least-squares.

Problem 4.2 (Lotka's law and least squares)

OptM Problem 6.3. (Hint: Do a variable transformation to make the problem look like a least-squares problem.)

Problem 4.3 (Model for enzyme kinetics)

OptM Problem 6.6 parts 1 and 2. (Note that Problem 6.6 is not given a title in OptM.)

Problem 4.4 (Solving least squares problems using the Moore-Penrose pseudoinverse)

In this problem you derive the result that the Moore-Penrose pseudoinverse can be used to solve least squares problems for overdetermined, underdetermined, and rank-deficient systems. Recall that least square problems consider the system of linear equations $Ax = b$ where $A \in \mathbb{R}^{m,n}$ and $\text{rank}(A) = r$. If the compact SVD of A is $A = U_r \Sigma V_r^T$ where $\Sigma \in \mathbb{R}^{r,r}$ is a positive definite diagonal matrix of (non-zero) singular values, $U_r \in \mathbb{R}^{m,r}$ and $V_r \in \mathbb{R}^{n,r}$ each contain orthonormal columns, then the Moore-Penrose pseudoinverse is $A^\dagger = V_r \Sigma^{-1} U_r^T$.

- (a) Recall that the *overdetermined* least squares problem considers an $A \in \mathbb{R}^{m,n}$ when $m > n$ (more constraints in the y vector than parameters in the x vector). Here the objective is to find an x that minimizes $\|Ax - y\|_2$. Show that an optimal solution is $x^* = A^\dagger y$. To show this recall that an optimal solution x^* must satisfy the “normal” equations $A^T A x^* = A^T y$. Verify that $x^* = A^\dagger y$ satisfies the normal equations.
- (b) Recall that the *underdetermined* least squares problem considers an $A \in \mathbb{R}^{m,n}$ when $m < n$ (fewer constraints in the y vector than parameters in the x vector). Here the objective is to find the x that minimizes $\|x\|_2$ while satisfying $Ax = y$ (equivalently, satisfying $\|Ax - y\|_2 = 0$). Show that the optimal solution $x^* = A^\dagger y$. To show this recall that the optimal solution x^* must satisfy two conditions: (i) $x^* \in \mathcal{R}(A^T)$ and (ii) $Ax^* = y$. Verify that $x^* = A^\dagger y$ satisfies (i) all the time. Under what conditions does x^* satisfy condition (ii)? (Hint, think about the rank of A .)

In the above two parts we haven't explicitly considered the role of the rank r of the A matrix. But we also note that the only place where the rank of A comes into the discussion of parts (a) and (b) is in the discussion of condition (ii) of part (b).

Recall that $r \leq \min\{m, n\}$. When $r = \min\{m, n\}$ the A matrix is full column rank in the overdetermined problem and is full row rank in the underdetermined problem. In both these full-rank cases x^* has the simple expression presented in class. Now we consider what happens when $r < \min\{m, n\}$.

- (c) In this part consider the situation where $\text{rank}(A) = r < m < n$. This is a “rank-deficient” underdetermined least squares problem. If we set $x^* = A^\dagger y$, what characteristics do Ax^* and $\|x^*\|_2$ satisfy? (Hint: this is a type of hybrid problem that at the same time can have characteristics of both overdetermined and underdetermined least squares.)

APPLICATION PROBLEMS

Problem 4.5 (Optimal control of a unit mass)

Consider a unit mass with position $x(t)$ and velocity $\dot{x}(t)$ subject to force $f(t)$, where the force is piecewise constant over intervals of duration one second, i.e., $f(t) = p_n$ for $n - 1 < t \leq n$, $n = 1, \dots, 10$; we consider the system for 10 seconds in total. Ignore friction. Assume the mass has zero initial position and velocity, i.e., $x(0) = \dot{x}(0) = 0$.

- (a) Derive the “state-space” equations that describe a discrete-time version of the dynamics of this system. In particular, derive relationships between $x(n)$ and $\dot{x}(n)$ in terms of $x(n - 1)$, $\dot{x}(n - 1)$, and the driving force p_n , for each $n \in \{1, 2, \dots, 10\}$. (You will need to use your knowledge of basic physics to determine these relationships.) The two equations you derive should be

$$\begin{aligned} x(n) &= x(n - 1) + \dot{x}(n - 1) + (1/2)p_n, \\ \dot{x}(n) &= \dot{x}(n - 1) + p_n, \end{aligned}$$

which you can then stack up in vector form to form the state-space equations

$$\begin{bmatrix} x(n) \\ \dot{x}(n) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_A \begin{bmatrix} x(n - 1) \\ \dot{x}(n - 1) \end{bmatrix} + \underbrace{\begin{bmatrix} 1/2 \\ 1 \end{bmatrix}}_b p_n. \quad (1)$$

We note that this continuous time system can only be discretized exactly because in this system the control effort $f(t)$ is held constant over each time interval. (At this point we recommend you remind yourself of the discussion of state-space models of linear dynamical systems in OptM Exercise 3.4.)

- (b) Find the p_n , $n \in [10]$ that minimizes

$$\sum_{n=1}^{10} p_n^2$$

subject to the following specifications: $x(10) = 1$, $\dot{x}(10) = 0$. Plot the optimal f , the resulting x and \dot{x} . (I.e., plot $f(t)$, $x(t)$, and $\dot{x}(t)$ for $0 \leq t \leq 10$.) Give a short intuitive explanation of what you see.

- (c) Suppose that we add one more specification $x(5) = 0$, i.e., we require the mass to be at position 0 at time 5. Plot the optimal f , the resulting x and \dot{x} . Give a short intuitive explanation of what you see.

OPTIONAL PROBLEMS

Problem 4.6 (CAT scan imaging)

In this problem you will obtain a basic understanding of the math behind computer aided tomography (CAT) scanning. Before proceeding we recommend you read the Example 6.6 (*CAT scan imaging*) in OptM to obtain a basic understanding about the CAT scan. In the following parts, we assume the same meaning for y , A and x as in Example 6.6. Download the file `scanImage.p` from the course website. Although you will observe that the file is encrypted, this file defines a MATLAB function called `scanImage(M)` that takes in one optional argument and produces one output. You can execute this function just the same way you execute other MATLAB functions. This function encapsulates a “virtual CAT scanner” that is able to scan 2-d images. Consider a grayscale image of height h and width w . Such an image can be represented as a $h \times w$ matrix M with values between 0 and 255. Let $n = hw$ be the total number of pixels (in CAT-scan terminology this is the number of “voxels”, as described in OptM). If you execute the function with M as the argument, it will return a vector $y \in \mathbb{R}^m$ which is the scan result. Similar to OptM Example 6.6, y is the vector of log-intensity ratios, and m corresponds to the number of beams used for the scan. Note that for the provided scanner, you have no control over the number of beams used or how beams are positioned across the image. However, the dimension m and the positioning of beams is a function only of h and w . This means that any two images which are of the same dimensions will be scanned with identical beam setups.

Similar to OptM, let us denote the vectorized image M by the n -dimensional column vector x . The input x can be related to the output y of the scanner by matrix $A \in \mathbb{R}^{m \times n}$ as $y = Ax$. In OptM Example 6.6, A is computed analytically, using geometric properties of the beam setup. For this numerical problem, even if you knew the positions (and angles) of the beams, computing A analytically would be ... tedious. We propose an alternative approach to estimate A . Observe that

$$[y_1 \ y_2 \ \cdots \ y_m]^T = \begin{bmatrix} a^{(1)} & a^{(2)} & \cdots & a^{(n)} \end{bmatrix} [x_1 \ x_2 \ \cdots \ x_n]^T, \quad (2)$$

where $a^{(i)}$ is the i th column of A . In the provided scanner, you have control over the input image. Consider the case where you obtain an image M_1 by setting the first pixel (the $(1, 1)$ coordinate of M) to 1 and all others to 0. The vectorized representation of M_1 will be $[1, 0, \dots, 0]^T$. If you put M_1 through the scanner, the resulting y should be equal to $a^{(1)}$ as per (2). This way, by turning each pixel on while all others are off, you can estimate each of the columns in A . As you may guess, this is not how things work with real-world CAT scanners, but eases our development herein.

- (a) Now, using the method described in the text, write a MATLAB script to estimate A when $h = 50$ and $w = 50$. In this case you will observe the dimension of the scanner output, i.e., m is 1950. Since A is an $m \times n$ matrix, we can visualize A by treating it as an image. Use the MATLAB function `imshow(A, [])` to display A as an image and include a scaled down version of the image with your answers. Note that if `imshow(A, [])` displays a mostly black image, `imshow(-A, [])` will display a mostly white image.

- (b) Till this point, you used the provided scanner function `scanImage(.)` to get the outputs for known images, yielding an estimate for A . In this part, you are given the scanner output y of an unknown image M_{un} , which has dimensions $h = 50$ and $w = 50$. The scanner output y for the image M_{un} can be obtained by executing `y=scanImage`, without passing an input to the function. (We use an encrypted function so that in this part you need to determine an estimate of the image.) Since y and A are known, you can solve for an x that satisfies $y = Ax$. You will find that $\text{rank } A < m < n$ so this is an underdetermined and rank-deficient system of equations. However, the matrix A may also have non-zero singular values that are extremely close to zero which, to avoid numerical instabilities, you will need to assume to be zero. To choose an appropriate effective rank to work with, plot the r singular values σ_i of A versus their index $i \in [r]$. Visually inspect your plot to choose an effective minimum non-zero singular value. After solving for x , obtain the image M_{un} and include with your answers. What is the hidden message in the scanned image?

In a real-world CAT scanner, the number of voxels (n) usually is much larger than the number of beams (m) used to produce the vector y . Note that the number of voxels considered is proportional to the resolution of the image obtained. A typical CAT scanner provides around 0.5mm resolution, which means two adjacent voxels are around 0.5mm apart.

For practical CAT image reconstruction, rather than least squares, the ‘inverse Radon transform’ is typically used to solve for the vector x . The principle behind Radon transform involves cool application for two-dimensional Fourier transform, if you are interested in reading up on this topic.