MA 402: Project 3

Richard Watson, Mountain Chan, Cole Nash

October 4, 2020

Instructions:

- Detailed instructions regarding submission are available on the class website[1].
- The zip file should contain three files hw3.pdf, hw3.tex, classnotes.sty.

Pen-and-paper exercises

The problems from this section total 30 points.

1) (10 points) Let the matrix $A \in \mathbb{R}^{3\times 2}$ have the SVD

$$A = \begin{bmatrix} 4 & 0 \\ -5 & -3 \\ 2 & 6 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & -2 & 2 \\ -2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}\begin{bmatrix} 6\sqrt{2} & 0 \\ 0 & 3\sqrt{2} \\ 0 & 0 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}.$$

Let $b = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^\top$. Compute the least squares solution in two different ways:

$$x^* = (A^TA)^{-1}A^Tb = V[\Sigma^{-1}0]U^Tb$$

Question: Do we have to show intermediary steps?

(a) Using the normal equation approach;

$$x^* = (A^TA)^{-1}A^Tb = \left(\begin{bmatrix} 4 & -5 & 2 \\ 0 & -3 & 6 \end{bmatrix}\begin{bmatrix} 4 & 0 \\ -5 & -3 \\ 2 & 6 \end{bmatrix}\right)^{-1}\begin{bmatrix} 4 & -5 & 2 \\ 0 & -3 & 6 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -\frac{1}{4} \\ \frac{5}{12} \end{bmatrix}$$

(b) Using the SVD of A.

$$x^* = V[\Sigma^{-1}0]U^Tb = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} \frac{1}{6\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{3\sqrt{2}} & 0 \end{bmatrix}\frac{1}{3}\begin{bmatrix} 1 & -2 & 2 \\ -2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -\frac{1}{4} \\ \frac{5}{12} \end{bmatrix}$$

[1] `https://github.ncsu.edu/asaibab/ma402/blob/master/project.md`

2) (10 points) Consider the line $f(t) = \alpha + \beta t$ that we seek to fit the data

$$\mathcal{D} = \{(t_1, b_1), \dots, (t_m, b_m)\}.$$

Show that the least squares coefficients α and β, obtained by solving the least squares problem, satisfy

$$\alpha = \frac{(\sum_i t_i^2)(\sum_i b_i) - (\sum_i t_i)(\sum_i b_i t_i)}{m(\sum_i t_i^2) - (\sum_i t_i)^2} \qquad \beta = \frac{m(\sum_i b_i t_i) - (\sum_i t_i)(\sum_i b_i)}{m(\sum_i t_i^2) - (\sum_i t_i)^2}.$$

$$\text{Let } A = \begin{bmatrix} 1 & t_1 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix}, b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \text{ and } x = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

$$x = (A^T A)^{-1} A^T b$$

$$\text{Thus } A^T A = \begin{bmatrix} m & \Sigma_i t_i \\ \Sigma_i t_i & \Sigma_i t_i^2 \end{bmatrix}$$

$$(A^T A)^{-1} = \frac{1}{det(A^T A)} \begin{bmatrix} \Sigma_i t_i^2 & -\Sigma_i t_i \\ -\Sigma_i t_i & m \end{bmatrix}$$

$$det(A^T A) = m\Sigma_i t_i^2 - (\Sigma_i t_i)^2$$

$$(A^T A)^{-1} = \frac{1}{m\Sigma_i t_i^2 - (\Sigma_i t_i)^2} \begin{bmatrix} \Sigma_i t_i^2 & -\Sigma_i t_i \\ -\Sigma_i t_i & m \end{bmatrix}$$

$$A^T b = \begin{bmatrix} \Sigma_i b_1 \\ \Sigma_i b_i t_i \end{bmatrix}$$

$$x = \frac{1}{m\Sigma_i t_i^2 - (\Sigma_i t_i)^2} \begin{bmatrix} \Sigma_i t_i^2 & -\Sigma_i t_i \\ -\Sigma_i t_i & m \end{bmatrix} \begin{bmatrix} \Sigma_i b_1 \\ \Sigma_i b_i t_i \end{bmatrix}$$

$$x = \frac{1}{m\Sigma_i t_i^2 - (\Sigma_i t_i)^2} \begin{bmatrix} \Sigma_i b_i \Sigma_i t_i^2 - \Sigma_i b_i \Sigma_i t_i \\ m\Sigma_i b_i t_i - \Sigma_i t_i \Sigma_i b_i \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

$$\text{Thus } \alpha = \frac{(\sum_i t_i^2)(\sum_i b_i) - (\sum_i t_i)(\sum_i b_i t_i)}{m(\sum_i t_i^2) - (\sum_i t_i)^2}$$

$$\beta = \frac{m(\sum_i b_i t_i) - (\sum_i t_i)(\sum_i b_i)}{m(\sum_i t_i^2) - (\sum_i t_i)^2}$$

3) (10 points) Let $A \in \mathbb{R}^{m \times n}$ and let $\text{rank}\,(A) = n$. Let x^* be the solution of the least squares problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2.$$

Define the residual corresponding to the optimal solution $r^* = b - Ax^*$.

(a) Starting with the normal equations $A^\top A x^* = A^\top b$, show that the x^*, r^* also satisfy

$$\begin{bmatrix} I & A \\ A^\top & 0 \end{bmatrix} \begin{bmatrix} r^* \\ x^* \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Remark: This system is known as the augmented equations and maybe beneficial to use when A is sparse (i.e., it has many zero entries).

We are given the following system of equations:

$$r^* + Ax^* = b$$
$$A^T r^* + 0x^* = 0$$

The first of these 2 equations is definitively true, based on the definition of the residual corresponding to the optimal solution given in the problem. The second equation can be shown to be true by using substitution:

$$\begin{aligned} A^T r^* + 0x^* &= 0 \\ A^T(b - Ax^*) &= 0 \\ A^T b - A^T A x^* &= 0 \\ A^T A x^* &= A^T b \\ x^* &= (A^T A)^{-1} A^T b \end{aligned}$$

This is the definition of x^* and thus true.

(b) Show: the vectors Ax^* and $b - Ax^*$ are orthogonal and

$$\|b\|_2^2 = \|b - Ax^*\|_2^2 + \|Ax^*\|_2^2.$$

In your words, provide an interpretation of this result.

First we show that Ax^* and $b - Ax^*$ are orthogonal by substituting $Pb = Ax^*$, keeping in mind that P is idempotent and symmetric:

$$\begin{aligned} (Ax^*)^T(b - Ax*) &= \\ (Pb)^T(b - Pb) &= \\ b^T P^T b - b^T P^T P b &= \\ b^T P b - b^T P P b &= \\ b^T P b - b^T P^2 b &= \\ b^T P b - b^T P b &= 0 \end{aligned}$$

Thus Ax^* and $b - Ax^*$ are orthogonal.

Next we will show that

$$\|b\|_2^2 = \|b - Ax^*\|_2^2 + \|Ax^*\|_2^2 :$$

Keep in mind that $Ax*$ is idempotent, so $Ax^* = (Ax^*)^2$:

$$\begin{aligned} \|b\|_2^2 &= \sum_{i=1}^{n} b_i^2 \\ \|b\|_2^2 &= \sum_{i=1}^{n} b_i^2 - 2(Ax^*)_i + 2(Ax^*)_i \\ \|b\|_2^2 &= \sum_{i=1}^{n} b_i^2 - 2(Ax^*)_i + 2(Ax^*)_i^2 \\ \|b\|_2^2 &= \sum_{i=1}^{n} b_i^2 - 2(Ax^*)_i + (Ax^*)_i^2 + \sum_{i=1}^{n} (Ax^*)_i^2 \\ \|b\|_2^2 &= \sum_{i=1}^{n} (b_i - (Ax^*)_i)^2 + \sum_{i=1}^{n} (Ax^*)_i^2 \\ \|b\|_2^2 &= \|b - Ax^*\|_2^2 + \|Ax^*\|_2^2 \end{aligned}$$

Numerical exercises

The problems from this section total 20 points, and were modified from Carl Meyer's book "Matrix Analysis and Applied Linear Algebra." MATLAB users may use "backslash" for solving the least squares problem, Python users may use `numpy.linalg.lstsq`.

4) (10 points) After studying a certain type of cancer, a researcher hypothesizes that in the short run the number (y) of malignant cells in a particular tissue grows exponentially with time (t). That is, $y = \alpha_0 e^{\alpha_1 t}$. The researcher records the data given below

t (days)	1	2	3	4	5
y (cells)	16	27	45	74	122

(a) Determine least squares estimates for the parameters α_0 and α_1 from the researcher's observed data.

```
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

# Model
def func(t, a, b):
  return a*np.exp(b * t)

# Initial data
xdata = np.array([1,2,3,4,5])
ydata = np.array([16,27,45,74,122])

# Curve fitting
param,ph = curve_fit(func, xdata, ydata)
```

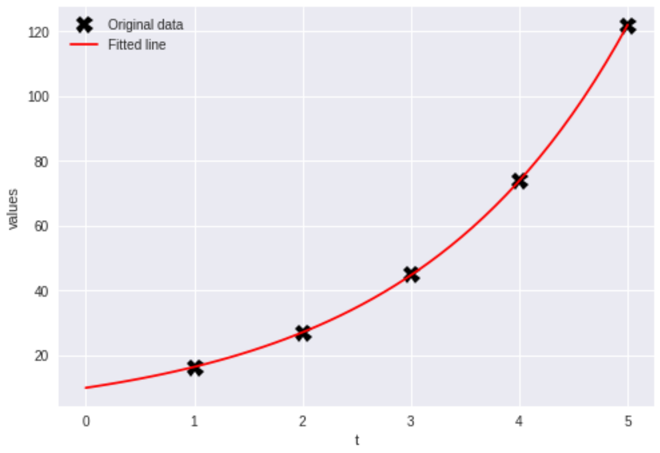
$\alpha_0 = 9.91943036$
$\alpha_1 = 0.50205373$

(b) Plot the data points (with black crosses) along with the "best" fit curve (as a solid red line).

Hint: what common transformation converts an exponential function into a linear function?

```
# Generate data points for model function
t = np.linspace(0, 5, 50)

# Plotting
plt.plot(xdata, ydata, 'X', label='Original data', markersize = 13,
c = 'black')
plt.plot(t, param[0]*np.exp(param[1]*t), 'r', label='Fitted line')
plt.legend()
plt.show()
```



5) (10 points) Consider the time (T) it takes for a runner to complete a marathon (26 miles and 385 yards). Many factors such as height, weight, age, previous training, etc. can influence an athlete's performance, but experience has shown that the following three factors are particularly important:

$$\begin{aligned} x_1 &= \text{Ponderal index} = \frac{\text{height (in.)}}{[\text{weight (lbs.)}]^{1/3}}, \\ x_2 &= \text{Miles run during the previous 8 weeks}, \\ x_3 &= \text{Age (years)}. \end{aligned}$$

A linear model hypothesizes that the time T (in minutes) is given by $T = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + r$, where r is a residual term that accounts for other factors.

T	x_1	x_2	x_3
181	13.1	619	23
193	13.5	803	42
212	13.8	207	31
221	13.1	409	38
248	12.5	482	45

(a) Determine the least squares estimates for the coefficients $\alpha_0, \dots, \alpha_3$ from the available data.

```
def func(a, x1, x2, x3):
  return (a[0] + a[1] * x1 + a[2]*x2 + a[3]*x3)

xdata = np.array([[1,1,1,1,1], [13.1, 13.5, 13.8, 13.1, 12.5],
[619, 803, 207, 409, 482], [23, 42, 31, 38, 45]]).T
ydata = np.array([181, 193, 212, 221, 248])


a = np.linalg.lstsq(xdata, ydata,rcond=None)[0]

# For validation
for i in range (5):
  print(func(a, xdata[i][1], xdata[i][2], xdata[i][3]))
```

$\alpha_0 = 4.92044229e + 02$
$\alpha_1 = -2.34354608e + 01$
$\alpha_2 = -7.61338932e - 02$
$\alpha_3 = 1.86243954e + 00$

(b) Estimate the expected marathon time for a 43-year-old runner of height 74 in., weight 180 lbs., who has run 450 miles during the previous eight weeks.

(c) Your instructor (height 68 in., weight 138 lbs, and 32 years old) wants to qualify for the Boston Marathon this year. How many miles should he run in an eight week period before the marathon to qualify for the Boston marathon? (In his age group, the cutoff for qualification is 3 hours 5 minutes).

```python
def func(a, x1, x2, x3):
  return (a[0] + a[1] * x1 + a[2]*x2 + a[3]*x3)

def coeff(a, age, height, weight, mile):
  x1 = height/ weight**(1/3)
  x2 = mile
  x3 = age
  return func(a, x1, x2, x3)

# Part b
age = 43
height = 74
weight = 180
mile = 450

ans = coeff(a, age, height, weight, mile)
print(ans)
```

Answer = 230.7208607989702

For part (c), feel free to use a different person (such as yourself, someone you admire, or a famous mathematician) than the instructor. The qualification times are available at this link: `https://www.baa.org/2019-boston-marathon-qualifier-acceptances`.

```
def func(a, x1, x2, x3):
  return (a[0] + a[1] * x1 + a[2]*x2 + a[3]*x3)

def coeff(a, age, height, weight, mile):
  x1 = height/ weight**(1/3)
  x2 = mile
  x3 = age
  return func(a, x1, x2, x3)

# Part c
age = 30
height = 76
weight = 160
mile = 200

ans = coeff(a, age, height, weight, mile)
print(ans)
```

Answer = 204.61012886187106