Homework 3 Solutions

November 13, 2017

1 Problem 1 Solution

(a)

$$\|T^k x\| \leq \|T\|^k \|x\|$$

Solution: The following was proved in Section 2.4.1 Lemma (1).

$$\|Ax\| \leq \|A\| \cdot \|x\|$$

Given the above, we have:

$$\begin{aligned}
\|T^k x\| &= \|T \times T^{k-1}x\| \leq \|T\| \cdot \|T^{k-1}x\| \\
\|T\| \cdot \|T^{k-1}x\| &= \|T\| \cdot \|T \times T^{k-2}x\| \leq \|T\| \cdot \|T\| \cdot \|T^{k-2}x\| = \|T\|^2 \cdot \|T^{k-2}x\| \\
\|T\|^2 \cdot \|T^{k-2}x\| &= \|T\|^2 \cdot \|T \times T^{k-3}x\| \leq \|T\|^2 \cdot \|T\| \cdot \|T^{k-3}x\| = \|T\|^3 \cdot \|T^{k-3}x\|
\end{aligned}$$

... (we continue on like this until we get)

$$\begin{aligned}
\|T\|^{k-1} \cdot \|T^{k-(k-1)}x\| &= \|T\|^{k-1} \cdot \|T \times T^{k-k}x\| \leq \|T\|^{k-1} \cdot \|T\| \cdot \|T^{k-k}x\| \\
&= \|T\|^k \cdot \|T^{k-k}x\| = \|T\|^k \cdot \|x\|
\end{aligned}$$

Because of transitivity, we can say:

$$\|T^k x\| \leq \|T\|^k \|x\|$$

(b) Show that the Jacobi method can be written as:

$$x^{(k+1)} = Tx^{(k)} + c$$

where $T = D^{-1}(L+U)$ (using the decomposition $A = D - L - U$)

Solution: If $A = D - L - U$, we can write the equation $Ax = b$ as follows:

$$
\begin{aligned}
Ax = b &= (D - L - U)x = b \\
\Rightarrow Dx &= (L+U)x + b \\
\Rightarrow x &= \underbrace{D^{-1}(L+U)}_{T}x + \underbrace{D^{-1}b}_{c}
\end{aligned}
$$

The above equation can be converted into the iteration:

$$x^{(k+1)} = Tx^{(k)} + c$$

(c) Define the error after the k^{th} iteration to be $e^{(k)} = x^{(k)} - x^\star$, where $x^\star$ is the exact solution to the equation $Ax = b$. Show that:

$$e^{(k+1)} = Te^{(k)}$$

Solution: Using the Jacobi iteration $x^{(k+1)} = Tx^{(k)} + c$, the error from $x^{(k)}$ is $e^{(k)} = x^{(k)} - x^\star$. The error at iteration $k+1$ can be found as:

$$e^{(k+1)} = Tx^{(k)} + c - (Tx^\star + c) = T(x^{(k)} - x^\star) = Te^{(k)}$$

(d) If A is diagonally dominant by rows, and T is the matrix defined in (b) above, show that $\|T\|_\infty < 1$.
Solution: Since A is diagonally dominant by rows, for each row, the entry a_{ii} is larger than the sum of the absolute values of all other entries in the respective row. Thus:

$$D^{-1} = \begin{bmatrix} 1/d_{11} & & & \\ & 1/d_{22} & & \\ & & \ddots & \\ & & & 1/d_{nn} \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & & & & & \\ l_{21} & 0 & & & O & \\ l_{31} & l_{32} & 0 & & & \\ l_{41} & l_{42} & l_{43} & 0 & & \\ \vdots & \vdots & \vdots & & \ddots & \\ l_{n1} & n_{n2} & l_{n3} & \dots & l_{n,n-1} & 0 \end{bmatrix}$$

$$U = \begin{bmatrix} 0 & u_{12} & u_{13} & \dots & u_{1n} \\ & & u_{23} & \dots & u_{2n} \\ & & & \dots & u_{3n} \\ & O & & \ddots & u_{n-1,n} \\ & & & & 0 \end{bmatrix}$$

When we perform $D^{-1}(L+U)$, we get:

$$L + U = \begin{bmatrix} 0 & u_{12} & u_{13} & \dots & u_{1n} \\ l_{21} & 0 & u_{23} & \dots & u_{2n} \\ l_{31} & l_{32} & 0 & \dots & u_{3n} \\ \vdots & \vdots & \vdots & & \ddots \\ l_{n1} & n_{n2} & l_{n3} & \dots & 0 \end{bmatrix}$$

$$D^{-1}(L+U) = \begin{bmatrix} 0 & u_{12}/d_{11} & u_{13}/d_{11} & \dots & u_{1n}/d_{11} \\ l_{21}/d_{22} & 0 & u_{23}/d_{22} & \dots & u_{2n}/d_{22} \\ l_{31}/d_{33} & l_{32}/d_{33} & 0 & \dots & u_{3n}/d_{33} \\ \vdots & \vdots & \vdots & & \ddots \\ l_{n1}/d_{nn} & n_{n2}/d_{nn} & l_{n3}/d_{nn} & \dots & 0 \end{bmatrix}$$

Since the values of d are larger than the values of u and the values of l, all the entries in the above matrix will be less than 1. Also, since, for each row, the value of d is larger than the sum of the values u and l, each row's sum will be less than 1 and the maximum of all the absolute row sums will also be less than 1. Q.E.D.

(e) Show that when A is diagonally dominant by rows, then:

$$\|e^{(k)}\|_\infty \leq \|T\|_\infty^k \|e^{(0)}\|_\infty$$

Explain why this result implies that the Jacobi method is guaranteed to converge for matrices that are diagonally dominant with rows.

Solution: We proved from (d) that $\|T\|_\infty < 1$. Thus, $\|T\|_\infty^k < 1$ because when you raise any number $c \mid c < 1$ to a positive power k, c becomes smaller and smaller. From (c), we have that:

$$e^{(k+1)} = Te^{(k)}$$

If we take the infinity norm of each side of the equation, we get:

$$\|e^{(k+1)}\|_\infty = \|Te^{(k)}\|_\infty \leq \|T\|_\infty \cdot \|e^{(k)}\|_\infty \quad \text{(Proved in (a))}$$

From the above statement, we can say that if $k = 0$,

$$\|e^{(1)}\|_\infty \leq \|T\|_\infty \cdot \|e^{(0)}\|_\infty$$

And, if $k = 1$,

$$\|e^{(2)}\|_\infty \leq \|T\|_\infty \cdot \|e^{(1)}\|_\infty \leq \|T\|_\infty^2 \cdot \|e^{(0)}\|_\infty$$

In general, for any $k > 0$,

$$\|e^{(k)}\|_\infty \leq \|T\|_\infty^k \cdot \|e^{(0)}\|_\infty$$

Since $\|T\|_\infty^k < 1$, the value of $\|e^{(0)}\|_\infty$ becomes smaller and smaller when it is multiplied by $\|T\|_\infty^k$. This result implies that Jacobi converges for matrices with diagonally dominant rows because it shows us that every time an iteration is performed, the error becomes smaller and smaller. With enough iterations, the error will converge to 0.

1.1 Breakdown of Points

The Points were broken down as follows:

Part A : 3 points
Part B : 2 points
Part C : 4 points
Part D : 6 Points
Part E : 5 Points

1.2 Common Mistakes

1.2.1 Part A

1. One common mistake in this part was proving $\|T^k x\| \leq \|T^k\| \|x\|$ instead of $\|T^k x\| \leq \|T\|^k \|x\|$. These are two different proofs.

1.2.2 Part D

1. Not being general when proving. Many people used specific examples to prove. This could lead to a loss of generality.

2. Failing to recognize that when dividing the other items in the matrix $L+U$ by the items in D, the row sum will be smaller because the items in D are larger in a diagonally dominant matrix.

2 Problem 2 Solution

The data given was as follows:

t	0.0	1.0	2.0	3.0	4.0	5.0
$p(t)$	1.0	2.7	5.8	6.6	7.5	9.9

We will construct different plots to show the fit to the data with Python:

```python
import numpy as np
import math
import matplotlib.pyplot as plt

x = np.matrix([[0.0],[1.0],[2.0],[3.0],[4.0],[5.0]])
y = np.matrix([[1.0],[2.7],[5.8],[6.6],[7.5],[9.9]])

for i in range(0,6):
    A = np.matrix(np.zeros((6,i+1)))
    for j in range(i+1):
        A[:,j] = np.power(x,j)
    a = np.transpose(A)
    dotprod = np.dot(a, A)
    b = np.dot(a,y)
    inverse = np.linalg.inv(dotprod)
```

```python
    poly = np.dot(inverse, b)
    matrix= np.transpose(poly)
    p = A[:,:i+1] * poly
    plt.plot(x,y,"o",x,p)
    plt.title("Polynomial of Degree %d "%(i))
    plt.show()
```

Listing 1: Python code

The plot show different levels of fitting. Although $n = 5$ provides the highest fit, i.e. the least sum of squared error, it might not be ideal because it will not generalize for new data points. This problem is referred to as “overfitting”.

3 Problem 3 Solution

3.1 Example

Python 2.7.10

```python
import numpy as np
A = np.array([[.16, .1],[.17, .11],[2.02, 1.29]])
b = np.array([[.26],[.28],[3.31]])
print np.linalg.solve(A.transpose().dot(A), A.transpose().
    dot(b))
b = np.array([[.27],[.25],[3.33]])
print np.linalg.solve(A.transpose().dot(A), A.transpose().
    dot(b))
print np.linalg.cond(A.transpose().dot(A))
print np.linalg.cond(A.transpose().dot(A),1)
print np.linalg.cond(A.transpose().dot(A),np.inf)
```

3.2 Part a

The result will be

$$x = \begin{bmatrix} 1. \\ 1. \end{bmatrix}$$

3.3 Part b

The result will be

$$x = \begin{bmatrix} 7.00888731 \\ -8.39566299 \end{bmatrix}$$

3.4 Part c

Following the hint, we use “`numpy.linalg.cond`” to find the condition number. According to 2.4, the condtion number of matrix A is computed by $\kappa(A) = \|A\| \cdot \|A^{-1}\|$. In the example, with the default 2-norm, $\kappa(A) = 1204591.14638$, and with the 1-norm/$\infty$-norm, $\kappa(A) = 1631924.81894$. The large condition number implies that matrix A is ill-conditioned, so even a small number in b may cause a large error in x.

3.5 Common Mistakes

3.5.1 Normal Equations

Some people obtained the wrong output without using the normal equations $\boxed{A^T Ax = A^T b}$ to obtain the least-squares approximation of $Ax \approx b$.

4 Problem 4 Solution

4.1 Part a

Show that the following matrix is singular:

$$A = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \\ 0.7 & 0.8 & 0.9 \end{bmatrix}$$

Describe the set of solutions to the equation $Ax = b$ if

$$\begin{bmatrix} 0.1 \\ 0.3 \\ 0.5 \end{bmatrix}$$

There are a couple different ways to show that the matrix is singular. One way is to show that the deteminant is 0. Instead of using the formula of the determinant, the solution will just show reduced echelon form of A. Note that when doing this by hand you could stop sooner:

$$\begin{bmatrix} 1 & 1.14 & 1.28 \\ 0 & 1 & 2.00 \\ 0 & 0 & 0 \end{bmatrix}$$

If we now expand the determinant of A along r_3, we find that $\det(A) = 0$. This shows that the matrix is singular.

Now we solve $A\boldsymbol{x} = \boldsymbol{b}$ (Here both sides have been multiplied by 10 for simplicity.):

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}$$

Going through the same reductions as above with the b vector attached, the augmented matrix produces:

$$\begin{bmatrix} 1 & 8/7 & 9/7 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 5/7 \\ 1/3 \\ 0 \end{bmatrix}$$

Leaving matrix notation we get the following equations:

$$x + \frac{8}{7}y + \frac{9}{7}z = \frac{5}{7} \tag{1}$$

$$y + 2z = \frac{1}{3} \tag{2}$$

Now solving (2) produces $y = {}^{1}\!/_{3} - 2z$. Plugging this into (1) and solving for x gives us the following set of infinite solutions:

$$\begin{aligned} x &= \frac{1}{3} + t \\ y &= \frac{1}{3} - 2t \\ z &= t \end{aligned}$$

where we take z to be any value. This describes the set of solutions for $A\boldsymbol{x} = \boldsymbol{b}$.

4.2 Part b

If we use Gaussian Elimination with partial pivoting to solve this system using *exact* arithmetic, at what point would the process fail?

Looking back above at the *ref* of A, Gaussian Elimination would fail the moment we had three zeros in the bottom row.

4.3 Part c

Because some of the entries of A are not exactly representable in a binary floating-point system, the matrix is no longer exactly singular when entered into a computer. Thus, solving the system by Gaussian Elimination will not necessarily fail. Compare the computed solution with your description of the solution set in part (a). What is the estimated value of the condition number $\kappa(a)$?

When you write a program to solve this question, you will find that it does indeed find a solution. The solution will be approximately correct due to machine rounding. The condition number is computed to be of magnitude 10^{16}.

4.4 Common Mistakes

4.4.1 Infinite Solutions

Many people stated that there would be infinitely many solutions or no solutions to the equation, but did not extend any further. Since there are indeed infinite solutions, the form of those equations is required.

4.4.2 Not Fully Answering Question

Points were deducted when any of the following were missing: failure location of gaussian elimination, condition number, or comparison of original solution to programmatic solution.

4.4.3 Use of `linalg` Library

Any use of the linalg library to solve the system of equations was not given credit since the question asks to use Gaussian Elimination.