School of Computing
National University of Singapore
Biometrics Course
July 2017

Assignment #3
Deadline: 2:00pm on 24 July 2017

Q1. Let $\mathbf{Q} = \mathbf{I} - (2\mathbf{v}\mathbf{v}^\top)/(\mathbf{v}^\top\mathbf{v})$. (4)

(a). Show that $\mathbf{Q}$ is symmetric.
(b). Show that $\mathbf{Q}$ is orthogonal.
(c). Let $\mathbf{y}$ be any vector orthogonal to $\mathbf{v}$, and $\mathbf{x}$ be any vector. Show that $\mathbf{x} - \mathbf{Q}\mathbf{x}$ is orthogonal to $\mathbf{y}$, i.e. $\mathbf{Q}$ projects through any vector that is orthogonal to $\mathbf{v}$.

Q2. Let $\mathbf{A} = \begin{bmatrix} 1 & 1 & 4 \\ -1 & 2 & 2 \end{bmatrix}$. Determine its four fundamental subspaces, i.e. $Col(\mathbf{A})$, $Col(\mathbf{A}^\top)$, $Null(\mathbf{A})$, $Null(\mathbf{A}^\top)$. Do not use Python or any other computing device. (4)

Q3. Just like for vectors, we may define a norm for matrices. A matrix norm $||\cdot||$ is any function with the following properties:

(i) $||\mathbf{A}|| \geq 0$, with equality iff $\mathbf{A} = \mathbf{0}$.

(ii) $||c\mathbf{A}|| = |c|\ ||\mathbf{A}||$, for any $c \in \mathbb{R}$.

(iii) $||\mathbf{A} + \mathbf{B}|| \leq ||\mathbf{A}|| + ||\mathbf{B}||$.

The common matrix norms are the p-norms:

$$||\mathbf{A}||_p = \max_{\mathbf{x} \neq \mathbf{0}} \frac{||\mathbf{A}\mathbf{x}||_p}{||\mathbf{x}||_p}, \quad p = 1, 2, 3, \ldots \tag{1}$$

That is, $||\mathbf{A}||_p$ measures the "magnifying power" of $\mathbf{A}$: how much $\mathbf{A}$ stretches the non-zero vector $\mathbf{x}$ under multiplication. Note that the right hand side of the above definition is a ratio of vector norms. (4)

(a) Prove that $||\mathbf{A}\mathbf{B}||_p \leq ||\mathbf{A}||_p\ ||\mathbf{B}||_p$

(b) Prove that $||\mathbf{A}||_2 = \sigma_{\max}$, the largest singular value of $\mathbf{A}$. *Hint:* Find the $\mathbf{x}$ that maximizes the ratio in Equation (1).

Q4. The *condition number* of a square matrix $\mathbf{A}$ is defined as: $k(\mathbf{A}) = ||\mathbf{A}||\ ||\mathbf{A}^{-1}||$. If $\mathbf{A}$ is singular, then by definition $k(\mathbf{A}) = \infty$. Since the condition number is defined in terms of matrix norms, there are different types of condition numbers corresponding to the different types of matrix norms. (8)

(a) Show that $k_2(\mathbf{A}) = \sigma_{\max}/\sigma_{\min}$, the ratio of the largest to the smallest singular value. Show also that $k_p(\mathbf{AB}) \leq k_p(\mathbf{A})\ k_p(\mathbf{B})$.

(b) Suppose we are trying to solve a system of linear equations $\mathbf{Ax} = \mathbf{b}$. Let $\Delta\mathbf{x}$ be a small change in $\mathbf{x}$, and $\Delta\mathbf{b}$ be the corresponding small change in $\mathbf{b}$. Thus, $\mathbf{A}(\mathbf{x}+\Delta\mathbf{x}) = \mathbf{b}+\Delta\mathbf{b}$. Another interpretation is that $\Delta\mathbf{b}$ is the error (or uncertainty) in $\mathbf{b}$, while $\Delta\mathbf{x}$ is the corresponding error in the solution $\mathbf{x}$. Prove that:

$$\frac{||\Delta\mathbf{x}||_p}{||\mathbf{x}||_p} \quad \leq \quad k_p(\mathbf{A})\ \frac{||\Delta\mathbf{b}||_p}{||\mathbf{b}||_p}$$

This says that the relative error in $\mathbf{x}$ is upper bounded by the condition number times the relative error in $\mathbf{b}$.

(c) Using Python, solve $\mathbf{Ax} = \mathbf{b}$ when $\mathbf{A} = \begin{bmatrix} 100 & 100 \\ 100 & 100.01 \end{bmatrix}$, and $\mathbf{b} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$. The Numpy functions `linalg.det, linalg.svd, linalg.solve` may be useful here. Read the reference for more info: `docs.scipy.org/doc/numpy/reference/routines.linalg.html`

You now discover that there was a round-off error in $\mathbf{b}$. The accurate value of $\mathbf{b}$ is, in fact, $\begin{bmatrix} 2 \\ 2.0001 \end{bmatrix}$. Re-calculate $\mathbf{x}$. What do you notice?

(d) For the $\mathbf{A}$ above, calculate its determinant and condition number, k_2. Using the result of part (b), explain what happened in part (c).

Q5. (Using SVD for compression.) Stepping gingerly from the spaceship onto Martian soil, you heave a sigh of relief. All these years of training at the School of Cosmology (SOC) has finally paid off: you are on your first mission to Mars. As you survey the barren and rocky landscape, your eyes spot something unusual in the distance. You move closer and discover what looks like a Martian flower. Very excitedly, you whip out your high resolution digital camera and take a few shots of the exotic plant. (8)

Back at the spaceship, you wonder how to transmit the images back to Earth. Each image is large, and because of bandwidth limitations, you can send only a small amount of data at a time. How best to do it? Fortunately, you remember the SVD technique. You decide to use it to transmit an image of the flower progressively: a coarse approximation at first, finer details later.

- In Python, import all the necessary libraries as following:

```
import Image
import ImageOps
import numpy
from numpy import linalg
import matplotlib
from matplotlib import pyplot
```

- Read the image using `flower = Image.open("flower.bmp")`
- To see the image, use: `flower.show()`
- This is an RGB image. Convert it to grayscale: `flower = ImageOps.grayscale(flower)`
- Convert image type to array using:
  `aflower = numpy.asarray(flower) # aflower is unit8`
  `aflower = numpy.float32(aflower)`
- Compute the SVD: `U,S,Vt = linalg.svd(aflower);`
- The singular values in S have been sorted in descending order. Plot it with the command:
  `pyplot.plot(S,'b.')`
  `pyplot.show()`
- Print out the plot and submit it. What do you notice?
- Let $K = 20$. Extract the first K singular values and their corresponding vectors in U and V:
  `K = 20`
  `Sk = numpy.diag(S[:K])`
  `Uk = U[:, :K]`
  `Vtk = Vt[:K, :]`
- Uk, Vk, Sk contain the compressed version of the image. To see this, form the compressed image using:
  `aImk = numpy.dot(Uk, numpy.dot( Sk, Vtk))`
  `Imk = Image.fromarray(aImk)`
  and display it: `Imk.show()`
- Print out a copy of this and submit it.
- Repeat for $K = 50, 100, 200$. Print and submit the compressed images for the four different values of K. Briefly describe what you notice.
- Thus, instead of transmitting the original image, you can transmit Uk, Vk, Sk, which should be much less data than the original. Is it worth transmitting when $K = 200$?

Note: Please read the file `printing.pdf` for printing tips.