Indian Institute of Science, Bangalore
Department of Computational and Data Sciences (CDS)

DS284: Numerical Linear Algebra

Assignment 2 [Posted Sept 12, 2022]

Faculty Instructor: Dr. Phani Motamarri

TAs: Ashish Rout, Dibya Nayak, Gourab Panigrahi, Nikhil Kodali

Submission Deadline: Sept 27, 2022 23:59 hrs | Max Points: 100

Notations: Vectors and matrices are denoted below by bold faced lower case and upper case alphabets respectively.

Problem 1 [20 marks]

Recall in IEEE single precision binary floating point representation, we use 32 bits to represent numbers 1 bit is for sign, 8 bits for the exponent and 23 bits for the mantissa. Using normalized binary scientific notation a floating point number in IEEE single precision can be represented as

$$(-1)^s \times (1.f)_2 \times 2^{(exponent-127)}$$

Here $s = 0$ for positive numbers and $s = 1$ for negative numbers. f represents the bits in mantissa. Note the digit 1 in $1.f$ and is explicitly shown for clarity and all binary representations are normalized to take the form $1.f$. The subscript 2 in the above $1.f$ denotes that we are representing the digits in base 2.

Now, in this exercise we will construct a dummy floating point number system where we use 5 bits of precision to represent numbers. In this simplified floating point number system, let us assume that we are representing numbers such that the exponent field admits values -1, 0 and 1 only. Imagine only positive numbers are represented in this system. Then the normalized binary scientific notation in this toy system would be

$$(1.f)_2 \times 2^{(exponent-1)}$$

Note here we use a biased representation in the exponent field instead of using a separate sign bit for exponent. The value of this bias is 1. Recall our toy floating point system admits -1, 0, 1 in the exponent field. Thus a value of -1 in the exponent field means $exponent = 0$, value of 0 in exponent field means $exponent = 1$, value of 1 in exponent field means $exponent = 2$. In this normalized binary scientific notation, 3 bits are used to store f, 2 bits are used to store the $exponent$.

In this backdrop answer the following questions for the toy floating point system we constructed above:

(a) How many numbers can this toy system describe?

(b) Create a table with 3 columns. First column should contain normalized binary scientific notation of the form

$$(1.f)_2 \times 2^{(exponent-1)}$$

of all the above numbers. (Make sure the numbers you are representing here just use 5 bits to store them). Second column should contain the usual binary representation. Third column should contain decimal representation (base 10 representation). Arrange the numbers in increasing order in the base 10 representation.

(c) From the table above, what is the minimum real number and maximum real number you can represent using our toy floating point number system.

(d) What can you say about absolute gaps between the numbers? Are they constant or do they change with the magnitude of the number you are representing?

(e) What can you say about machine epsilon for our toy floating point system?

(Hint: Pick $\mathbf{x} \in \mathbb{R}$, there exists $\mathbf{x}' \in \mathbb{F}$ such that $\frac{|\mathbf{x}-\mathbf{x}'|}{|\mathbf{x}|} \leqslant \epsilon_{machine}$)

Problem 2 [10 marks]

Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be invertible matrix and consider the solution of the problem $\mathbf{Ax} = \mathbf{b}$ for some given non-zero $\mathbf{b} \in \mathbb{R}^n$.

(a) Derive the relative condition number of the problem of computing $\mathbf{x}$ given $\mathbf{b}$ with respect to perturbations in $\mathbf{b}$.

(b) Find the value of the tight lower bound of the relative condition number obtained in (a).

Problem 3 [15 marks]

Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be invertible matrix and suppose that $\mathbf{x}$ solves $\mathbf{Ax} = \mathbf{b}$ for some given non-zero $\mathbf{b}$. Consider perturbations $\Delta\mathbf{A} \in \mathbb{R}^{n\times n}$ of $\mathbf{A}$ satisfying the following in some given matrix norm induced by the vector norm $\|.\|$,

$$K(\mathbf{A})\|\Delta\mathbf{A}\| < \|\mathbf{A}\|$$

where $K(\mathbf{A})$ is the condition number of $\mathbf{A}$ in the given norm. Consider also some perturbation $\Delta\mathbf{b} \in \mathbb{R}^n$ of $\mathbf{b}$ and let $\mathbf{x} + \Delta\mathbf{x}$ solve

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}$$

Prove that

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leqslant \left[\frac{K(\mathbf{A})}{1 - K(\mathbf{A})\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}}\left[\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|} + \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}\right]\right]$$

Problem 4 [20 marks]

Recall the following from class:

(i) For all $x \in \mathbb{R}$ there exists $|\epsilon| \leqslant \epsilon_{machine}$ such that $fl(x) = x(1+\epsilon)$, where $fl(x)$ denotes floating point representation of x.

(ii) For all $x, y \in \mathbb{F}$ there exists $|\epsilon| \leqslant \epsilon_{machine}$ such that $x \circledast y = (x * y)(1+\epsilon)$ where $*$ denotes one of the operators $+, -, \times, \div$ and let $\circledast$ be its floating point analogue. Note $\mathbb{F}$ is a discrete subset of $\mathbb{R}$ which denote floating point representation of the real numbers.

Each of the following describes an algorithm implemented on a computer satisfying the properties (i) and (ii) described above. State with proper arguments whether the following algorithms are backward stable, stable but not backward stable, or unstable?

(a) Input data, $x \in \mathbb{R}$, computation of $2x$ as $x \oplus x$.

(b) Input data, $x \in \mathbb{R}$, computation of x^2 as $x \otimes x$.

(c) Input data, $x \in \mathbb{R} \setminus \{0\}$, computation of 1 as $x \oslash x$.

(d) Input data, $\mathbf{A} \in \mathbb{R}^{m \times m}$, computation of full SVD of $\mathbf{A}$ i.e., $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ with $\mathbf{U}$ and $\mathbf{V}$ being orthogonal matrices and $\mathbf{\Sigma}$ being diagonal.

- In the above, what would it mean for this algorithm to be backward stable. Verify that this algorithm can not be backward stable.
- Standard algorithms for computing SVD are stable. What does stability mean for such algorithms which compute SVD?

(e) Input data, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$, computation of the inner product $\mathbf{x}^T\mathbf{y}$ as $(x_1 \otimes y_1) \oplus (x_2 \otimes y_2) \oplus (x_3 \otimes y_3) \oplus ...(x_m \otimes y_m)$.

(f) Input data $\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, computation of eigen-values of $\mathbf{A}$ by evaluating the roots of characteristic polynomial.

[Hint: You need to examine the stability by looking at how the eigen-values of perturbed matrix $\mathbf{A} + \delta\mathbf{A}$ can be computed by finding the roots of the corresponding characteristic polynomial]

Problem 5 [20 marks]

This exercise will walk you through the steps in proving the existence of SVD of any rectangular matrix $\mathbf{A}$ of size $m \times n$ with rank r.

(a) Matrices of the form $\mathbf{G} = \mathbf{A}^T\mathbf{A}$ are called Gram matrices where $\mathbf{A} \in \mathbb{R}^{m\times n}$. Show that $\mathbf{x}^T\mathbf{G}\mathbf{x} \geqslant 0 \ \forall \ \mathbf{x} \in \mathbb{R}^n$ and hence show that all eigen-values of $\mathbf{G}$ are non-negative.

(b) Show that $\mathbf{A}$ and $\mathbf{A}^T\mathbf{A}$ have the same rank.

(c) Show that a vector $\mathbf{u}$ of the form $\mathbf{A}\mathbf{v}/\sigma$ $(\sigma > 0)$ is a unit eigen-vector of $\mathbf{A}\mathbf{A}^T$ where $\mathbf{v}$ and σ^2 form the eigen-vector, eigen-value pair of $\mathbf{A}^T\mathbf{A}$.

(d) Note that i^{th} eigen-vector, eigen-value pair of $\mathbf{A}^T\mathbf{A}$ can be written as $\mathbf{A}^T\mathbf{A}\mathbf{v}_i = (\sigma_i^2)\mathbf{v}_i$.

Consider the case of a full rank matrix $\mathbf{A}$ ie. $(\sigma_i > 0 \ \ \forall i)$, if we define a new vector $\mathbf{u}_i = \frac{\mathbf{A}\mathbf{v}_i}{\sigma_i}$, show that $\mathbf{A}$ can be written as $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{m\times m}$ is an orthogonal matrix, $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is diagonal matrix and $\mathbf{V} \in \mathbb{R}^{n\times n}$ is again an orthogonal matrix, $\mathbf{u}_i$ is i^{th} column of $\mathbf{U}$, and $\mathbf{v}_i$ is i^{th} column of $\mathbf{V}$

[Note: In low rank scenario some $\sigma_i = 0$ if other non-zero σ_i are sorted, we can compute $\mathbf{U}$ by adding additional column vectors that span $\mathbb{R}^m$ and add rows of 0-vector to $\mathbf{\Sigma}$.]

Problem 6 [15 marks]

You are one of the scientists working at NASA's Goddard Space Flight Center in Greenbelt, Maryland and have been researching Wide-Field Slitless Spectroscopy to capture galaxy spectra of the distant universe. With the help of NASA's James Webb Space Telescope, you have successfully captured the deepest and sharpest infrared image of the distant universe to date. It is an image of the galaxy cluster SMACS 0723 and has been named Webb's First Deep Field.

Unfortunately, due to some technical difficulties, the space telescope has not been able to transmit full-resolution images to Earth. However, an onboard computer can be programmed remotely from Earth to transmit the image in a compressed format until the difficulties are resolved. The control station on Earth has decided to use SVD to compress the image. As a scientist tasked with programming the onboard computer, think about the following:

(a) How many singular values are required to approximate the image i.e., make it look indistinguishable from the original image? (Hint: Load the image in Python or Matlab or Octave or Julia and the matrix representation of the image will be accessible to you. For $r \times r$ pixel image, the image will have $r \times r \times 3$ matrix entries

with the number 3 corresponding to color depth of the image representing Red, Blue, Green.)

(b) Based on your observation in (a), how many entries need to be transmitted to earth to reconstruct the approximate image as opposed to sending the original image?

[Perform the tasks in a programming environment comfortable to you like Matlab/Octave/Python/Julia. You can use inbuilt functions for computing SVD.]

(c) What is the 2-Norm and Frobenius-Norm error between the matrix representation of the original image and the approximate image obtained for different number of singular values. Check if the following theorems hold for these errors:

For the matrix $\mathbf{A}$ of rank r, with singular values σ_1, σ_2, σ_3, ... σ_r, $\mathbf{A}_v$ is the v-rank approximation of $\mathbf{A}$. ($\mathbf{A}_v = \sum_{i=1}^{v} \sigma_i \mathbf{u}_i \mathbf{v}_i$) such that $1 < v < r$, then: $\|\mathbf{A} - \mathbf{A}_v\|_2 = \sigma_{v+1}$, $\|\mathbf{A} - \mathbf{A}_v\|_F = \sqrt{\sigma_{v+1}^2 + \sigma_{v+2}^2 + ... + \sigma_r^2}$

The image Webb's First Deep Field is as below and also downloadable from Teams assignment page as a png file.