Fundamentals of Data Science
ICME Summer Workshops
Linear Algebra

Day 1 Exercises

Monday, August 2, 2021

Instructors: Professor Margot Gerritsen & Laura Lyman

Throughout these problems, we always assume that matrices and vectors have real-valued entries.

1. *(Solving a system of equations).* Suppose we have the following system of equations,

$$\begin{aligned} 2x_1 + 3x_2 - x_3 &= 4 \\ x_1 + 2x_3 &= 3 \\ 2x_2 + 3x_3 &= 5. \end{aligned}$$

   (a) Let $\vec{\boldsymbol{x}} = (x_1, x_2, x_3)^T$. Write the system of equations in the form $A\vec{\boldsymbol{x}} = \vec{\boldsymbol{b}}$ for a 3×3 matrix A and 3×1 vector $\vec{\boldsymbol{b}}$.
   (b) Solve $A\vec{\boldsymbol{x}} = \vec{\boldsymbol{b}}$ for the 3×1 solution $\vec{\boldsymbol{x}}$. Is this solution unique?
   (c) Let $\vec{\boldsymbol{a}}_1$, $\vec{\boldsymbol{a}}_2$, $\vec{\boldsymbol{a}}_3$ be your 3 columns of A. Verify that

$$A\vec{\boldsymbol{x}} = x_1\vec{\boldsymbol{a}}_1 + x_2\vec{\boldsymbol{a}}_2 + x_3\vec{\boldsymbol{a}}_2.$$

   Hint. This follows directly from matrix-vector multiplication, so you don't need to use the $\vec{\boldsymbol{b}}$ vector.

   Bonus question. Can you show this is true in general for an $n \times n$ matrix A and an $n \times 1$ vector $\vec{\boldsymbol{x}}$?

2. *(Vector norms, induced matrix norms).* In lecture, we saw that the *Euclidean norm* or the *vector 2-norm* $\|\cdot\|_2$ is defined by the formula

$$\|\vec{\boldsymbol{x}}\|_2^2 = \vec{\boldsymbol{x}}^T\vec{\boldsymbol{x}} = x_1^2 + x_2^2 + \cdots + x_n^2$$

for any $n \times 1$ vector $\vec{\boldsymbol{x}}$.
We can actually extend this idea to the norm of a matrix! For an $m \times n$ matrix A, define $\|A\|_2$ as the *induced matrix 2-norm* by the formula

$$\max_{\|\vec{\boldsymbol{x}}\|_2=1} \|A\vec{\boldsymbol{x}}\|_2.$$

Intuitively the matrix 2-norm asks this: If I'm given a vector on the unit circle (since $\|\vec{\boldsymbol{x}}\|_2 = 1$), how big can it make if I multiply by A?

(a) Let A be the matrix

$$\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

for any θ angle. What is $\|A\|_2$?

(b) Let $A = 3I$, where I is the $n \times n$ *identity matrix*

$$I = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$$

with 1s on its diagonal and 0s everywhere else. What is $\|A\|_2$?

Bonus. What about when $A = \alpha I$ for a generic number α?

(c) In fact, there are many norms beyond the usual Euclidean $\|\cdot\|_2$ one! For instance, the vector 1-norm is defined by the formula

$$\|\vec{\boldsymbol{x}}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$$

for any $n \times 1$ vector $\vec{\boldsymbol{x}}$. Here, $|x_i|$ means the *absolute value* of entry x_i.

i. Draw a picture of all the 2×1 vectors $\vec{\boldsymbol{x}}$ such that $\|\vec{\boldsymbol{x}}\|_1 = 1$.
ii. Let $\vec{\boldsymbol{x}} = (1, 2, 3)^T$. Compute $\|\vec{\boldsymbol{x}}\|_1$.
iii. **Bonus.** Now let $\vec{\boldsymbol{x}} = (1, 2, \ldots, n)^T$ for some $n > 1$. What is $\|\vec{\boldsymbol{x}}\|_1$?

Hint. You may need to remember a summation usually taught in calculus. If you haven't seen this in some time, you can read the famous anecdote about Carl Friedrich Gauss as a school boy.

(d) **Bonus.** Similar to before, the *induced matrix 1-norm* $\|A\|_1$ for an $m \times n$ matrix A is

$$\max_{\|\vec{\boldsymbol{x}}\|_1=1} \|A\vec{\boldsymbol{x}}\|_1.$$

Let

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

What is $\|A\|_1$?

3. *(Application! Doc-term matrices).* In today's workshop, Margot introduced the *document-term* or *doc-term* matrix. As a reminder, a *doc-term* matrix A describes which terms in a "word bank" occur in a collection of documents. Here's a tiny example. If our documents $D1$ and $D2$ are

   D1 = `"I love matrices"`   D2 = `"I hate matrices"`

   and our word bank is {I, love, hate, matrices}, then the doc-term matrix would be

	$D1$	$D2$
I	1	1
love	1	0
hate	0	1
matrices	1	1

   Document-term matrices can describe document collections in the traditional sense (books, articles, etc.), but they *also* can be used on web content, where each document corresponds to a webpage! Internet document-term matrices can have millions of rows and billions of columns. In this problem, we will play with a small doc-term matrix.

   Suppose our bots scrape text from 3 webpages. After removing some filler words (e.g. "the", "a", "and", etc.), we find that the following words were mentioned frequently in these webpages:

   `ICME, data, coffee, love, hate, matrices, morning`.

   For simplicity, let's restrict our word bank to only contain these words.

   For the examples thus far, the entries of the doc-term matrices we have seen have only been 0s or 1s. In this example, we define doc-term A to be the $m \times n$ matrix whose (i, j)th entry equals

   $$A_{ij} = \#\text{ of times word } i \text{ was mentioned in page j.}$$

   That is, if words occur more than once in a document, the entries in the document-term matrix will reflect that. In practice, more sophisticated weighting systems can be used to reflect the importance of each term, but for this problem, we simply measure importance by these counts.

   Our bots give us the following doc-term matrix.

$A =$

	Page1	Page 2	Page 3
love	2	0	0
hate	3	0	0
morning	2	0	1
matrices	0	2	4
ICME	0	6	0
coffee	5	1	0
data	1	5	5

(a) Suppose we add a fourth web page that mentions the word "love" 4 times and "data" 3 times. What does A look like now?

(b) For two nonzero vectors $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{y}}$, we learned that the *angle* θ *between them* is given by the formula

$$\cos(\theta) = \frac{\vec{\boldsymbol{x}}^T \vec{\boldsymbol{y}}}{\|\vec{\boldsymbol{x}}\|_2 \|\vec{\boldsymbol{y}}\|_2}.$$

In the workshop, Margot discussed that one way to judge how similar two webpages are is by considering the angle between their page vectors. The closer the vectors are to one another, the more similar they are predicted to be. Which two of the **original 3 sites** do we expect to be the most alike?

Note. As is true for many real world problems, the computations do not reduce to "nice" simplified numbers. Unless you are a champion at estimating square roots in your head, you will need a computer at the end.