Math 5600 Spring 2021 Notes of 1/22/21

Announcements

- As you know, my formal office hours are right before class. I’ll be here at 9:00, but I’ll mute my mike and turn off my camera, to work on other things when nobody is there. If you’d like to talk with me just say something and I’ll activate my connection.

- As discussed, class meetings are being recorded, and (undited and raw) videos will be available on Canvas

- If I forget to turn on recording at the beginning please remind me!
Matrix Multiplication

- let

\[ AB = C \]

where

\[ A \in \mathbb{R}^{m \times p}, \quad B \in \mathbb{R}^{p \times n}, \quad \text{and} \quad C \in \mathbb{R}^{m \times n} \]

- We know that

\[ c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj} \]

- Examples:

\[ \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 10 & 8 \\ 8 & 6 \end{bmatrix} = A \begin{bmatrix} 3 & 2 \\ 1 & 2 \end{bmatrix} = BA \]

\[ AB \neq BA \]

\[ \begin{bmatrix} 2 \times 3 \end{bmatrix} \times \begin{bmatrix} 3 \times 3 \end{bmatrix} = \begin{bmatrix} 2 \times 2 \end{bmatrix} \]

\[ \begin{bmatrix} 3 \times 3 \end{bmatrix} \times \begin{bmatrix} 2 \times 3 \end{bmatrix} = \text{DNE} \]

\[ \begin{bmatrix} 2 \times 3 \end{bmatrix} \begin{bmatrix} 3 \times 2 \end{bmatrix} = \begin{bmatrix} 2 \times 2 \end{bmatrix} \]

\[ \begin{bmatrix} 3 \times 2 \end{bmatrix} \begin{bmatrix} 2 \times 3 \end{bmatrix} = \begin{bmatrix} 3 \times 3 \end{bmatrix} \]
• There is a better way.

\[ A = m \times p \quad B = p \times n \]

\[ C = A \cdot B = m \times n \]

\[ i^{th} \Rightarrow \begin{bmatrix} x & \cdots & x \end{bmatrix} \begin{bmatrix} C_{i1} \\ \vdots \\ C_{in} \end{bmatrix} \]

\[ A \times B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 8 \end{bmatrix} \]

why not
Interpretation of Matrix Product

- $c_{ij}$ is the dot product of the $i$-th row of $A$ and the $j$-th column of $B$:

$$c_{ij} = \text{row}_i(A) \cdot \text{col}_j(B).$$

- The $j$-th column of $C$ equals $A$ multiplied with the $j$-th column of $B$:

$$\text{col}_j(C) = A \text{col}_j(B).$$

- The $i$-th row of $C$ equals the $i$-th row of $A$ multiplied with $B$:

$$\text{row}_i(C) = \text{row}_i(A)B.$$

- $C$ is a sum of rank 1 matrices:

$$A = \sum_{k=1}^{p} \text{col}_k(A)\text{row}_k(B).$$

- Note that column vector times a row vector gives a matrix (of rank 1).
Why do we multiply matrices this way?

• The quick answer: Matrices are linear functions, and matrix multiplication means function composition.
• Everything flows from there!
• A function
  \[ T : \mathbb{R}^n \rightarrow \mathbb{R}^m \]
  is linear if
  \[ T(u + v) = T(u) + T(v) \]
  and
  \[ T(ku) = kT(u) \]
  for all \( u \) and \( v \) in \( \mathbb{R}^n \) and real numbers \( k \).
• Given an \( m \times n \) matrix \( A \) we can define \( T \) by
  \[ T(x) = Ax. \]
• The function
  \[ T : \mathbb{R}^n \rightarrow \mathbb{R}^m \]
  is linear!
• Remarkably, we can also go the other way.
• Suppose \( T \) is a linear function.
• Let \( e_i \) be the \( i \)-th standard basis vector in \( \mathbb{R}^n \), i.e.,
  \[ e_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}. \]
• Then we can write a vector

\[ x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \sum_{i=1}^{n} x_i e_i \]

and

\[ T(x) = T \left( \sum_{i=1}^{\mathbb{N}} x_i e_i \right) = \sum_{i=1}^{\mathbb{N}} x_i T(e_i) = Ax \]

where

\[ A = \begin{bmatrix} T(e_1) & T(e_2) & \ldots & T(e_{\mathbb{N}}) \end{bmatrix} \]
• Example: Rotation in the Plane

\[
T(e_1) = \begin{bmatrix} \cos \alpha \\ \sin \alpha \end{bmatrix}
\]

\[
T(e_2) = \begin{bmatrix} -\sin \alpha \\ \cos \alpha \end{bmatrix}
\]

\[
T \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}
\]
Again: Why do we multiply matrices this way?

- Linear Functions can be composed!
- The composition $f \circ g$ of two linear functions $f$ and $g$ is linear, and its matrix is the product of the matrices of the constituent functions.

$$f \circ g(x) = Cx \quad x \in \mathbb{R}^n \quad Cx \in \mathbb{R}^m$$

$\mathbb{R}^n \xrightarrow{B} \mathbb{R}^p \xrightarrow{A} \mathbb{R}^m$

Exercise: Work out the details!

- Note the switch in the sequence. $B$ comes first in the diagram and second in the product, just like $g$ comes first in the composition.
Eigenvalues and Eigenvectors

- Suppose \( A \in \mathbb{R}^{n \times n} \), \( x \) is a (possibly complex) vector, \( \lambda \) is a (possibly complex) number, and

\[
Ax = \lambda x.
\]

Then \( x \) is an eigenvector of \( A \), and \( \lambda \) is the corresponding eigenvalue.

- “eigen” is the German word for “own”.

- The eigenvalues are the roots of the characteristic polynomial of \( A \):

\[
p(\lambda) = \det(A - \lambda I) = 0
\]

where \( I \) is the \( n \times n \) identity matrix. \( p \) is a polynomial of degree \( n \) with leading term \((-\lambda)^n\) (exercise). The equation (1) is the characteristic equation of the matrix \( A \).
• Computing the eigenvalues by solving the characteristic equation only works for very small matrices (since the roots of a polynomial are very sensitive to small perturbations of the coefficients of the polynomial).

• However, the reverse process, computing the roots of a polynomial by solving a related eigenvalue problem, works very well.

• Suppose that the polynomial $p$ is given by

\[ p(\lambda) = (-1)^n (\lambda^n - \alpha_{n-1}\lambda^{n-1} - \alpha_{n-2}\lambda^{n-2} - \ldots - \alpha_0) \]

• The companion matrix $C$ of $p$ is defined by

\[
C = \begin{bmatrix}
\alpha_{n-1} & \alpha_{n-2} & \ldots & \alpha_1 & \alpha_0 \\
1 & 0 & \ldots & 0 & 0 \\
0 & 1 & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 1 & 0 
\end{bmatrix}
\]

• Exercise: Show that

\[ p(\lambda) = \det(C - \lambda I). \]
The Gershgorin Theorem

- Among all mathematical theorems the Gershgorin Theorem maximizes the ratio of utility and notoriety.


- Suppose $\lambda$ is an eigenvalue of $A$. Then, for some $i$ in $\{1, 2, \ldots, n\}$

$$|\lambda - a_{ii}| \leq \sum_{j=1, j \neq i}^{n} |a_{ij}|.$$  

- In other words all eigenvalues of $A$ lie in circles around the diagonal entries, with the radii being the sum of the absolute values of the off diagonal entries in the corresponding row.
• This is easy to see:

\[
\begin{align*}
A x &= 2 x \\
\max_j |x_j| &= 1 = x_i \\
\sum_{j=1}^{n} a_{ij} x_j &= 2 x_i = 2 \\
\sum_{j=1}^{n} a_{ij} x_j &= 2 x_i - a_{ii} x_i = 2 - a_{ii} \\
|2 - a_{ii}| &= \left| \sum_{j \neq i} a_{ij} x_j \right| \\
&\leq \sum_{j \neq i} |a_{ij}| |x_j| \\
&\leq \sum_{j \neq i} |a_{ij}|
\end{align*}
\]
Floating Point Arithmetic

• The wikipedia article for “floating point arithmetic” is a good source of information on this topic.

Computers are finite

• This implies that only finitely many numbers can be expressed on a computer. Most numbers can’t.

• A floating point number $z$ is of the form

$$z = \pm 0.m_1m_2\ldots m_k \times b^{n-L}$$

where

$$1 \leq n \leq M \quad \text{and} \quad 0 \leq m_i < b, \quad i = 1, \ldots, k$$

for some values of $b$, $k$, $N$, and $M$.

• $b$ is the base, $n$ is the exponent, and $0.m_1m_2\ldots m_k$ is the mantissa or significand. The integer $L$ is the exponent bias.

• Typically the base is 2. The exponent in that case is also expressed as a binary number. The digits and the signs of the mantissa and the exponent each correspond to one bit.

• Most computers use the IEEE 754 standard where a single double precision floating point number (the most common type) occupies 64 bits and the significand has 53 bits. (There are also single and quadruple precision.)

Hewlett Packard calculators use base 10.

In binary systems the leading bit of the mantissa is usually 1, and not stored. It is referred to as the hidden bit.

• Example: (modified IEEE 754). Consider 64 bit floating point numbers of the form:

$$z = \pm 0.1b_2b_3\ldots b_{52} \times 2^{e_1e_2\ldots e_{10}-L}$$
where the $b_i$ and $e_i$ are binary digits and $L = 1023$.

If all the $e_i$ are zero then $z$ is considered zero, if all the $e_i$ are one $z$ is considered to be NaN, not a number.

Note that all floating point numbers are in fact **rational numbers**.

Floating point numbers are bunched at the origin.

- For illustration consider the much smaller set of floating point numbers of the form

$$z = \pm 0.1b_1b_2 \times 2^{e_1e_2e_3-L}$$

where $L = 3$ and, as before, the $b_i$ and $e_i$ are binary digits

- There are six bits and hence 64 possible numbers. They are all rational with a common denominator of 64. Half of these are negatives of the others. The following Table lists the positive numbers in increasing sequence.
<table>
<thead>
<tr>
<th>count</th>
<th>$b_1$</th>
<th>$b_2$</th>
<th>$e_1$</th>
<th>$e_2$</th>
<th>$e_3$</th>
<th>64z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>56</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>96</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>112</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>128</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>160</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>192</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>224</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>256</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>320</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>384</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>448</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>512</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>640</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>768</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>896</td>
</tr>
</tbody>
</table>
Notes

- The (positive) numbers that can be represented in this system range from

\[ z_{\text{min}} = \frac{4}{64} = \frac{1}{16} \]

...to...

\[ z_{\text{max}} = \frac{896}{64} = 56. \]

- The spacing among neighboring numbers starts at $1/64$, and then keeps doubling until it reaches 2.
What is $\sin \left( 10^{20}\pi \right)$?
My first program

- the very first program I ever wrote was meant to compute a certain result for 10 values of the probability

\[ x = 0.1, \ 0.2, \ \ldots, \ 1.0. \]

- The very simple FORTRAN program

```fortran
Do 1 i = 1,10
   x = i/10.0
   write(*,2) x, sin(x)
1 continue
2 format(f5.2,3x,f7.4)
end
```

does just that. It computes the sine function of \( x \) for those values and produces the output

<table>
<thead>
<tr>
<th>( x )</th>
<th>( \sin(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0.0998</td>
</tr>
<tr>
<td>0.20</td>
<td>0.1987</td>
</tr>
<tr>
<td>0.30</td>
<td>0.2955</td>
</tr>
<tr>
<td>0.40</td>
<td>0.3894</td>
</tr>
<tr>
<td>0.50</td>
<td>0.4794</td>
</tr>
<tr>
<td>0.60</td>
<td>0.5646</td>
</tr>
<tr>
<td>0.70</td>
<td>0.6442</td>
</tr>
<tr>
<td>0.80</td>
<td>0.7174</td>
</tr>
<tr>
<td>0.90</td>
<td>0.7833</td>
</tr>
<tr>
<td>1.00</td>
<td>0.8415</td>
</tr>
</tbody>
</table>

- That’s not how I wrote the program, however. I wrote something like

```fortran
x = 0.0
1 continue
x = x + 0.1
write(*,3) x, sin(x)
if (x .eq. 1.0) go to 2
   go to 1
2 continue
3 format(f5.2,3x,f7.4)
end
```
The output from that second program traumatized me for life! No kidding!

- What does the second program do?