PRINCIPLES OF DATA SCIENCE

2021 - 2022

Assoc. Prof. Dr. Salha Alzahrani

Department of Computer Science

CHAPTER 4: BASIC MATHEMATICS

- BASIC SYMBOLS AND TERMINOLOGY
  - Vectors and matrices
  - Arithmetic symbols
    - Summation
    - Proportional
    - Dot product
  - Graphs
  - Logarithms/exponents
  - Set theory
- LINEAR ALGEBRA
  - Matrix multiplication

MATHEMATICS AS A DISCIPLINE

Mathematics as a discipline

- Mathematics, as a science, is one of the oldest known forms of logical thinking by mankind.
- Since ancient Mesopotamia and likely before (3,000 BCE), humans have been relying on arithmetic and more challenging forms of math to answer life’s biggest questions.
- Whether or not you are consciously using the principles of math, the concepts live deep inside everyone's brains. It's my job as a math teacher to get you to realize it.

BASIC SYMBOLS AND TERMINOLOGY

Vectors and matrices

- **A vector** is defined as an object with both magnitude and direction.
- For our purpose, a vector is simply a 1-dimensional array representing a series of numbers.
- In another way, a vector is a list of numbers.
- It is generally represented using an arrow or bold font, as shown:

$$\vec{x} \quad or \quad x$$

- Vectors are broken into components, which are individual members of the vector.
- We use index notations to denote the element that we are referring to, as illustrated:

$$\text{If} \quad \vec{x} = \begin{pmatrix} 3 \\ 6 \\ 8 \end{pmatrix} \quad \text{then} \quad x_1 = 3$$

Vectors and matrices

In math, we generally refer to the first element as index 1, as opposed to computer science, where we generally refer to the first element as index 0. It is important to remember what index system you are using.

- In Python, we can represent arrays in many ways. We could simply use a Python list to represent the preceding array:

```
x = [3, 6, 8]
```

- However, it is better to use the **numpy** array type to represent arrays, as shown, because it gives us much more utility when performing vector operations:

```
import numpy as np
x = np.array([3, 6, 8])
```

Vectors and matrices

- Consider that we measure the average satisfaction rating (0-100) of employees for three departments of a company as being 57 for HR, 89 for engineering, and 94 for management. We can represent this as a vector with the following formula:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 57 \\ 89 \\ 94 \end{pmatrix}$$

- This vector holds three different bits of information about our data. This is the perfect use of a vector in data science.
- You can also think of a vector as being the theoretical generalization of Panda's `Series` object. So, naturally, we need something to represent the `Dataframe`.
- We can extend our notion of an array to move beyond a single dimension and represent data in multiple dimensions.

Vectors and matrices

- A matrix is a 2-dimensional representation of arrays of numbers.
- Matrices (plural) have two main characteristics that we need to be aware of. The dimension of the matrix, denoted as n x m (n by m), tells us that the matrix has n rows and m columns.
- Matrices are generally denoted using a capital, bold-faced letter, such as **X**. Consider the following example:

$$\begin{pmatrix} 3 & 4 \\ 8 & 55 \\ 5 & 9 \end{pmatrix}$$

- This is a 3 x 2 (3 by 2) matrix because it has three rows and two columns.
- If a matrix has the same number of rows and columns, it is called a square matrix.

Vectors and matrices

- Revisiting our previous example, let's say we have three offices in different locations, each with the same three departments: HR, engineering, and management. We could make three different vectors, each holding a different office's satisfaction scores, as shown:
- However, this is unscalable. What if you have 100 different offices? Then we would need to have 100 different 1-dimensional arrays to hold this information.
- This is where a matrix alleviates this problem. Let's make a matrix where each row represents a different department and each column represents a different office, as shown:
- This is much more natural. Now, let's strip away the labels, and we are left with a matrix!

$$x = \begin{pmatrix} 57 \\ 89 \\ 94 \end{pmatrix}, y = \begin{pmatrix} 67 \\ 87 \\ 84 \end{pmatrix}, z = \begin{pmatrix} 65 \\ 98 \\ 60 \end{pmatrix}$$

	Office 1	Office 2	Office 3
HR	57	67	65
Engineering	89	87	98
Management	94	84	60

$$X = \begin{pmatrix} 57 & 67 & 65 \\ 89 & 87 & 98 \\ 94 & 84 & 60 \end{pmatrix}$$

Arithmetic symbols

Summation

- The uppercase sigma symbol Σ is a universal symbol for addition. Whatever is to the right of the sigma symbol is usually something iterable, meaning that we can go over it one by one.
- For example, let's create the representation of a vector: X = [1, 2, 3, 4, 5]
- To find the sum of the content, we can use the following formula:

$$\sum x_i = 15$$

- In Python, we can use the following formula:

```
In [1]:  x = [1,2,3,4,5]
         sum(x)

Out[1]:  15
```

Arithmetic symbols

Proportional

- The lowercase alpha symbol α represents values that are proportional to each other.
- This means that as one value changes, so does the other. Values can either vary directly or indirectly. If values vary directly, they both move in the same direction (as one goes up, so does the other). If they vary indirectly, they move in opposite directions (if one goes down, the other goes up).

- Consider the following examples:
  - The sales of a company vary directly with the number of customers. This can be written as :

    $$Sales\ \alpha\ Customers$$

  - Gas prices vary (usually) indirectly with oil availability, meaning that as the availability of oil goes down, gas prices will go up. This can be denoted as :

    $$Gas\ \alpha\ Oil\ Availability$$

Arithmetic symbols

Dot product

- The dot product is an operator like addition and multiplication. It is used to combine two vectors, as shown:

$$\begin{pmatrix} 3 \\ 7 \end{pmatrix} \cdot \begin{pmatrix} 9 \\ 5 \end{pmatrix} = 3*9+7*5=62$$

- So, what does this mean? Let's say we have a vector that represents a customer’s sentiments towards three genres of movies—comedy, romantic, and action.

[When using a dot product, note that the answer is a single number, known as a scalar.]

Arithmetic symbols

Dot product Example

- Consider that, on a scale of 1-5, a customer loves comedies, hates romantic movies, and is alright with action movies. We might represent this as follows: $\begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix}$
- Now, let's assume that we have two new movies, one of which is a romantic comedy and the other is a funny action movie. The movies would have their own vector of qualities, as shown:

$$m_1 = \begin{pmatrix} 4 \\ 5 \\ 1 \end{pmatrix} \text{ and } m_2 = \begin{pmatrix} 5 \\ 1 \\ 5 \end{pmatrix}$$

Here, m_1 is our romantic comedy and m_2 is our funny action movie

Arithmetic symbols

Dot product Example (cont.)

- In order to make a recommendation, we will apply the dot product between the customer's preferences for each movie. The higher value will win and, therefore, will be recommended to the user.
- Let's compute the recommendation score for each movie. For movie 1, we want to compute:

Customer: M_1

$$\begin{pmatrix}5\\1\\3\end{pmatrix} \cdot \begin{pmatrix}4\\5\\1\end{pmatrix} = \begin{array}{l} (5.4) \rightarrow \text{user loves comedies and this move is funny} \\ + \\ (1.5) \rightarrow \text{user hates romance but this move is romantic} \\ + \\ (3.1) \rightarrow \text{user doesn't mind action and the move is not action packed} \\ \hline 28 \end{array}$$

Arithmetic symbols

Dot product Example (cont.)

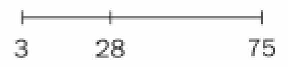
The answer we obtain is *28*, but what does this number mean? On what scale is it? Well, the best score anyone can ever get is when all values are *5*, making the outcome as follows:

$$\begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix} . \begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix} = 5^2 + 5^2 + 5^2 = 75$$

The lowest possible score is when all values are *1*, as shown:

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 1^2 + 1^2 + 1^2 = 3$$

So, we must think about *28* on a scale from *3-75*. To do this, imagine a number line from *3* to *75* and where *28* would be on it. This is illustrated as follows:



Arithmetic symbols

Dot product Example (cont.)

Not that far. Let's try for movie 2:

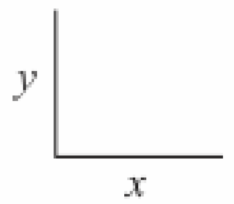
$$\begin{pmatrix}5\\1\\3\end{pmatrix} \cdot \begin{pmatrix}5\\1\\5\end{pmatrix} = (5*5)+(1*1)+(3*5) = 41$$

This is higher than *28*! Putting this number on the same timeline as before, we can also visually observe that it is a much better score, as shown:

So, between movie *1* and movie *2*, we will definitely recommend movie *2* to our user.

Graphs

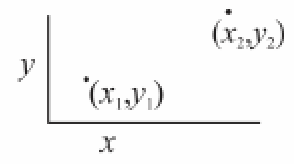
- No doubt you have encountered dozens, if not hundreds, of graphs in your life so far.



- This is a basic Cartesian graph (*x* and *y* coordinate). The *x* and *y* notation are very standard but sometimes do not entirely explain the big picture. We sometimes refer to the *x* variable as being the independent variable and the *y* as the dependent variable.
- This is because when we write functions, we tend to speak about them as being y is a function of x, meaning that the value of y is dependent on the value of x. This is what a graph is trying to show.

Graphs

- Suppose we have two points on a graph, as shown:



We refer to the points as (x_1, y_1) and (x_2, y_2).

- The slope between these two points is defined as follows:

$$slope = m = \frac{y_2 - y_1}{x_2 - x_1}$$

Graphs

- You have probably seen this formula before, but it is worth mentioning if not for its significance. The slope defines the rate of change between the two points.
- Rates of change can be very important in data science, specifically in areas involving differential equations and calculus.
- Rates of change are a way of representing how variables move together and to what degree.
- Consider that we are modeling the temperature of your coffee in relation to the time that it has been sitting out. Perhaps we have a rate of change as follows:

$$-\frac{2\,degrees\ F}{1\,minute}$$

  - *This rate of change is telling us that for every single minute, our coffee's temperature is dropping by two degrees Fahrenheit.*

Logarithms/exponents

- An exponent tells you how many times you have to multiply a number to itself.

$$\underbrace{2}_{\text{base}}^{\overbrace{4}^{\text{exponent}}} = 2 \cdot 2 \cdot 2 \cdot 2 = 16$$

- A logarithm is the number that answers the question: "what exponent gets me from the base to this other number?"

$$\log_{\underbrace{2}_{\text{base}}}(16) = \underbrace{4}_{\text{logarithm}}$$

$$\log_2(16) = 4 \leftrightarrow 2^4 = 16$$

$$\log_3 81 = 4 \text{ because } 3^4 = 81$$
$$\log_5 125 = 3 \text{ because } 5^3 = 125$$

Set theory

- The set theory involves mathematical operations at a set level. It is sometimes thought of as a basic fundamental group of theorems that governs the rest of mathematics. For our purpose, we use the set theory in order to manipulate groups of elements.
- A set is a collection of distinct objects.

```
In [3]:  s = set()
         s = set([1, 2, 2, 3, 2, 1, 2, 2, 3, 2])
         # the set will remove duplicates from a list
         print(s)

         {1, 2, 3}
```

Set theory

Note that, in Python, the curly braces – {, } – can denote either a set or a dictionary.

Remember that a dictionary in Python is a set of key-value pairs, for example:

```
In [5]:  dict = {"dog": "human's best friend", "cat": "destroyer of world"}
         dict["dog"]

Out[5]:  "human's best friend"
```

```
In [6]:  # but if we try to create a pair with the same key as an existing key
         # It will override the previous value
         # dictionaries cannot have two values for one key.
         dict["dog"] = "Arf"
         dict

Out[6]:  {'dog': 'Arf', 'cat': 'destroyer of world'}
```

Set theory

- The **magnitude** of a set is the number of elements in the set and is represented as follows:

$$|A| = magnitude\ of\ A$$

- If we wish to denote that an element is **within** a set, we use the epsilon notation, as shown:

$$2 \in \{1, 2, 3\}$$

- If one set is entirely inside another set, we say that it is a **subset** of its larger counterpart.

$$A = \{1, 5, 6\}, B = \{1, 5, 6, 7, 8\}$$

$$A \subseteq B$$

Set theory

- The **intersection** of two sets is a set whose elements appear in both the sets. It is denoted as shown:

```
user1 = {"Target","Banana Republic","Old Navy"}
# note that we use {} notation to create a set
# compare that to using [] to make a list

user2 = {"Banana Republic","Gap","Kohl's"}
```

$$user1 \cap user2 = \{Banana\ Republic\}$$

$$|user1 \cap user2| = 1$$

- The **union** of two sets is a set whose elements appear in either set. It is denoted as shown:

$$user1 \cup user2 = \{Banana\ Republic, Target, Old\ Navy, Gap, Kohl's\}$$

$$|user1 \cup user2| = 5$$

Set theory

- When looking at the **similarity** between user1 and user2, we should use a combination of the union and the intersection of their sets. user1 and user2 have one element in common out of a total of five distinct elements between them. So, we can define the similarity between the two users as follows:

$$\frac{|user1 \cap user2|}{|user1 \cup user2|} = \frac{1}{5} = .2$$

- In fact, this has a name in the set theory. It is called the **jaccard measure**. In general, for the sets A and B, the jaccard measure (jaccard similarity) between the two sets is defined as :

$$JS(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

- It can also be defined as the magnitude of the intersection of the two sets divided by the magnitude of the union of the two sets.

LINEAR ALGEBRA

Linear algebra

- Remember the movie recommendation engine we looked at earlier? What if we had 10,000 movies to recommend and we had to choose only 10 to give to the user? We'd have to take a dot product between the user profile and each of the 10,000 movies.

- Linear algebra provides the tools to make these calculations much more efficient.
- It is an area of mathematics that deals with the math of matrices and vectors. It has the aim of breaking down these objects and reconstructing them in order to provide practical applications.

- Let's look at a few linear algebra rules before proceeding.

Matrix multiplication

- Like numbers, we can multiple matrices together. Multiplying matrices is, in essence, taking several dot products at once. Let's, for example, try to multiple the following matrices:

$$\begin{pmatrix} 1 & 5 \\ 5 & 8 \\ 7 & 8 \end{pmatrix} \cdot \begin{pmatrix} 3 & 4 \\ 2 & 5 \end{pmatrix}$$

- You can multiple matrices together if the second number in the first dimension pair is the same as the first number in the second dimension pair.

$$3 \times \boxed{2 \cdot 2} \times 2$$

- The resulting matrix will always have dimensions equal to the outer numbers in the dimension pairs (the ones you did not circle in the second point). In this case, the resulting matrix will have a dimension of 3 x 2.

How to multiply matrices

- To multiply matrices, there is actually a quite simple procedure. Essentially, we are performing a bunch of dot products.

$$\begin{pmatrix} 1 & 5 \\ 5 & 8 \\ 7 & 8 \end{pmatrix} \cdot \begin{pmatrix} 3 & 4 \\ 2 & 5 \end{pmatrix}$$

- We know that our resulting matrix will have a dimension of 3 x 2. So, we know it will look something like the following:

$$\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \\ m_{31} & m_{32} \end{pmatrix} \Rightarrow \begin{array}{l} m_{11} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 2 \end{pmatrix} = 13 \\ m_{12} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 5 \end{pmatrix} = 29 \end{array} \cdots \Rightarrow \begin{pmatrix} 13 & 29 \\ 31 & 60 \\ 37 & 68 \end{pmatrix}$$

Movie recommendation example.

- Recall the user's movie genre preferences of comedy, romance, and action, which are illustrated as :

$$U = user\ prefs = \begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix}$$

- Now suppose we have 10,000 movies, all with a rating for these three categories. To make a recommendation, we need to take the dot product of the preference vector with each of the 10,000 movies. We can use matrix multiplication to represent this.

$$M = movies = 3x10,000 \text{ dimension matrix.}$$

- So, now we have two matrices, one is 3 x 1 and the other is 3 x 10,000. We can't multiply these matrices as they are because the dimensions do not work out. We can take the transpose of the matrix (turning all rows into columns and columns into rows).

$$U^T = transpose\, of\, U = (513)$$

$$\begin{pmatrix} & 513 \end{pmatrix} \cdot \begin{pmatrix} 452 & \\ & \cdots \\ 151 & \end{pmatrix}$$

$$1 \times \boxed{3 \qquad 3} \times 10000$$

جامعة الطائف
TAIF UNIVERSITY

Q&A

s.zahrani@tu.edu.sa