Introduction to Matrix Algebra with Python, Numpy, and Scipy

Ryan Pennell

Ivy Tech

contact@ryanpennell.xyz

March 10, 2019

Matrix Addition

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \hdotsfor{5} \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} \end{bmatrix} + \begin{bmatrix} y_{11} & y_{12} & y_{13} & \dots & y_{1n} \\ y_{21} & y_{22} & y_{23} & \dots & y_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{d1} & y_{d2} & y_{d3} & \dots & y_{dn} \end{bmatrix}$$

$$= \begin{bmatrix} x_{11} + y_{11} & x_{12} + y_{12} & x_{13} + y_{13} & \dots & x_{1n} + y_{1n} \\ x_{21} + y_{21} & x_{22} + y_{22} & x_{23} + Y_{23} & \dots & x_{2n} + y_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{d1} + y_{d1} & x_{d2} + y_{d2} & x_{d3} + x_{d3} & \dots & x_{dn} + Y_{dn} \end{bmatrix}$$

Addition Example

Let

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

Then

$$A + B = \begin{bmatrix} 1+4 & 2+3 \\ 3+2 & 4+1 \end{bmatrix} = \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix}$$

Addition Example ctd.

Notice

$$A + B = \begin{bmatrix} 1+4 & 2+3 \\ 3+2 & 4+1 \end{bmatrix} = B + A \begin{bmatrix} 4+1 & 2+3 \\ 2+3 & 4+1 \end{bmatrix} = \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix}$$

Example

```
>>> A = np.matrix('1 2; 3 4')
>>> B = np.matrix('4 3; 2 1')
>>> print(A+B)
>>> print(B+A)
```

$$A + B = B + A$$

Matrix addition and subtraction is commutative. However, the dimensions of the matrices must be equal.

General rules of addition for matrices

Matrix addition and subtraction is commutative. However, the dimensions of the matrices must be equal.
General Rule:

$$\boxed{A + B = \sum_{ij}(a_{ij} + b_{ij})_{ij}}$$

Matrix multiplication

General Rule:

$$(AB)_{ij} = \sum_{k=1}^{n} (a_{ik} b_{kj})$$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

Then

$$AB = \begin{bmatrix} 1 \cdot 4 + 2 \cdot 2 & 1 \cdot 3 + 2 \cdot 1 \\ 3 \cdot 4 + 4 \cdot 2 & 3 \cdot 4 + 4 \cdot 1 \end{bmatrix} = \begin{bmatrix} 8 & 5 \\ 20 & 13 \end{bmatrix}$$

However

$$BA = \begin{bmatrix} 4 \cdot 1 + 3 \cdot 3 & 4 \cdot 2 + 3 \cdot 4 \\ 2 \cdot 1 + 1 \cdot 3 & 2 \cdot 2 + 1 \cdot 4 \end{bmatrix} = \begin{bmatrix} 13 & 20 \\ 5 & 8 \end{bmatrix}$$

So,

$$AB \neq BA$$

This means that matrix multiplication is not commutative.

Example (code)

```
>>> print(A*B)
>>> print(B*A)
```

Dimensions

If A is $n \times m$ and B is $n \times m$ then $A + B$ will be of dimensions $n \times m$. For matrix addition and subtraction the dimensions of A and B have to be the same.

For matrix multiplication the columns of the first matrix have to be equal to the rows of the second. So, if A has dimensions of $n \times m$ then B dimensions of $m \times k$, $m \times n$ or $m \times m$.

Identity and null

Identity matrix is like the number one and can be as large or small as you need it. Always square.

$$I = [1] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \dots & 0 \\ 0 & 1 & \dots \\ 0 & \dots & 1 \end{bmatrix}$$

Zero matrix (does not have to be square).

$$\begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Example (code)

```
>>> np.identity(1)
>>> np.identity(2)
>>> np.identity(3)
>>> np.identity(1000000000)
>>> MemoryError


>>> np.zeros((4))
>>> np.zeros((2,4))
>>> np.zeros((3,4))
>>> np.zeros((4,4))
```

Dividing?

Inverse of a matrix:

$$A \cdot A^{-1} = I = A^{-1} \cdot A$$

example

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$A \cdot A^{-1} = I$$

$$\implies A^{-1} = \begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$$

Example (code)

```
>>>from scipy import linalg
>>>linalg.inv(A)
```

system of equations

$$x + 3y + 5z = 10$$
$$2x + 5y + z = 8$$
$$2x + 3y + 8z = 3$$

Let

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 5 & 1 \\ 2 & 3 & 8 \end{bmatrix} \text{ and } B = \begin{bmatrix} 10 \\ 8 \\ 3 \end{bmatrix}$$

Then

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -9.28 \\ 5.16 \\ 0.76 \end{bmatrix}$$

Example (code)

```
>>> A = np.matrix('1 3 5; 2 5 1; 2 3 8')
>>> B = np.matrix('10; 8; 3')
>>> np.linalg.solve(A, B)
```

Transpose

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, A^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

Example (code)

$>>>$ np.transpose(A)

More Adv. topics???

Determinants
Scalar Multiplication
Eigen-Values
Eigen-Vectors
Cross-Products
Markov-Chains
Stable-Vectors