Data 345

Applied Linear Algebra for Statistical Learning

Class 4 (Sep. 8, 2025)

Matrices

- A **real** $m \times n$ **matrix** A is a two-dimensional array of (real) numbers. Said differently, it is an $m \cdot n$-tuple of numbers a_{ij} (where $1 \leq i \leq m$ and $1 \leq j \leq n$), which is ordered according to a rectangular scheme of m rows and n columns. The set of all real $m \times n$ matrices is sometimes written $\mathbb{R}^{m \times n}$.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad (a_{ij} \in \mathbb{R})$$

- Matrices embody a similar structure to vectors. The **matrix sum** $A + B$ of two like-sized matrices is an element-wise sum, and the **scalar multiple** cA of a matrix multiplies each entry of the matrix by c.

$$A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{bmatrix}$$

$$cA = \begin{bmatrix} ca_{11} & ca_{12} & \cdots & ca_{1n} \\ ca_{21} & ca_{22} & \cdots & ca_{2n} \\ \vdots & \vdots & & \vdots \\ ca_{m1} & ca_{m2} & \cdots & ca_{mn} \end{bmatrix}$$

Matrices

- For an $m \times n$ matrix A, there are m "rows" and n "columns." When we isolate a single row of a matrix, we treat it as a vector called a **row vector**. Similarly, when we isolate a column, we treat it as a vector called a **column vector**.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$A = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_m \end{bmatrix}, \text{ where } \mathbf{r}_i = [a_{i1} \quad a_{i2} \quad \cdots \quad a_{in}]$$

(Each $\mathbf{r}_i$ is a vector of dimension n)

$$A = [\mathbf{c}_1 \quad \mathbf{c}_2 \quad \cdots \quad \mathbf{c}_n], \text{ where } \mathbf{c}_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix}$$

(Each $\mathbf{c}_j$ is a vector of dimension m)

Matrices

- Remember earlier that we defined a vector *dot product*, which is not a true product (because the output is a scalar, not a vector).
- Matrices, however, are different. We can use the vector dot product to define a **matrix product** for two matrices A and B (so that the output of $A \cdot B$ is a matrix).
- Suppose that A is an $m \times n$ matrix, and B is an $n \times k$ matrix. Write A in terms of its row vectors $\mathbf{r}_i$ and B in terms of its column vectors $\mathbf{c}_j$. Define the **matrix product** AB to be the $m \times k$ matrix consisting of numbers d_{ij}, where $d_{ij} = \mathbf{r}_i \cdot \mathbf{c}_j$.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_m \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & b_{22} & \cdots & b_{2k} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nk} \end{bmatrix} = [\mathbf{c}_1 \quad \mathbf{c}_2 \quad \cdots \quad \mathbf{c}_k]$$

$$AB = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1k} \\ d_{21} & d_{22} & \cdots & d_{2k} \\ \vdots & \vdots & & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mk} \end{bmatrix}, \text{ where } d_{ij} = \mathbf{r}_i \cdot \mathbf{c}_j.$$

Matrices

- Example:

$$AB = \begin{bmatrix} 1 & -2 \\ 3 & 6 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} -7.5 & 1 & -11 \\ 25.5 & 15 & -9 \\ 15.5 & 1 & 15 \end{bmatrix}$$

$$\begin{bmatrix} 0.5 \\ 4 \end{bmatrix} \quad \begin{bmatrix} 3 \\ 1 \end{bmatrix} \quad \begin{bmatrix} -7 \\ 2 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & -2 \\ 3 & 6 \\ -1 & 4 \end{bmatrix}, B = \begin{bmatrix} 0.5 & 3 & -7 \\ 4 & 1 & 2 \end{bmatrix}$$

$$d_{11} = (\text{row 1 of } A) \cdot (\text{column 1 of } B) = [1 \quad -2] \cdot \begin{bmatrix} 0.5 \\ 4 \end{bmatrix} = (1 \cdot 0.5) + (-2 \cdot 4) = -7.5$$

$$d_{12} = (\text{row 1 of } A) \cdot (\text{column 2 of } B) = [1 \quad -2] \cdot \begin{bmatrix} 3 \\ 1 \end{bmatrix} = (1 \cdot 3) + (-2 \cdot 1) = 1$$

$$d_{13} = (\text{row 1 of } A) \cdot (\text{column 3 of } B) = [1 \quad -2] \cdot \begin{bmatrix} -7 \\ 2 \end{bmatrix} = (1 \cdot -7) + (-2 \cdot 2) = -11$$

$$\vdots$$

$$d_{33} = (\text{row 3 of } A) \cdot (\text{column 3 of } B) = [-1 \quad 4] \cdot \begin{bmatrix} -7 \\ 2 \end{bmatrix} = (-1 \cdot -7) + (4 \cdot 2) = 15$$

$$\longrightarrow AB = \begin{bmatrix} -7.5 & 1 & -11 \\ 25.5 & 15 & -9 \\ 15.5 & 1 & 15 \end{bmatrix}$$

Matrix Multiplication

- If A is an $m \times k$ matrix and B is a $k \times n$ matrix then AB is an $m \times n$ matrix.
- Because matrix multiplication is defined in terms of a dot product, the two vectors being multiplied must have matching dimensions. So, the dimension of a row vector in A must match the dimension of a column vector in B.
- In other words, the **number of columns in the first matrix must be the same as the number of rows in the second.**
- We cannot define matrix multiplication in the instance that A is an $m \times k$ matrix and B is an $l \times n$ matrix and $k \neq l$.
- For instance, if A is 3×2 and B is 3×2, then AB **is not defined**. So, you cannot necessarily multiply two matrices of the same shape.

$$(3 \times 2 \text{ matrix}) \cdot (3 \times 2 \text{ matrix})$$

Must match

Matrix Multiplication

- In our earlier example, A was a 3×2 matrix and B was a 2×3 matrix. So AB was a 3×3 matrix.
- If we consider the matrix product BA instead, we are multiplying a 2×3 matrix by a 3×2 matrix, so the result is a 2×2 matrix.
- This means matrix multiplication is **automatically prohibited from being commutative**.

```
A = matrix([[1,-2], [3,6], [-1,4]])
B = matrix([[0.5, 3, -7], [4, 1, 2]])
print("AB:", A@B)
print("BA:", B@A)
```

✓ 0.0s Python

```
AB: matrix([-7.5  1.0  -11.0]
        [25.5  15.0  -9.0]
        [15.5  1.0  15.0])
BA: matrix([16.5  -11.0]
        [5.0  6.0])
```

Matrix Multiplication

- If A is a 3×3 matrix and B is a 3×3 matrix, then both AB and BA will be 3×3 matrices as well.
- (Side note: matrices with the same number of rows and columns are said to be **square**.)
- In the case that A and B are same-dimension square matrices, is it true that $AB = BA$?

$$A = \begin{bmatrix} -2 & 4 & -2 \\ 4 & -8 & 3 \\ 1 & -2 & 1 \end{bmatrix} \quad B = \begin{bmatrix} -8 & 3 & -3 \\ -2 & 1 & -1 \\ -2 & 0 & -3 \end{bmatrix}$$

```
A = matrix([[-2, 4, -2], [4, -8, 3], [1, -2, 1]])
B = matrix([[-8, 3, -3], [-2, 1, -1], [-2, 0, -3]])
print("AB:", A@B)
print("BA:", B@A)
```

✓ 0.0s Python

```
AB: matrix([12.0  -2.0  8.0]
        [-22.0  4.0  -13.0]
        [-6.0  1.0  -4.0])
BA: matrix([25.0  -50.0  22.0]
        [7.0  -14.0  6.0]
        [1.0  -2.0  1.0])
```

Other Matrix Things

- Let's suppose A and B are the following:

$$A = \begin{bmatrix} -1 & 4 \\ 4 & -8 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$AB = \begin{bmatrix} (-1 \cdot 1 + 4 \cdot 0) & (-1 \cdot 0 + 4 \cdot 1) \\ (4 \cdot 1 + (-8 \cdot 0)) & (4 \cdot 0 + (-8 \cdot 1)) \end{bmatrix} = \begin{bmatrix} -1 & 4 \\ 4 & -8 \end{bmatrix} = A$$

$$BA = \begin{bmatrix} (1 \cdot (-1) + 0 \cdot 4) & (1 \cdot 4 + 0 \cdot 8) \\ (0 \cdot (-1) + 1 \cdot 4) & (0 \cdot 4 + 1 \cdot (-8)) \end{bmatrix} = \begin{bmatrix} -1 & 4 \\ 4 & -8 \end{bmatrix} = A$$

- In this case, $AB = BA$. This is because B is a special matrix called the **identity matrix**. More generally, the n-dimensional identity matrix is an $n \times n$ matrix of the form:

$$I_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

(I_n is made up of entries e_{ij}, where $e_{ij} = 1$ if $i = j$ and $e_{ij} = 0$ otherwise.)

Other Matrix Things

- Suppose we have a square matrix A in $\mathbb{R}^{n\times n}$. A matrix B in $\mathbb{R}^{n\times n}$ with the property that $AB = I_n = BA$ is called the **inverse** of A, denoted by A^{-1}.

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 4 & 4 & 5 \\ 6 & 7 & 7 \end{bmatrix} \qquad B = \begin{bmatrix} -7 & -7 & 6 \\ 2 & 1 & -1 \\ 4 & 5 & -4 \end{bmatrix}$$

```
A = matrix([[1,2,1], [4,4,5], [6,7,7]])
B = matrix([[-7,-7, 6], [2, 1, -1], [4, 5, -4]])
print("AB:", A@B)
print("BA:", B@A)
```

✓ 0.0s Python

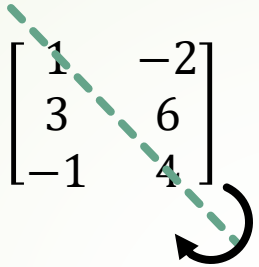
```
AB: matrix([1.0  0.0  0.0]
        [0.0  1.0  0.0]
        [0.0  0.0  1.0])
BA: matrix([1.0  0.0  0.0]
        [0.0  1.0  0.0]
        [0.0  0.0  1.0])
```

- **Note:** if a matrix A has an inverse B, then automatically A is the inverse to the matrix B. In other words, $(A^{-1})^{-1} = A$.

Other Matrix Things

- Now, consider the following two matrices A and B.

$$A = \begin{bmatrix} 1 & -2 \\ 3 & 6 \\ -1 & 4 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 3 & -1 \\ -2 & 6 & 4 \end{bmatrix}$$



- For $A \in \mathbb{R}^{m \times n}$, let $B \in \mathbb{R}^{n \times m}$ be defined as $b_{ij} = a_{ji}$. Then, we call B the **transpose** of the matrix A. We often write this as A^T.
- A matrix A is said to be **symmetric** if $A^T = A$. Naturally, only square matrices may be symmetric.

Goals

- Define a new class <matrix> in Python which takes a list of lists as an input, corresponding to the rows of the matrix. Has a <data> attribute which stores the input and a <shape> attribute which contains the dimensions of the matrix.
- Define a <__repr__> method for <matrix> so that the output is readable.
- Define <__getitem__> and <__setitem__> methods so that we can access/set items by index tuples.
- Define <__add__>, <__mul__>, <__sub__> methods that behave like the operations in <vector>.
- Define a <__matmul__> method which multiplies two matrices of the appropriate size.

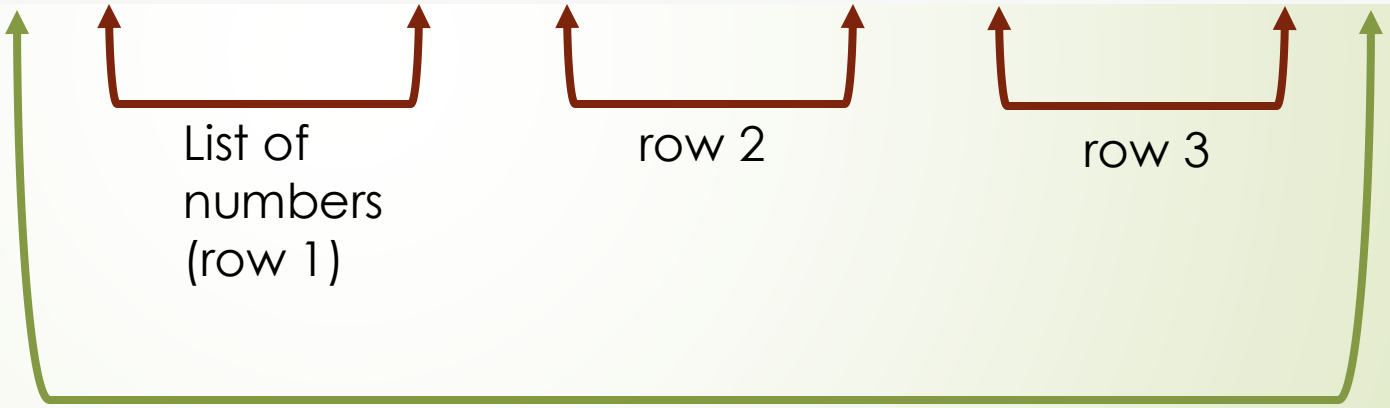
Defining <vector> vs <matrix>

- In defining <vector> we expected a *list* of numbers.

```
self.data = [float(x) for x in data]
```

- Ideally, we would define <matrix> in a similar way. We should pass *multiple* lists, each of which would represent rows in the matrix (a list of lists).

```
A = matrix([  [-2, 4, -2],   [4, -8, 3],   [1, -2, 1]  ])
```



First brackets [ ] indicating we are passing a list to our constructor. Rather than a list where the elements are numbers, we are passing a list where each individual element *is also a list*. Those smaller lists should only contain numbers.