

06 Matrices



Class Survey

Fill out the survey at this link! :)

<https://forms.gle/E7sQc2URLSvkJcqu7>

Unit 1: Vectors, Book ILA Ch. 1-5

Unit 2: Matrices, Book ILA Ch. 6-11 + Book IMC Ch. 2

- **06 Matrices**
- 07 Linear Equations
- 08 Linear Dynamical Systems
- 09 Matrix Multiplication
- 10 Matrix Inverse

Unit 3: Least Squares, Book ILA Ch. 12-14 + Book IMC Ch. 8

Unit 4: Eigen-decomposition, Book IMC Ch. 10, 12, 19

Outline: 06 Matrices

- **Matrices**
- **Matrix-vector multiplication**
- **Examples**

Matrices

Definition: A matrix is a rectangular array of numbers, e.g.:

$$A = \begin{bmatrix} 0 & 1 & -2.3 \\ 1.3 & 4 & -0.1 \end{bmatrix}$$

- Its size, or shape, is: (row dimension) x (column dimension).
 - **Example:** Matrix above has size 2 x 3.
- Its elements are called: entries, coefficients.
- $A_{i,j}$ refers to element at i th row and j th column in matrix A .
 - i is the row index and j is the column index.

In **Python**, we use `numpy` and `np.array` to build matrices. The shape of the matrix can be accessed via the function `shape`.

In [7]:

```
import numpy as np

A = np.array([
    [0, 1, -2.3],
    [1.3, 4, -0.1]
])
print(np.shape(A)); print(A.shape)
v = np.array([1, 2, 3, 4]); print(len(v))
```

```
(2, 3)
(2, 3)
4
```

In **Python**, we can access the elements of the matrix.

In [18]:

```
v = [1.1, 2.2, 3.3, 4.3]; #print(len(v))

for v_index in range(len(v)):
    print(v[v_index])

for v_value in v:
    print(v_value)
```

```
1.1
2.2
3.3
4.3
```

In [29]:

```
print(A); print(A[0][2]); print(A[0, 2])
# Implement code that prints all entries of the matrix A
print("Begin for loop:")
print(A.shape)
n_rows = A.shape[0]; n_columns = A.shape[1]
n_rows, n_columns = A.shape

for i in range(n_rows):
    for j in range(n_columns):
        print(A[i, j])
```

```
[[ 0.   1.  -2.3]
 [ 1.3  4.  -0.1]]
-2.3
-2.3
Begin for loop:
(2, 3)
0.0
1.0
-2.3
```

```
1.3
4.0
-0.1
```

Sizes/Shapes of Matrices

Definitions: A $m \times n$ matrix A is:

- tall if $m > n$,
- wide if $m < n$,
- square if $m = n$.

In [15]:

```
A = ...
```

Matrices, Vectors and Scalars

Definitions:

- A 1×1 matrix is a number or scalar.
- A $n \times 1$ matrix is an n -vector.
- A $1 \times n$ matrix is a n -row-vector.

Starting now, we will distinguish vectors and row vectors.

Columns and rows of a matrix

Notations: Take A a $m \times n$ matrix with entries A_{ij} for $i = 1, \dots, m$ and $j = 1, \dots, n$.

- Its j th column is the m -vector:

$$\begin{bmatrix} A_{1j} \\ \vdots \\ A_{mj} \end{bmatrix}$$

- Its i th row is the n -row-vector: $[A_{i1}, \dots, A_{in}]$.

Slices of a matrix

Definition The slice of matrix $A_{p:q,r:s}$ is the matrix:

$$\begin{bmatrix} A_{pr} & A_{p,r+1} & \dots & A_{ps} \\ \dots & \dots & \dots & \dots \\ A_{qr} & A_{q,r+1} & \dots & A_{qs} \end{bmatrix}$$

In Python, we can extract rows, columns and slices:

```
In [9]: """Extract rows, columns and slices from matrix A."""
```

```
Out[9]: 'Extract rows, columns and slices from matrix A.'
```

Block matrices

Definition: A matrix A composed from other matrices is called a block matrix:

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix}$$

where B, C, D, E are called submatrices or blocks of A .

Column/row representations

Notations: Take A a $m \times n$ matrix with entries A_{ij} for $i = 1, \dots, m$ and $j = 1, \dots, n$.

- A is the block matrix of its columns a_1, \dots, a_n :
 - $A = [a_1 \dots a_n]$
- A is the block matrix of its rows b_1, \dots, b_m :

- $A = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}.$

Examples in ECE and beyond

- Images: A_{ij} is intensity value at i, j .

[In Python](#), we can plot an image represented by a matrix.

```
In [20]: ...
```

```
Out[20]: Ellipsis
```

Examples in ECE and beyond

- Weather: A_{ij} is rainfall data at location i on day j .
- Finances: A_{ij} is the return of asset i in period j .

Exercise: In each of these, what do the rows and columns mean?

Special Matrices

Definition: The $m \times n$ zero matrix (resp. ones-matrix) is the matrix with all entries equal to 0 (resp. to 1).

Definition: The identity matrix I is the square matrix with $I_{ii} = 1$ and $I_{ij} = 0$ if $i \neq j$, for example:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

In Python:

```
In [11]: """Code zeroes matrices, ones matrices and the identity matrix"""  
Out[11]: 'Code zeroes matrices, ones matrices and the identity matrix'
```

Diagonal Matrices

Definition: A diagonal matrix A is a square matrix with $A_{ij} = 0$ for $i \neq j$.

- $\text{diag}(a_1, \dots, a_n)$ denotes the diagonal matrix with $A_{ii} = a_i$:

$$\text{diag}(0.2, -3, 1.2) = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 1.2 \end{bmatrix}$$

In Python:

```
In [12]: np.diag([0.2, -3, 1.2])  
Out[12]: array([[ 0.2,  0. ,  0. ],  
                [ 0. , -3. ,  0. ],  
                [ 0. ,  0. ,  1.2]])
```

Triangular Matrices

Definition: A lower triangular matrix A is a matrix such that $A_{ij} = 0$ for $i < j$. An upper triangular matrix A is a matrix such that $A_{ij} = 0$ for $i > j$.

Example: $\begin{bmatrix} 0.2 & 1.2 & 10 \\ 0 & -3 & 0 \\ 0 & 0 & 1.2 \end{bmatrix}$ (upper-triangular)

Transpose

Definition: The transpose of an $m \times n$ matrix A is written A^T and is defined by:

$$(A^T)_{ij} = A_{ji}, \quad i = 1, \dots, n \quad j = 1, \dots, m$$

Example: $\begin{bmatrix} 0.2 & 1.2 & 10 \\ 0 & -3 & 0 \end{bmatrix}^T = \begin{bmatrix} 0.2 & 0 \\ 1.2 & -3 \\ 10 & 0 \end{bmatrix}$

In Python:

```
In [10]: """Transpose a vector or a matrix."""
```

```
Out[10]: 'Transpose a vector or a matrix.'
```

Addition, Substraction and Scalar Multiplication

Just like vectors:

- we can add or subtract matrices of the same size,
- we can multiply a matrix by a scalar.

Property: The transpose verifies:

- $(A^T)^T = A$.
- $(A + B)^T = A^T + B^T$.

Matrix norm

Definition: For a $m \times n$ matrix A , we define the matrix norm as:

$$\|A\| = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2}.$$

Remark: This definition agrees with the definition of norm of vectors when $n = 1$ or $m = 1$.

Exercise: Compute the matrix norm of $A = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}$.

In Python, there are many ways to find the norm of a matrix.

```
In [14]: ...
```

```
Out[14]: Ellipsis
```

Remark: As opposed to vectors which had 1 axis, matrices have several axes. Observe below how `numpy` functions adapt to the case of several axes.

```
In [13]: ...
```

```
Out[13]: Ellipsis
```

Distance between two matrices

Definition: The distance between two matrices A and B is defined as:

$$\text{dist}(A, B) = \|A - B\|.$$

Remark: This means that the clustering algorithm, which only needs a notion of "distance", works on the matrices.

Outline: 06 Matrices

- [Matrices](#)
- **Matrix-vector multiplication**
- [Examples](#)

Matrix-vector multiplication

Definition: The matrix-vector multiplication y of $m \times n$ matrix A and n -vector x is denoted $y = Ax$ and is defined as:

$$y_i = A_{i1}x_1 + \dots + A_{in}x_n, \quad i = 1, \dots, m$$

Exercise: Let A be the $n \times n$ identity matrix and x an n -vector. Compute Ax .

Exercise: Let A be the $n \times n$ zeroes matrix and x an n -vector. Compute Ax .

Exercise: Compute the matrix-vector multiplication of:

$$A = \begin{bmatrix} 0 & 2 & -1 \\ -2 & 1 & 1 \end{bmatrix}; \quad v = \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix}$$

In Python:

In [5]:

```
...
```

Out[5]:

Ellipsis

Exercise: Take $m \times n$ matrix A , and one-hot vector e_i for some i in $1, \dots, m$. Compute Ae_i .

Exercise: Take $m \times n$ matrix A , and one-hot vector e_i for some i in $1, \dots, n$. Compute $A1_n$.

- [Matrices](#)
- **Matrix-vector multiplication**
- **Examples**

Math operations as matrices

Example: The $(n - 1) \times n$ difference matrix:

$$D = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}$$

allows us to compute the $(n - 1)$ vector of differences between consecutive entries of x :

$$Dx = \begin{bmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{bmatrix}$$

In Python, let be given a vector listing the days at which earthquakes have happened in California. Compute the vector giving the number of days in-between earthquakes.

In []:

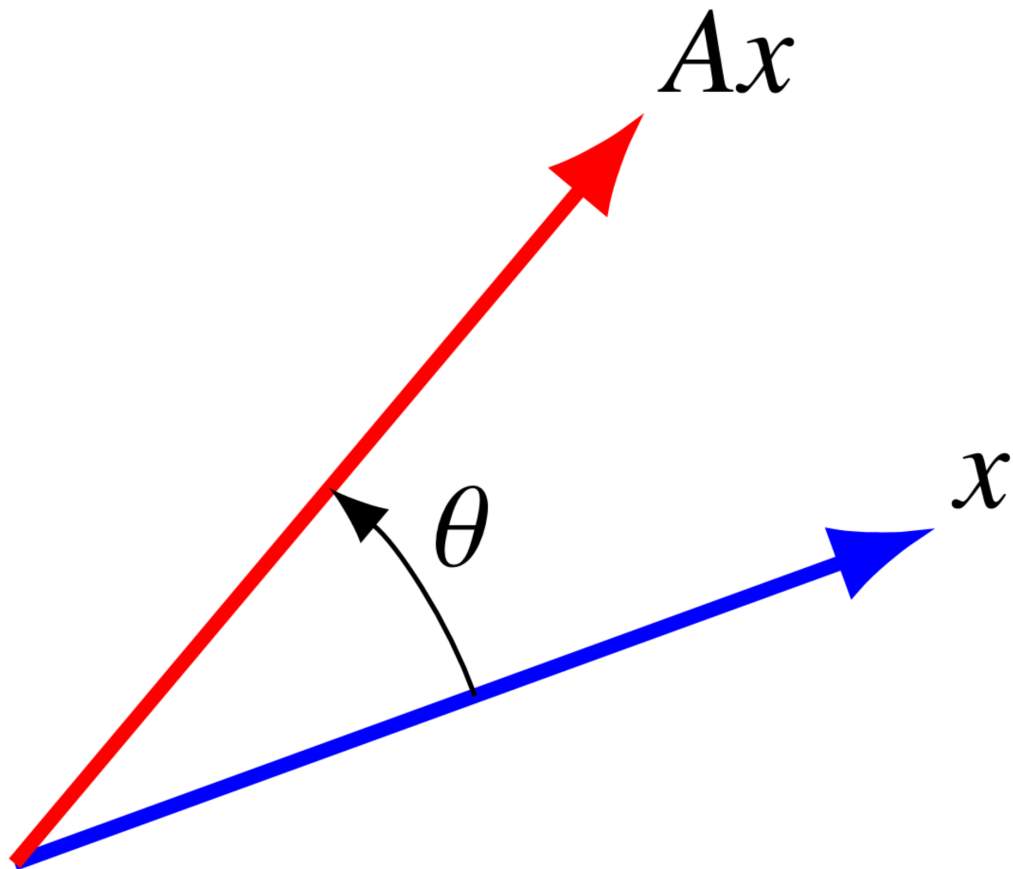
...

Geometric transformations as matrices

Many geometric transformations and mappings of 2-D and 3-D vectors can be represented as matrices A . They can transform vectors x through matrix-vector multiplication $y = Ax$.

Example: For example, the rotation by angle θ can rotate the 2D-vector x through the rotation matrix R_θ :

$$R_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad y = R_\theta x$$



In Python:

In [32]:

...

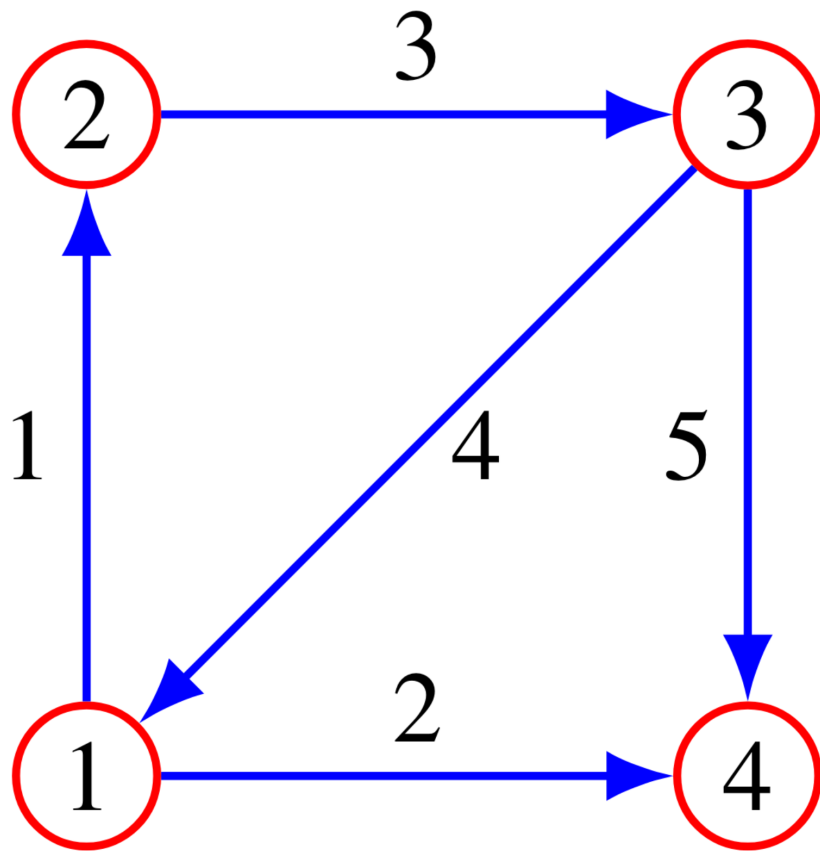
Out[32]:

Ellipsis

(Social) graphs as matrices

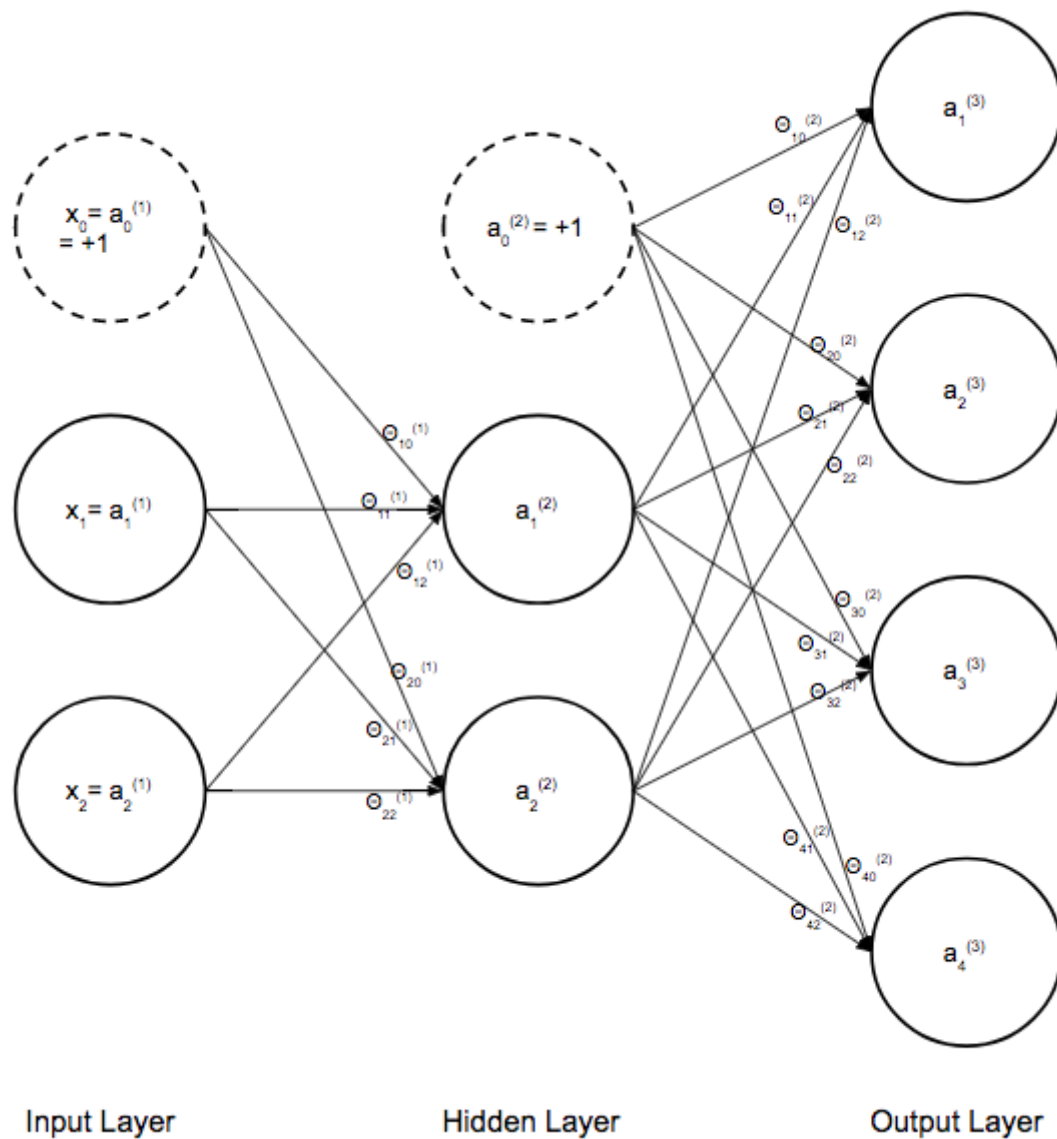
Any (social network) graph with n nodes and m edges can be represented by $n \times m$ matrix A defined defined as:

$$A_{ij} = \begin{cases} 1 & \text{if edge } j \text{ points to node } i \\ -1 & \text{if edge } i \text{ points to node } j \\ 0 & \text{otherwise} \end{cases}$$



Exercise: Compute the incidence matrix associated to this graph.

Deep Learning and Convolutions



By writing $a^{(1)}$ the vector of the first layer, and $a^{(2)}$ the vector of the second layer, we see that:

$$a^{(2)} = W a^{(1)},$$

where W_{ij} is the weight on the edge going from the j th element of the first layer $a_j^{(1)}$ to the i th element of the second layer $a_i^{(2)}$.

In Python, we can implement the layer of a neural network.

```
In [33]: def simple_nn(input_x):
...     
```

Outline: 06 Matrices

- [Matrices](#)
- [Matrix-vector multiplication](#)
- [Examples](#)

Resources: ILA, Ch. 6, 7.

In []: