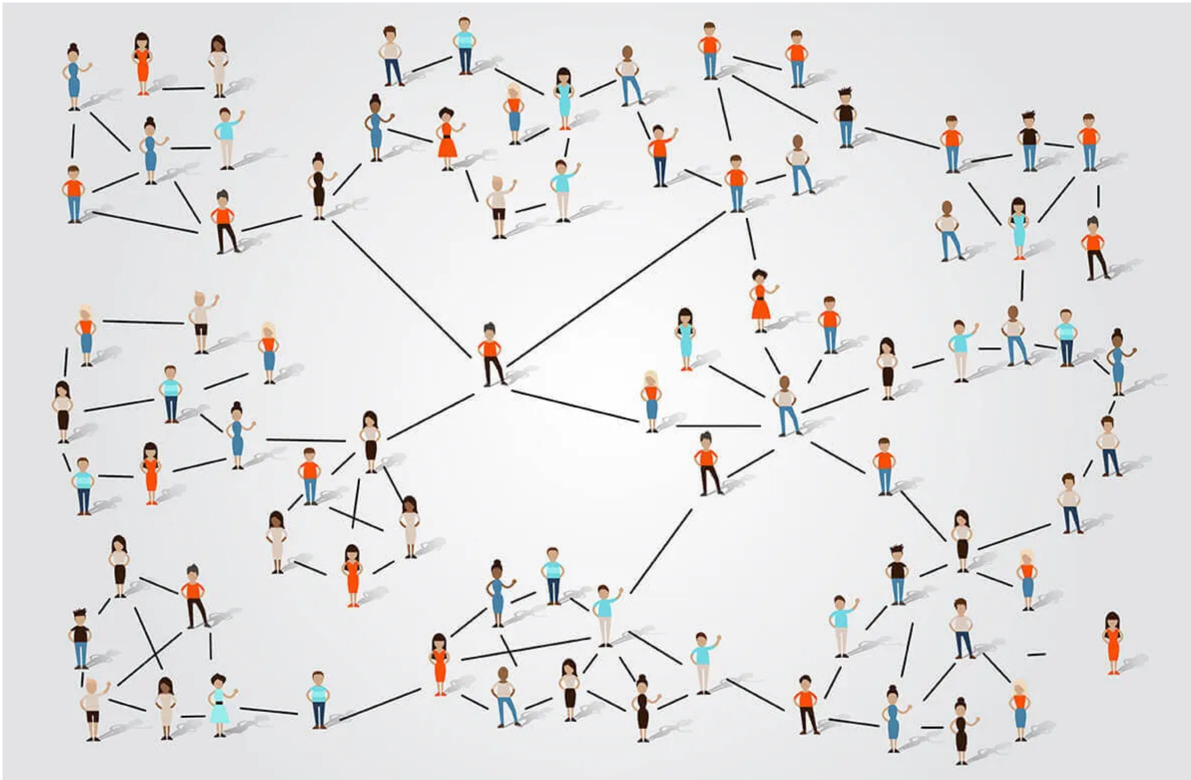
09 Matrix Multiplication



Six degrees of separation theory

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: 09 Matrix Multiplication

Matrix Multiplication

Definition: Consider $m \times p$ matrix A and $p \times n$ matrix B. The matrix multiplication, written $C = AB$ is defined as:

$$C_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj} \text{ for } i = 1, \ldots m, j = 1, \ldots n$$

Special Cases of Matrix Multiplication

For α a scalar, x, y vectors, and A a matrix:

- vector-scalar product, i.e. $x\alpha$ (note that α is on the right!)
- inner-product $x^T y$
- matrix vector multiplication Ax

Definition: The outer product of m-vector x and n-vector y is defined as xy^T, and is a special case of matrix multiplication.

Properties

Properties: For matrices A, B, C and identity matrix I:

- Associativity: $(AB)C = A(BC)$
- Distributivity: $A(B + C) = AB + AC$
- $(AB)^T = B^T A^T$
- $AI = A$ and $IA = A$.

Important remark: $AB = BA$ does NOT hold in general. We say that the matrix multiplication is NOT commutative.

Exercises

Exercise: Let A be a matrix and I be the identity matrix. Show that $AI = A$.

Exercise: Given m-vector x and n-vector y, write the outer product xy^T using the entries of x and y.

In Python, we us `np.matmul` or `@` to compute the matrix multiplication. Verify that the matrix multiplication is generally NOT communitative.

In [9]:

```
import numpy as np
A = np.array([[1, 2],[-3, 0]])
B = np.array([[3, 1], [7, 9]])
print(np.matmul(A, B))
```

```
print(A @ B)
print(B @ A)
```

```
[[17 19]
 [-9 -3]]
[[17 19]
 [-9 -3]]
[[  0   6]
 [-20  14]]
```

Inner-product interpretation

Properties: Consider matrices A, B with a_i^T the rows of A nd b_j the columns of B. We have:

$$AB = \begin{bmatrix} a_1^T b_1 & a_1^T b_2 & \dots & a_1^T b_n \\ a_2^T b_1 & a_2^T b_2 & \dots & a_2^T b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_m^T b_1 & a_m^T b_2 & \dots & a_m^T b_n \end{bmatrix}$$

The matrix product gathers the inner-products of rows of A and columns of B.

Building Matrices from Matrices

Definition: Let A be an $m \times n$ matrix with columns $a_1, .., a_n$. The Gram matrix G of A is:

$$G = A^T A = \begin{bmatrix} a_1^T a_1 & a_1^T a_2 & \dots & a_1^T a_n \\ a_2^T a_1 & a_2^T a_2 & \dots & a_2^T a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n^T a_1 & a_n^T a_2 & \dots & a_n^T a_n \end{bmatrix}$$

Properties:

- The Gram matrix gives all inner products of columns of A
- If $G = I$ then the columns of A are orthonormal.

Outline: 09 Matrix Multiplication

- Matrix Multiplication
- **Composition of linear functions**
- Matrix powers
- QR factorization

Definition: Consider $f : \mathbb{R}^p \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$. The function $h : \mathbb{R}^n \to \mathbb{R}^m$ defined as: $h(x) = f(g(x))$ for all $x \in \mathbb{R}^n$ is called the composition of f and g.

Definition: Assume functions f, g are linear functions, thus can be written as: $f(u) = Au$ and $g(x) = Bx$ for n-vector x and p-vector u. Then the composition h can be written as: $h(x) = (AB)x$.

Remark: This means that:

- the composition of linear functions is a linear function
- the associated matrix is product of matrices of the functions.

Example: Second Difference Matrices

Example: Consider D_n the $(n-1) \times n$ difference matrix such that:

$$D_n x = (x_2 - x_1, \ldots, x_n - x_{n-1}),$$

and D_{n-1} the $(n-2) \times (n-1)$ different matrix such that:

$$D_{n-1} y = (y_2 - y_1, \ldots, y_{n-1} - y_{n-2}).$$

Then $\Delta = D_{n-1} D_n$ is called the $(n-2) \times n$ second difference matrix:

$$\Delta x = (x_2 - 2x_1 + x_3, \ldots, x_n - 2x_{n-1} + x_n).$$

Compute Δ for $n = 5$.

Outline: 09 Matrix Multiplication

Matrix Power

Definition: The square of a matrix A, written A^2 is defined as: $A^2 = AA$.

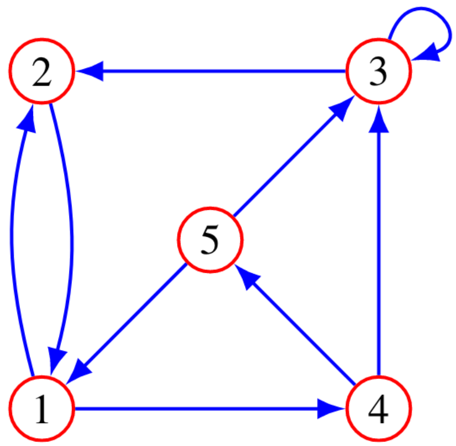
Definition: For a positive integrer k, the kth power of a matrix A, written A^k is defined as: $A^k = A \ldots A$ with k matrices A multiplied. By convention, $A^0 = I$ the identity matrix.

Property: For integers k, l we have: $A^k A^l = A^{k+l}$.

Remark: We will see negative integers later; fractional integers will be seen in other courses.

Example: Directed graph

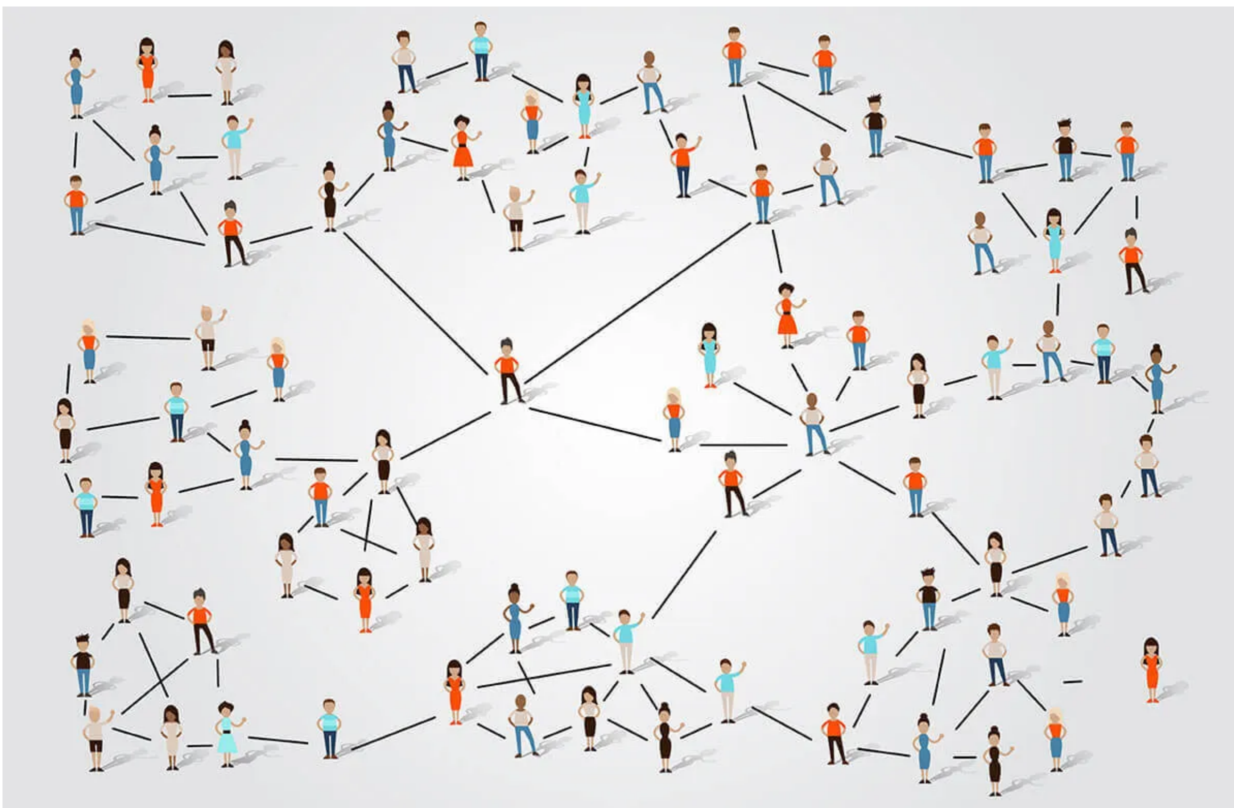
Example: Consider the adjacency matrix associated to the directed graph:



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- $(A^2)_{ij}$ is the number of paths of length 2 going from j to i,
- $(A^l)_{ij}$ is the number of paths of length l going from j to i.

Example: Give an idea on how to prove the six degrees separation theory.



Example: Consider an epidemic modeled by the SIR linear dynamical model with dynamic matrix A. What is $A^k x_0$ where x_0 is the initial state?

Outline: 09 Matrix Multiplication

QR Factorization

Proposition: Any matrix A can be decomposed into two matrices Q, R such that $A = QR$ and:

- Q is a matrix such that: $Q^T Q = I$
- R is a upper triangular matrix.

Remark: We will not learn how to compute it manually. We will rather learn how to compute it in Python, and use it in practice... for example to solve any type of linear equations!

In Python, the QR decomposition of a matrix A can be computed using `np.linalg.qr` :

In [20]:

```
A = np.array([
    [1, 2, 3],
    [4, 5, 6]
])
```

Outline: 09 Matrix Multiplication

- Matrix Multiplication
- Composition of linear functions
- Matrix powers
- **QR factorization**

Resources: Book ILA Ch. 10