

MASSIVE GRAPH MANAGEMENT & ANALYTICS

PRELIMINARIES

Nacéra Seghouani

Computer Science Department, CentraleSupélec
Laboratoire Interdisciplinaire des Sciences du Numérique, LISN
nacera.seghouani@centralesupelec.fr

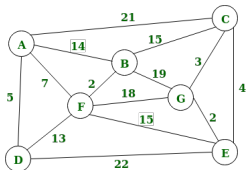
2024-2025

GRAPH THEORY PRELIMINARIES

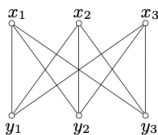
Graph Typology

☞ $\mathcal{G}(V, E)$, V set of vertices, $E = \{(v_i, v_j) | v_i, v_j \in V\}$ set of edges, $|V| = n, |E| = m$

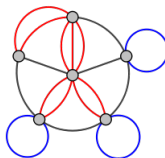
- ✓ **Undirected** - edge: symmetric pair of vertices - **Directed** - edge: asymmetric pair of vertices -
- ✓ **weighted** vertex $w_v : V \rightarrow \mathbb{R}$ or edge $w_e : E \rightarrow \mathbb{R}$
- ✓ **labeled** vertex $w_v : V \rightarrow \mathbb{L}$ or edge $w_e : E \rightarrow \mathbb{L}$
- ✓ **Bipartite** - $V = V_1 \cup V_2$, $E = \{(v_i, v_j) | v_i \in V_1, v_j \in V_2\}$ - Generalization to **k-partite**
- ✓ **Multigraph** or **Multidigraph** - $r : E \rightarrow V_i, v_j \in V$ where r assigns to each $e \in E$ a pair of vertices -
- ✓ **Hypergraph** - edge: relates a subset of vertices -
- ✓ **Complete graph**: $\forall (v_i, v_j) \in V \times V, (v_i, v_j) \in E$



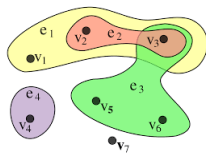
weighted graph



bipartite graph



multigraph



hypergraph

Graph Properties

- Let $\mathcal{G}(V, E)$ a directed graph, d_i^+, d_i^- denote resp. the number of edges coming out and coming to v_i . The degree of v_i :

$$d_i = d_i^+ + d_i^-$$

- $\mathcal{N}_i^+, \mathcal{N}_i^-$ denote resp. the set of the successors and predecessors of v_i . The set of the neighbors of v_i :

$$\mathcal{N}_i = \mathcal{N}_i^+ \cup \mathcal{N}_i^-$$

- A (directed) path $(v_i \rightsquigarrow v_j)$ is a sequence of vertices in the graph (v_i, v_k, \dots, v_j) where each consecutive vertices pair $\in E$
- A (directed) cycle is $(v_i \rightsquigarrow v_j = v_i)$
- The length of a path $(v_i \rightsquigarrow v_j)$ is the number of the edges in $(v_i \rightsquigarrow v_j)$.
- A distance between (v_i, v_j) is the shortest path length between (v_i, v_j)

$$\text{dist}(v_i, v_j) = \text{Min}_{v_i \rightsquigarrow v_j} \text{length}(v_i \rightsquigarrow v_j)$$

Graph Properties

- ☞ The eccentricity ecc of v : the greatest distance between v and any other vertex;

$$ecc(v) = \max_{s \in V} dist(v, s)$$

- ☞ The diameter of G is

$$\max_{v, s \in V} dist(v, s)$$

It is also the maximum eccentricity of any v in G

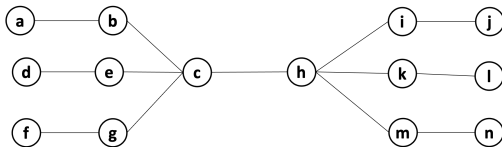
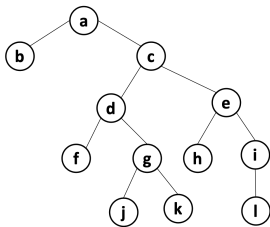
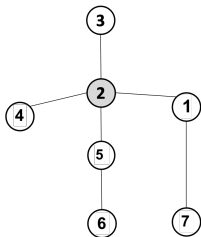
- ☞ The radius of G is the minimum eccentricity of any vertex

$$\min_{v \in V} ecc(v)$$

- ☞ The center of a graph is the set of all vertices of minimum eccentricity, equal to the graph's radius.

Graph Properties

Examples:



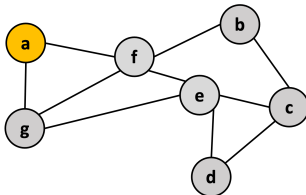
Graph Properties

- ☞ $G'(V, E' \subset E)$ is a partial graph of $G(V, E)$
- ☞ $G'(V' \subset V, E' \subset E)$ is a subgraph of $G(V, E)$
- ☞ $G(V, E)$ is a connected graph $\iff \forall (v_i, v_j) \in V \exists (v_i \rightsquigarrow v_j)$
- ☞ (strongly) connected component of $G(V, E)$ is a subgraph - $G_{cc}(V_{cc}, E_{cc})$ where $\exists (v_i \rightsquigarrow v_j)$, a (directed) path between each v_i and $v_j \in V_{cc}$,
- ☞ $G(V, E)$ is a tree $\iff G$ is a connected graph without cycle \Rightarrow graph with $m = n - 1$ edges
- ☞ $G(V, E)$ is a forest \iff each connected component is a tree

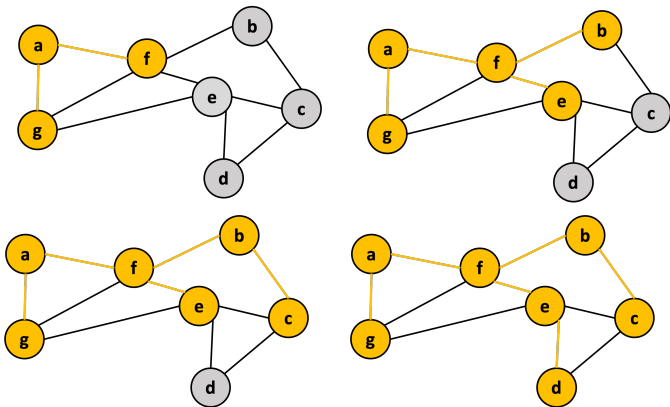
Breadth First Search (BFS)

👉 Queue data structure: an element first added in the list first removed out the list FIFO (First In First Out)

```
1: procedure BFS( $\mathcal{G}(V, E), r$ )
2:    $Q \leftarrow \emptyset$ , enqueue( $Q, r$ ),
3:    $r.label = true$ 
4:   while  $Q \neq \emptyset$  do
5:      $v \leftarrow dequeue(Q)$ 
6:     for  $w \in \mathcal{N}_v$  do
7:       if  $\neg w.label$  then
8:         enqueue( $Q, w$ )
9:          $w.label = true$ 
10:      end if
11:    end for
12:  end while
13: end procedure
```



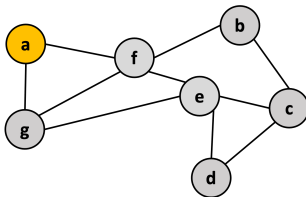
Breadth First Search (BFS)



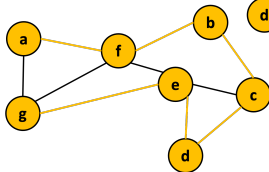
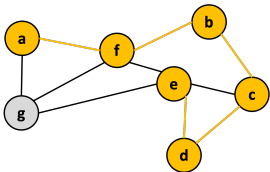
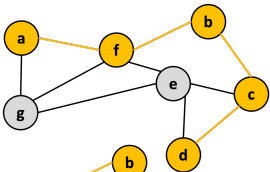
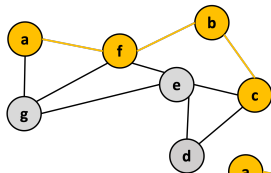
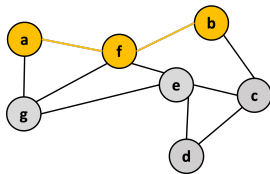
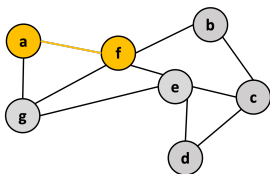
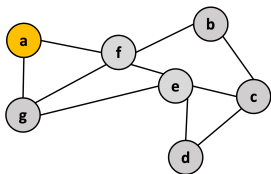
Depth First Search (DFS)

Recursive DFS

```
1: procedure DFS*( $\mathcal{G}(V, E), r$ )  
2:    $r.label = true$   
3:   for  $v \in \mathcal{N}$  do  
4:     if  $\neg v.label$  then  
5:       DFS*( $\mathcal{G}(V, E), v$ )  
6:     end if  
7:   end for  
8: end procedure
```

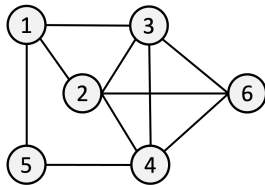


Breadth First Search (BFS)



Depth and Breadth First Search

Another example



Graph Representation using Matrices

☞ $\mathcal{G}(E, V)$ with n vertices and m edges can be encoded using:

- ✓ Adjacency Matrix $\mathbf{A}(n \times n)$, $n = |V|$

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E, \\ 0 & \text{otherwise} \end{cases}$$

Symmetric matrix if \mathcal{G} is an undirected (without loops)

- ✓ Adjacency list \mathbf{L} each vertex holds a list of its neighbours

$$\forall v_i \in V, \mathbf{L}_i = \{v_j | (v_i, v_j) \in E\}$$

If \mathcal{G} is directed the choice of the direction depends on analytic needs

- ✓ Incidence matrix \mathbf{B} , $n \times m$

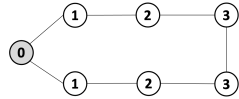
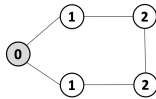
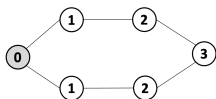
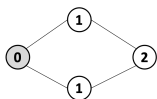
$$\mathbf{B}_{ij} = \begin{cases} 1 & \text{if } e_j = (v_i, v_k) \in E, \\ 0 & \text{otherwise} \end{cases}$$

GRAPH THEORY PRELIMINARIES

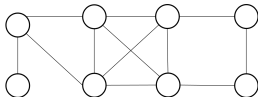
Some Exercises

Exercise: Breadth-First Search and Bipartite graphs

- 1) Using graph traversal algorithms, propose an algorithm that computes the number of edges between a given vertex and all other vertices.
- 2) Given the following cycles with even and odd length (with the distances or depths from the grey vertex), what do you think about the case of graphs with an **odd** cycle (in number of edges)? Is this a characteristic property? State the general case.



- 3) Propose an algorithm that determines if a graph contains an odd cycle.
- 4) In a bipartite graph, can there be a cycle with an odd number of edges? Is this a characteristic property? Justify your answer.
- 5) Propose an algorithm that allows to determine if a graph is bipartite. Test your algorithm on the following graph. Is it bipartite? Justify your answer



Exercise: Breadth-First Search and Bipartite graphs

```
1)
1: procedure BFS_DISTANCE( $\mathcal{G}(V, E), s$ )
2:    $depth \leftarrow \{s : 0\}$ 
3:    $next \leftarrow [s]$ 
4:   while  $len(next) > 0$  do
5:      $n \leftarrow pop(next)$ 
6:     for  $v \in \mathcal{N}_n$  do
7:       if  $v \notin next$  then
8:          $push(v, next)$ 
9:          $depth[v] \leftarrow depth[n] + 1$ 
10:      end if
11:    end for
12:  end while
13:  return depth **works only for connected graphs**
14: end procedure
```

2)

Lemma: A graph contains a cycle C with an odd number of edges iff: $\exists (x, y) \in E \mid depth(x) = depth(y)$

First we know that all edges connect vertices of "neighboring" depths: $\forall (x, y) \in E \mid depth(x) - depth(y) \leq 1$

- \Rightarrow by contrapositive, we suppose that $\forall (x, y) \in C \mid depth(x) \neq depth(y)$, then $\forall (x, y) \in C \mid depth(x) = depth(y) \pm 1$, we will have along the cycle a node of even depth, followed by a node of odd depth and so on and we will not be able to close a cycle of odd size!
- \Leftarrow if there exists an edge $(x, y) \in E$ for which $depth(x) = depth(y)$. We consider the path tree that was used to annotate the depths. In this tree x and y have a first ancestor z in common (possibly the root) from which we can form an odd cycle of size $2(depth(x) - depth(z)) + 1$ by adding the edge (x, y) to this subtree starting at z .

Exercise: Breadth-First Search and Bipartite graphs

```
3)
1: procedure HASODDCYCLE( $(G(V \neq \emptyset, E))$ )
2:    $depth \leftarrow BFS\_depth(G, V[0])$ 
3:   for  $(n_1, n_2) \in E$  do
4:     if  $depth[n_1] == depth[n_2]$  then
5:       return true
6:     end if
7:   end for
8:   return false
9: end procedure
```

4)
⇒ If the graph is bipartite, any path alternates between each vertex of each partition to create a cycle ending by the initial vertex. This means an even number of edges. In a bipartite graph all cycles are pair.

⇐ We just consider the partition of vertices with even depth V_1 on one side and the partition of vertices with odd depth on the other V_2 . We assume that there is no odd cycle then (from question 2): $\forall (x, y) \in E, depth(x) = depth(y) \pm 1$ the all the edges are in $V_1 \times V_2$. The graph is obviously bipartite.

5)
From the previous questions
Bipartite $\Leftrightarrow_{question4}$ no odd cycle
 $\Leftrightarrow_{question2}$ no pair of connected vertices have the same depth

The proposed algorithm selects randomly a vertex to compute the depths. The graph is bipartite iff no pair of connected vertices have the same depth

Exercise: Depth-First Search and 2-colorable graphs

Graph coloring is a way of coloring the vertices of a graph such that no two adjacent vertices share the same color. A 2-colorable graph is a graph that can be colored with only 2 colors.

- 1) What is the link with the previous exercise? Justify your answer.
- 2) We want to write an algorithm, inspired by DFS search which takes as input a graph $G(V, E)$ and which returns a pair `(result, color)` where *result* is *true* if the graph is colorable, *false* otherwise and *color* is a dictionary associating a color 0 or 1 to each vertex. This algorithm should *stop as soon as possible* when the graph is not 2-colorable. Propose an **iterative** version or a **recursive** version.

Exercise: Depth-First Search and 2-colorable graphs

1)

Theorem : a graph is 2-colorable if and only if it is bipartite.

\Rightarrow we partition V into $V_1 \cup V_2$ such that V_1 gathers all white nodes and V_2 all black nodes ($V_1 \cap V_2 = \emptyset$). Any edge connects a white node (in V_1) to a black node (in V_2).

\Leftarrow it is enough to choose the white for V_1 and the black for V_2 and as all the edges are in $V_1 \times V_2$, we are done.

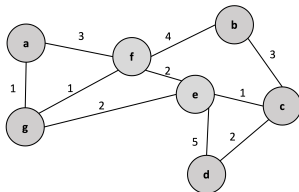
2)

```
1: procedure COLORING2ITER( $(G, s = V[0])$ )
2:    $color \leftarrow \{s : 0\}$ 
3:    $next \leftarrow [s]$ 
4:   while  $len(next) > 0$  do
5:      $n \leftarrow pop(next)$ 
6:     for  $v \in \mathcal{N}_n$  do
7:       if  $v \notin color$  then
8:          $push(v, next)$ 
9:          $color[v] \leftarrow 1 - color[n]$ 
10:      else
11:        if  $color[v] == color[n]$  then
12:          return false, color
13:        end if
14:      end if
15:    end for
16:  end while
17:  return true, color
18: end procedure
```

Exercise: Shortest path

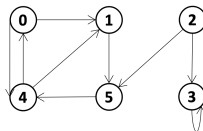
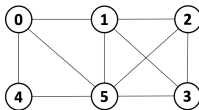
Compute the shortest path using Dijkstra algorithm

```
1: procedure DIJKSTRA( $\mathcal{G}(V, E, W), s$ )  
2:    $dist \leftarrow \{s : 0\}$   
3:    $P \leftarrow \emptyset$   
4:   for  $v \in V \wedge v \neq s$  do  
5:      $dist[v] \leftarrow +\infty$   
6:   end for  
7:    $w \leftarrow \text{select}(v \in V - P \wedge dist[v] = \min_v dist[v])$   
8:    $P \leftarrow P \cup \{w\}$   
9:   for  $v \in \mathcal{N}_w \wedge v \notin P$  do  
10:    if  $dist[w] + w_{(v,w)} < dist[v]$  then  
11:       $predecessor(v) \leftarrow w$   
12:       $dist[v] \leftarrow w_{(v,w)} + dist[w]$   
13:    end if  
14:  end for  
15: end procedure
```



Exercise: Matrix Multiplication & Power

- 1) Give the different representations of these graphs.
- 2) Compute \mathbf{A}^2 , \mathbf{A}^3 . What \mathbf{A}_{ij}^r represents?
- 3) What is the complexity of \mathbf{A}^r , Is it possible to reduce it?



Exercise: Matrix Multiplication & Power

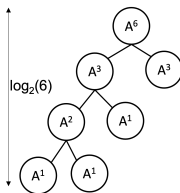
Using the adjacency matrix \mathbf{A} , the number of r -length paths:

$$\mathbf{A}^r = \mathbf{A} \mathbf{A}^{r-1} \forall r > 1$$

$\mathbf{A}_{ij}^r = \sum_k \mathbf{A}_{ik}^{r-1} \mathbf{A}_{kj}$, the number of r -length paths between nodes i and j

Computing \mathbf{A}^r is $O(n^3)$: requires r multiplications, each one $O(n^3)$.

If r is pair $\mathbf{A}^r = (\mathbf{A}^{\frac{r}{2}})^2$, otherwise $\mathbf{A}^r = \mathbf{A}(\mathbf{A}^{\frac{r-1}{2}})^2$. Computing \mathbf{A}^r requires $O(\log_2(r))n^3$ multiplications.



The distance between v_i and v_j is the smallest d such that $\mathbf{A}_{ij}^d \neq 0$

Useful to compute shortest paths, triangles, ...

LINEAR ALGEBRA PRELIMINARIES

Vector Norms

☞ A function f that measures the size of a vector is called a norm.

☞ A vector norm has to satisfy the following:

$$f(\mathbf{x}) = 0 \rightarrow \mathbf{x} = \mathbf{0}$$

$$f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}) \text{ triangle inequality}$$

$$\forall \alpha \in \mathbb{R} \quad f(\alpha \mathbf{x}) = |\alpha| f(\mathbf{x})$$

☞ The general form of vector norm: $\|\mathbf{x}\|_p = \sqrt[p]{\sum_i |x_i|^p}$

☞ The most commonly used is euclidian norm $\|\mathbf{x}\|_2 = \sqrt{\sum_i |x_i|^2}$

Vectors and Matrices

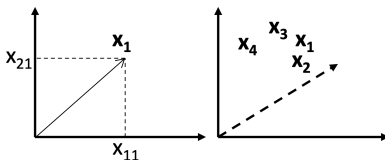
☞ Let \mathbf{x}_i $i \in [1..m]$ be m vectors of n elements:

$$\mathbf{x}_1 = [x_{11} \quad x_{21} \quad \dots \quad x_{n1}], \mathbf{x}_2 = [x_{12} \quad x_{22} \quad \dots \quad x_{n2}], \dots \mathbf{x}_m = [x_{1m} \quad x_{2m} \quad \dots \quad x_{nm}]$$

☞ Matrix representation of size $n \times m$, as a row vector or column vector (transpose)

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_m] = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \dots \\ \mathbf{x}_m \end{bmatrix}^T$$

☞ Each $\mathbf{x}_{i \in [1..m]}$ is n -dimensional point



Matrix Transpose

👉 Transpose of a $n \times m$ matrix \mathbf{X} is a $m \times n$ matrix \mathbf{X}^T : $x_{ij}^T = x_{ji}$

$$\begin{bmatrix} 1 & 2 \end{bmatrix}^T = \begin{bmatrix} 1 \\ 2 \end{bmatrix}; \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}; \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}; \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 3 & 4 & 5 \end{bmatrix}^T = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 3 & 4 & 5 \end{bmatrix}$$

👉 If \mathbf{X} is symmetric $x_{ij} = x_{ji} \Leftrightarrow \mathbf{X} = \mathbf{X}^T$

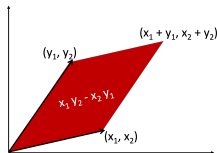
👉 Let \mathbf{X} and \mathbf{Y} be matrices and c be a scalar, some properties:

$$(\mathbf{X}^T)^T = \mathbf{X}; (\mathbf{X} + \mathbf{Y})^T = \mathbf{X}^T + \mathbf{Y}^T; (\mathbf{XY})^T = \mathbf{Y}^T \mathbf{X}^T; (c\mathbf{X})^T = c\mathbf{X}^T$$

Matrix Determinant

- ☞ Let a 2×2 square matrix $\begin{bmatrix} \mathbf{x} & \mathbf{y} \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$, its determinant is a scalar value which characterises some properties of what (\mathbf{x}, \mathbf{y}) represents.

$$\det \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \text{ or } \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} = x_1 y_2 - y_1 x_2 = \|\mathbf{x}\| \cdot \|\mathbf{y}\| \cdot \sin \theta$$



- ✓ The absolute value of the determinant is equal to the area of the parallelogram defined by \mathbf{x}, \mathbf{y}
- ✓ The determinant is $= 0$ if and only if \mathbf{x}, \mathbf{y} are co-linear (line-parallelogram)
- ☞ In the case of 3×3 matrix $(\mathbf{x}, \mathbf{y}, \mathbf{z})$. The absolute value of the determinant is the volume of the parallelepiped defined by $\mathbf{x}, \mathbf{y}, \mathbf{z} = 0$ if they are on the same plan (plat parallelepiped)

$$\det [\mathbf{x}, \mathbf{y}, \mathbf{z}] = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = x_1 \begin{vmatrix} y_2 & z_2 \\ y_3 & z_3 \end{vmatrix} - y_1 \begin{vmatrix} x_2 & z_2 \\ x_3 & z_3 \end{vmatrix} + z_1 \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix}$$

- ☞ Generalisation to vectorial space of dimension n on \mathbb{R} where the vectors define a parallelotope.

Matrix Determinant

☞ Many uses of matrix determinant for ex.:

- ✓ Coefficients in a system of linear equations $\mathbf{Y} = \mathbf{AX}$, can be used to solve these equations (Cramer's rule), other methods computationally much more efficient.
- ✓ Characteristic polynomial of \mathbf{X} , whose roots are the eigenvalues.

☞ Some properties:

$|\mathbf{I}| = 1$ where \mathbf{I} identity matrix (1 on the diagonal and 0 elsewhere);

If $|\mathbf{X}| = 0$ \mathbf{X} is a singular matrix;

$$|\mathbf{XY}| = |\mathbf{X}| |\mathbf{Y}| ; |\mathbf{X}^T| = |\mathbf{X}| ; |\mathbf{A}^n| = |\mathbf{A}|^n$$

Invertible Matrix

☞ A square matrix \mathbf{X} is invertible (non-singular or non-degenerate), its inverse denoted \mathbf{X}^{-1} , if $\exists \mathbf{Y}$ such that $\mathbf{XY} = \mathbf{YX} = \mathbf{I}$

⇔ Its vectors are linearly independent.

⇔ $|\mathbf{X}| \neq 0$

⇔ \mathbf{X}^T is invertible

⇔ 0 is not an eigenvalue (see next)

☞ Other properties:

$$(\mathbf{X}^{-1})^{-1} = \mathbf{X} ; (\mathbf{X}^T)^{-1} = (\mathbf{X}^{-1})^T ; (\mathbf{XY})^{-1} = \mathbf{Y}^{-1}\mathbf{X}^{-1} ; (c\mathbf{X})^{-1} = \frac{1}{c}\mathbf{X}^{-1} \text{ with } c \neq 0 ; |\mathbf{X}^{-1}| = \frac{1}{|\mathbf{X}|} ; |\mathbf{X}^T| = |\mathbf{X}|$$

☞ Analytic inversion of 3×3 matrix using the comatrix transpose

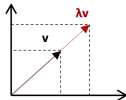
$$\mathbf{X}^{-1} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^{-1} = \frac{1}{|\mathbf{X}|} \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}^T = \frac{1}{|\mathbf{X}|} \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix}$$

$$\begin{array}{lllll} A = (ei - fh) & B = -(di - fg) & C = (dh - eg) & D = -(bi - ch) & E = (ai - cg) \\ F = -(ah - bg) & G = (bf - ce) & H = -(af - cd) & I = (ae - bd) & \end{array}$$

☞ Different methods of matrix inversion: Gaussian elimination, Newton's method, Cayley–Hamilton method
Cholesky decomposition and also Eigen decomposition

Eigenvectors and eigenvalues

- An eigenvector (or characteristic vector) of a linear transformation T is a nonzero vector that changes at most by a scalar factor λ (eigenvalue): $T(\mathbf{v}) = \lambda\mathbf{v}$.



- There is a direct correspondence between $n \times n$ square matrices and linear transformations of an n -dimensional vector space into itself, given any vector space basis.
The T representation of \mathbf{A} $n \times n$ matrix: $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$.
- Finding all eigenvalues: Solving a polynomial function of λ called the characteristic polynomial of \mathbf{A} : $(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = 0$ has a nonzero solution \mathbf{v} iff $|\mathbf{A} - \lambda\mathbf{I}| = 0$
- Example: $\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$; $|\mathbf{A} - \lambda\mathbf{I}| = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = 3 - 4\lambda + \lambda^2 = (\lambda - 1)(\lambda - 3) = 0$

$$v_1 + v_2 = 0 \text{ if } \lambda_1 = 1 \text{ and } -v_1 + v_2 = 0 \text{ if } \lambda_2 = 3$$

The eigenvectors define $eigenSpace(\lambda_1)$ and $eigenSpace(\lambda_2)$ ($t \in \mathbb{R}^*$): $\mathbf{v}_{\lambda_1} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} t$, $\mathbf{v}_{\lambda_2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} t$

- An infinity of collinear eigenvectors (cross product null) for λ_1 and also for λ_2 .
Digression: Strictly speaking, we talk about the eigenvector associated with one given eigenvalue
Note that not all matrices have eigenvalues.

Eigenvectors and eigenvalues

- Algebraic multiplicity t_i of eigenvalue λ_i

$$(\lambda - \lambda_1)^{t_1} (\lambda - \lambda_2)^{t_2} \dots (\lambda - \lambda_k)^{t_k} = 0$$

$\sum_i t_i = n$ where $t_i \in \mathbb{N}^*$ satisfying the algebraic multiplicity of λ_i

- A** can have at most n distinct eigenvalues (complex or real).
- If eigenvalues are distinct eigenvalues of **A**, then the corresponding eigenvectors are linearly independent (non collinear).

Eigenvectors and eigenvalues

☞ Example: $\begin{bmatrix} -1 & 1 & 0 \\ -4 & 3 & 0 \\ 1 & 0 & 2 \end{bmatrix}$

$(\lambda - 1)^2(\lambda - 2) = 0$ has two eigenvalues: $\lambda_1 = 1$ and $\lambda_2 = -2$ with algebraic multiplicities 2 and 1, resp. and \mathbf{v}_{λ_1} is defined by $-2v_1 + v_2 = 0$ and $v_1 + v_3 = 0$

the $\text{eigenSpace}(\lambda_1)$ is $\mathbf{v}_{\lambda_1} = \begin{bmatrix} -t \\ -2t \\ t \end{bmatrix} \forall t \in \mathbb{R}^*$. This set has dimension 1.

☞ Example: $\begin{bmatrix} 4 & 6 & 0 \\ -3 & -5 & 0 \\ -3 & -6 & 1 \end{bmatrix} (\lambda - 1)^2(\lambda + 2) = 0$

$\begin{bmatrix} 2t \\ -t \\ u \end{bmatrix} \forall (t, u) \in \mathbb{R}^*$ $\text{eigenSpace}(\lambda = 1)$ dimension 2. $\begin{bmatrix} -t \\ t \\ t \end{bmatrix} \forall t \in \mathbb{R}^*$ $\text{eigenSpace}(\lambda = -2)$ dimension 1.

☞ The dimension of $\text{eigenSpace}(\lambda_i)$ is referred to as the geometric multiplicity of λ_i . The geometric multiplicity of an eigenvalue is at most its algebraic multiplicity.

Eigenvectors and eigenvalues

What about:

$$\rightarrow \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 1 & -1 & 4 \\ 3 & 2 & -1 \\ 2 & 1 & -1 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 6 & -2 & 2 \\ -2 & 3 & -1 \\ 2 & -1 & 3 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 0 & -1 & -1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

Eigen Decomposition

- Another representation of eigenvalues and eigenvectors

$$\mathbf{A}\mathbf{X} = \mathbf{X}\mathbf{\Lambda}$$

where $\mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_n]$ (each \mathbf{x}_i is an eigenvector) and $\mathbf{\Lambda} = \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ \dots \\ \Lambda_n \end{bmatrix} = \begin{bmatrix} \lambda_{11} & 0 & \dots & 0 \\ 0 & \lambda_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_{nn} \end{bmatrix}$ a diagonal matrix (each λ_{ii} is an eigenvalue).

- A matrix \mathbf{A} is diagonalizable if there exist n linearly independent eigenvectors, i.e., if the matrix \mathbf{X} is invertible:

$$\mathbf{X}^{-1}\mathbf{A}\mathbf{X} = \mathbf{\Lambda}$$

- Leading to the eigen-decomposition of the matrix

$$\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$$

Orthogonal Matrix

☞ A real matrix \mathbf{U} is orthogonal

$$\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I}$$

⇔ \mathbf{U}^T is orthogonal

$$\Leftrightarrow \mathbf{U}^T = \mathbf{U}^{-1}$$

$$\Leftrightarrow |\mathbf{U}| = +1 \text{ or } -1$$

⇔ \mathbf{U} 's eigenvectors are orthogonal, the pairwise dot product is 0), with a norm = 1

⇔ \mathbf{U} is diagonalizable ...

☞ Example

$$\mathbf{I} \text{ (is orthogonal)}, \frac{1}{3} \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & -2 \\ -2 & 2 & -1 \end{bmatrix}; \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}; \text{ (Permutation of coordinates),}$$

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \text{ (rotation), } \begin{bmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{bmatrix} \text{ (reflection)}$$

☞ For complex square matrix we talk about unitary matrix (and conjugate transpose)

Positive (semi-)definite matrices

- ☞ **A** is said to be positive semi-definite when it can be obtained as the product of a matrix by its transpose:
 $\mathbf{A} = \mathbf{X}\mathbf{X}^T$.
- ☞ A positive semi-definite matrix is always symmetric $\mathbf{A}^T = (\mathbf{X}\mathbf{X}^T)^T = \mathbf{A}$. A symmetric matrix **A** is said positive semi-definite if all its eigenvalues are non negative.
- ☞ A positive semi-definite matrix implies:
 - ⇒ $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$, and its eigenvectors are pairwise orthogonal when their eigenvalues are different.
 - ⇒ The eigenvectors are also composed of real values.
 - ⇒ The multiplicity of an eigenvalue λ is the dimension of the space of its eigenvectors $\text{eigenspace}(\lambda)$.
- ☞ Because eigenvectors are orthogonal, it is possible to store all the eigenvectors in an orthogonal matrix. This implies $\mathbf{U}^{-1} = \mathbf{U}^T$ where $\mathbf{U}^T \mathbf{U} = \mathbf{I}$ are the normalized eigenvectors ; if they are not normalized then it is a diagonal matrix.
- ☞ Therefore, the eigen-decomposition of a positive semi-definite matrix **A** could be: $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$
- ☞ Example:
$$\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ \sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ \sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} \end{bmatrix}$$
- ☞ As a consequence, the eigen-decomposition of a positive semi-definite matrix is often referred to as its diagonalization. $\mathbf{\Lambda} = \mathbf{U}\mathbf{A}\mathbf{U}^T$

Positive (semi-)definite matrices - Another definition

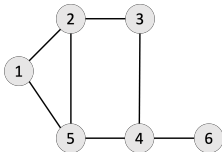
- ✎ A matrix A is said to be positive semi-definite if we observe the following relationship for any non-zero vector \mathbf{x} : $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0 \ \forall \mathbf{x}$.
When > 0 the matrix is positive definite.
When ≤ 0 the matrix is negative semi-definite.
- ✎ **A matrix rank** is the dimension of the vector space generated (or spanned) by its columns. This corresponds to the maximal number of linearly independent columns of A . A matrix whose rank is equal to its size is called a full rank matrix. Only full rank matrices have an inverse.
- ✎ The sum of the eigenvalues of a matrix is the sum of the elements of its main diagonal
- ✎ The product of the eigenvalues is equal to the determinant of the matrix.

Laplacian Matrix

👉 Laplacian Matrix for undirected graph :

$$L_{ij} = \begin{cases} -1, (v_i, v_j) \in E \\ 0, (v_i, v_j) \notin E \\ d_i, i = j \end{cases}$$

or equivalently $L = D - A$ where D is the degree matrix of A where $D_{ii} = \sum_j A_{ij}$



$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

LINEAR ALGEBRA PRELIMINARIES

Some Exercises

Some Exercises

- ☞ what could you say about these matrices: $\begin{pmatrix} -1 & \frac{3}{2} \\ 1 & -1 \end{pmatrix} \begin{pmatrix} -1 & \frac{3}{2} \\ \frac{2}{3} & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- ☞ Show that $\mathbf{A}^n = \mathbf{X}\mathbf{\Lambda}^n\mathbf{X}^{-1}$
- ☞ Find the eigenvalues and unit eigenvectors of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$ with $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ Fibonacci matrix
- ☞ Without multiplying $\mathbf{S} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$ find the determinant, the eigenvalues and eigenvectors, why \mathbf{S} is positive definite
- ☞ For what numbers c and d such that \mathbf{S} and \mathbf{T} are positive definite $\mathbf{S} = \begin{pmatrix} c & 1 & 1 \\ 1 & c & 1 \\ 1 & 1 & c \end{pmatrix} \mathbf{T} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & d & 4 \\ 3 & 4 & 5 \end{pmatrix}$
- ☞ Show if $\lambda_1, \lambda_2, \dots, \lambda_n$ are the eigenvalues of a matrix \mathbf{A} , then \mathbf{A}^m has Eigenvalues $\lambda_1^m, \lambda_2^m, \dots, \lambda_n^m$
- ☞ What is the determinant of any orthogonal matrix ?
- ☞ For an undirected graph both the adjacency matrix and the Laplacian matrix are symmetric. Show that Laplacian is positive semi-definite matrix. Show that Laplacian that 0 is an eigenvalue (the smallest one).
- ☞ Provide a python notebook to compute the main properties of a square matrix.