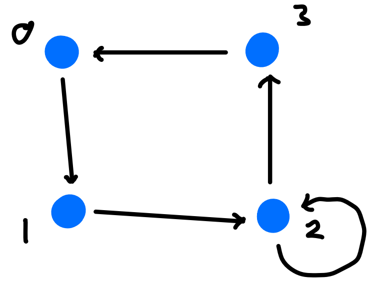
Network Representations and Python

6/9/2022

Agenda:
- Network matrices and network representations
- Python!

We already showed 1 way to represent a network numerically: using an adjacency matrix.

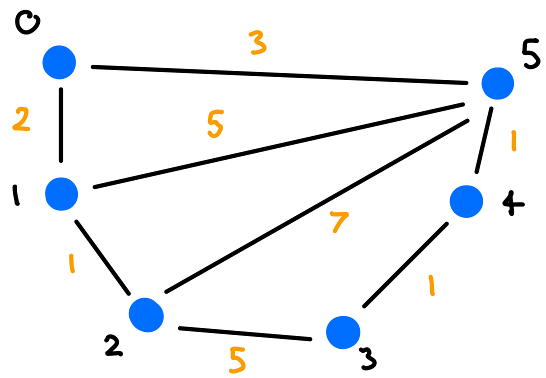
ex)



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

We've seen how to do this for an unweighted graph, or a graph whose edges do not have weights. The opposite kind of graph would be called a weighted graph, a graph where each of the edges has a numerical weight attached to it

ex) Weighted graph



node labels: black
edge weights: orange

Each of the weights are associated with an edge.

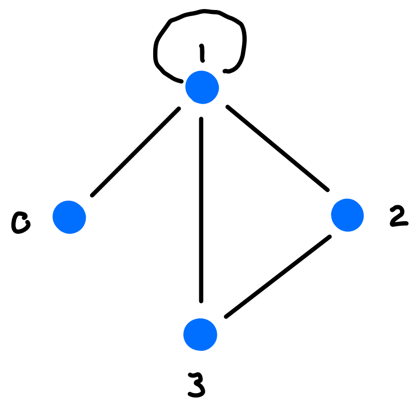
Adjacency matrix

$$A = \begin{bmatrix} 0 & 2 & 0 & 0 & 0 & 3 \\ 2 & 0 & 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 5 & 0 & 7 \\ 0 & 0 & 5 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 3 & 5 & 7 & 0 & 1 & 0 \end{bmatrix}$$

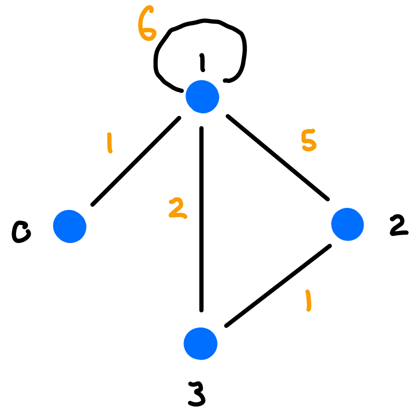
- Degree matrix: Matrix which contains information about the degree of each node (or the number of edges attached to each node).

$$D_{i,j} = \begin{cases} 0 & \text{if } i \neq j \\ \text{degree(node } i) & \text{if } i = j \end{cases}$$

Undirected, unweighted network



Undirected, weighted network



Degree matrix:

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Degree matrix:

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

weighted degree matrix:

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 14 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

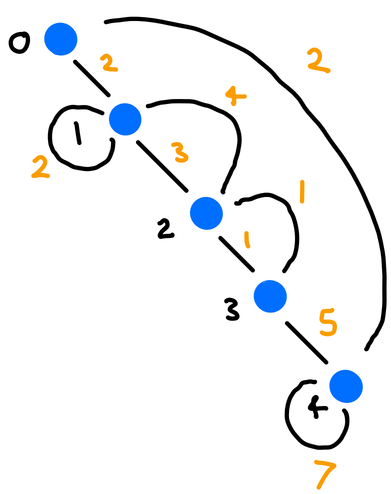
sum all weights going into or leaving each node

- Laplacian matrix (AKA graph laplacian): A matrix that is related to useful properties of the graph. For example, in machine learning you can use the eigenvalues and eigenvectors of the Laplacian matrix to efficiently store a network.

interesting project ideas here...

Calculate by $L = D - A$.

ex



Compute: Adjacency matrix
degree matrix
weighted degree matrix
Laplacian matrix

Compute before looking below at the answer.

$$A = \begin{bmatrix} 0 & 2 & 0 & 0 & 2 \\ 2 & 2 & 7 & 0 & 0 \\ 0 & 7 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 5 \\ 2 & 0 & 0 & 5 & 7 \end{bmatrix} \qquad D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

$$D_w = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 11 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 14 \end{bmatrix} \qquad L = \begin{bmatrix} 2 & -2 & 0 & 0 & -2 \\ -2 & 2 & -7 & 0 & 0 \\ 0 & -7 & 4 & -2 & 0 \\ 0 & 0 & -2 & 3 & -5 \\ -2 & 0 & 0 & -5 & -4 \end{bmatrix}$$

Next we are going to review python. Check out the jupyter notebook I created...