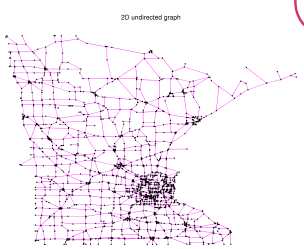
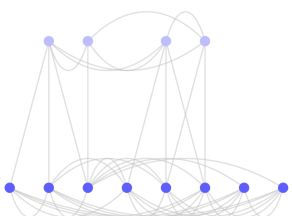
Lecture 1 EN. 553.744 Prof. Luana Ruiz 1/22

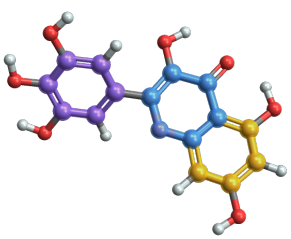
Welcome to Data Science Methods for Large-Scale Graphs!

Today:
- why DS? why LS. graphs?
- what are graphs/ graph signals?
- graph diffusion processes
- adjacency matrix & Laplacian
- total variation energy
- canonical frequencies & oscillation modes

What is graph data science?







Why large-scale graphs?

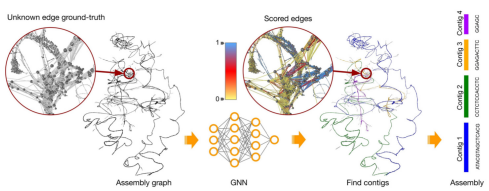
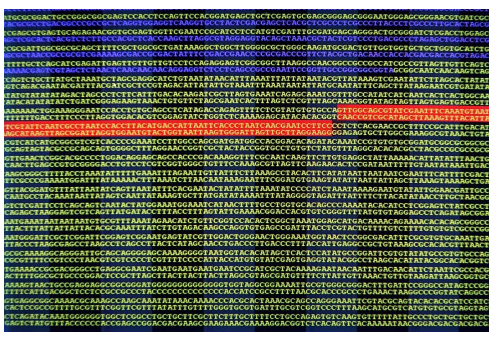
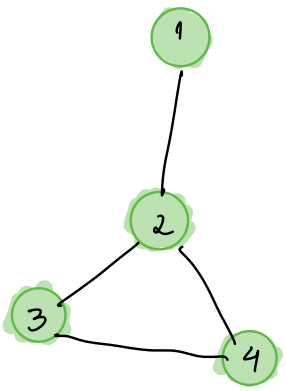


Figure 1. Overview of the assembly workflow with GNNome. An assembly graph is passed to a trained neural network which produces a probability for each edge. The probabilities are visualized in the zoomed-in region in the middle, with the color scheme that increases the visibility of the edges with probability around 0.5. The graph shown here represents entangled chromosomes 21 and 22 of CHM13 (Nurk et al. 2022) generated with hifiasm (Cheng et al. 2021), as well as the four longest contigs found by GNNome. The graph was visualized with Graphia (Freeman et al. 2022).



Graphs & graph signals



- A graph $\mathcal{G}$ is a triplet

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$$

where $\mathcal{V} = \{1, \dots, n\}$, $|\mathcal{V}| = n$

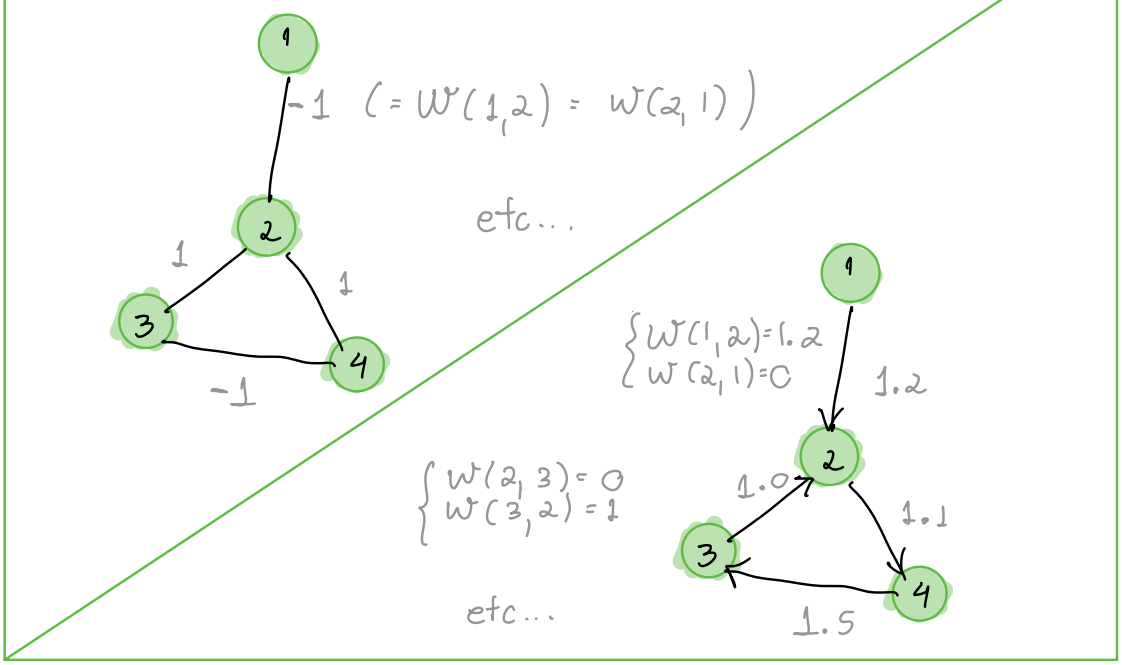
$$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$$

$$\mathcal{W} : \mathcal{E} \to \mathbb{R}$$

- A graph is undirected if : $\forall (i,j) \in \mathcal{E}$, $\mathcal{W}(i,j) = \mathcal{W}(j,i)$

It is directed otherwise.

Example :



- A graph is unweighted if: if $w: E \to \{0, 1\}$ (no edge: 0, edge: 1)

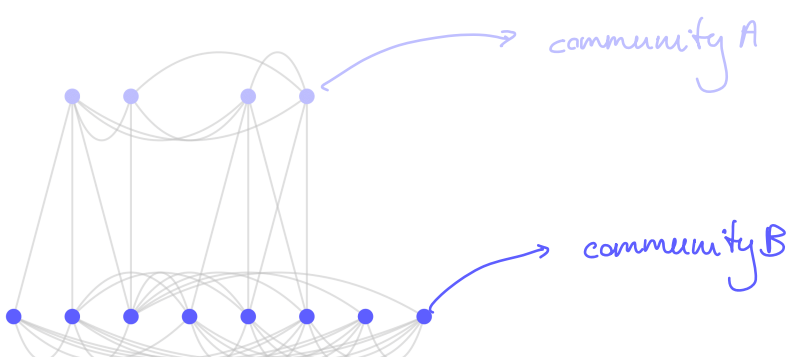
e.g.: friendship networks, citation networks, etc...

If is weighted if $w : E \to \mathbb{R}$

e.g.: transportation networks (TSP), correlation networks, etc...

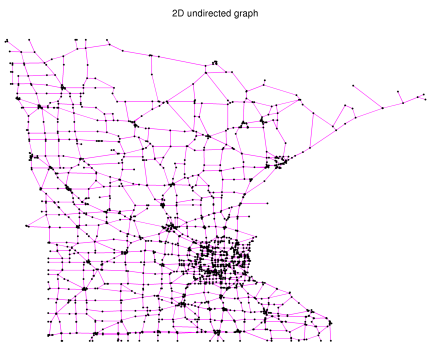
- The nodes of a graph often have info associated with them

↳ fixed node properties or features (type 1)



such as nodes' community assignments

↳ graph signals (often implies variability; can be interpreted as variables) (type 2)



e.g.: traffic counts at Minnesota road network intersections

Both types of node data are represented as vectors:

$$\begin{cases} x \in \mathbb{R}^n \\ [x]_i = \text{the signal value at node } i \end{cases}$$

We'll call such vectors graph signals

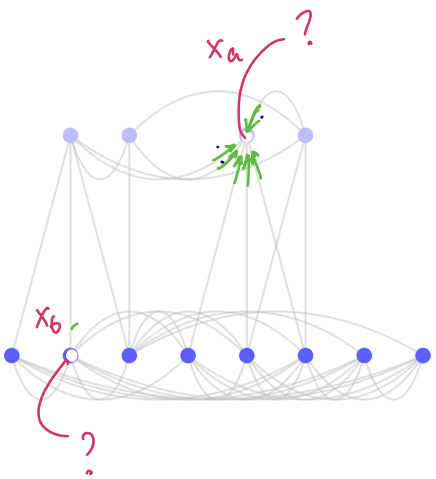
↳ the signal exists on the graph G.
This is often implicit (when obvious from context) but we'll sometimes use the notation (G, x)

- How do we process graph signals (G, x)? How do signals x interact with the graph G?

Through a network diffusion process

(aka message passing; aka aggregation)

E.g.:



local (node-level) estimation of attributes; limited comms imposed by G

$$\begin{cases} \text{if } \bullet, & x = 0 \\ \text{if } \bullet, & x = 1 \end{cases}$$

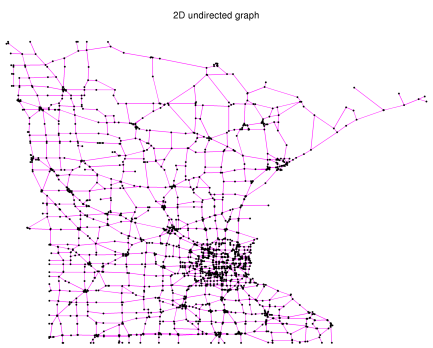
How would you estimate x_a & x_b?

Define $N_a = \{ j \in V \text{ s.t. } (a,j) \in E \}$

Then $\hat{x}_a = \left(\sum_{i \in N_a} x_i \right) \div |N_a| = \frac{3.0 + 3.1}{6} = 0.5$?

$\hat{x}_b = \frac{1.0 + 7.1}{8} \approx 0.875 \rightarrow$ likely •

E.g.: signal forecasting, such as traffic flow



→ suppose G is directed and weighted, with $\sum_j W(i,j) = 1$ at each node i

i.e., $W(i,j)$ is the probability of moving from i to j

→ If $x(t)$ is traffic at time t, how would you estimate traffic at $t+1$?

Define $\mathcal{N}_i^{in} = \{ j \in \mathcal{V} \text{ s.t. } (j,i) \in \mathcal{E} \}$

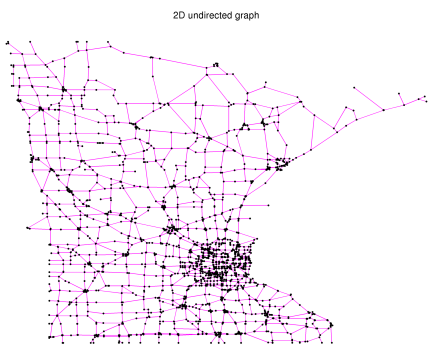
$\mathcal{N}_i^{out} = \{ j \in \mathcal{V} \text{ s.t. } (i,j) \in \mathcal{E} \}$

Estimate $x(t+1)$ as $\hat{x}_i(t+1) = \sum_{j \in \mathcal{N}_i^{in}} W(j,i)\, x_j(t)$

→ Diffusion processes are local processes: they can be defined locally at each node.

But we can also represent them globally using graph matrix representations

E.g., consider once again the Minnesota road network



→ suppose G is directed and weighted, with $\sum_j W(i,j) = 1$ at each node i

i.e., $W(i,j)$ is the probability of moving from i to j

and consider the network's adjacency matrix:

(DEF) $[A]_{ij} = \begin{cases} W(j,i) \in (j,i) \in \mathcal{E} & \forall\, i,j \in \mathcal{V} \\ 0 & \text{o.w.} \end{cases}$

Then, $x(t+1)$ can be expressed as a function of $x(t) \in \mathbb{R}^n$ as: $x(t+1) = A\,x(t)$ $\quad [Ax(t+1)]_i = \sum_{j \in \mathcal{V}} [A]_{ij} (x(t))_j = \sum_{j \in \mathcal{N}_i^{in}} w(j,i)\,(x(t))_j$

This is perhaps the simplest network diffusion process. We can define more general diffusion processes by defining the matrix $S \in \mathbb{R}^{n \times n}$ s.t.

(DEF) $\quad [S]_{ij} \neq 0 \iff (i,j) \in \mathcal{E}$

OR

$i = j$

This matrix is called the graph shift operator (GSO)

$z = Sx$ is a (graph) "shift" or diffusion of x by S

Examples of GSOs:

	undirected	directed
adjacency matrix	$A_{ji} = A_{ij} = w(i,j)$ $= w(j,i)$, symmetric	$A_{ij} = w(j,i)$ $A_{ji} = w(i,j)$
degree matrix	$D_{ii} = d(i)$ (degree of i) $D = \text{diag}(A \cdot \mathbb{1})$	$D^{out} = \text{diag}(A \cdot \mathbb{1})$ $D^{in} = \text{diag}(A^T \cdot \mathbb{1})$
Laplacian matrix	$L_{ij} = \begin{cases} -A_{ij} & \text{if } i \neq j \\ d(i) & \text{if } i = j \end{cases}$ $L = D - A$	asymmetric, numerical issues; many definitions... - based on in-degree - " " out-degree - " " symmetrized adj. → symmetric, PSD

Random walk (RW) matrix

$$P_{ij} = \frac{A_{ij}}{d(i)} \qquad P_{ij} = \frac{A_{ij}}{d_{out}(i)}$$

$$P = D^{-1} A \qquad P = D_{out}^{-1} A$$

RW Laplacian

$$L^{rw} = I - P$$

normalized adjacency

$$\bar{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

either $\bar{A} = P$ or $A \cdot D_{in}^{-1}$

normalized Laplacian

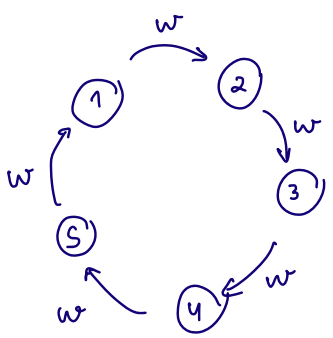
$$\bar{L} = I - \bar{A}$$

Obs: The most common GSOs in large-scale graph data science are the adjacency & Laplacian

Obs.: We can recover local implementation as

$$z_i = \sum_j S_{ij} x_j = \sum_{j \in N_i} S_{ij} x_j$$

E.g.: Interpretation of the (left) graph Laplacian.



5-cycle digraph

"5-sample discrete-time graph"

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