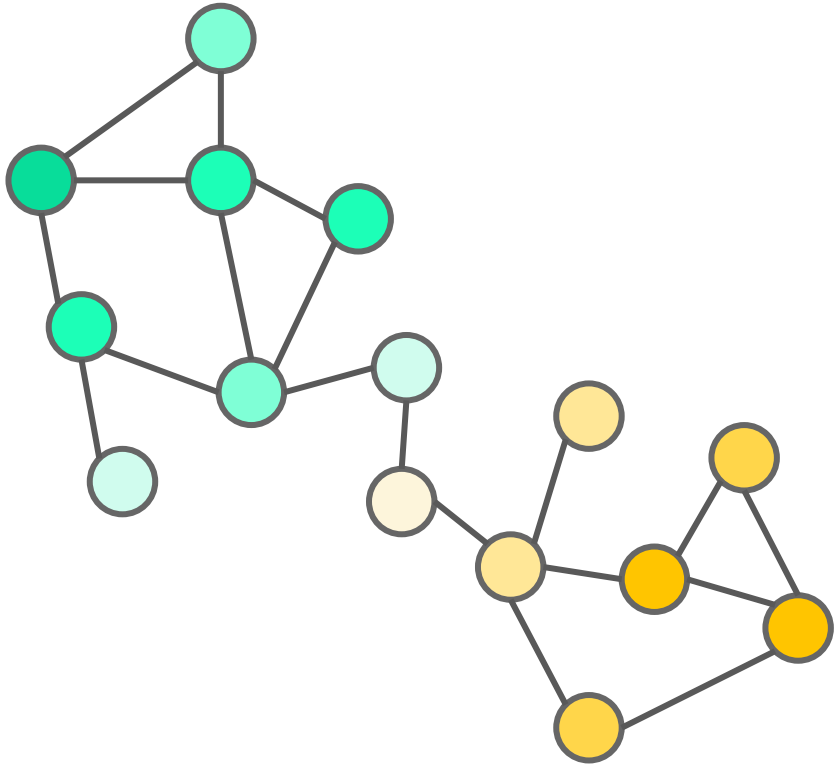
Class core values

1. Be **respect**ful to yourself and others
2. Be **confident** and believe in yourself
3. Always do your **best**
4. Be **cooperative**
5. Be **creative**
6. Have **fun**
7. Be **patient** with yourself while you learn
8. Don’t be shy to **ask “stupid” questions**
9. Be **inclusive** and **accepting**



Week 7, Lecture 1

Of Graphs & Proteins

Learning Objectives

1. Describe graph, nodes, and edges
2. Calculate general features of a graph such as diameter, length, radius, centers
3. Calculate degree and adjacency matrix
4. Describe the basic idea behind graph convolution
5. Generate a graph from a protein

Types of input data for proteins

1. Simple input — SVM, random forest, dense neural net

protein$_1$	25 kDa	pI=7.5	310 residues	…	2.5 hr half-life	Stability$_1$
protein$_2$	10 kDa	pI=4	50 residues	…	10 hr half-life	Stability$_2$
protein$_3$	100 kDa	pI=8	1200 residues	…	2 hr half-life	Stability$_3$
…						

Types of input data for proteins

1. Simple input — SVM, Random Forest, dense neural net
2. 2D image — CNN

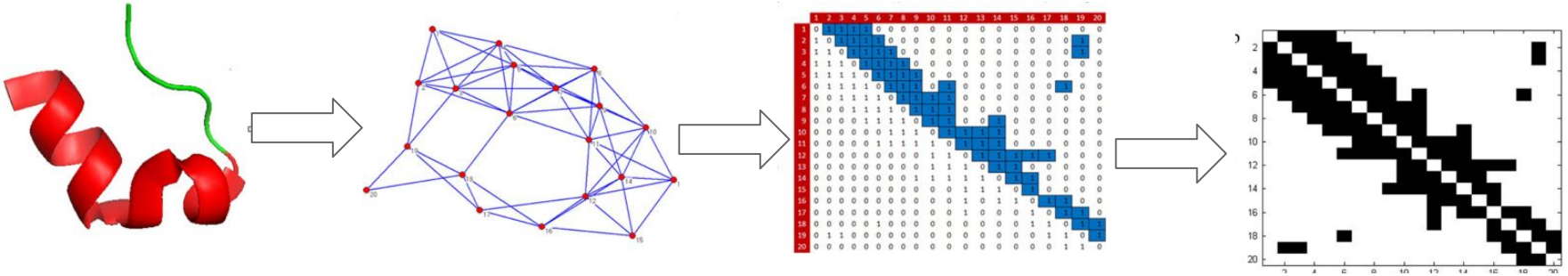


Image from *Physcia A*, **2017**, volume 456, 782-791

Types of input data for proteins

1.	Simple input	SVM, Random Forest, dense neural net
2.	2D image	CNN
3.	String of amino acids	Natural language processing

p_1 MGLTDILGFNREFDILAV…SPLFG s_1

p_2 MLKPTRVNMSERCGHITDENVCSR…TLVRF s_2

p_3 MIKRTVIHGRDFRWNYTSPL…GMNSWQ s_3

…

Features: charge, pKa, size, functional groups, hydrogen bond status, …

Types of input data for proteins

1. Simple input — SVM, Random Forest, dense neural net
2. 2D image — Convolutional neural nets
3. String of amino acids — Natural language processing (RNN, LSTM, Transformers)
4. Graphs

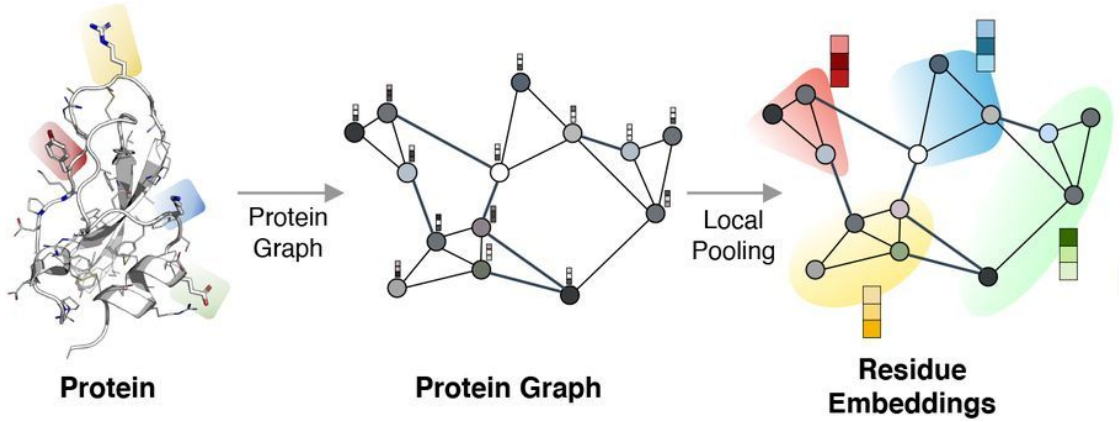
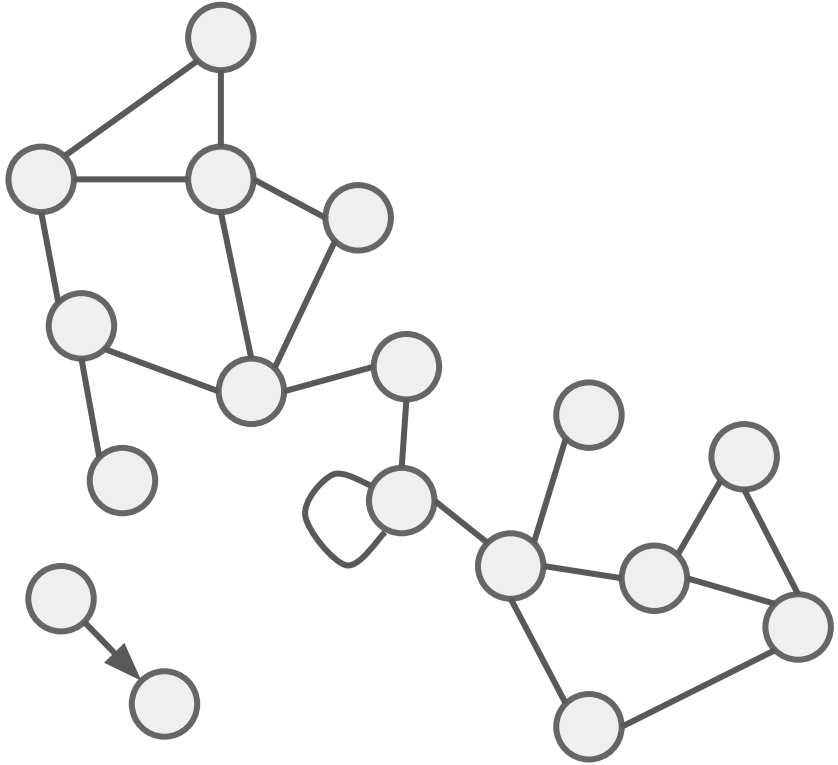
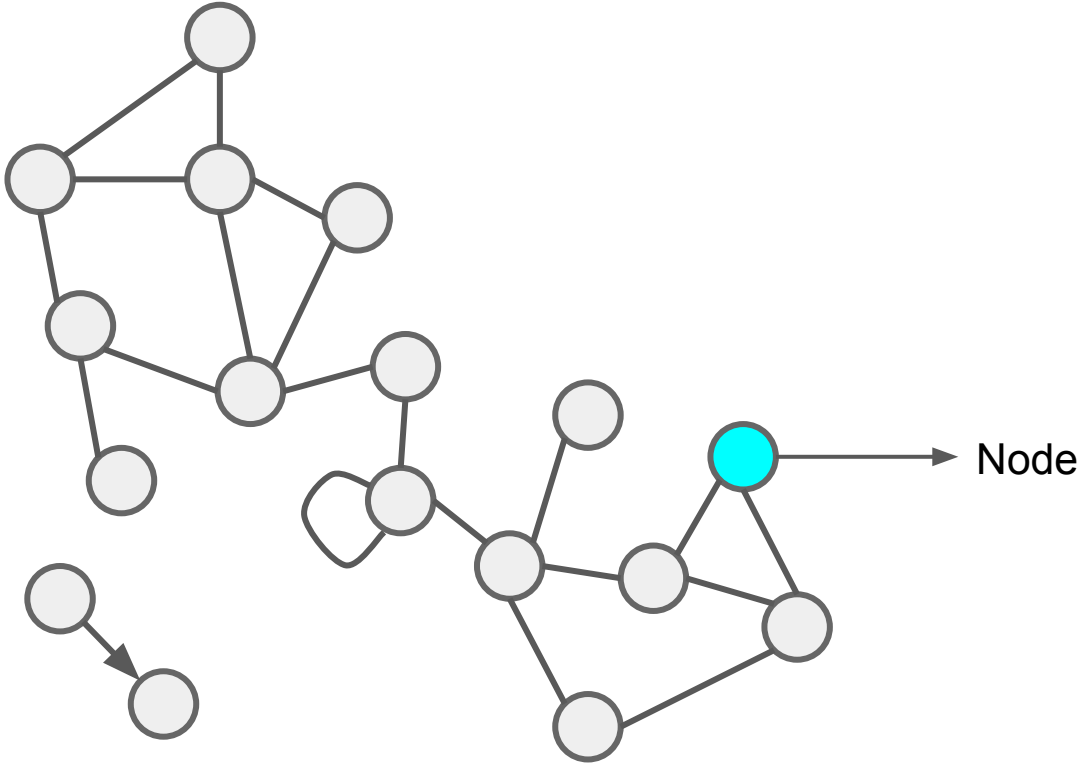


Image from bioRxiv 2020, ProteinGNC

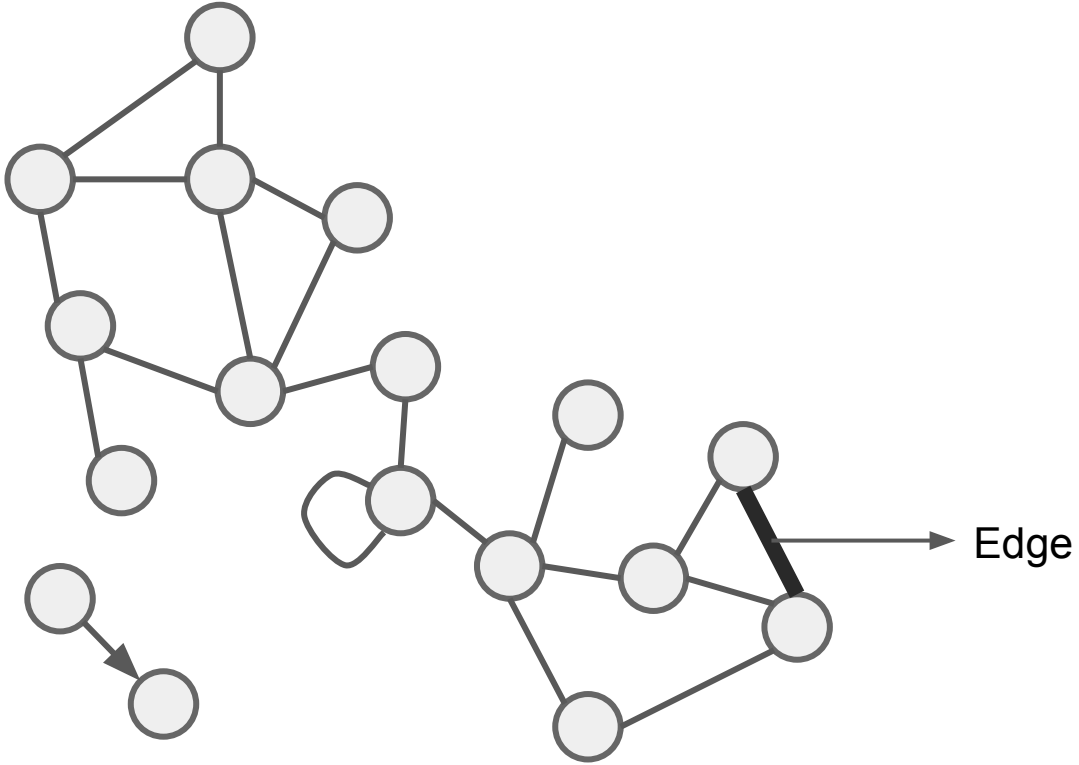
Let’s talk about graphs



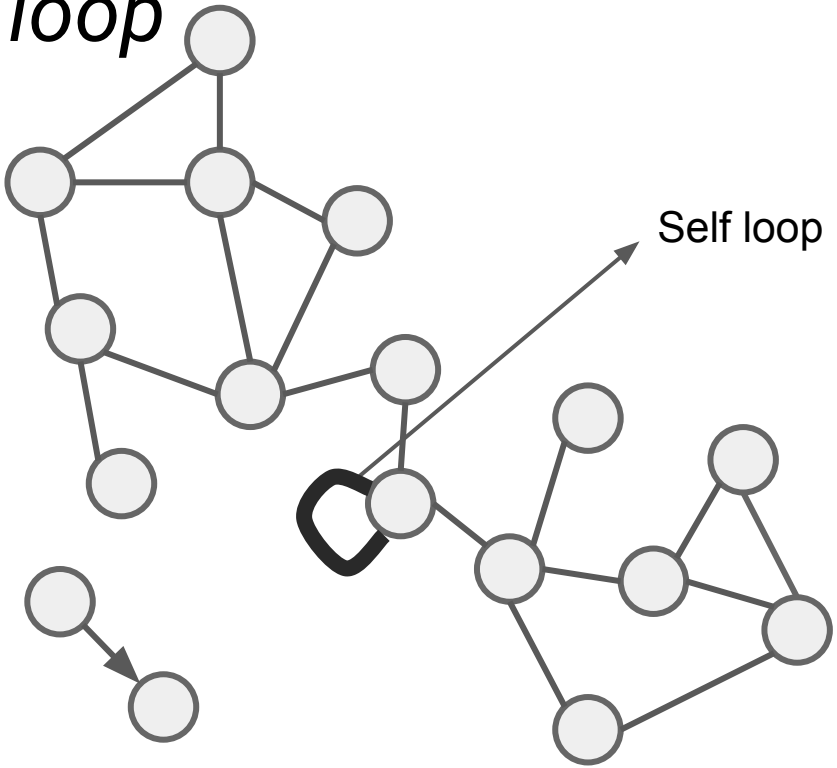
Nodes are basic building blocks of a graph



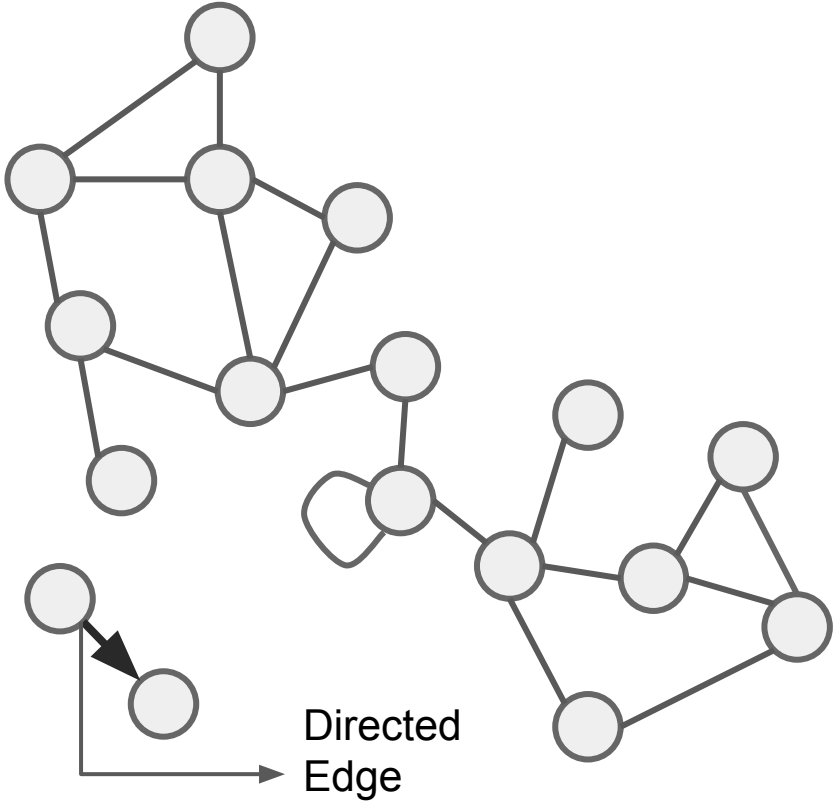
Nodes are connected via edges



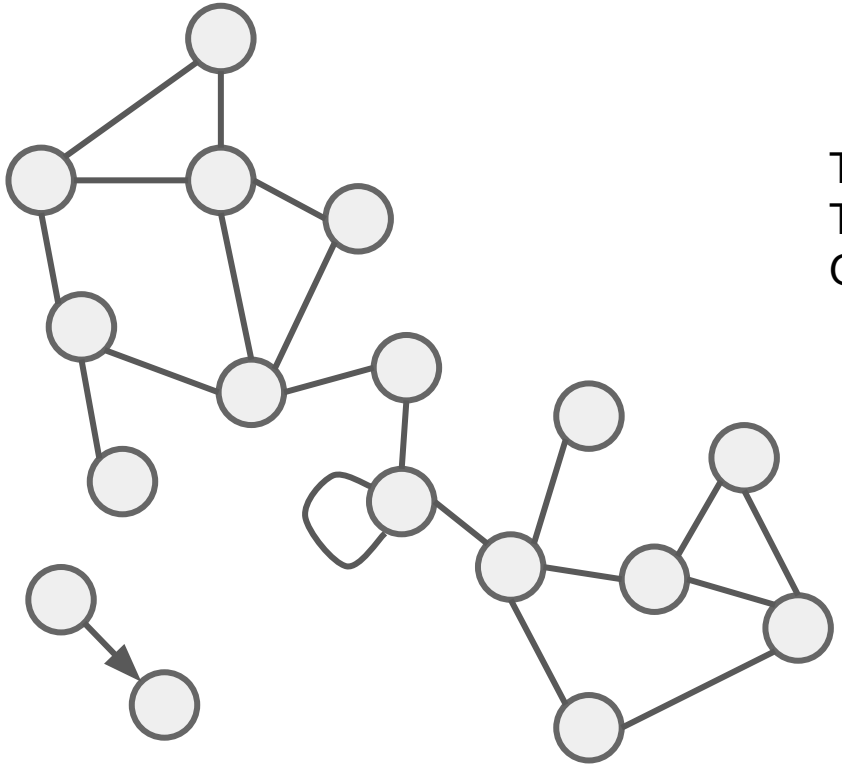
An edge that connects a node to itself is a *self loop*



Edges can be directed

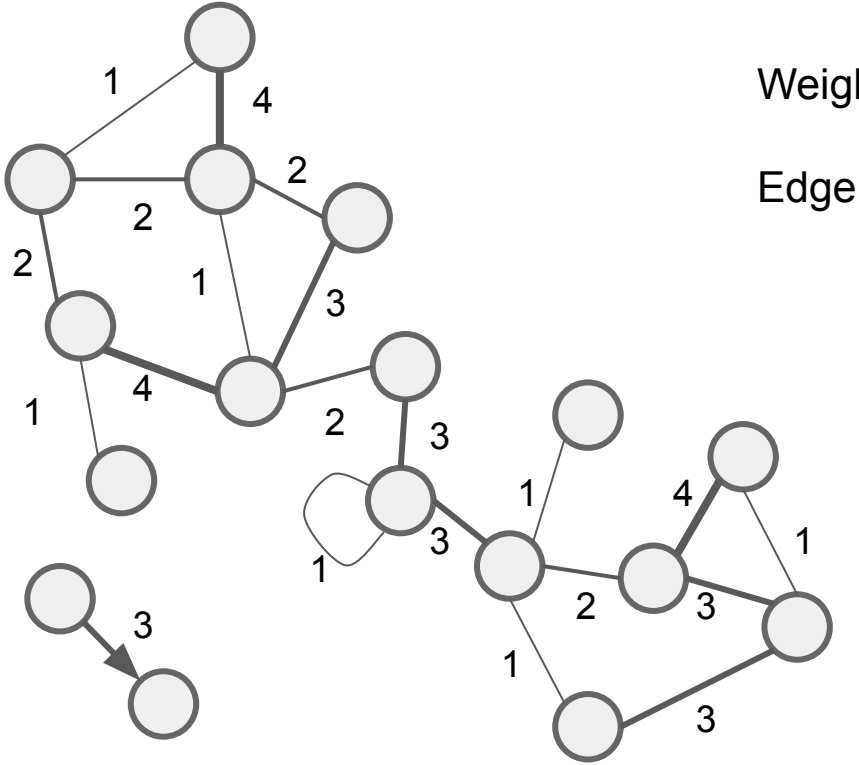


Graph *length* is the number of edges in graph



Total nodes = 17
Total edges = 21
Graph length = total edges = 21

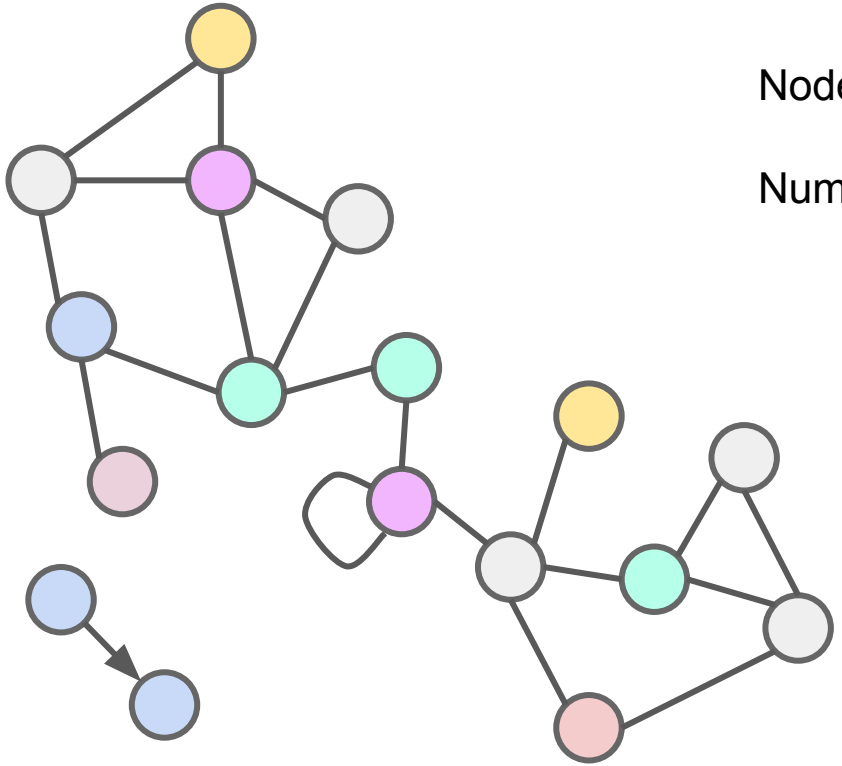
Edge weights describe the strengths of edge



Weighted graph

Edge weight = strength of the edge

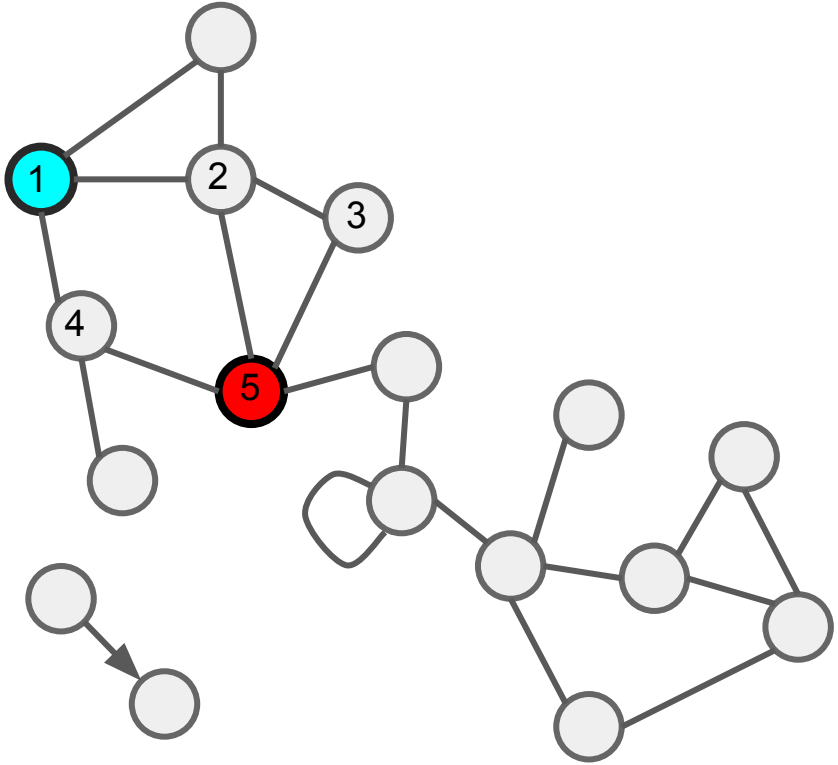
Eah node can have features/attributes



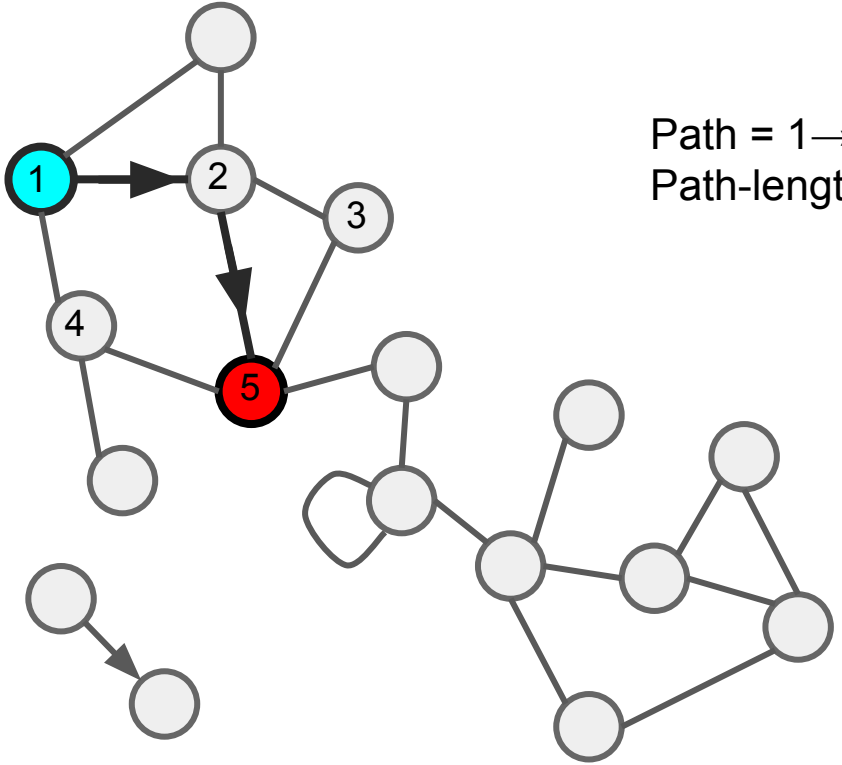
Node features

Numbers/values assigned to nodes

A path defines how one can go from node a to b

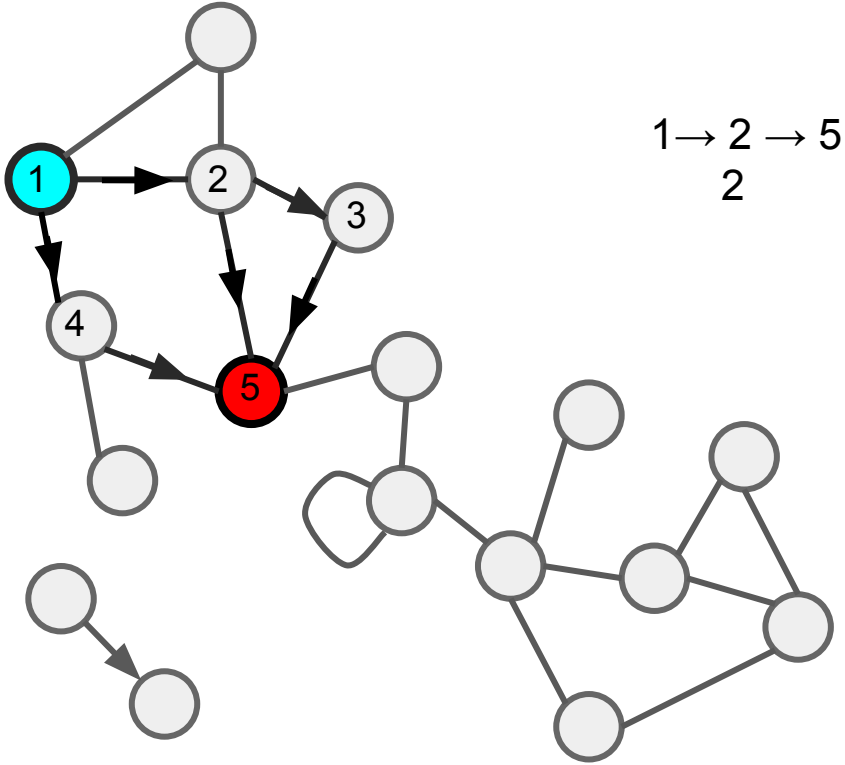


A path defines how one can go from node a to b



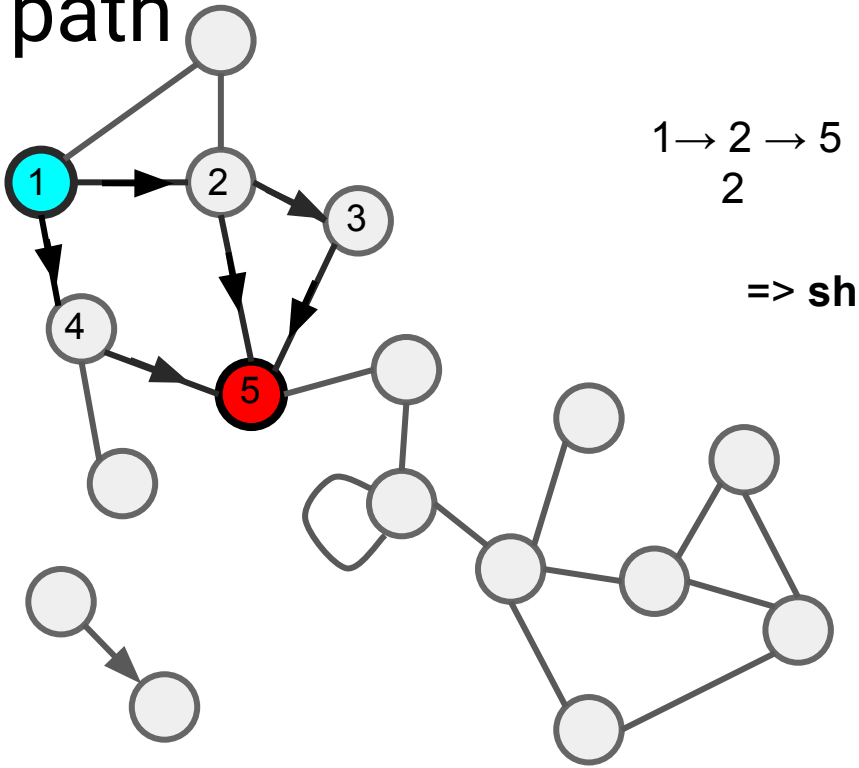
Path = 1→ 2 → 5
Path-length = 2

A path defines how one can go from node a to b



$1 \xrightarrow{2} 2 \xrightarrow{2} 5$ or $1 \xrightarrow{2} 2 \xrightarrow{3} 3 \xrightarrow{2} 5$ or $1 \xrightarrow{4} 4 \xrightarrow{2} 5$

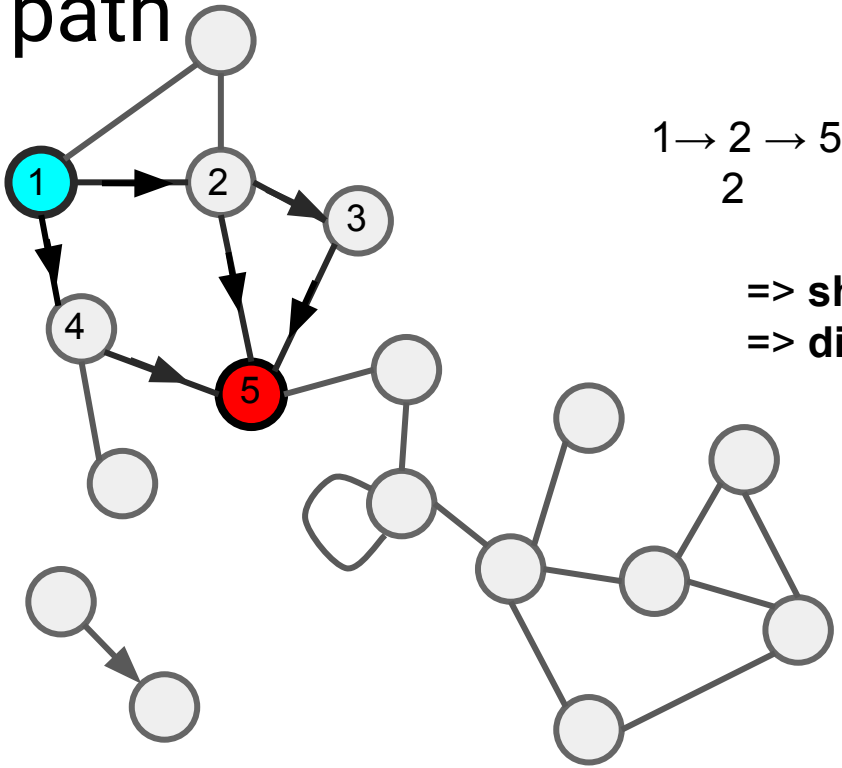
Distance between two nodes is their shortest path



$1 \xrightarrow{2} 2 \xrightarrow{2} 5$ or $1 \xrightarrow{2} 2 \xrightarrow{3} 3 \xrightarrow{5} 5$ or or $1 \xrightarrow{4} 4 \xrightarrow{2} 5$

=> **shortest path** has length **2**

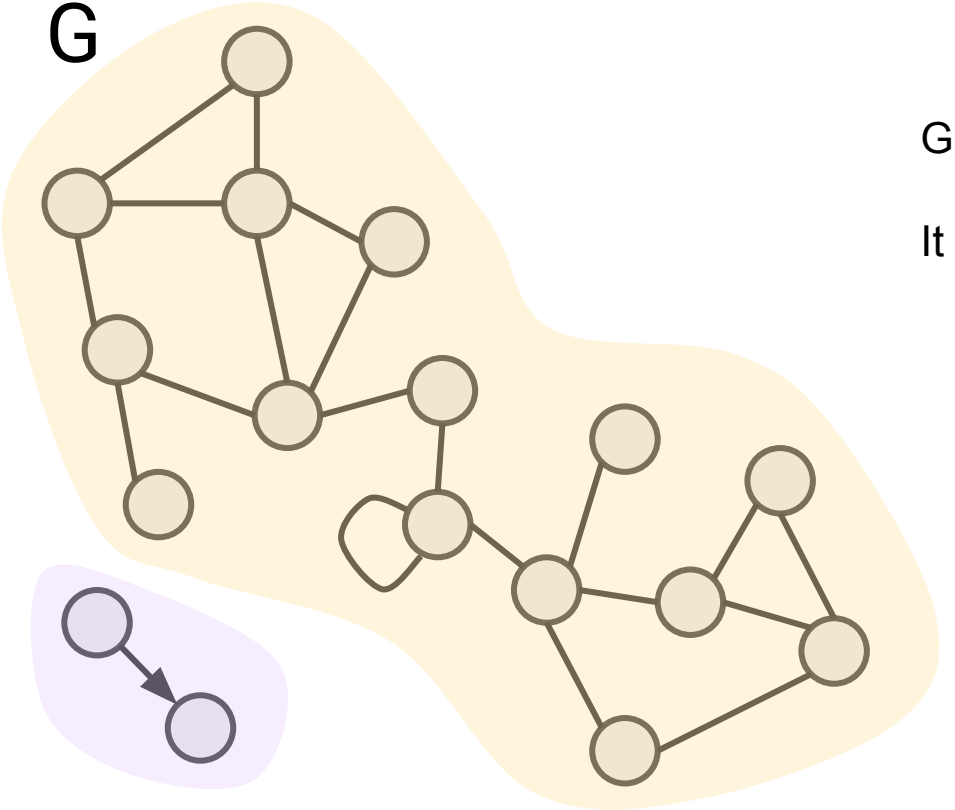
Distance between two nodes is their shortest path



$1 \xrightarrow{2} 2 \xrightarrow{2} 5$ or $1 \xrightarrow{3} 2 \xrightarrow{3} 3 \xrightarrow{2} 5$ or or $1 \xrightarrow{2} 4 \xrightarrow{2} 5$

=> **shortest path** has length **2**
=> **distance** between 1,5 is **2**

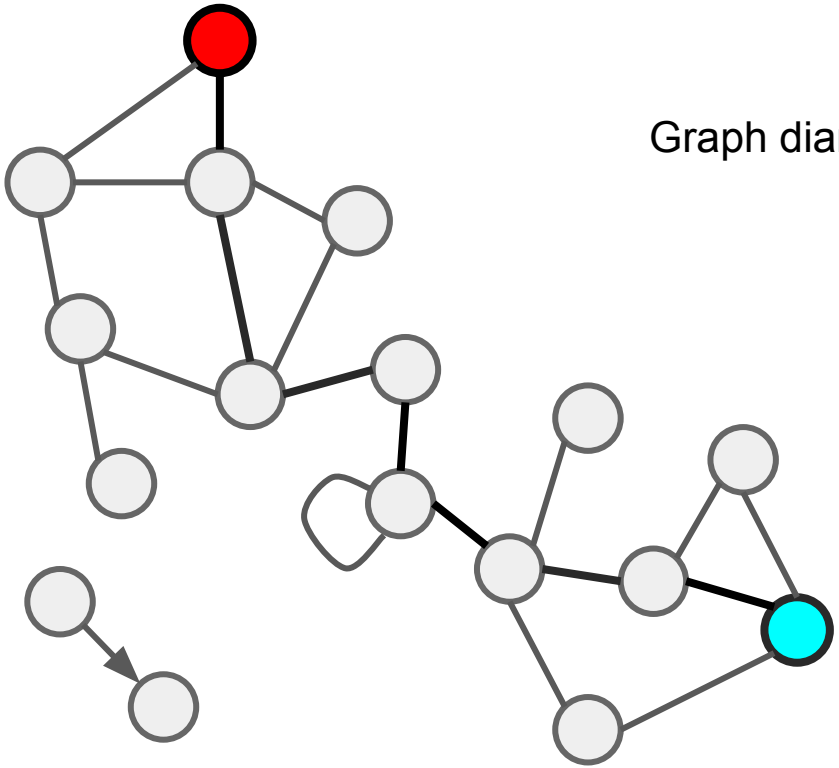
There’s a path between all nodes in connected G



Graph is ***disconnected***

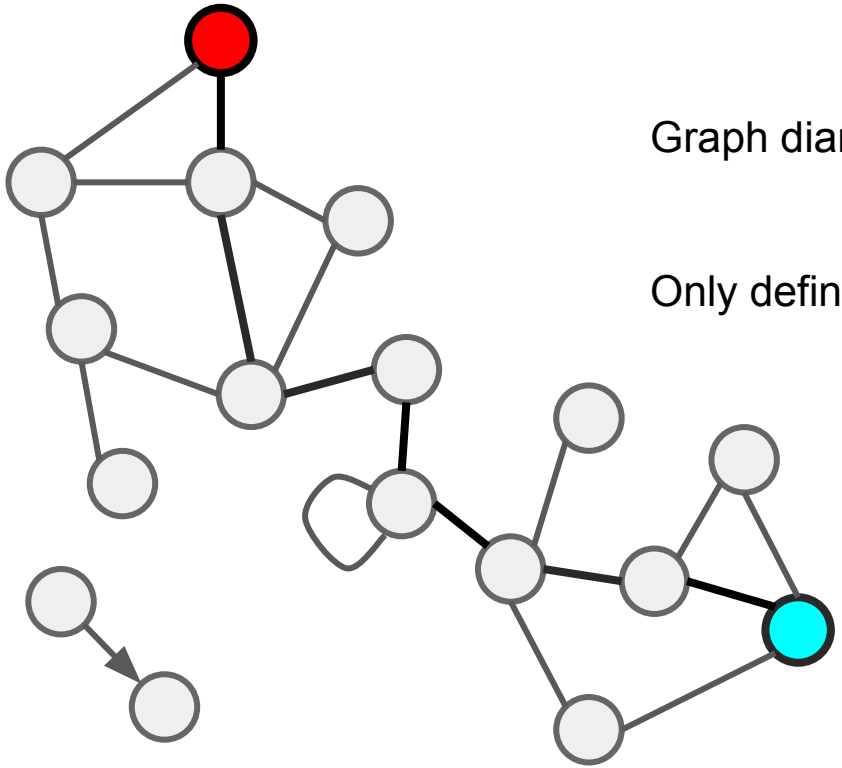
It has **2** connected components

Graph *diameter* is defined based on distances



Graph diameter = longest distance between any two nodes
=

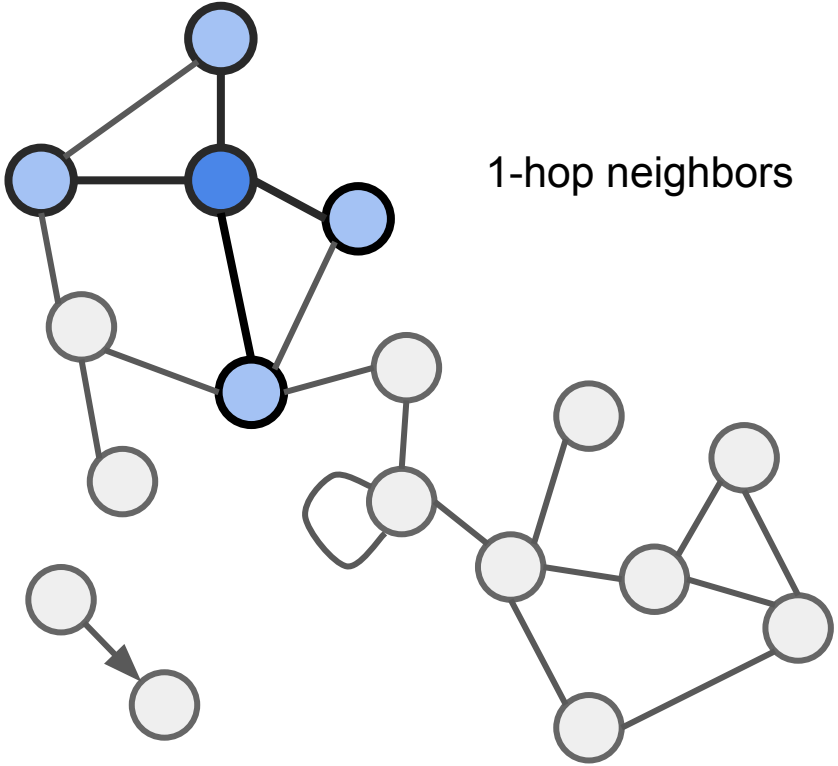
Graph *diameter* is defined based on distances



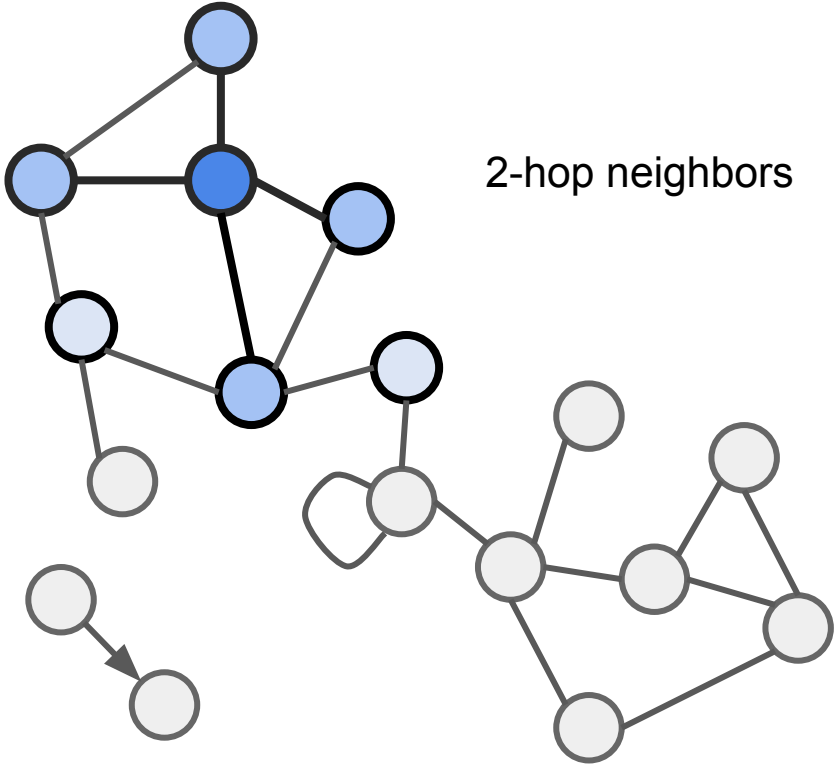
Graph diameter = longest distance between any two nodes
= 7

Only defined in connected components

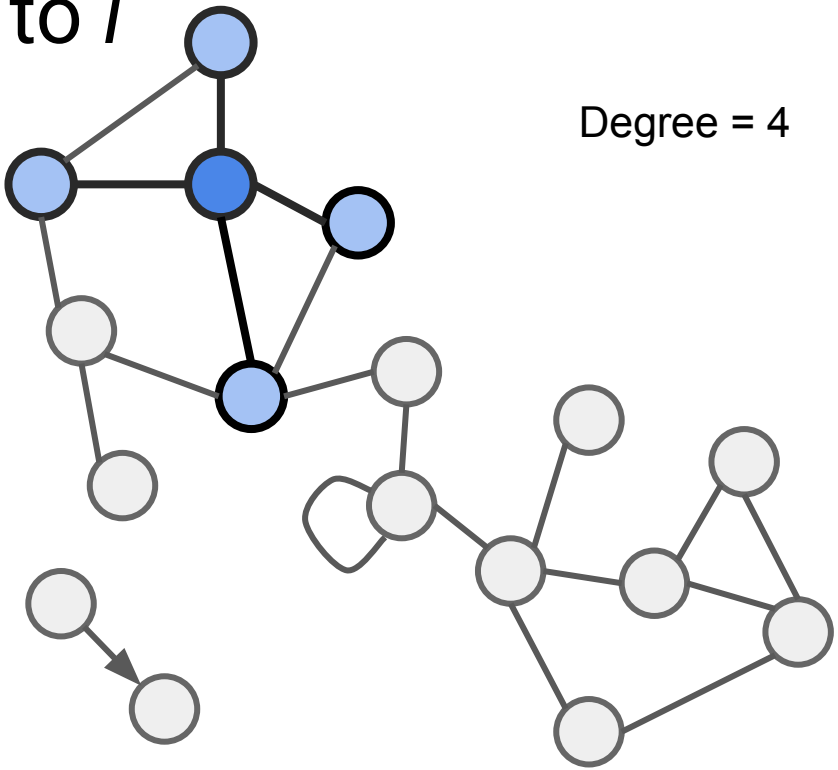
Neighbors of *i* are nodes connected to *i*



Neighbors of *i* are nodes connected to *i*

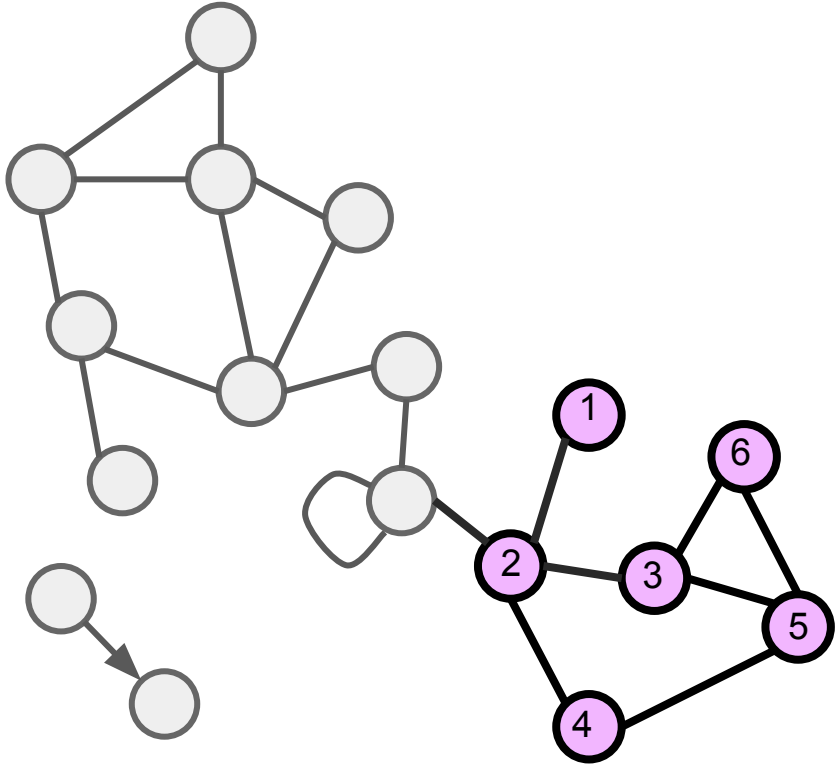


Degree of *i* is the number of edges connected to *i*



Degree = 4

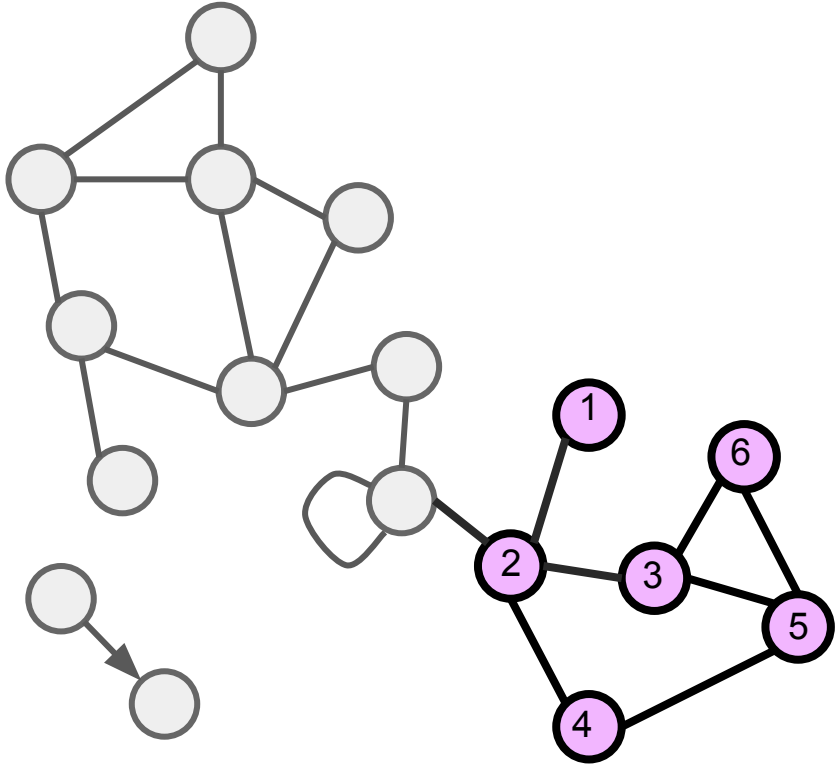
Degree matrix shows all matrices in graph



Degree matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

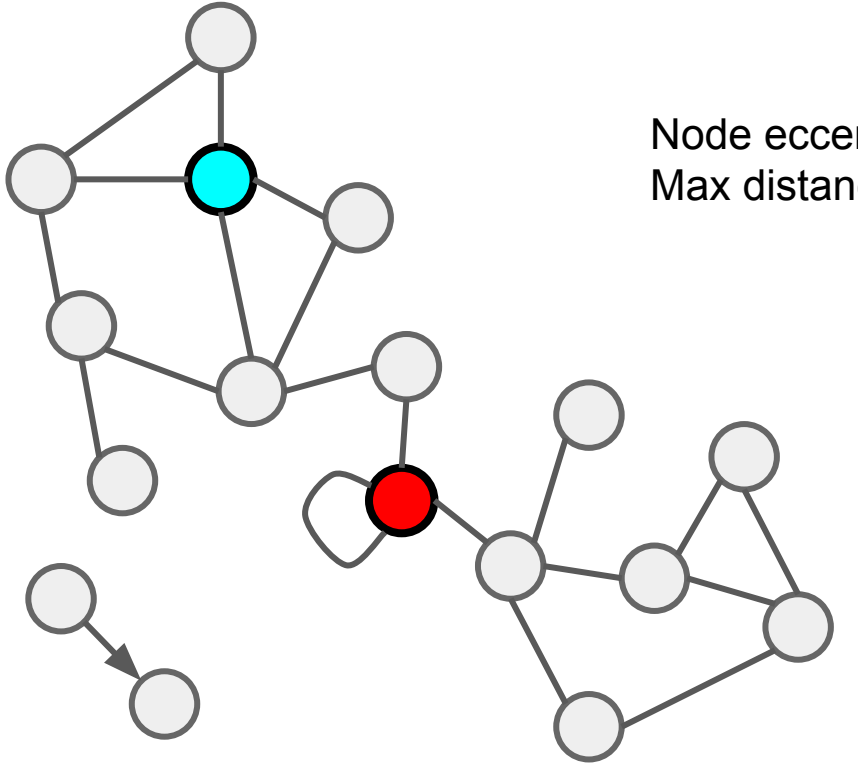
Adjacency matrix gives a sense of connection



Adjacency matrix

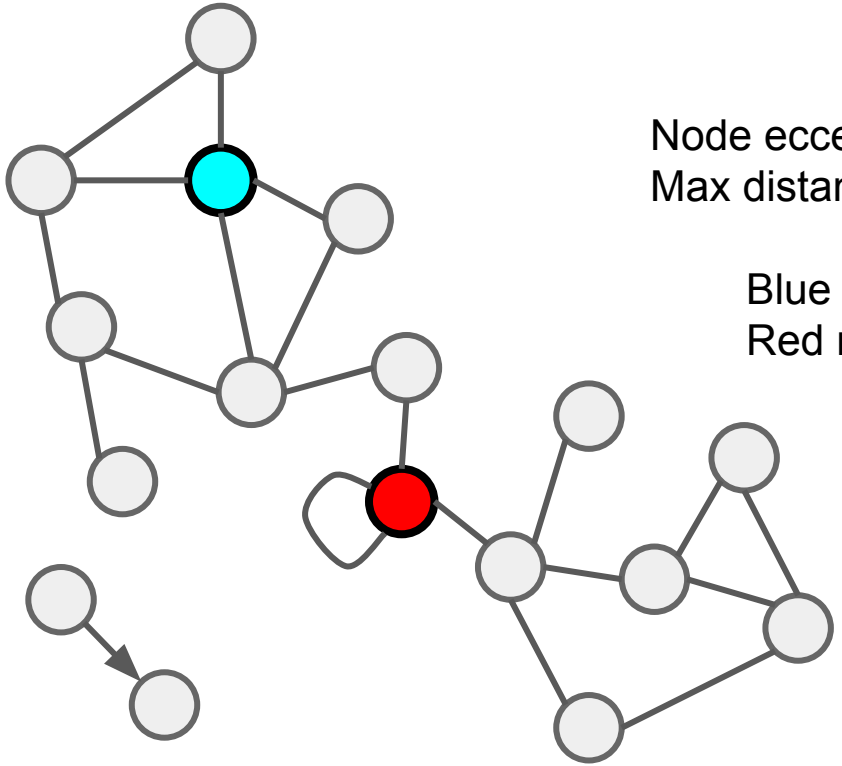
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Eccentricity describes the centrality of *i*



Node eccentricity =
Max distance between node and other nodes in graph

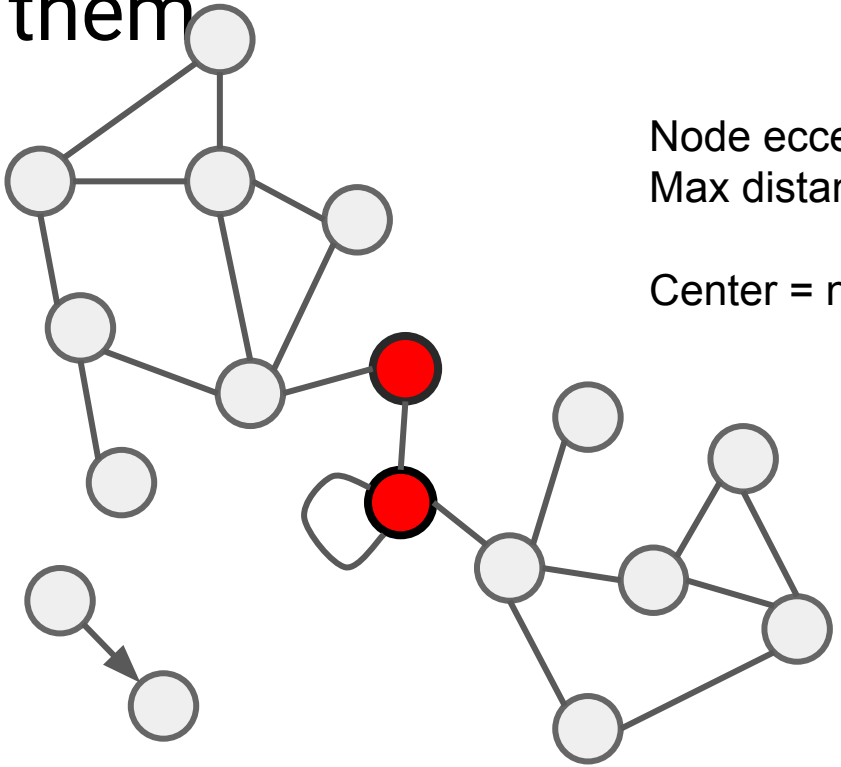
Eccentricity describes the centrality of *i*



Node eccentricity =
Max distance between node and other nodes in graph

Blue node → 6
Red node → 4

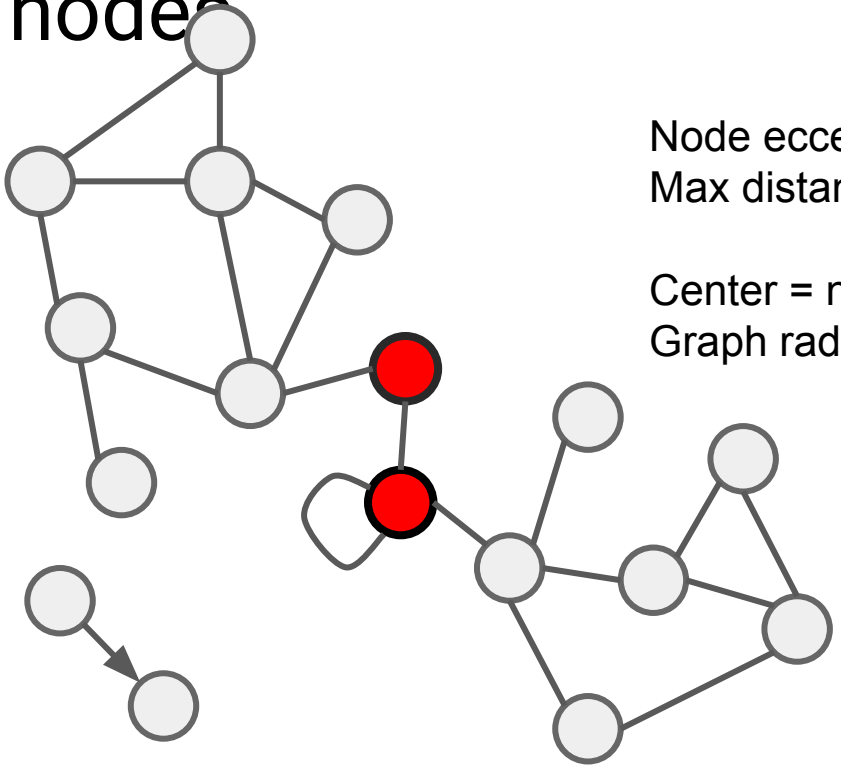
Centers are nodes that most paths go through them



Node eccentricity =
Max distance between node and other nodes in graph

Center = nodes with lowest eccentricity

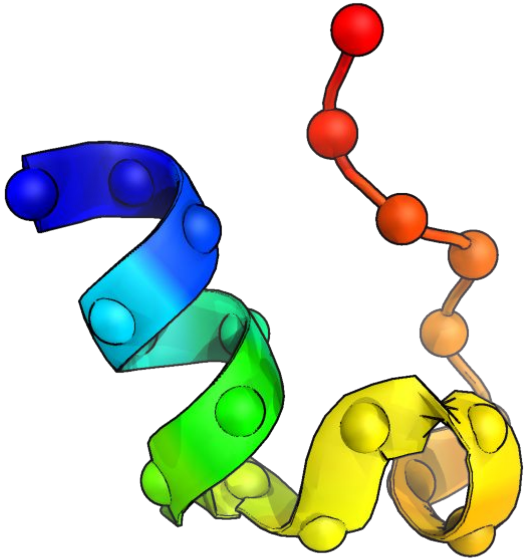
Graph radius is lowest eccentricity among nodes



Node eccentricity =
Max distance between node and other nodes in graph

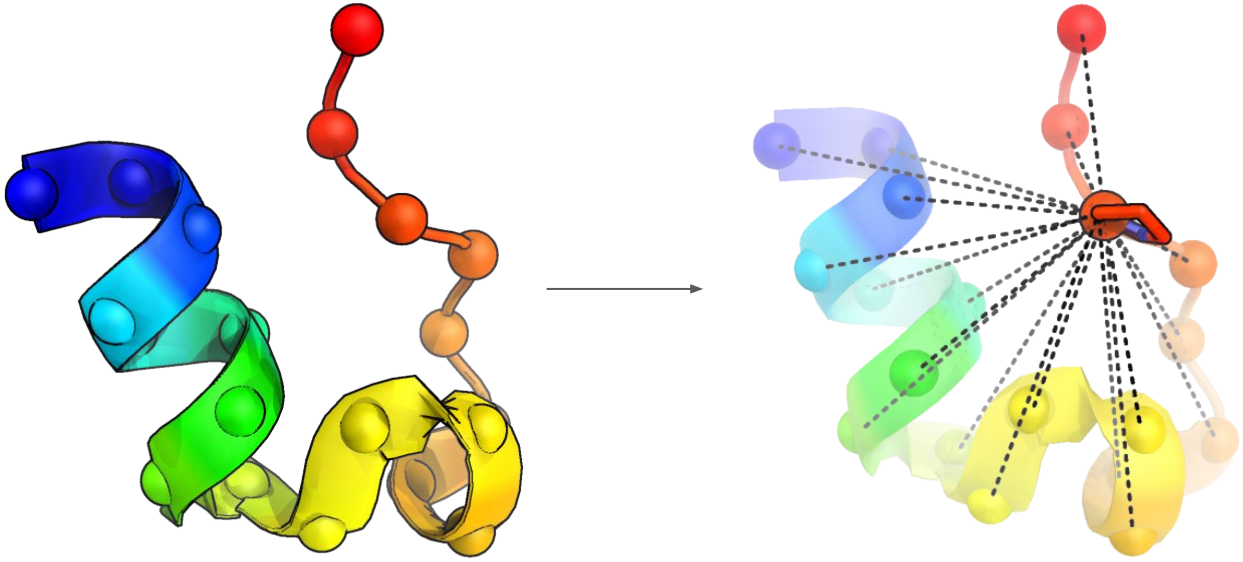
Center = nodes with lowest eccentricity
Graph radius = lowest eccentricity of all nodes
= 4

Proteins and small molecules can be seen as graphs



PDB 1L2Y

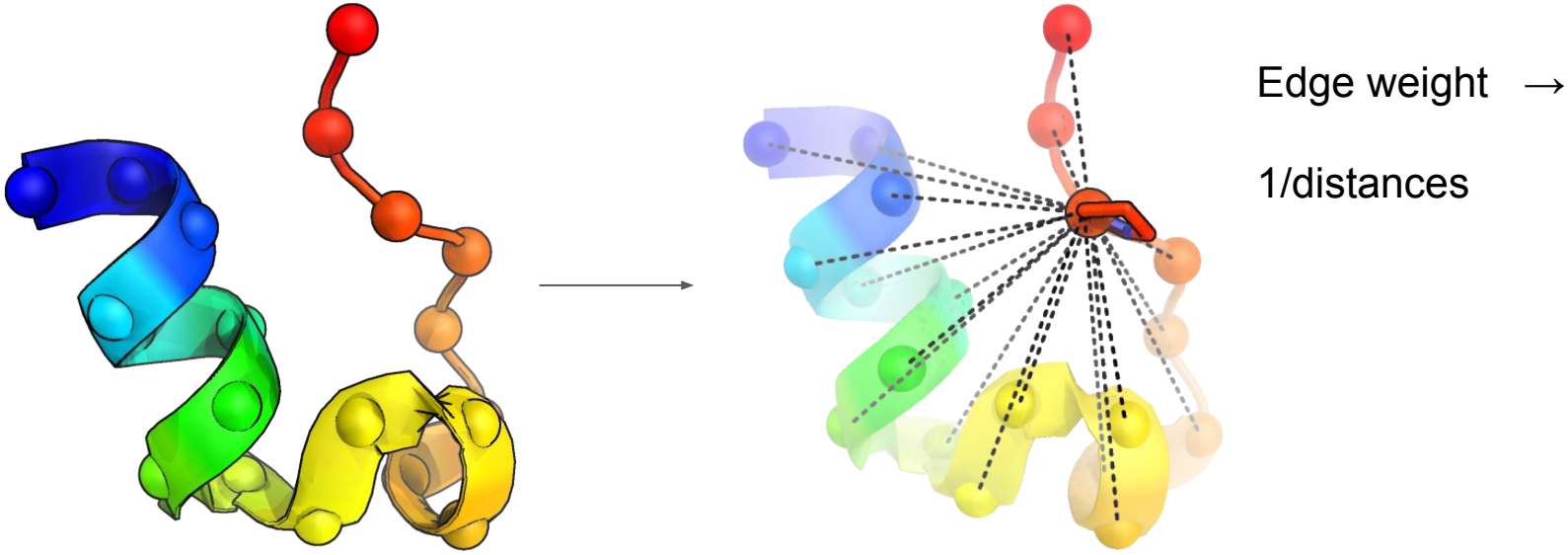
Proteins and small molecules can be seen as graphs



PDB 1L2Y

Measured distances

Proteins and small molecules can be seen as graphs

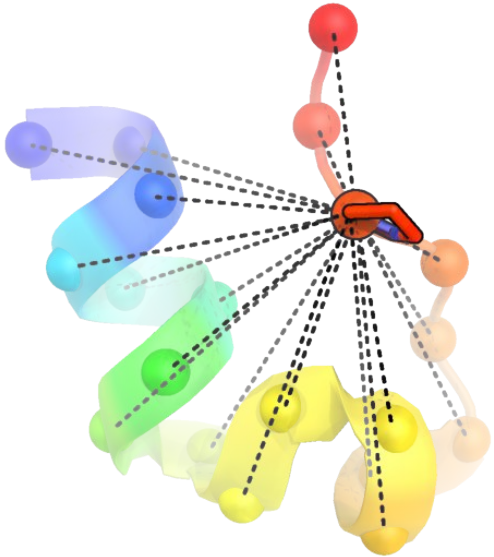


PDB 1L2Y

Measured distances

In-class activity

Protein to distance map to graph



Learning problems on graphs

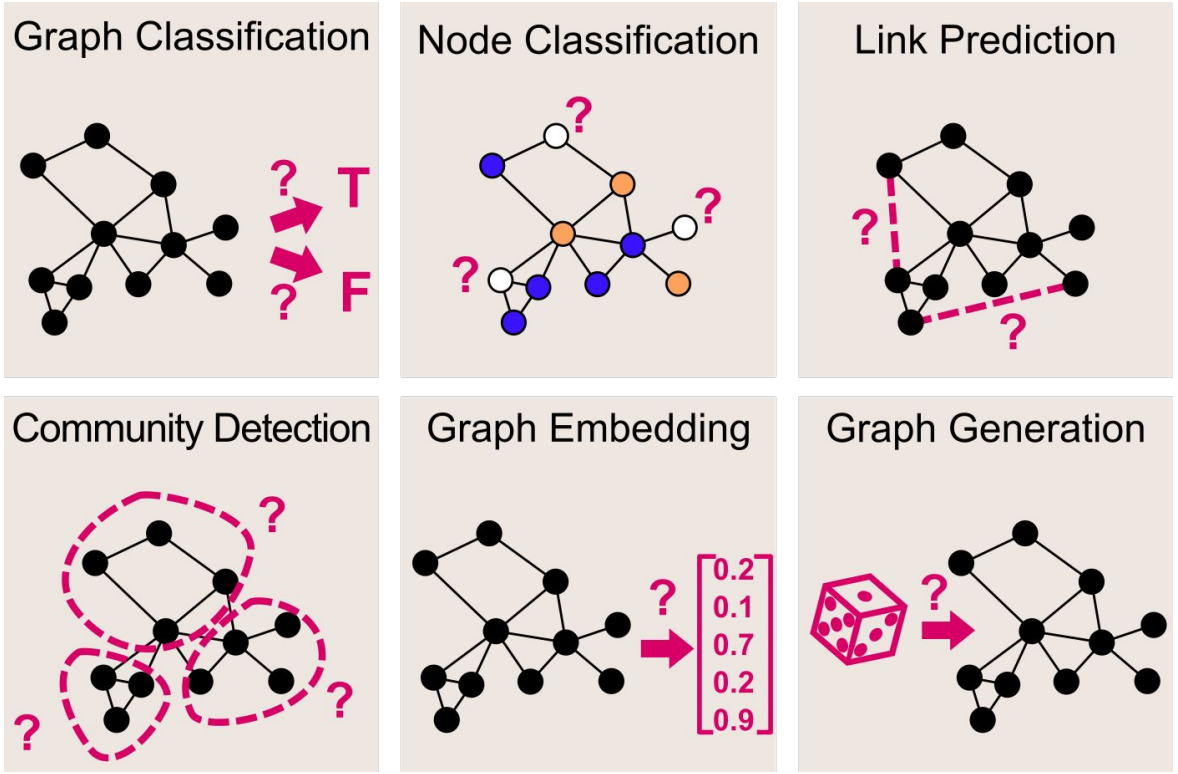


Image from Fracesco Casalengo, TDS – Graph Convolutional Networks

Learning problems on graphs

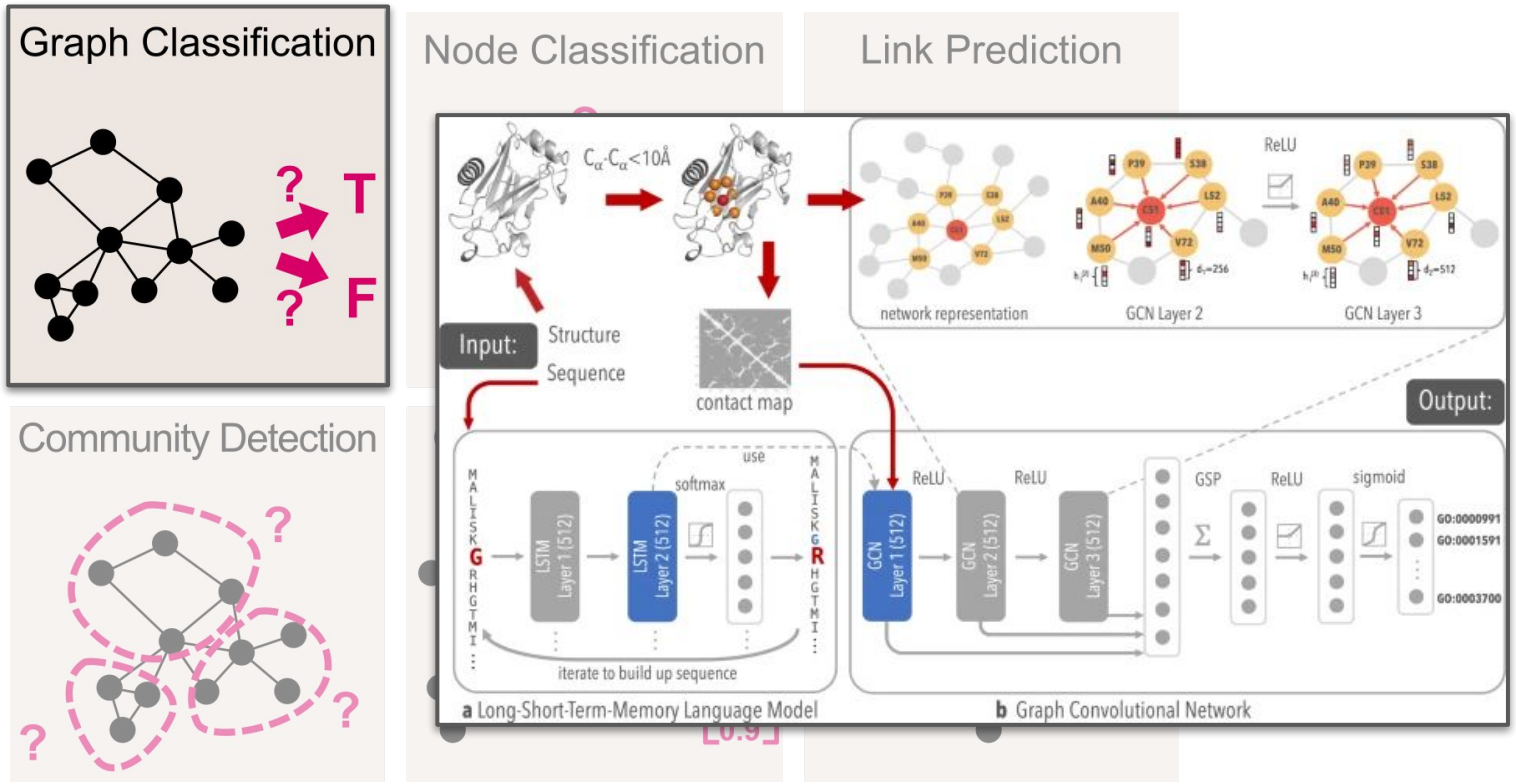
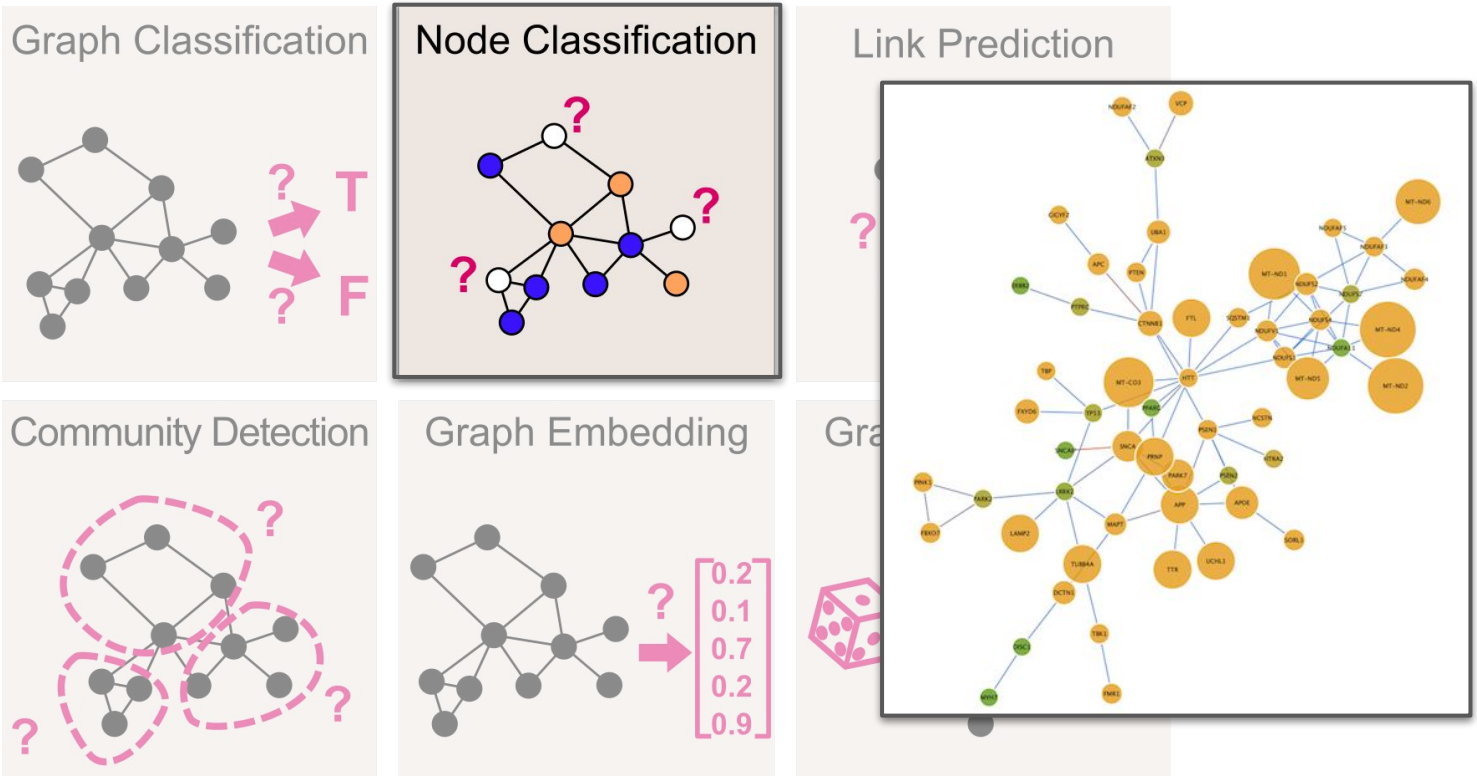
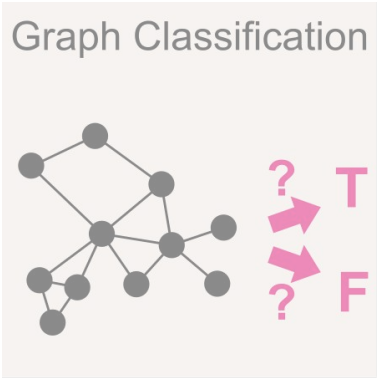


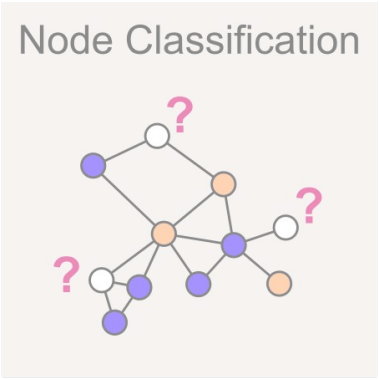
Image from Gligorijevic et al, *Nature comms* **2021** 12(3168)

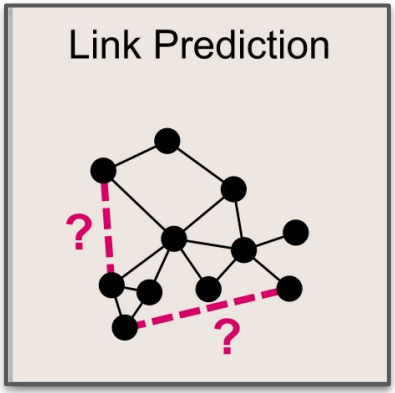
Learning problems on graphs



Learning problems on graphs





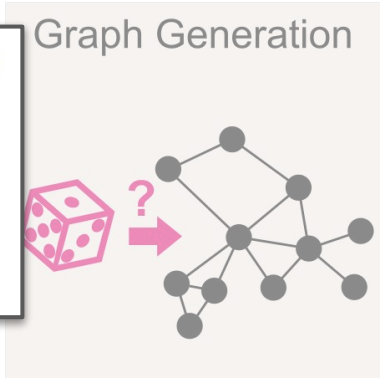


Graph-based prediction of Protein-protein interactions with attributed signed graph embedding

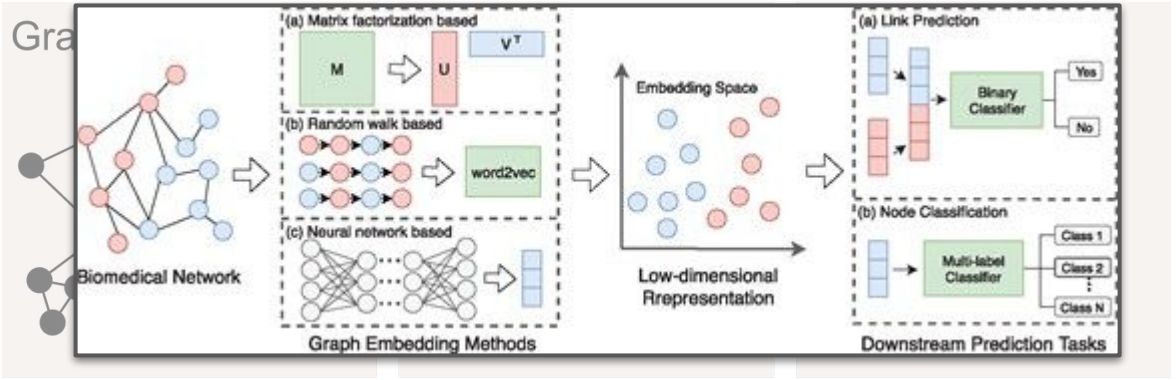
Fang Yang, Kunjie Fan, Dandan Song & Huakang Lin

BMC Bioinformatics **21**, Article number: 323 (2020) | Cite this article

10k Accesses | **18** Citations | **2** Altmetric | Metrics



Learning problems on graphs



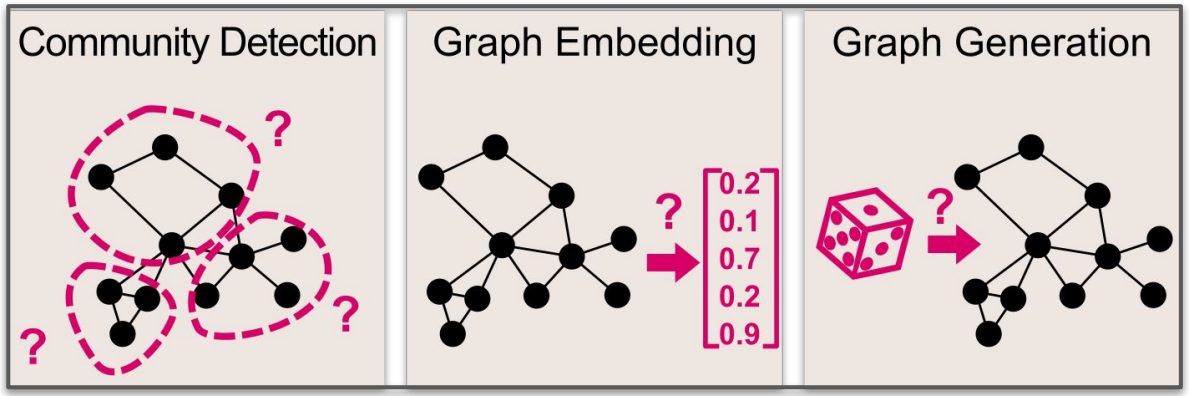
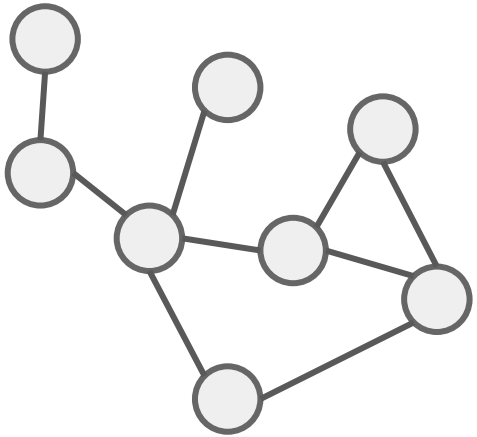
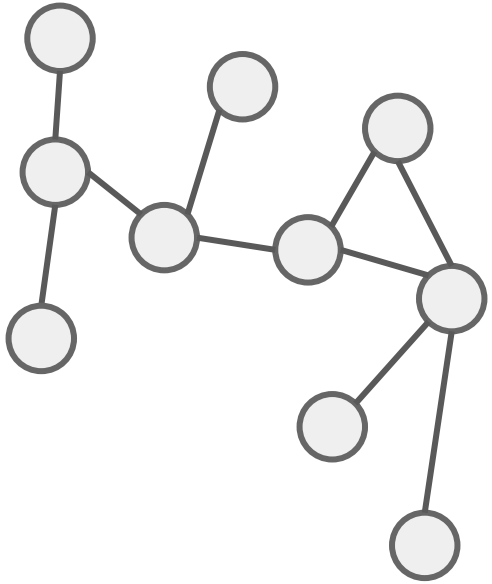


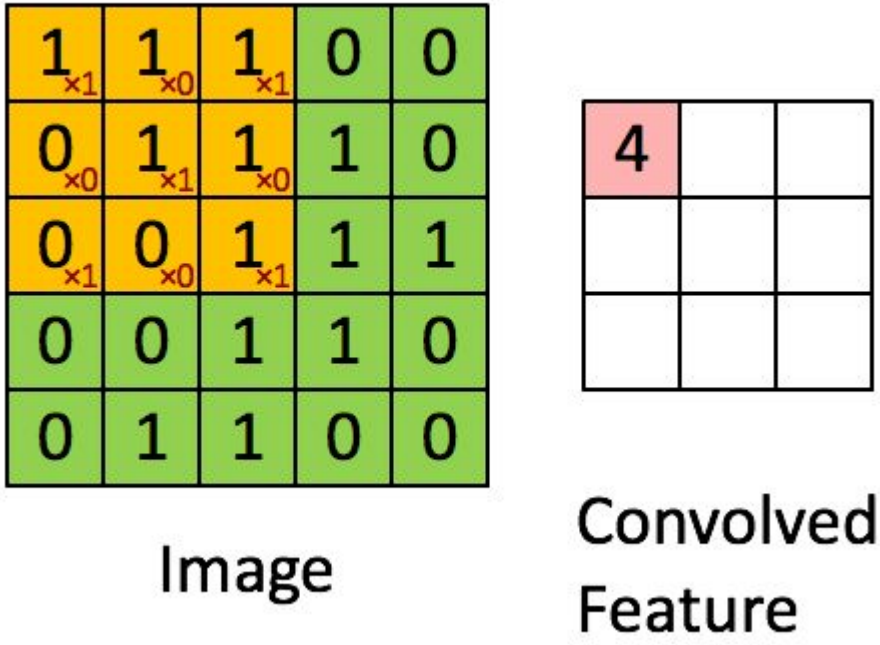
Image from Yue et al, *Bioinformatics* **2020** 36(4)

Graphs are unstructured input data

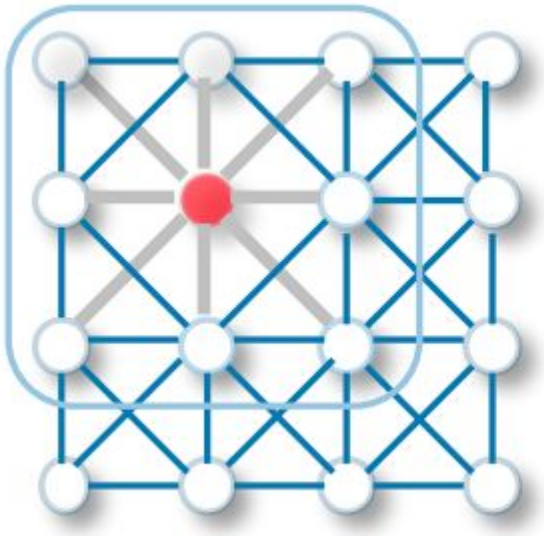




Why convolution?



Convolution on graphs



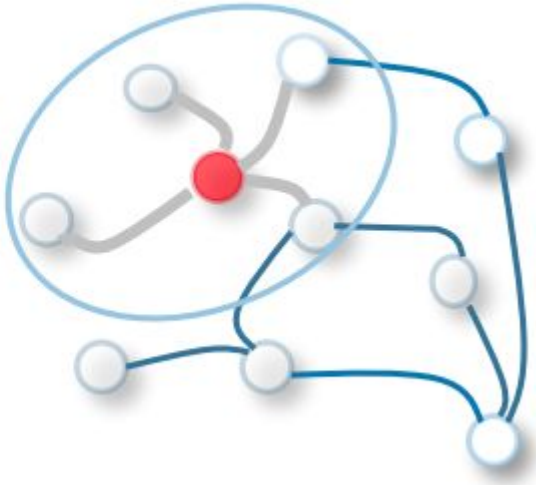
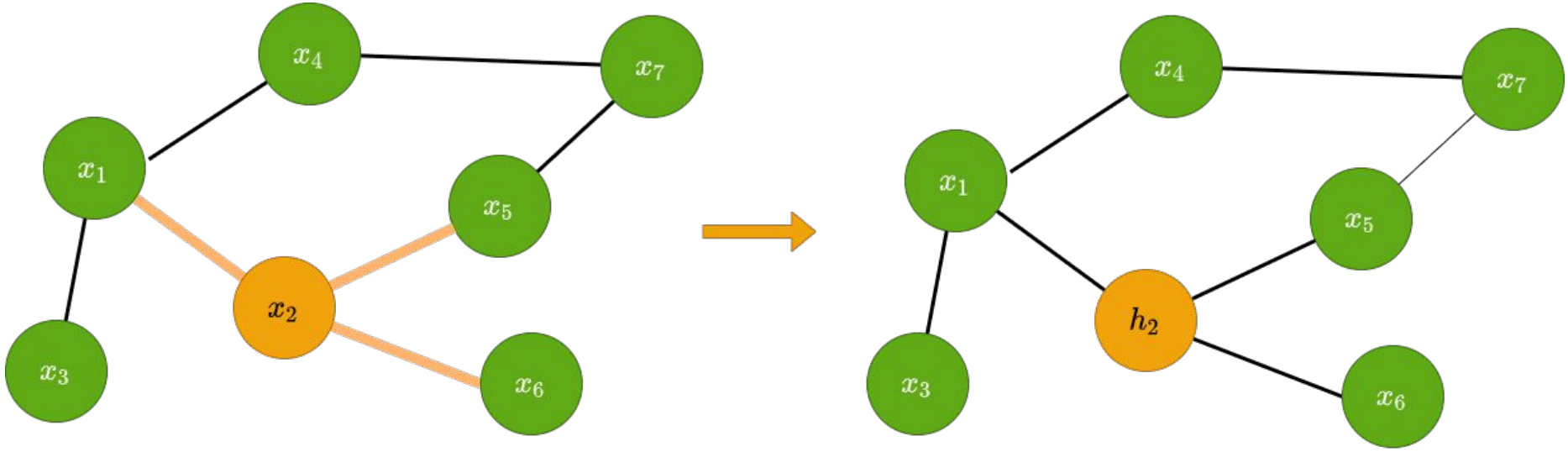


Image from Wu et al, *arXiv* **2018** a comprehensive survey on graph neural networks

Convolution on graphs is a method to learn information from other nodes



$$h_2 = g(x_1, x_5, x_6)$$

Image from the AI summer – GNN architectures

Convolution on graphs is inspired by signal/wave propagation

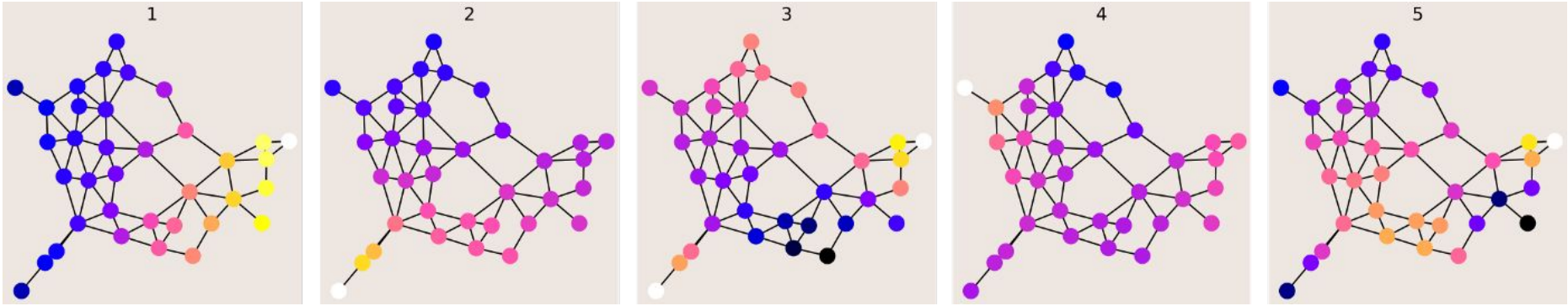
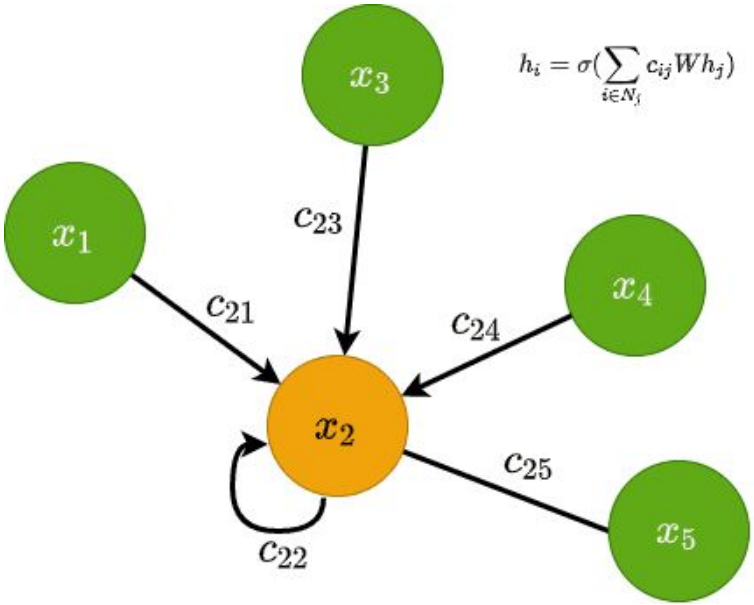
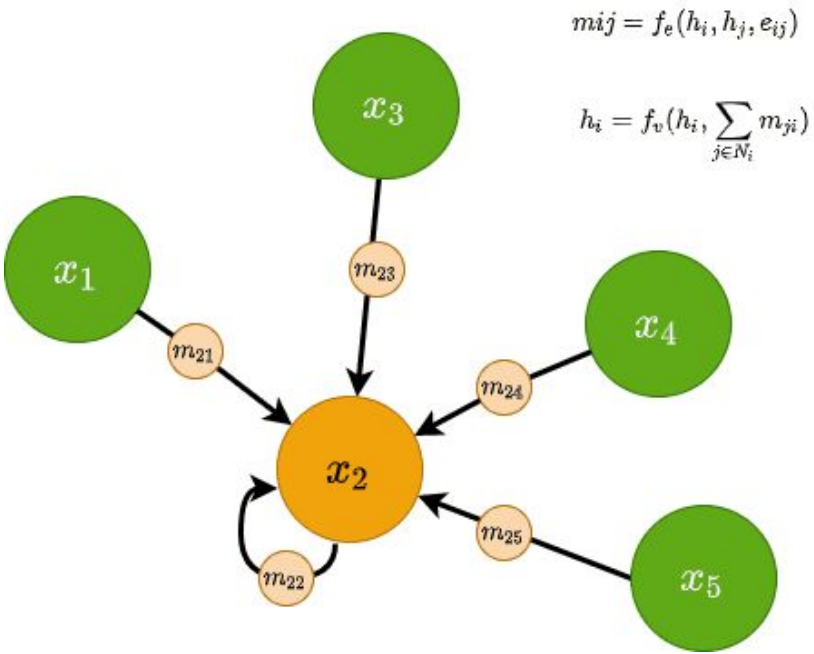


Image from Fracesco Casalengo, TDS – Graph Convolutional Networks

Two methods of aggregating info from other nodes



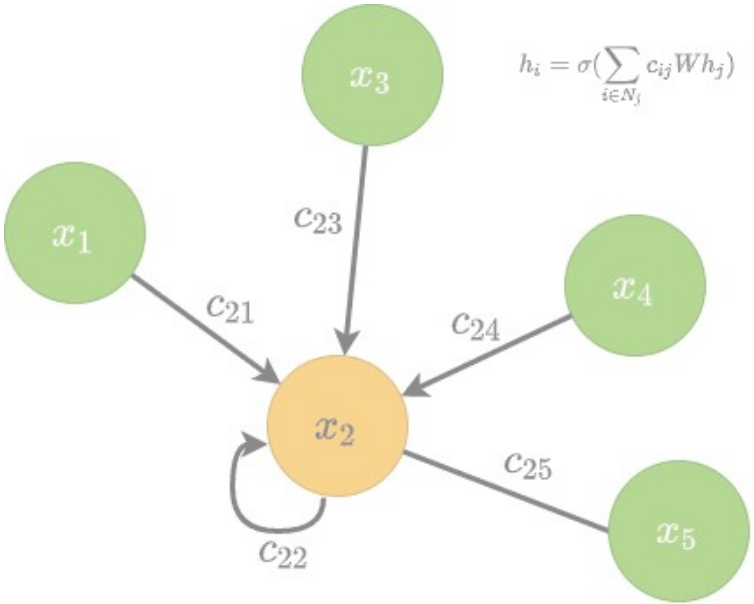
GCN



MPNN

Image from the AI summer – GNN architectures

Two methods of aggregating info from other nodes



GCN

- Add self loop to edges
- Calculate adjacency matrix (A)
- Multiply A by X (features)
- Calculate degree matrix
- Multiply Degree matrix by others

$$\text{normalized features} = D^{-1}AX$$

Image from the AI summer – GNN architectures
Equation from Inneke Mayachita, TDS - Understanding GCN for node classification

Learning happens through convolution and pooling

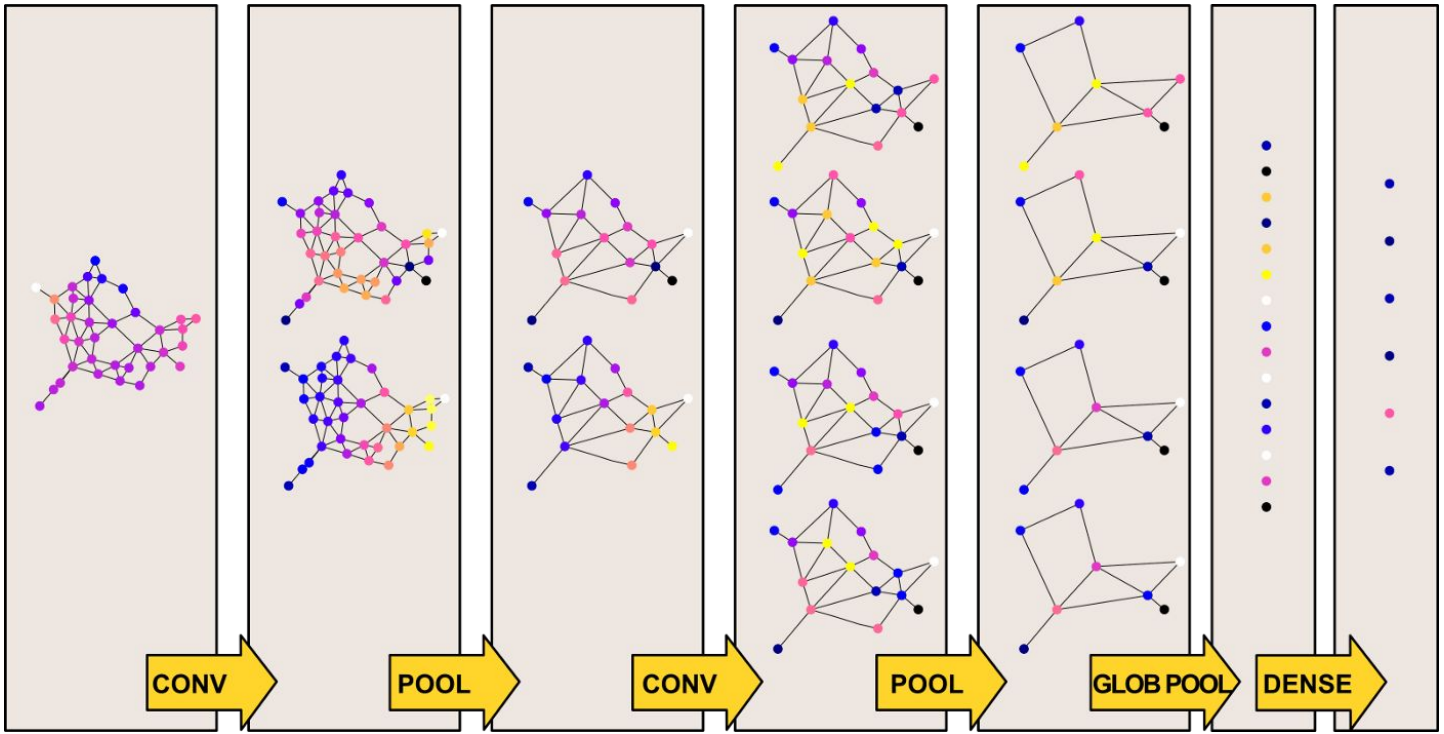
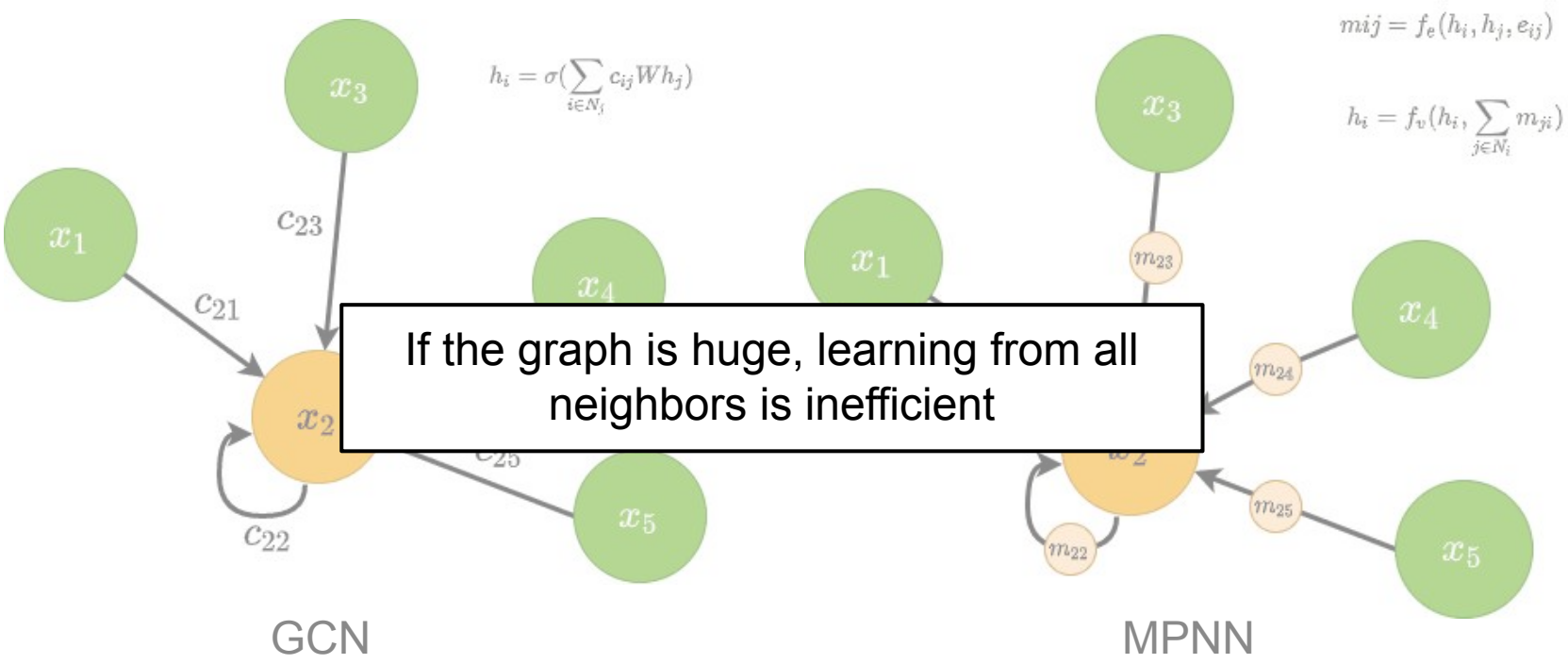


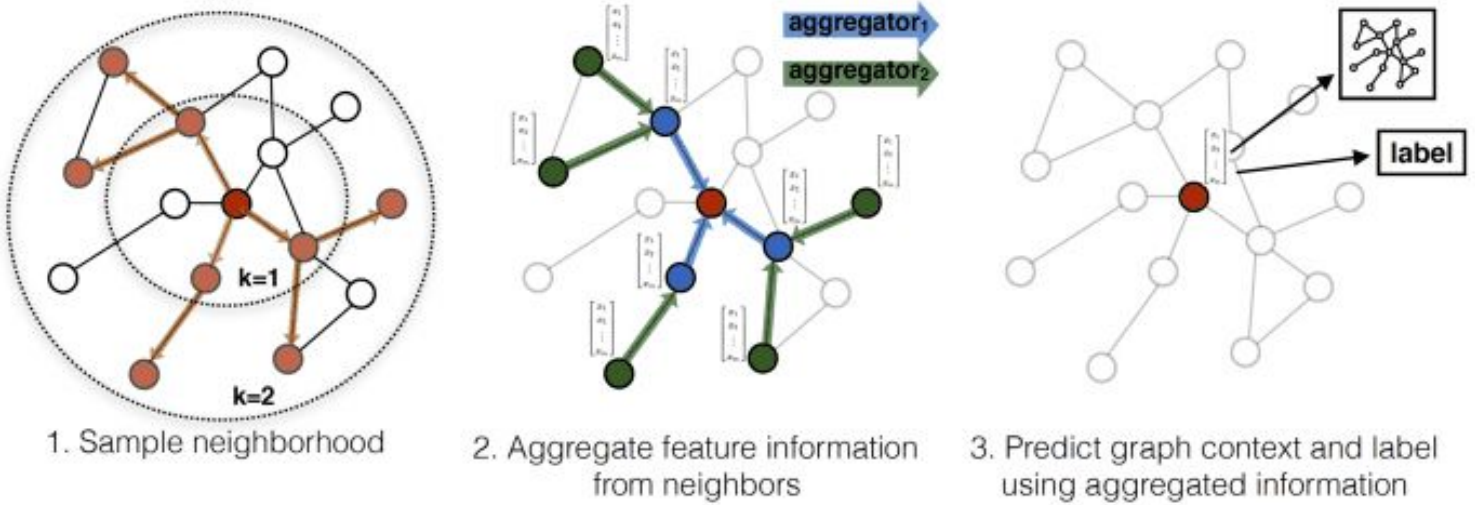
Image from Fracesco Casalengo, TDS – Graph Convolutional Networks

Two methods of aggregating info from other nodes



Sampling methods help use a subset of neighbors for learning

GraphSAGE



1. Sample neighborhood

2. Aggregate feature information from neighbors

3. Predict graph context and label using aggregated information

Image from Hamilton, et al. ArXiv:1706.02216 [Cs, Stat], Sept. 2018

Sampling methods help use a subset of neighbors for learning

PinSAGE

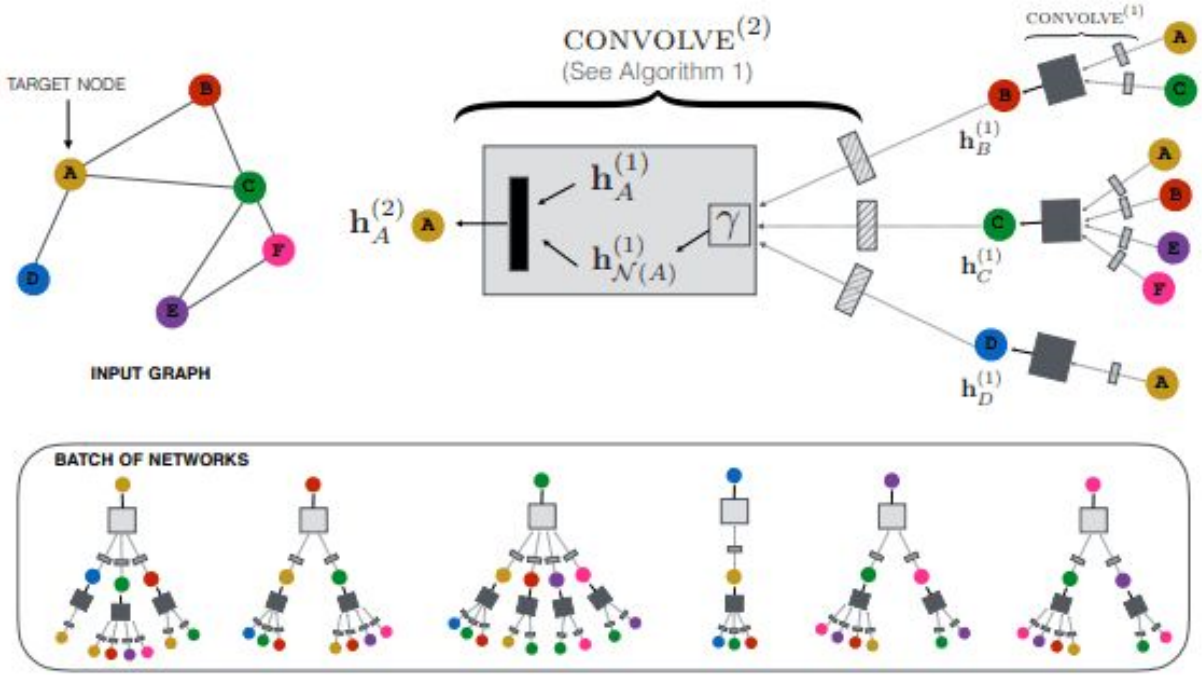


Image from Ying, et al. ArXiv:1806.01973 [Cs, Stat], June 2018

Inductive vs transductive learning

Inductive

Model only sees training data

Label prediction for unseen data

→ generalize well, hard to capture the complete structure

Inductive vs transductive learning

Inductive

Model only sees training data

Label prediction for unseen data

→ generalize well, hard to capture the complete structure

Transductive

Model has already seen all data, test and training!

→ need to retrain if new nodes are added

Next lecture:

GCNs in action