DATA MINING

Graph Mining

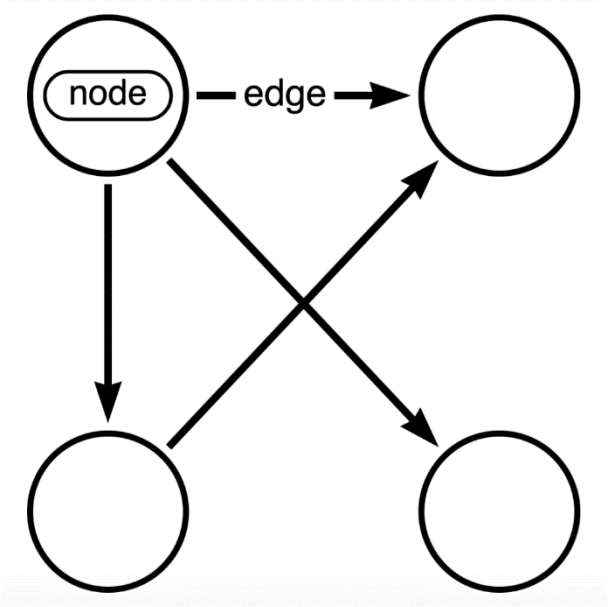
Ashish Pujari

Lecture Outline

1. Graph Data Structure
2. Graph Applications
3. Graph Algorithms
4. Graph Mining

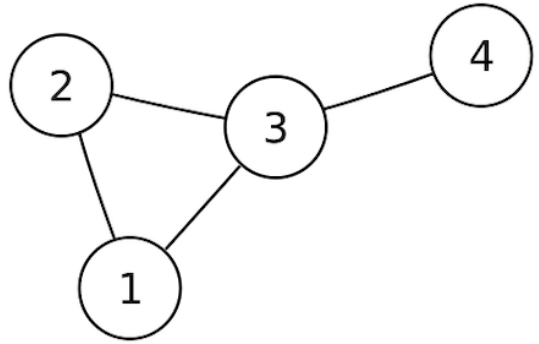
Graph

- Abstract representation with Nodes and Edges
- Nodes represent Entities
  - Person, Place, Thing, etc.
- Edges represent Relationships
  - Arrow represents directionality
- Graph $G = (V, E)$
  - V: set of vertices
  - E: set of edges $\subseteq (V \times V)$

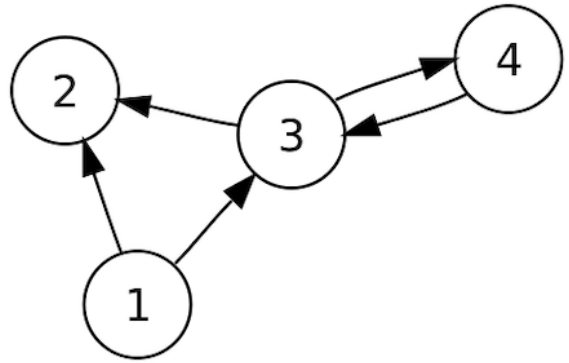


Types of Graphs

- Undirected Graph:
  - edge $(u, v) = (v, u)$; for all $v, (v, v) \notin E$ (No self loops.)

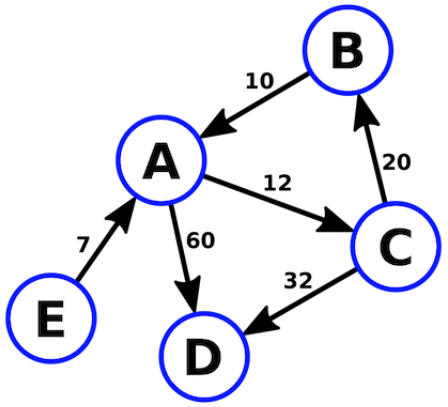


- Directed Graph:
  - (u, v) is edge from u to v, denoted as $u \rightarrow v$. Self loops are allowed.

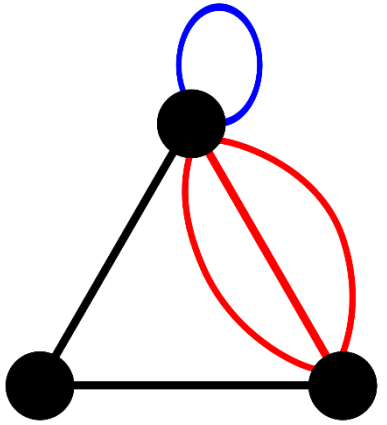


Types of Graphs

- Weighted Graph:
  - Each edge has an associated weight, given by a weight function $w : E \rightarrow R$.

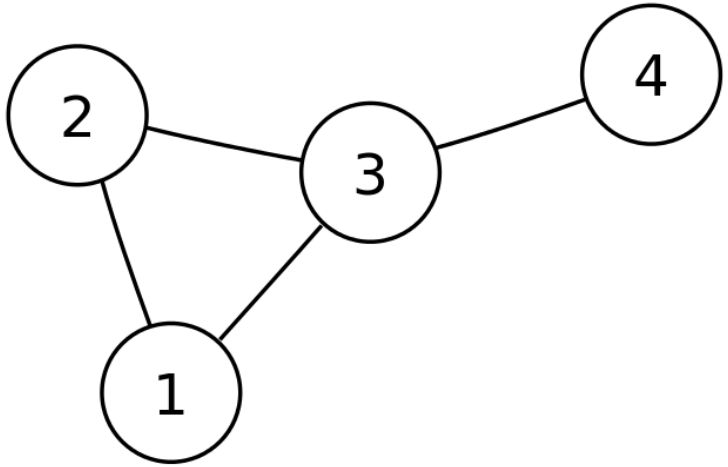


- Multi-Graph:
  - Graph in which multiple edges between nodes are either permitted or required

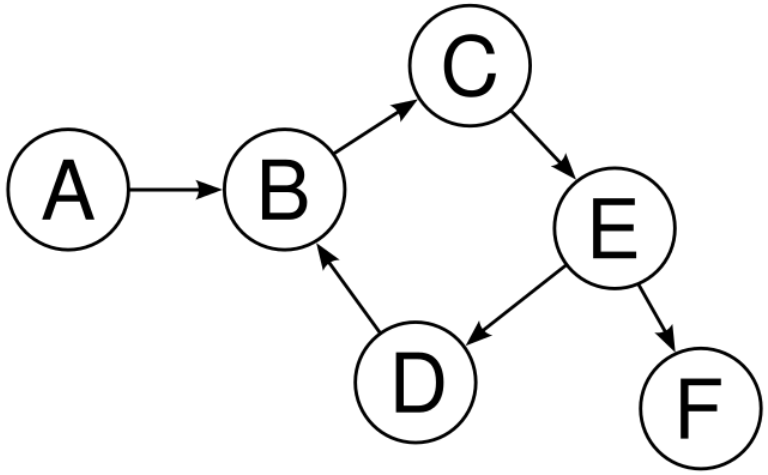


Directed Vs Undirected Graphs

Undirected Graph



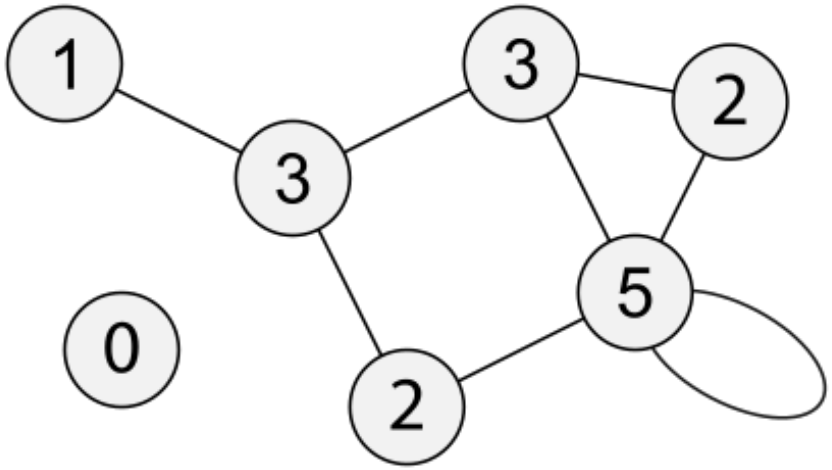
Directed Graph



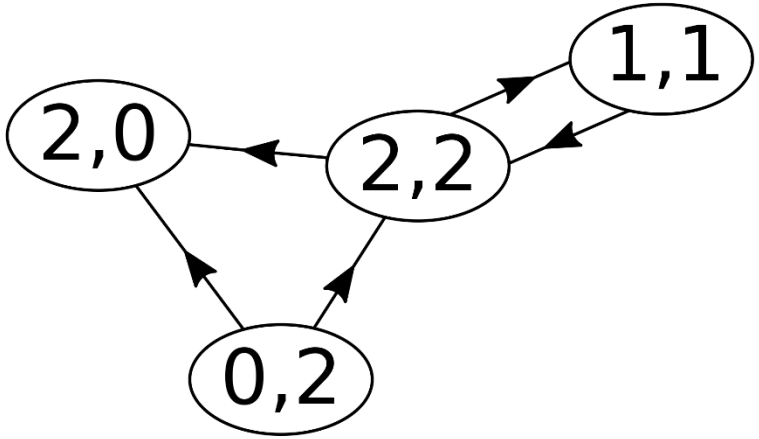
Sets	Undirected Graph	Directed Graph
Vertices	$\{1, 2, 3, 4\}$	$\{A, B, C, D, E, F\}$
Edges	$\{(1,2), (2,1), (2,3), (3,2), (1,3), (3,1), (3,4), (4,3)\}$	$\{(A,B), (B,C), (C,E), (E,D), (D,E), (E,F)\}$

Degree

- Degree (or valency) of a vertex of a graph is the number of edges that are incident to the vertex
- In a multigraph, a loop contributes 2 to a vertex's degree, for the two ends of the edge.



A graph with a loop having vertices labeled by degree

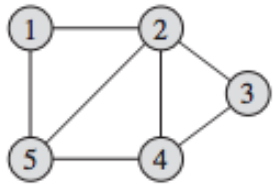


A directed graph with vertices labeled (indegree, outdegree)

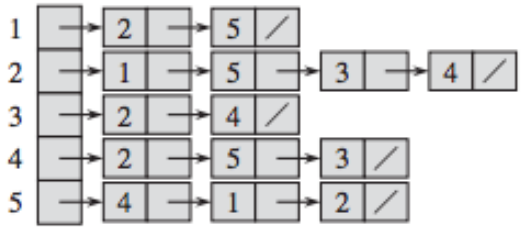
Graph Representation

There are two standard ways to represent a graph $G = (V, E)$:

- A collection of adjacency lists
  - best for **sparse** graphs
  - $|E| << |V|^2$
- An adjacency matrix
  - preferred when the graph is **dense**
  - $|E| \approx |V|^2$



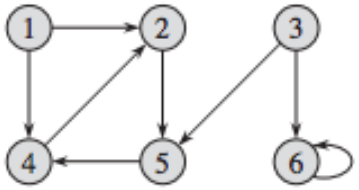
(a)



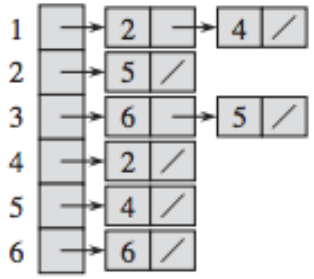
(b)

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

(c)



(a)



(b)

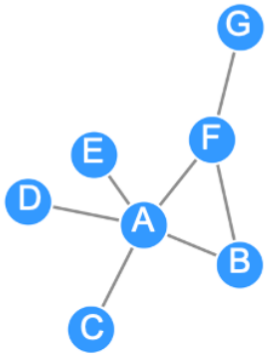
	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

(c)

https://notes.shichao.io/clrs/

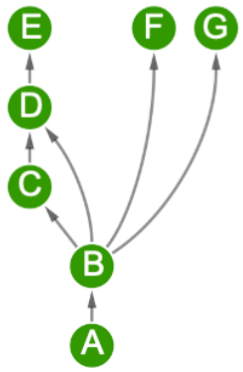
Adjacency Matrices

Undirected



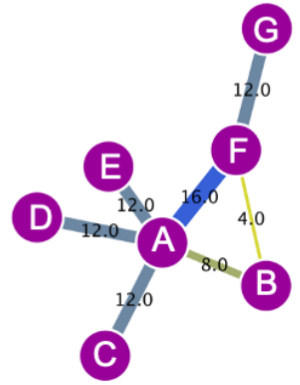
	A	B	C	D	E	F	G	Degree
A	0	1	1	1	1	1	0	5
B	1	0	0	0	0	1	0	2
C	1	0	0	0	0	0	0	1
D	1	0	0	0	0	0	0	1
E	1	0	0	0	0	0	0	1
F	1	1	0	0	0	0	1	3
G	0	0	0	0	0	1	0	1

Directed



	A	B	C	D	E	F	G	Out-degree
A	0	1	0	0	0	0	0	1
B	0	0	1	1	0	1	1	4
C	0	0	0	1	0	0	0	1
D	0	0	0	0	1	0	0	1
E	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0

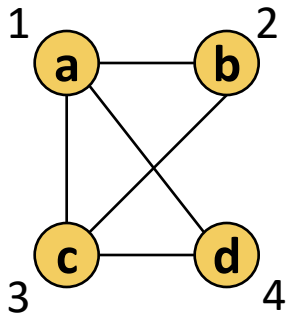
Weighted



	A	B	C	D	E	F	G	Degree
A	0	8	12	12	12	16	12	72
B	8	0	0	0	0	4	0	12
C	12	0	0	0	0	0	0	12
D	12	0	0	0	0	0	0	12
E	12	0	0	0	0	0	0	12
F	16	4	0	0	0	0	12	32
G	12	0	0	0	0	12	0	24

Adjacency Relationship

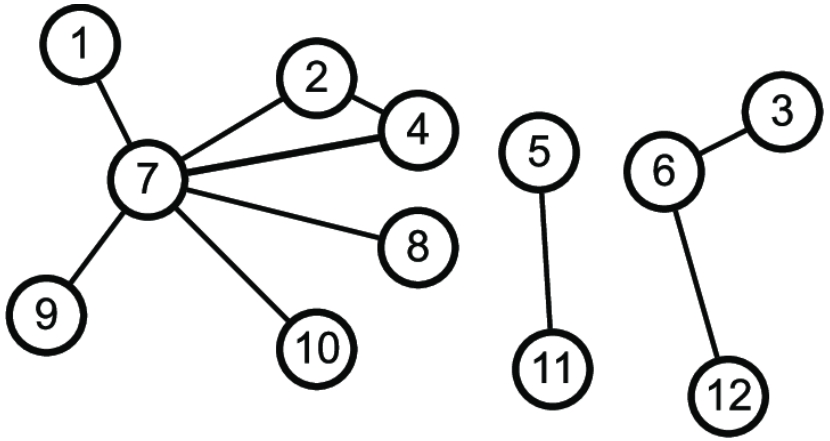
- If $(u, v) \in E$, then vertex v is adjacent to vertex u.
- Adjacency relationship is:
  - Symmetric if G is undirected.
  - Not necessarily so if G is directed.
- If G is connected:
  - There is a path between every pair of vertices.
  - $|E| \geq |V| - 1$.
  - Furthermore, if $|E| = |V| - 1$, then G is a tree.



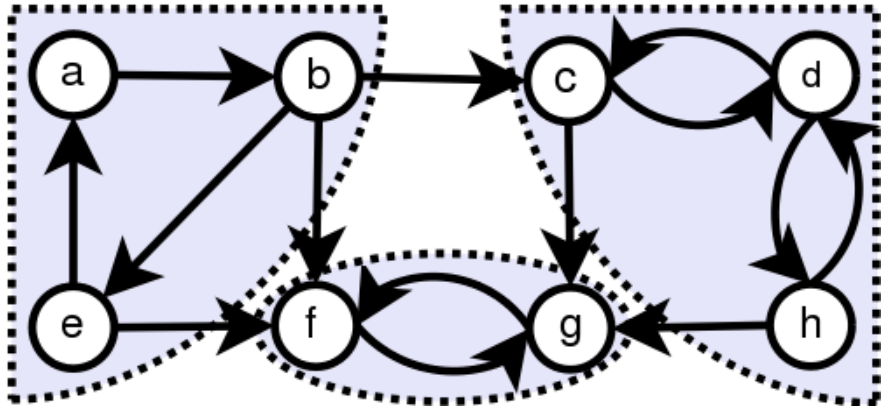
	1	2	3	4
1	0	1	1	1
2	1	0	1	0
3	1	1	0	1
4	1	0	1	0

Connected Components

- Undirected Graphs
  - Subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the super-graph.



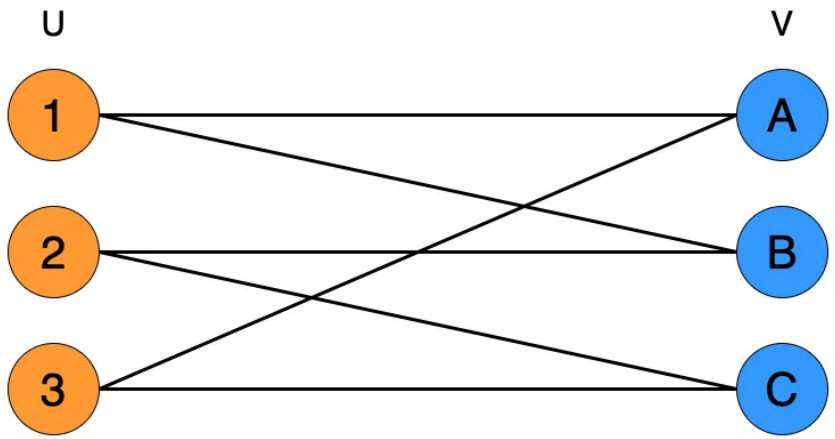
- Directed graphs
  - Strongly connected means there exists a directed path between every pair of nodes.
  - Weakly connected means that there exist a path between every pair of nodes, regardless of direction
  - A strongly connected graph is also weakly connected but the converse is not true.



Source: Wikipedia

Bipartite Graph

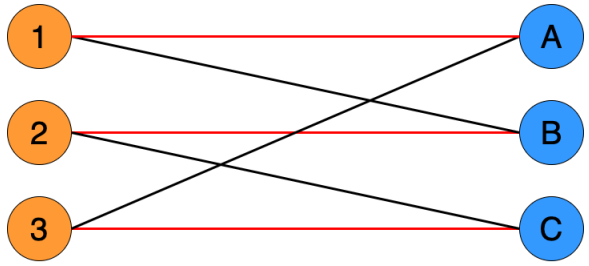
- A graph whose vertices can be divided into two disjoint and independent sets U and V (known as bipartitions) such that every edge connects a vertex in U to one in V.
- Each edge is incident on one vertex in U and one vertex in V. There will not be any edges connecting two vertices in U or two vertices in V.



Bipartite Graph

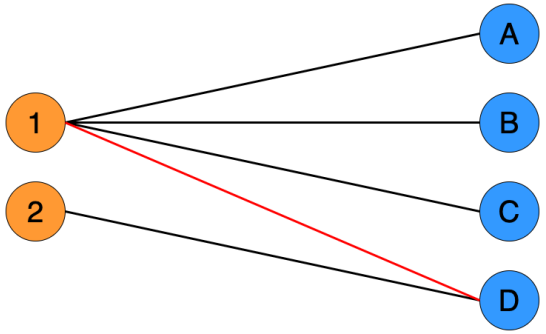
Perfect Matching

- Each node has exactly one edge incident on it. For a perfect matching, we should have |U|=|V|.



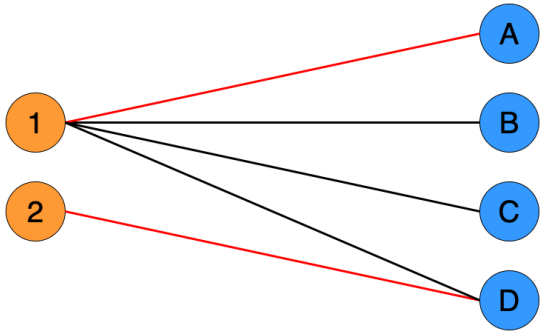
Maximal Matching

- No other edges can be added to the matching because every vertex is matched to another vertex.



Maximum Cardinality Matching

- Contains the largest possible number of edges.



Storage Requirement

Directed graphs

- Sum of lengths of all adj. lists is
- $\sum$ out-degree$(v) = |E|$
- Total storage: $\Theta(V + E)$
  - E is No. of edges leaving v
  - $v \in V$

Undirected graphs

- Sum of lengths of all adj. lists is
- $\sum$ degree(v) = 2|E|
- Total storage: $\Theta(V + E)$
  - E is No. of edges incident on v.
  - Edge (u,v) is incident on vertices u and v.
  - $v \in V$

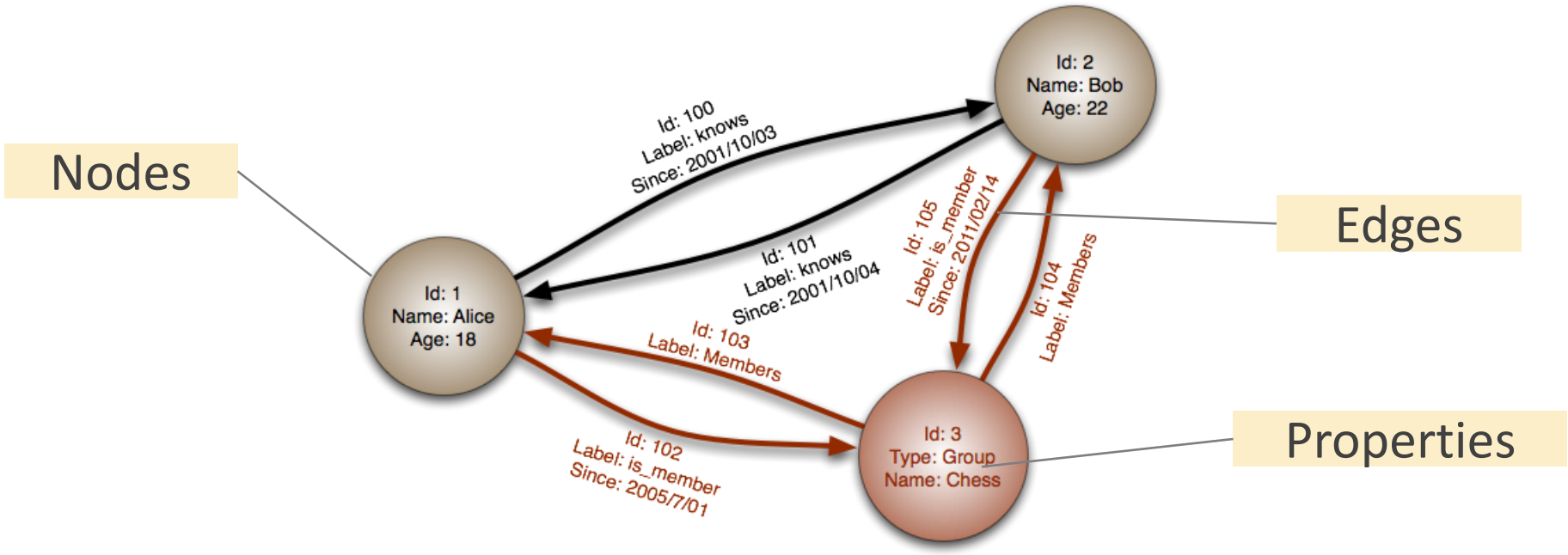
Space and Time

- Space: $\Theta(V^2)$.
  - Not memory efficient for large graphs.
- Time: to list all vertices adjacent to u: $\Theta(V)$.
- Time: to determine if $(u, v) \in E$: $\Theta(1)$.
- Can store weights instead of bits for weighted graph.

GRAPH APPLICATIONS

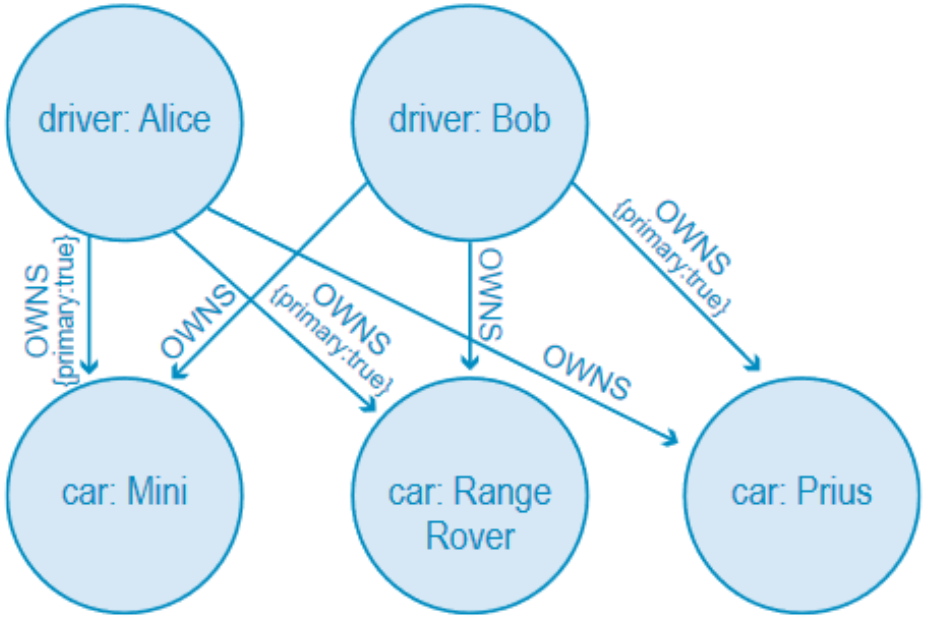
Examples and Applications

Graph Example

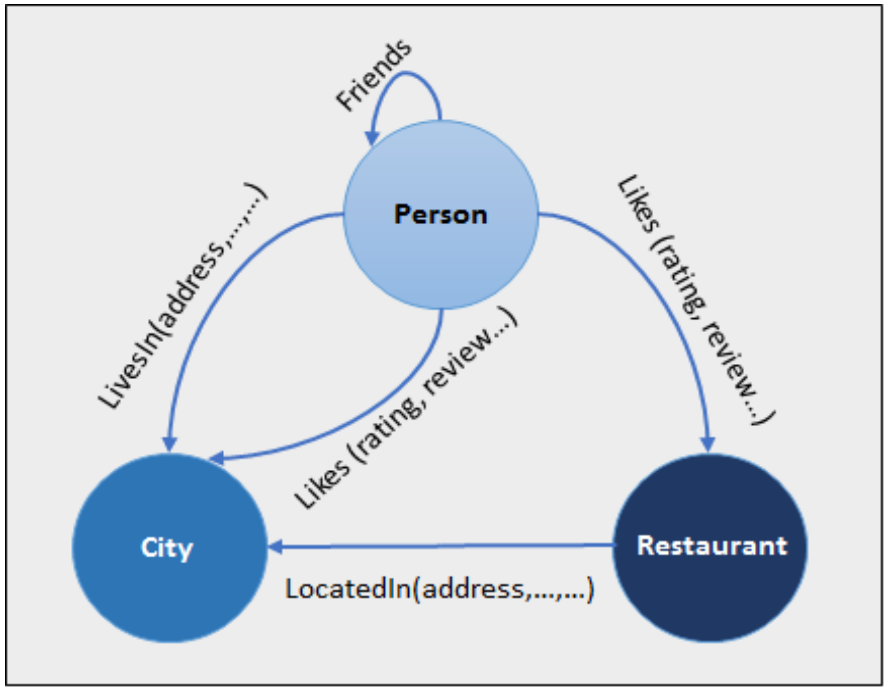


Source: Wikipedia

Graph Example: Ownership, Ratings

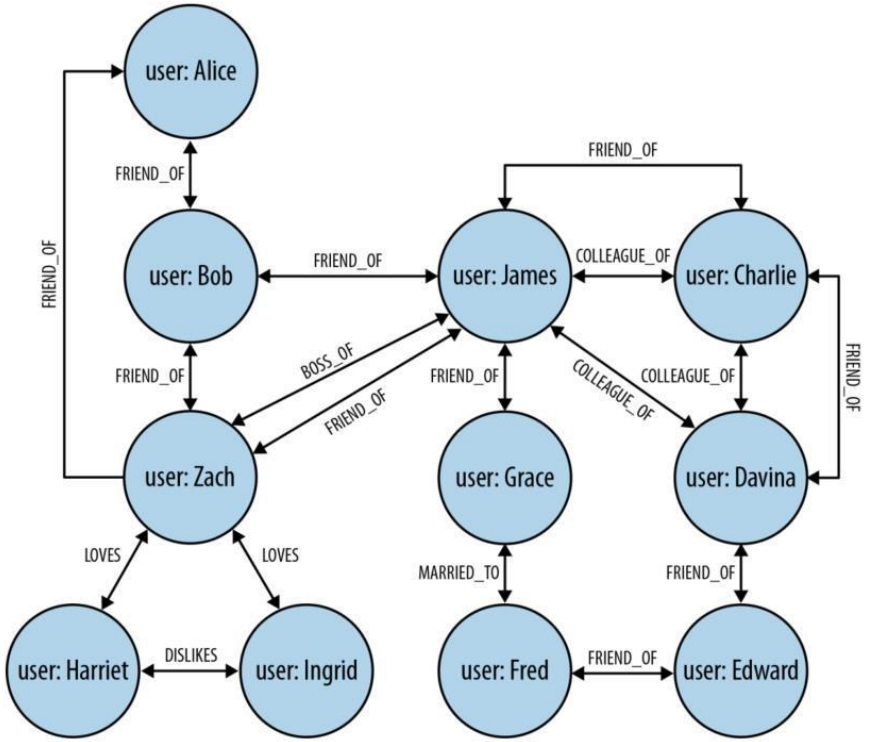


Car Ownership Graph

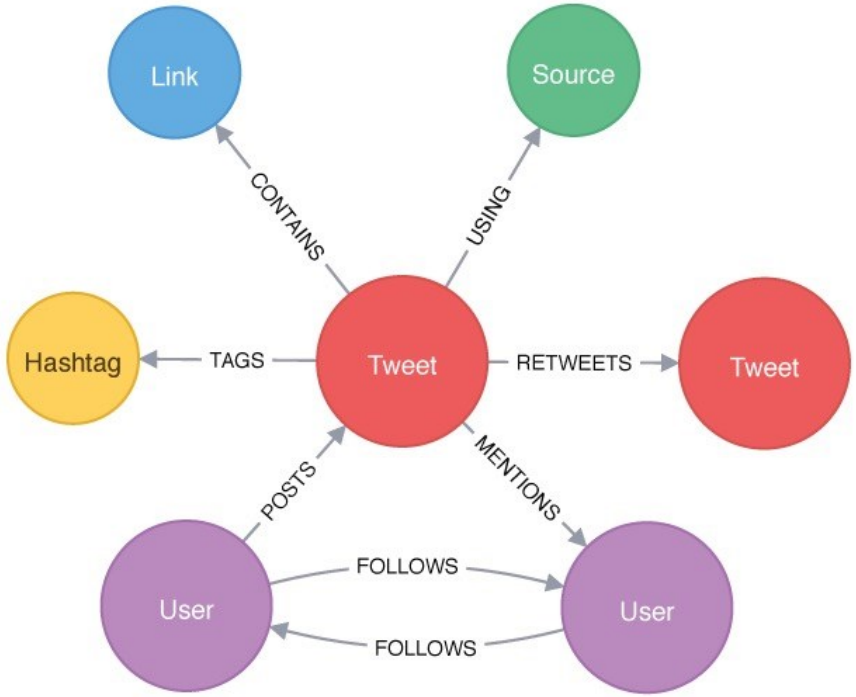


Restaurant, Person Graph

Graph Example: Social Networks

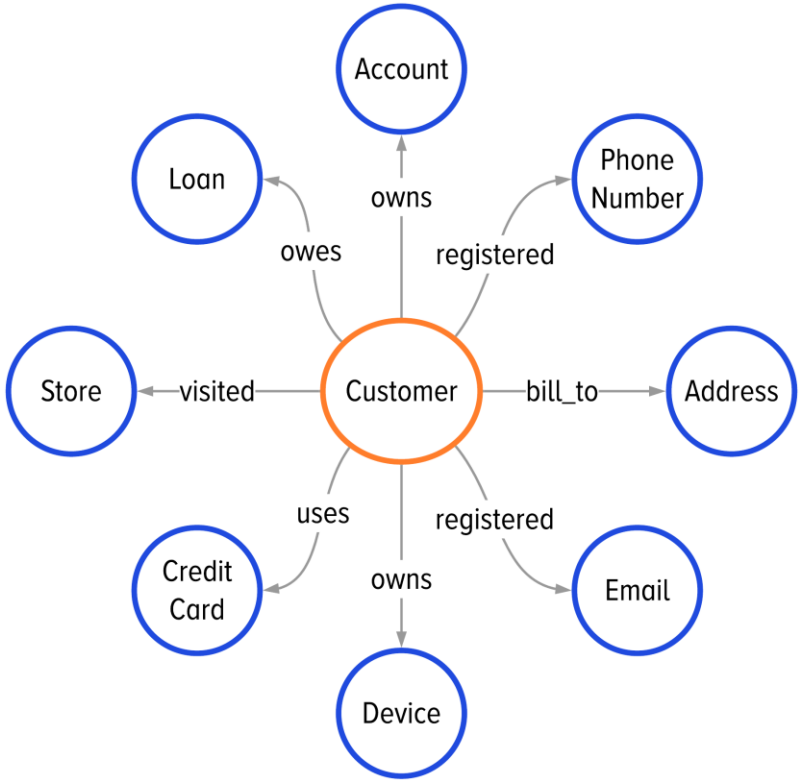


Social Network Graph

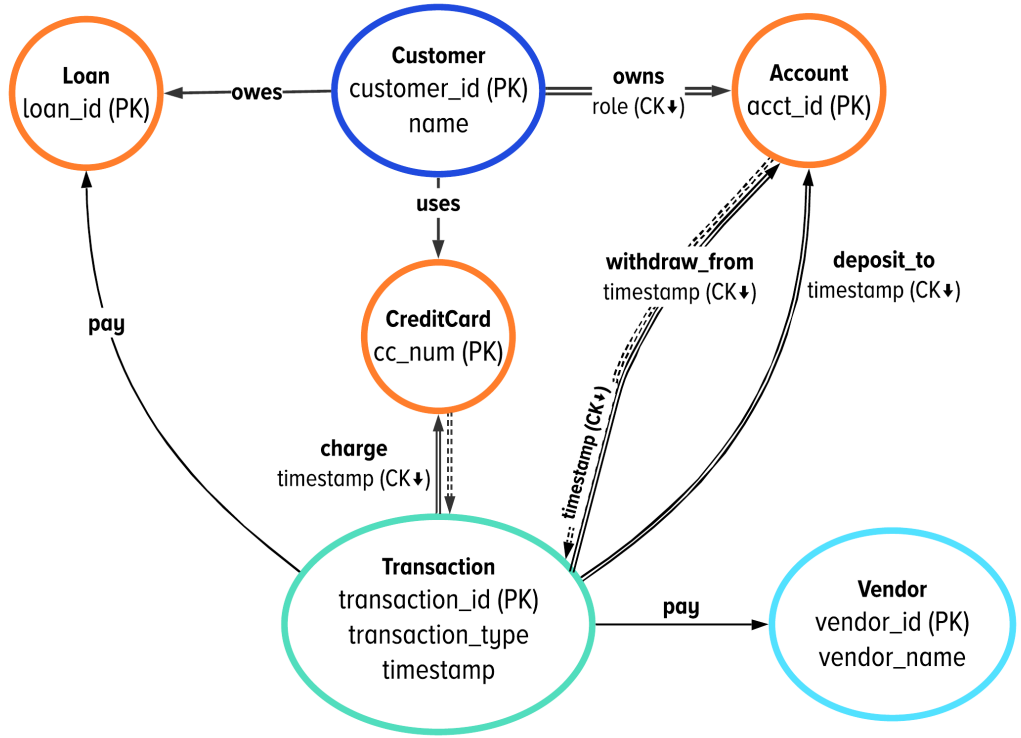


Twitter Graph

Graph Example: Financial



Customer Graph

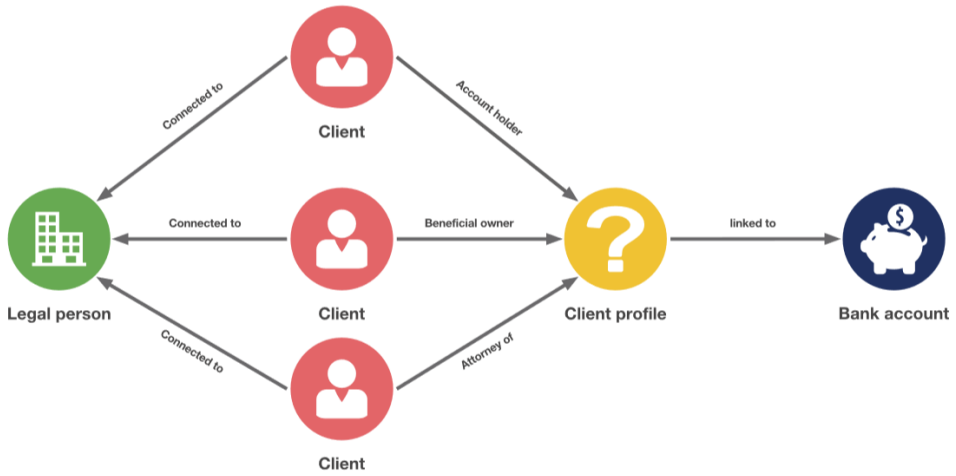


Transaction Graph

The Practitioner's Guide to Graph Data

Financial Fraud Detection

- Anti-Money Laundering
  - ICIJ - Panama Papers
  - 100,000 clients from 203 countries
  - 275,000 nodes with 400,000 relationships

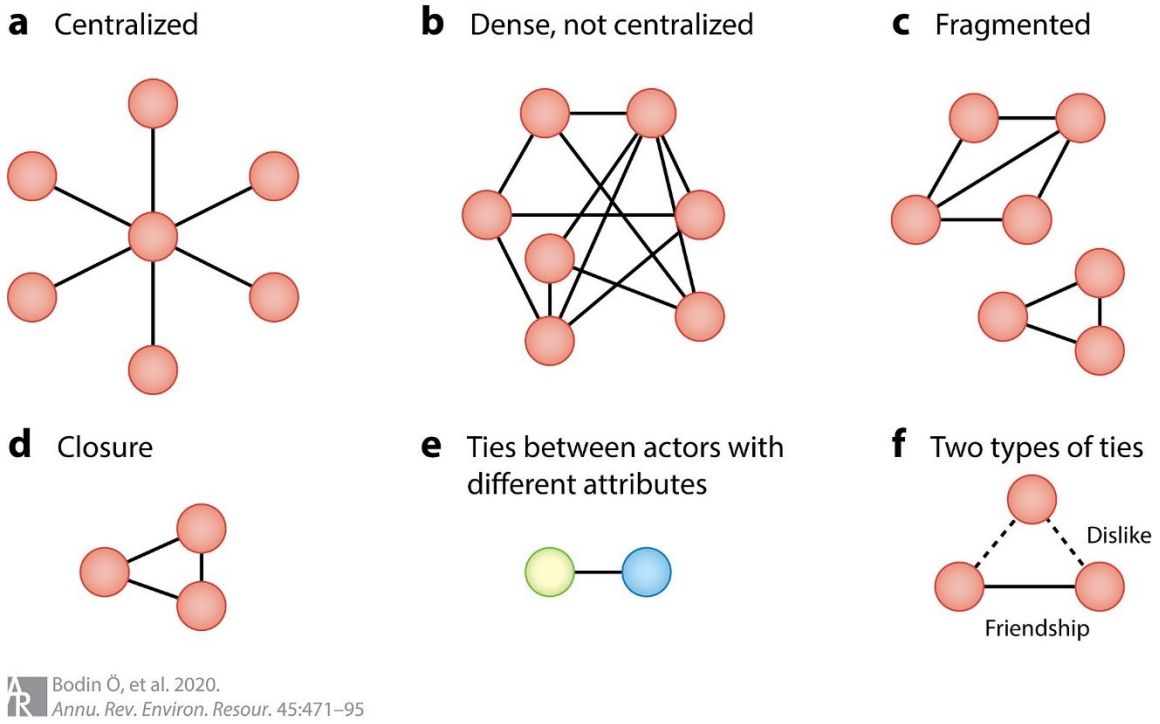


Schema representing the entities and connections within the Swiss Leaks dataset.

https://www.neo4j.com

Social Network Analytics

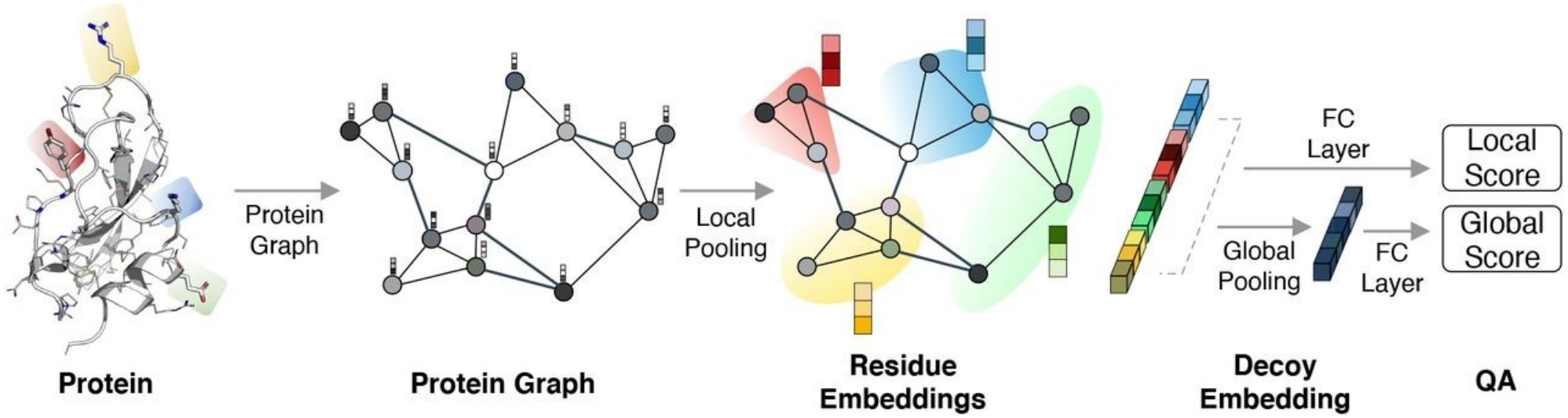
- Network analysis applications
  - Network modeling and sampling
  - Network propagation modeling
  - User behavior analysis
  - Location-based interaction analysis
  - Recommender systems
  - Link prediction (future collaboration)
  - Entity resolution
  - Social Networking Potential (SNP)



Source: Wikipedia

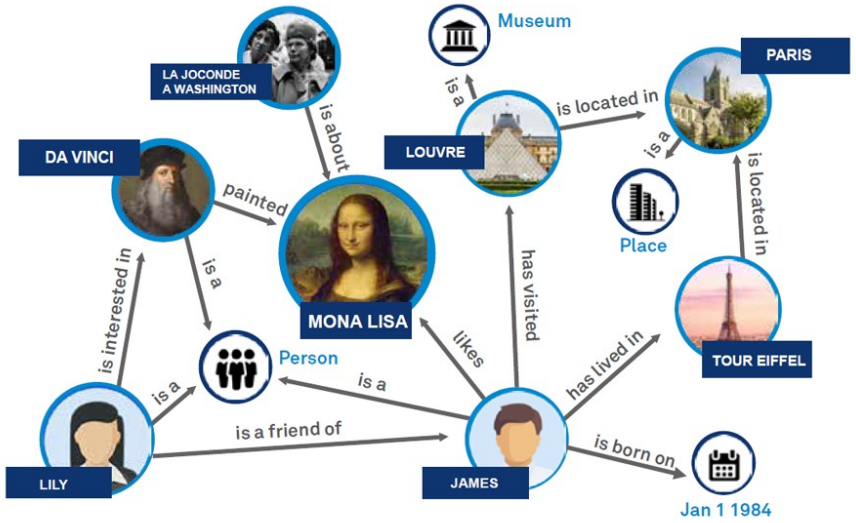
Protein Modeling

- Predicting protein function (graph classification)
- Searching for compounds with certain substructures (graph similarity search)
- Protein model refinement



https://www.biorxiv.org/content/10.1101/2020.04.06.028266v1.full

Knowledge Graphs



Mahatma Gandhi

Indian lawyer

Mohandas Karamchand Gandhi was an Indian lawyer, anti-colonial nationalist, and political ethicist, who employed nonviolent resistance to lead the successful campaign for India's independence from British rule, and in turn inspired movements for civil rights and freedom across the world. Wikipedia

Born: 2 October 1869, Porbandar

Full name: Mohandas Karamchand Gandhi

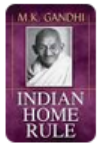
Assassinated: 30 January 1948, New Delhi

Spouse: Kasturba Gandhi (m. 1883–1944)

Books

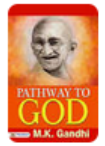
View 40+ more

Satya ke Prayog
1927



Hind Swaraj or Indian H...
1909

The Essential Gandhi
1962



Pathway to God
1971

Quotes

View 7+ more

An eye for eye only ends up making the whole world blind.

Happiness is when what you think, what you say, and what you do are in harmony.

The weak can never forgive. Forgiveness is the attribute of the strong.

Google

Technology company

google.com

Google LLC is an American multinational technology company that specializes in Internet-related services and products, which include online advertising technologies, a search engine, cloud computing, software, and hardware. Wikipedia

CEO: Sundar Pichai (2 Oct 2015–) Trending

Founded: 4 September 1998, Menlo Park, California, United States

Parent organization: Alphabet Inc.

Headquarters: Mountain View, California, United States

Subsidiaries: YouTube, Google China, YouTube TV, Fitbit, MORE

Founders: Larry Page, Sergey Brin

Signs

Ip address

Disclaimer

Profiles

LinkedIn YouTube Twitter Instagram

Graph Databases

- Online DBMS with CRUD methods that exposes a graph data model
- Can store native graphs - Nodes, Edges, Properties
- Generally built for use with OLTP systems

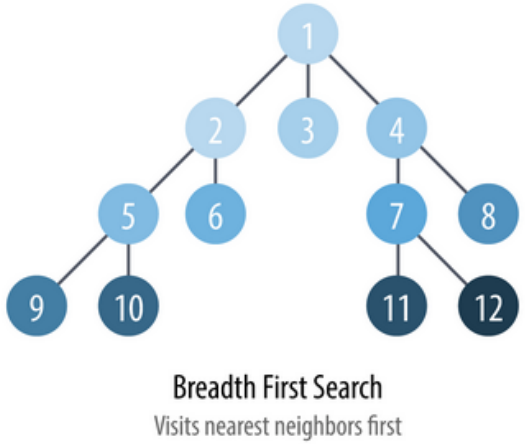
GRAPH ALGORITHMS

Graph Traversal Algorithms

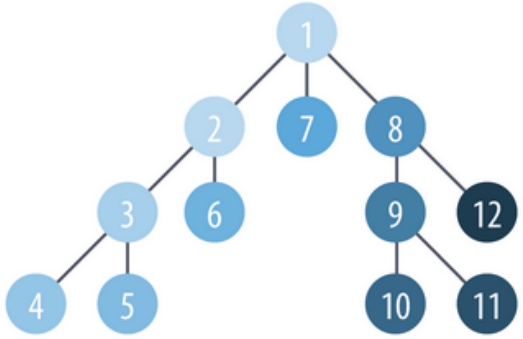
- Traversal (searching) is the process of accessing each vertex (node) of a data structure in a systematic well-defined order
- Used to discover the structure of a graph.
- Standard graph-searching algorithms.
  - Breadth-first Search (BFS).
  - Depth-first Search (DFS).

Graph Traversal Algorithms

Algorithm	Description	Example Usage
Breadth First Search	Traverses a tree structure by fanning out to explore the nearest neighbors and then their sublevel neighbors	Locating neighbor nodes in GPS systems to identify nearby places of interest
Depth First Search	Traverses a tree structure by exploring as far as possible down each branch before backtracking	Discovering an optimal solution path in gaming simulations with hierarchical choices



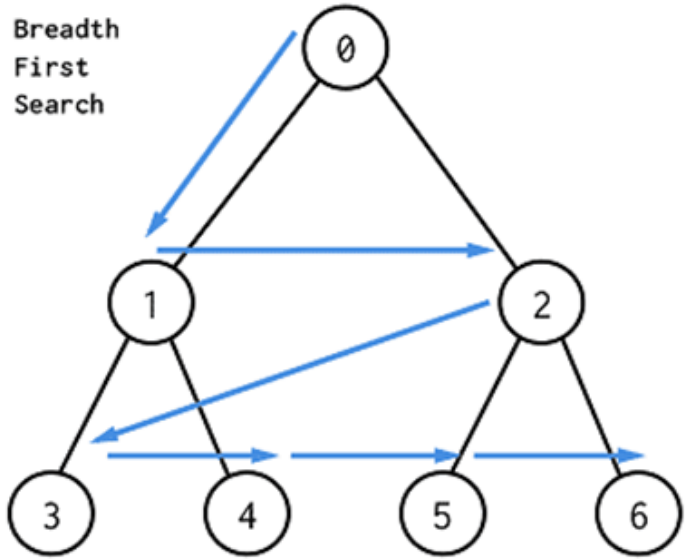
Breadth First Search
Visits nearest neighbors first



Depth First Search
Walks down each branch first

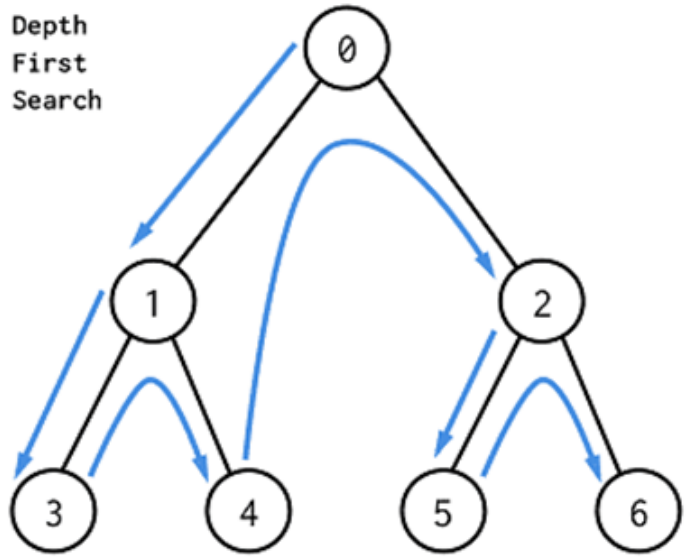
Breadth-first Search (BFS): Algorithm

- Expands the frontier between discovered and undiscovered vertices uniformly across the breadth of the frontier.
  - A vertex is “discovered” the first time it is encountered during the search.
  - A vertex is “finished” if all vertices adjacent to it have been discovered.
- Colors the vertices to keep track of progress.
  - White – Undiscovered.
  - Gray – Discovered but not finished.
  - Black – Finished.
    - Colors are required only to reason about the algorithm. Can be implemented without colors.



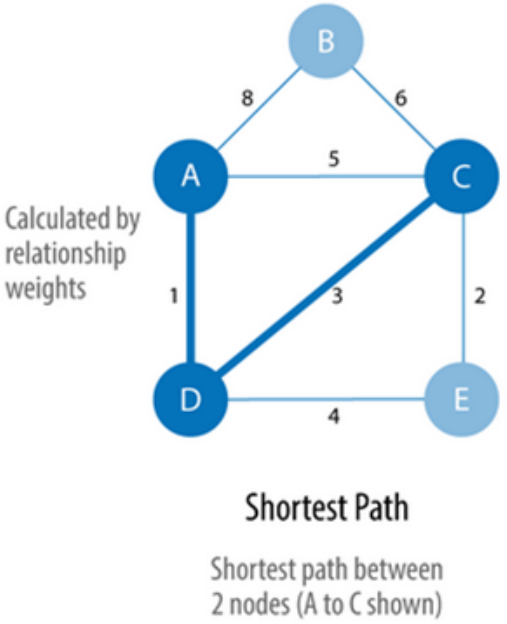
Depth-first Search (DFS): Algorithm

- "Search as deep as possible first."
- Explore edges out of the most recently discovered vertex v.
- When all edges of v have been explored, backtrack to explore other edges leaving the vertex from which v was discovered (its predecessor).
- Continue until all vertices reachable from the original source are discovered.
- If any undiscovered vertices remain, then one of them is chosen as a new source and search is repeated from that source.
- Uses the same coloring scheme for vertices as BFS.



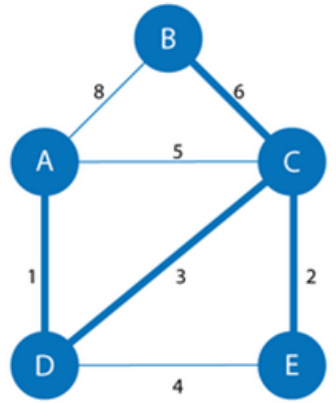
Path Finding Algorithms

Algorithm	Description	Example Usage
Shortest Path	Shortest path between a pair of nodes	Driving directions between two locations
All Pairs Shortest Path	Shortest path between all pairs of nodes in the graph	Evaluating alternate traffic routes
Single Source Shortest Path	Shortest path between a single root node and all other nodes	Least cost routing of phone calls



Path Finding Algorithms

Algorithm	Description	Example Usage
Minimum Spanning Tree	Calculates the path in a connected tree structure with the smallest cost for visiting all nodes	Optimizing connected routing, such as laying cable or garbage collection
Random Walk	Returns a list of nodes along a path of specified size by randomly choosing relationships to traverse.	Augmenting training for machine learning or data for graph algorithms.



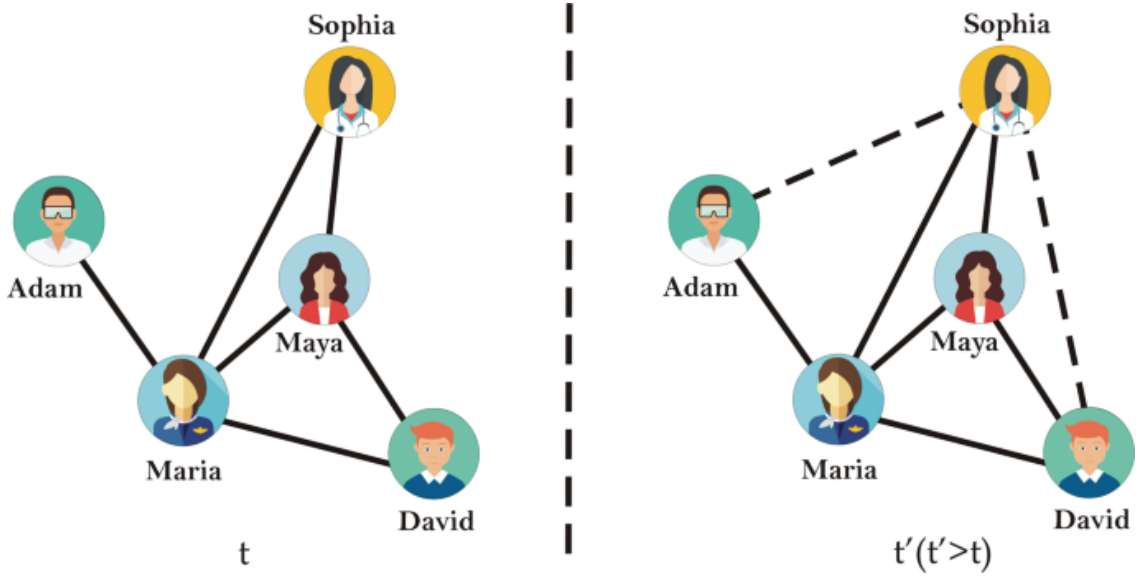
Minimum Spanning Tree

Shortest path **connecting all nodes**
(A start shown)

Traverses to the next unvisited node via the lowest weight **from any visited node**

Link Prediction

- Understand the if/what relationship between entities in graphs.
- Applications
  - Social networks
  - Recommender system
  - Predict fraud, criminal associations, etc.
  - Predict the spread of epidemic diseases
  - Development of vaccination strategies



https://www.nature.com/articles/s41598-019-57304-y

Link Prediction Algorithms

- Adamic Adar algorithm (2003)
  - where $N(u)$ is the set of nodes adjacent to u.
  - A value of 0 indicates that two nodes are not close, while higher values indicate nodes are closer.

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$$

- Common neighbors
  - Two strangers who have a friend in common are more likely to be introduced than those who don't have any friends in common.
  - where $N(x)$ is the set of nodes adjacent to node x, and $N(y)$ is the set of nodes adjacent to node y

$$CN(x,y) = |N(x) \cap N(y)|$$

Centrality Algorithms

- Centrality is a helpful measure for identifying key players in a network.
- Network centrality is among the most well-known social network analysis metrics, measuring the degree to which a person or organization is central to a network.
- Algorithms
  - Degree Centrality
  - Closeness centrality
  - Betweenness centrality
  - Eigenvector Centrality

Degree Centrality

- The degree centrality of a node is simply its degree - the number of edges it has.
- Approach
  - The higher the degree, the more central the node is.
  - Sometimes scaled from 0-1.
- Pros/Cons
  - Simple but crude popularity measure
  - Effective measure since many nodes with high degrees also have high centrality by other measures.
  - Does not recognize a difference between quantity and quality

$$C_D(G) = \sum_{v \in G} \frac{|\deg(v^*) - \deg(v)|}{|H|}$$

$where\ v^*$: vertex with the highest degree

$$H = (|V| - 1)(|V| - 2)$$

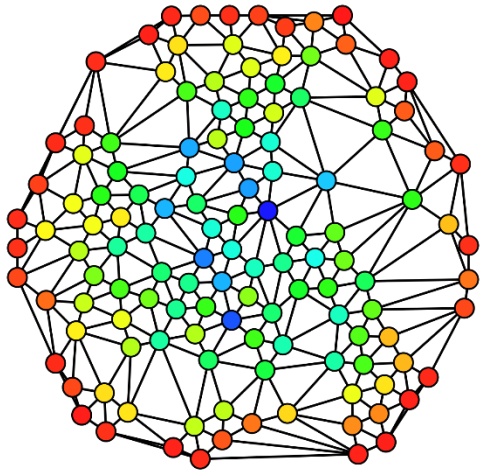
Freeman's general formula

Betweenness Centrality

- Detects the amount of influence a node has over the flow of information in a graph.
- Measure of centrality in a graph based on shortest paths.
- Approach
  - Finds nodes that serve as a bridge from one part of a graph to another. i.e., extent to which a vertex lies on paths between other vertices.
  - Calculates shortest paths between all pairs of nodes in a graph.

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{\mathrm{st}}}$$

where σ_{st} is the total number of shortest paths from node s to t
$\sigma_{st}(v)$ is the number of those paths that pass-through v (not where v is an end point)



Betweenness centrality of each vertex from least (red) to greatest (blue).

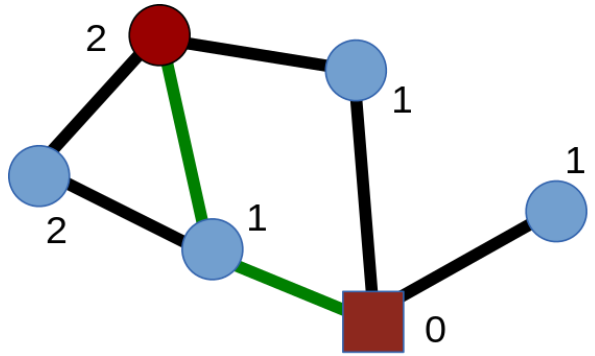
Source: Wikipedia

Closeness Centrality

- In a connected graph, the normalized closeness centrality (or closeness) of a node is the average length of the shortest path between the node and all other nodes in the graph.
- Approach
  - It is usually expressed as the normalized inverse of the sum of the topological distances in the graph. This sum is also known as the farness of the nodes.
  - This normalization allows comparisons between nodes of graphs of different sizes

$$C(v) = \frac{N-1}{\sum_u d(u,v)}$$

where N is the number of nodes in the graph
where $d\ (u\ ,v\ )$ is the distance between vertices $u\ and\ v$



The number next to each node is the shortest path to the square red node. The green edges illustrate one of the two shortest paths between the red square node and the red circle node. The closeness of the red square node is therefore 5/(1+1+1+2+2) = 5/7.

Source: Wikipedia

Eigenvector Centrality

- Measures the transitive influence of nodes.
- A high eigenvector score means that a node is connected to many nodes who themselves have high scores.
- Approach
  - Uses the adjacency matrix and computes the eigenvector associated with the largest absolute eigenvalue.
  - Relationships originating from high-scoring nodes contribute more to the score of a node than connections from low-scoring nodes.
  - The PageRank algorithm is a variant of Eigenvector Centrality with an additional jump probability.

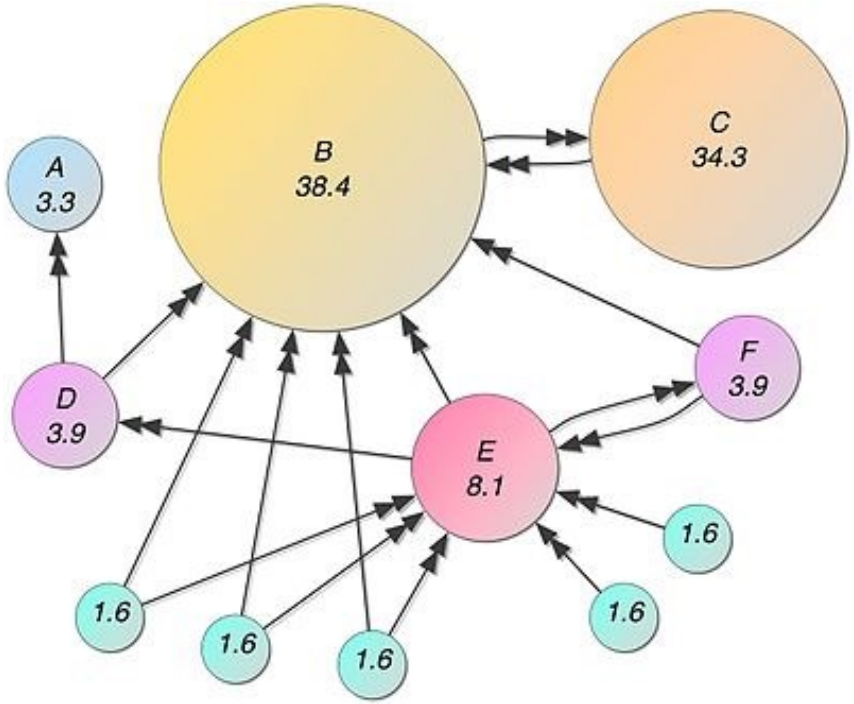
Source: Wikipedia

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in W} a_{v,t} x_t$$

where x_v is centrality of vertex v
$a_{v,t}$ is the adjacency matrix
$M(v)$ is the set of neighbors of v
λ is a constant (eigenvalue)

PageRank

- Link analysis algorithm that assigns a numerical weighting to each element of a hyperlinked set of documents (e.g. www) with the purpose of measuring its relative importance within the set
- Approach
  - PR judges the “value of a page” by looking at the quantity and quality of other pages that link to it
  - If a URL (page) is referenced the most by other URLs then its rank increases, because being referenced means that it is important.
  - If an important URL references other URLs this will also increase the destination’s ranking
- Applications
  - Search engines use PR for ordering page display during a search.
  - Enterprise information retrieval



http://infolab.stanford.edu/~backrub/google.html

PageRank: Algorithm

- PR outputs a probability distribution that represents the likelihood that a person randomly clicking on web links will arrive at a particular web page
- Any page's PR is derived in large part from the PR of other pages
- We assume page A has pages B, C, D which point to it (i.e., are citations).
  - The PageRank of a page A is given as follows:

$$PR(A) = \frac{1\text{-}\,d}{N} + d\,(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} \ldots\ )$$

  - N: total number of documents in the collection
  - d is a damping factor which can be set between 0 and 1. (usually set to 0.85)
  - $L(A)$ is defined as the number of links going out of page A.

PageRank: Algorithm

- PageRank Computation

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

- Adjacency function
  - $\ell\ (p_i, p_j)$ is the ratio between number of links outbound from page j to page i to the total number of outbound links of page j.
  - $\ell\ (p_i, p_j) = 0$ if the pages don't link to each other.

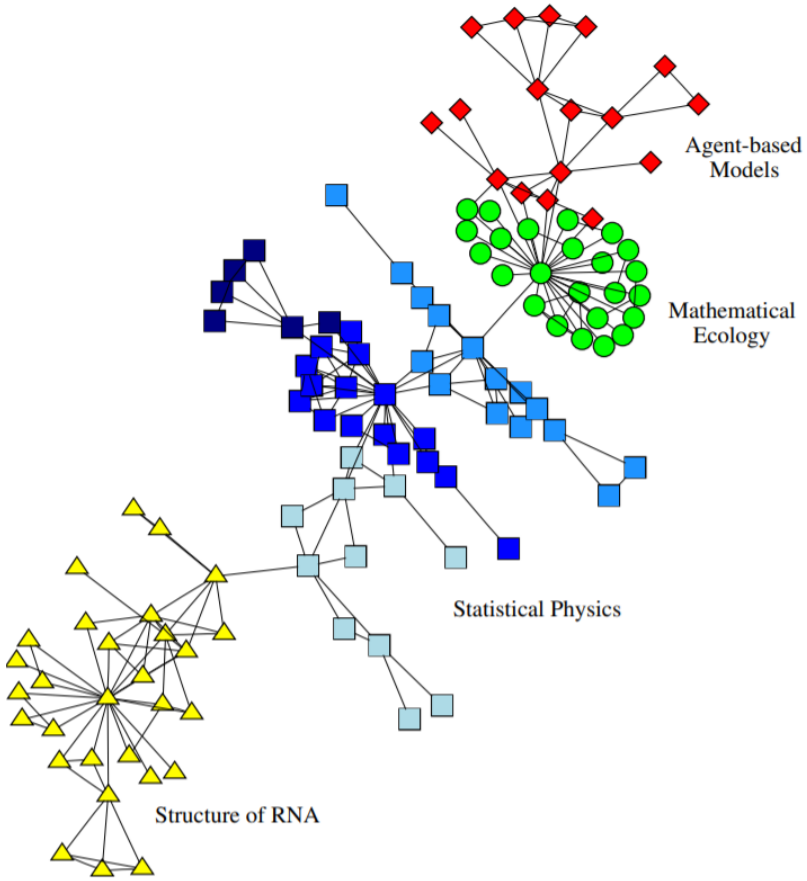
$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

- PageRank values are the entries of the dominant right eigenvector

$$\mathbf{R} = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix}$$

Clustering Algorithms

- A cluster is defined as a group of nodes that are more connected within themselves than with the rest of the network.
- Clustering algorithms (community detection) exclusively use the topology of the network
- Algorithmically complex for large networks hence many approximation methods are used.



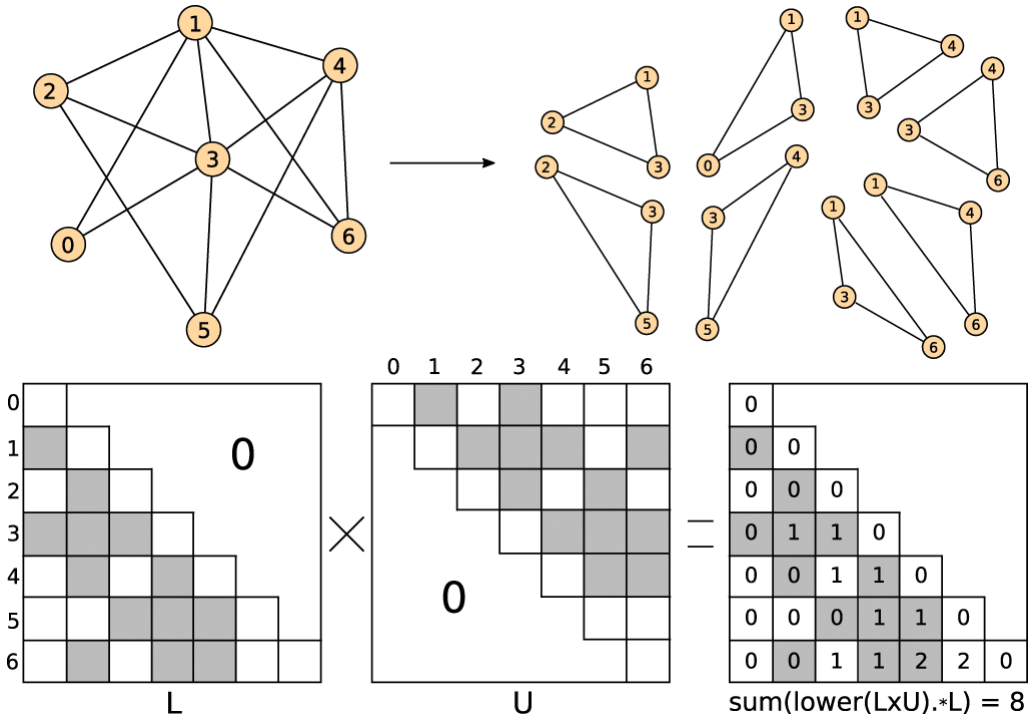
Clustering Algorithms

- Agglomerative Methods
  - Start with an empty graph that consists of nodes of the original graph but no edges.
  - Add edges are one-by-one, starting from “stronger” to “weaker” edges. This strength of the edge, or the weight of the edge, can be calculated in different ways.
- Divisive Methods
  - Start with the complete graph and take off the edges iteratively. The edge with the highest weight is removed first. E.g., Girvan-Newman Algorithm

- Algorithms:
  - Clique-percolation method
  - Markov Clustering Algorithm (MCL)
  - Fuzzy C-Means
  - Affinity Propagation
  - Chinese Whispers Clustering
  - Label Propagation Clustering
  - Newman-Girvan fast greedy algorithm

Triangle Counting

- Counts the number of triangles for each node in the graph.
- Counting the number of triangles in a graph is computationally expensive, therefore some algorithms provide approximate counts
- Applications
  - Detection of spamming activity
  - Link recommendation in Social Networks
  - Uncovering thematic structure of the web



https://www.computer.org/csdl/journal/td/2017/12/08000612/13rRUxYINeZ

Transitivity/Clustering Coefficient

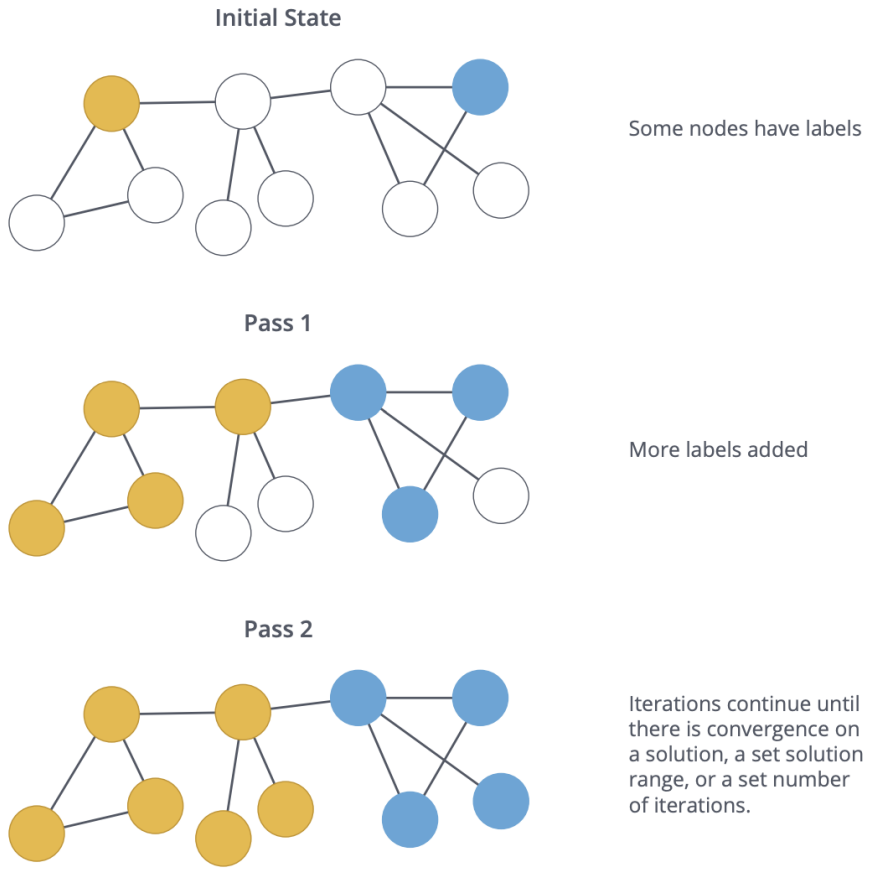
- Transitivity is closely related to the clustering coefficient of a graph, as both measure the relative frequency of triangles
- The transitivity T of a graph is based on the relative number of triangles in the graph, compared to total number of connected triples of nodes

$$T = \frac{3 \times number\ of\ triangles\ in\ the\ network}{number\ of\ connected\ triples\ of\ nodes\ in\ the\ network}$$

- The factor of three accounts for the fact that each triangle contributes to three different connected triples in the graph, one centered at each node of the triangle.
- With this definition, $0 \leq T \leq 1, and\ T = 1$ if the network contains all possible edges.

Label Propagation Algorithm (LPA)

- LPA is a fast semi-supervised algorithm for finding communities in a graph.
- Approach
  - Detects these communities using network structure alone as its guide and doesn't require a pre-defined objective function or prior information about the communities.
  - Nodes can be assigned preliminary labels to narrow down the range of solutions generated.
  - As labels propagate, densely connected groups of nodes quickly reach a consensus on a unique label.
  - LPA reaches convergence when each node has the majority label of its neighbors.



Raghavan et al in 2007, in "Near linear time algorithm to detect community structures in large-scale networks"

GNN Machine Learning