Data Science with Graphs

— Graph Structure Analysis 1 —

Matteo Lissandrini – University of Verona

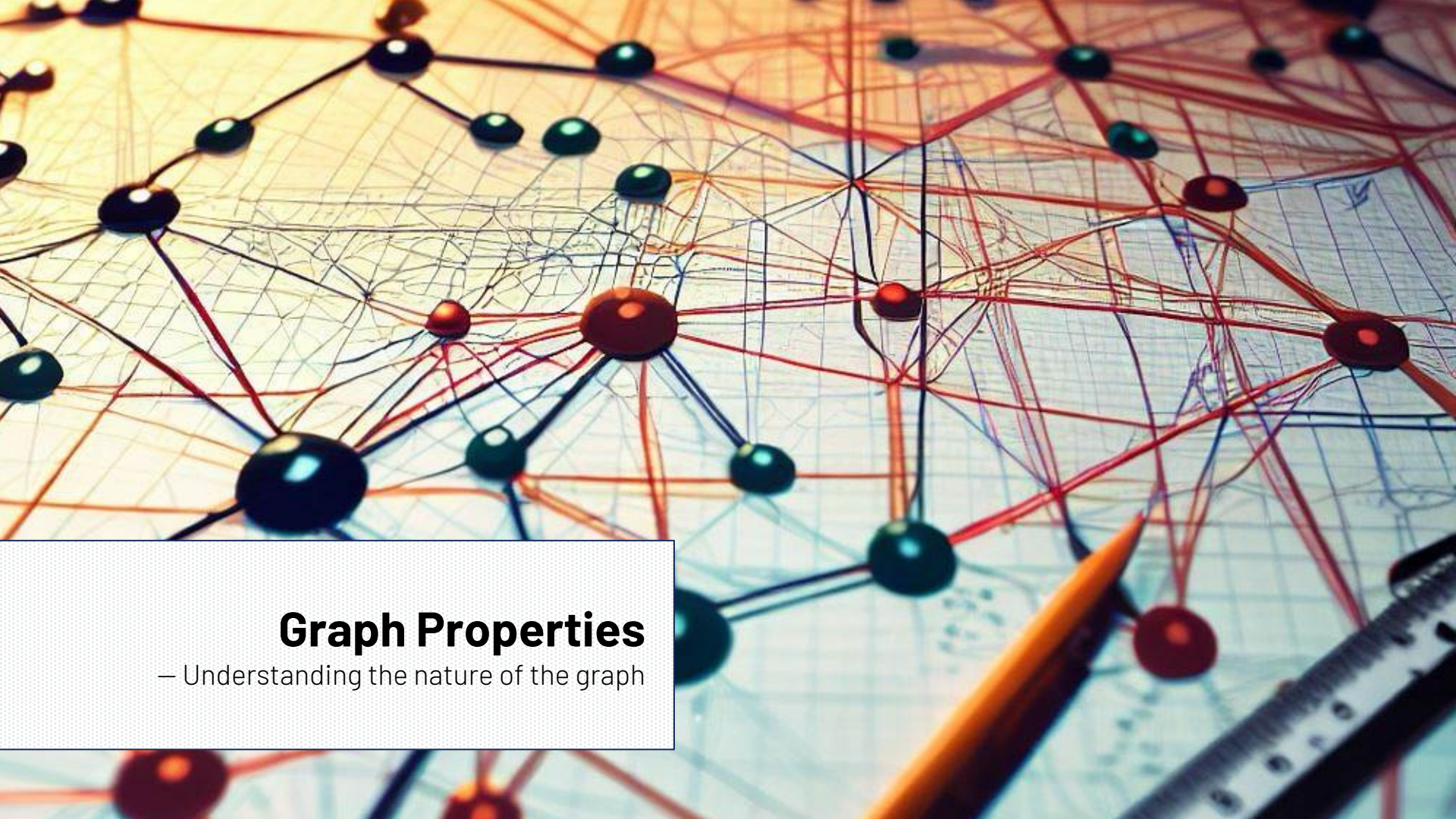
UNIVERSITÀ di VERONA

Outline

1. **Graph Properties**
   - Scale Free Networks
   - Preferential Attachment
   - Small world property
   - Erdös Number
   - Density/Diameter/Eccentricity
   - Clustering Coefficient/ Wiener Index

2. **Centrality Measures**
   - Degree/Closeness
   - Betweenness Centrality
   - Katz Centrality
   - Prestige / H-index

3. **Page Rank**
   - Random Walk & Transition Probability
   - Markov Model
   - Algebraic representation
   - Power Iteration
   - Personalized Page Rank
   - Particle Filtering
   - SimRank



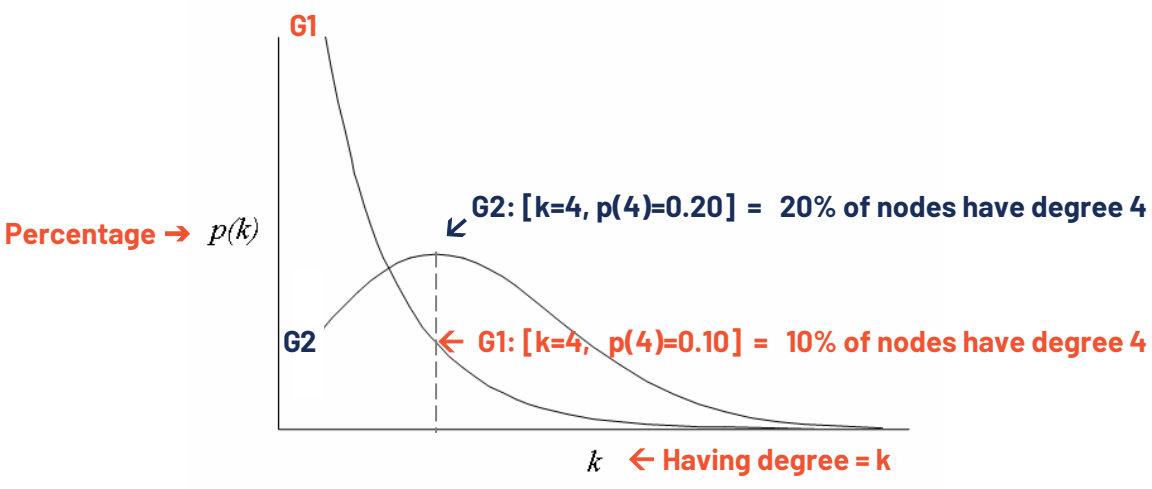
Graph Properties

— Understanding the nature of the graph

Degree Distribution

- **Node Degree:** number of nodes connected
- **What is the Degree Distribution in a Graph?**

  Plot the ratio of nodes having a specific Node-Degree



Scale Free Network/Graph

- **Node Degree:** number of nodes connected
- **What is the Degree Distribution in a Graph?**

Plot Number of nodes having a specific Node-Degree

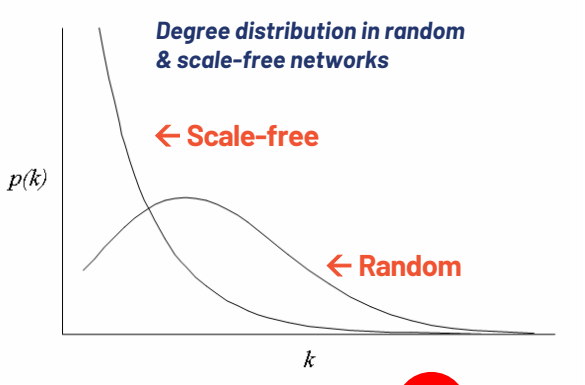
$P(k)$ *proportion of nodes with degree =* k

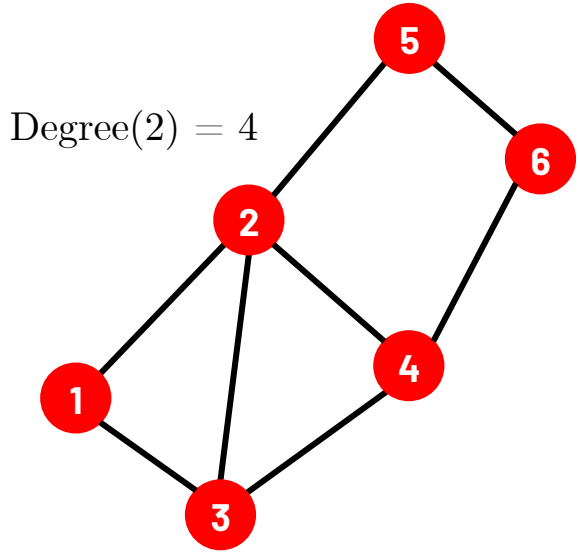
A scale-free network is a network whose degree distribution follows a power law.

The fraction P(k) *of nodes in the network having* k *connections to other nodes follows approximately*

$$P(k) \sim k^{-\gamma}$$

Typically $2 < \gamma < 3$





https://en.wikipedia.org/wiki/Scale-free_network

Cause of Scale-free: Preferential attachment

Rich gets Richer

1. New nodes are added to the network one at a time.
2. Each new node is connected to existing ***nodes with a probability that is proportional to the number of links*** that the existing nodes already have

– **Barabási–Albert model**

the probability p_i that the new node is connected to node i

$$p_i = \frac{k_i}{\sum_j k_j}$$

← Sum of all degrees = 2*|E|

where k_i is the degree of node i

Random Graphs instead follow The Erdos-Renyi model

https://en.wikipedia.org/wiki/Preferential_attachment
https://en.wikipedia.org/wiki/Barab%C3%A1si%E2%80%93Albert_model

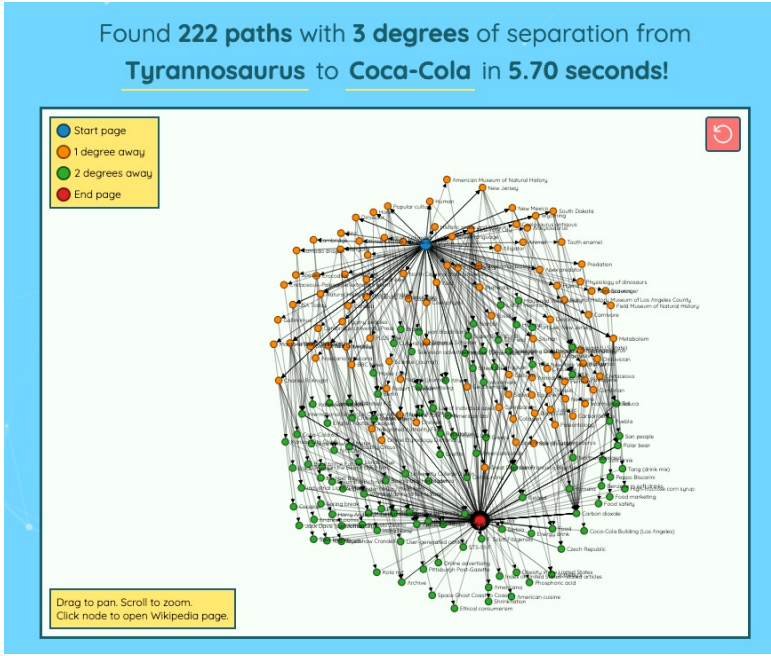
Small World Property

Shortest path: the path with the smallest number of links (edges) between 2 selected nodes.

Small world networks:

the average shortest path length between any two nodes in the network is relatively small.
Any node can be reached within a small number of edges, e.g., 4~5 hops.

7 degrees of separation: *in a social network there are at most 7 "handshakes" between you and any other person in the world*



https://www.sixdegreesofwikipedia.com

Erdős Number

the "collaborative distance" between mathematician Paul Erdős and another person

Erdős number, *the number of steps in the shortest path between a mathematician and Erdős in terms of co-authorships.*

Co-author/Collaboration Network: *An undirected graph representing authors as nodes, an edge exists between A and B if it exist a publication where A and B are co-authors*

Citation Network: a directed graph representing scientific publications as nodes, an edge goes from A to B if A has a reference to B. This is a ***directed acyclic graph*** (DAG)

Similar Concept: Bacon Number
https://en.wikipedia.org/wiki/Six_Degrees_of_Kevin_Bacon

Fun Fact: Natalie Portman has both Erdős Number and Bacon Number!

https://en.wikipedia.org/wiki/Paul_Erd%C5%91s

Paul Erdős in 1992
authored~ 1,500
mathematical papers

https://oakland.edu/enp/compute/
https://www.csauthors.net/distance/

How Compact is a Graph? (I)

- **Eccentricity of node:** the greatest distance (length of shortest path) between a node N_i and any other vertex

$$\text{Eccentricity}(1) = 3$$

- **Radius of a graph:** the minimum eccentricity of any node

$$\text{Radius} = 2$$

- **The diameter of a graph:** the maximum eccentricity of any vertex in the graph. (the maximum distance between any 2 nodes)

$$\text{Diameter} = 4$$

- **Density of a graph:** fraction between number of edges and maximal number of edges

$$\text{Density} = 2*11/56 = 0.39$$

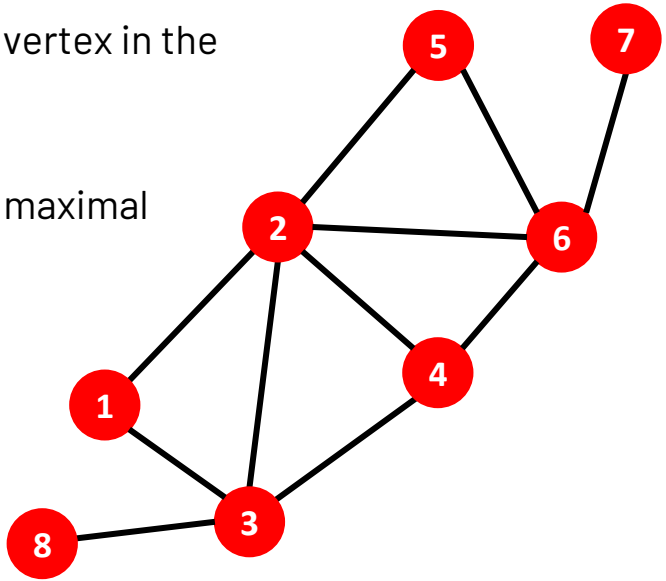
These measures ignore directions of edges (except for density)

$$D = \frac{|E|}{\binom{|V|}{2}} = \frac{2|E|}{|V|(|V|-1)}$$

Undirected

$$D = \frac{|E|}{2\binom{|V|}{2}} = \frac{|E|}{|V|(|V|-1)}$$

Directed



https://en.wikipedia.org/wiki/Combination

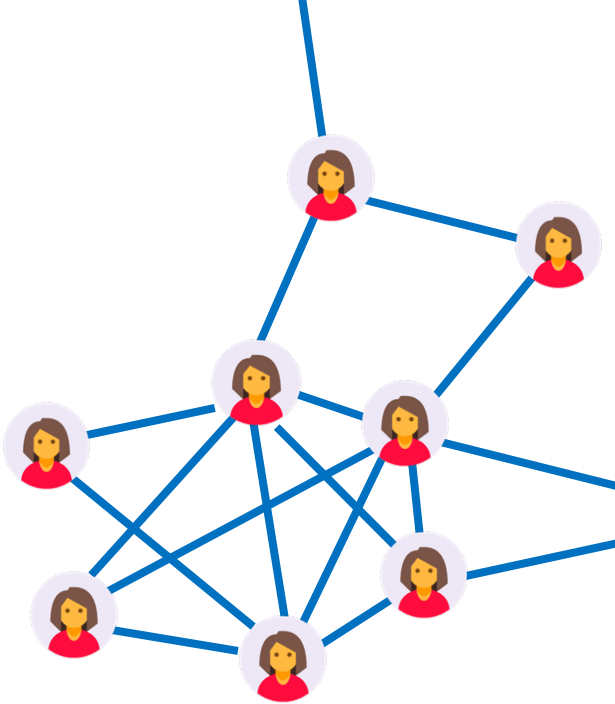
Wiener Index: Closeness of a graph

How tightly connected is a graph?

Wiener Index:
the sum of pairwise shortest-path-distances between nodes in the graph G

$$\sum_{(u,v)\in G} d(u,v)$$

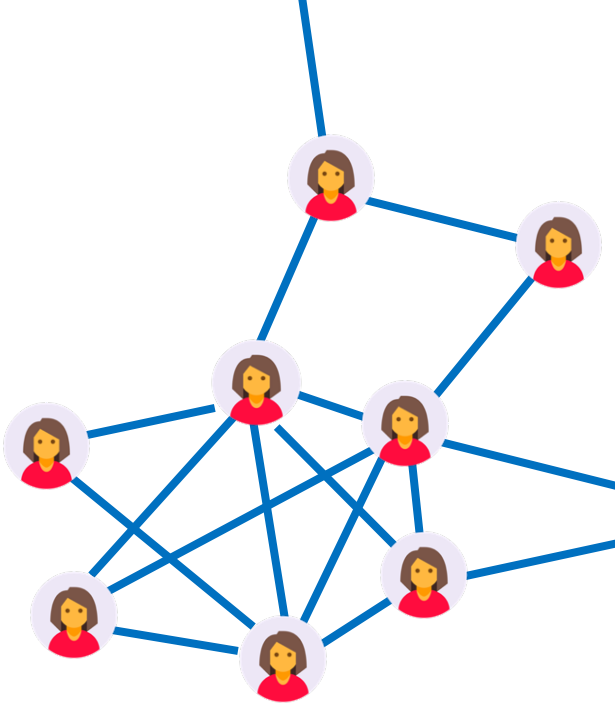
d(u, v) is the shortest-path distance



How Compact is a Graph? (II)

Characterize the structure of a graph:

1. **Average Diameter L:** *average length of the shortest paths connecting any two nodes*
2. *Effective Diameter: 90th Percentile of shortest path length*
3. **Clustering coefficient C:** *the average local density* *(see next slide)*.



Small World Graphs have
relatively small *L* & a relatively large *C*.

Clustering Coefficient: Local Density

How dense is the neighborhood of a node:
The fraction pairs of neighbors of the node that are themselves connected

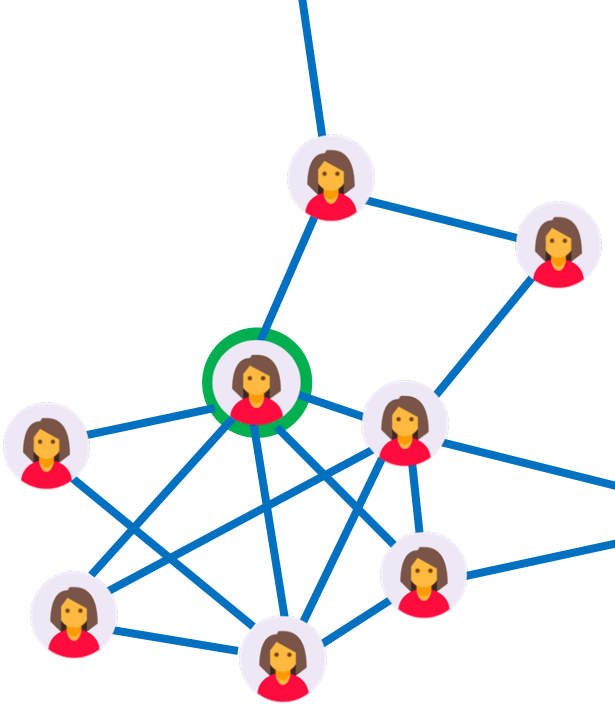
Density of a graph: fraction between number of edges and maximal number of edges

The clustering coefficient is Equivalent to the density of the subgraph when considering ONLY the neighbors of n (ignoring n)

Given node **n**

$$C_n = \frac{\text{\# edges between the neighbors of } \mathbf{n}}{\text{degree}(\mathbf{n}) * (\text{degree}(\mathbf{n}) - 1)}$$

$$C_n = \frac{2 * (\text{\# edges between the neighbors of } \mathbf{n})}{\text{degree}(\mathbf{n}) * (\text{degree}(\mathbf{n}) - 1)}$$



Directed $D = \frac{|E|}{2\binom{|V|}{2}} = \frac{|E|}{|V|(|V|-1)}$

Undirected $D = \frac{|E|}{\binom{|V|}{2}} = \frac{2|E|}{|V|(|V|-1)}$

Outline

1. **Graph Properties**
   - Scale Free Networks
   - Preferential Attachment
   - Small world property
   - Erdös Number
   - Density/Diameter/Eccentricity
   - Clustering Coefficient/ Wiener Index

2. **Centrality Measures**
   - Degree/Closeness
   - Betweenness Centrality
   - Katz Centrality
   - Prestige / H-index

3. **Page Rank**
   - Random Walk & Transition Probability
   - Markov Model
   - Algebraic representation
   - Power Iteration
   - Personalized Page Rank
   - Particle Filtering
   - SimRank

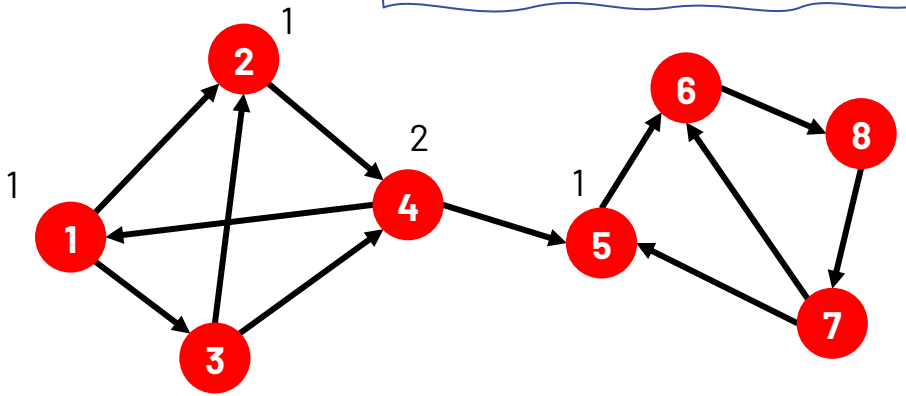
Outline

1. **Graph Properties**
   - Scale Free Networks
   - Preferential Attachment
   - Small world property
   - Erdös Number
   - Density/Diameter/Eccentricity
   - Clustering Coefficient/ Wiener Index
2. **Centrality Measures**
   - Degree/Closeness
   - Betweenness Centrality
   - Katz Centrality
   - Prestige / H-index
3. **Page Rank**
   - Random Walk & Transition Probability
   - Markov Model
   - Algebraic representation
   - Power Iteration
   - Personalized Page Rank
   - Particle Filtering
   - SimRank

The Random Walk

1. **Pick a node**
2. **Select a neighbour at random:** take a step
3. Keep making steps until we are "tired"
4. **Take note of the node where we stop and how often we visit each node**

Random Walk: *traversal of the graph by selecting neighbours at random. It is possible to visit the same edge/node multiple times. We keep note of the "frequency" with which each node is visited*



Directed Graph:
We need to follow the directions

The Random Walk Gamble

Let's play a game

1. **I pick a random node** (not telling which one)

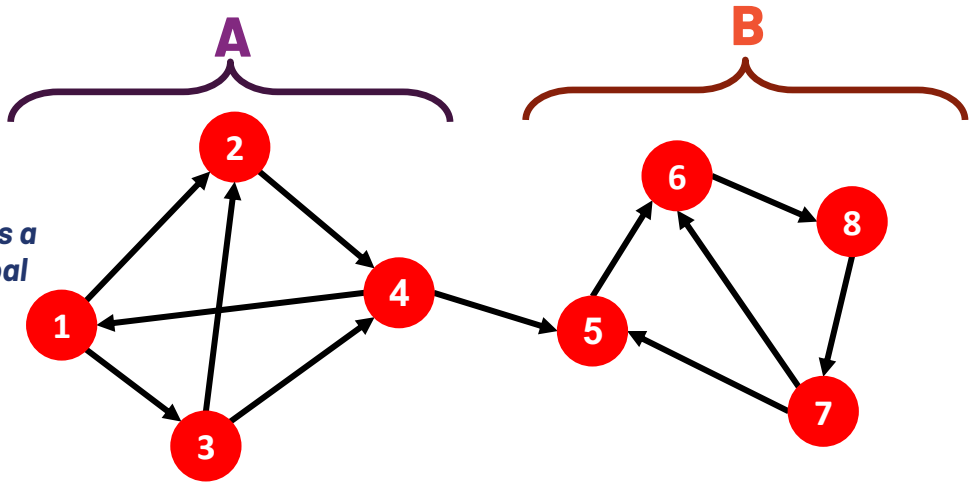
2. **I perform a random walk** (not telling how many steps, let's say >3)
3. Your guess: where am I on the graph? Group A or Group B

	Probability
6	0.233843
8	0.222960
7	0.213162
5	0.139355
4	0.068738
2	0.046465
1	0.043432
3	0.032045

~ 0.81

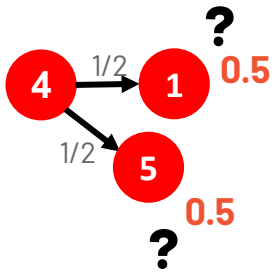
This probability is a measure of "global centrality"

~ 0.19



Transition Probability

Given a node N_i, assuming to pick a random step, what is the probability to end up in its neighbour N_j?

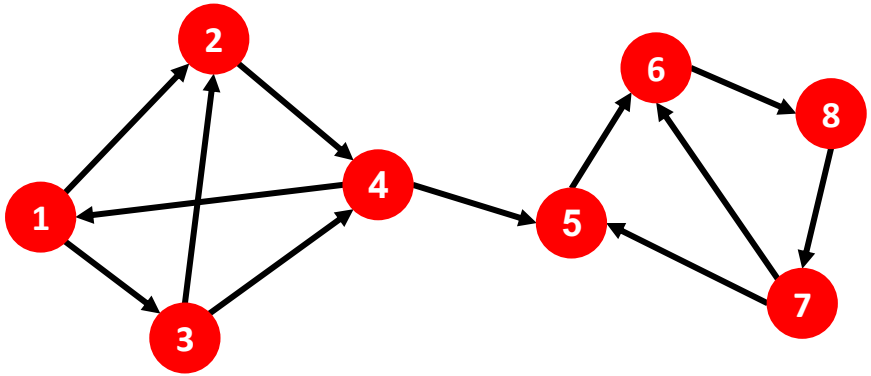


1. **If all edges are treated equally:** the probability depends on the number of outgoing edges, i.e., the degree of N_i – uniform transition probability
2. **Transition probabilities for all the edges of a single node always sum to 1**
3. **With weighted edges,** in that case we can have non-uniform probabilities

In which cases we could have weighted edges?

Transition Probability

The probability, given a node N_i, to traverse its outgoing edge $e_{i,j}$ and reach neighbour node N_j

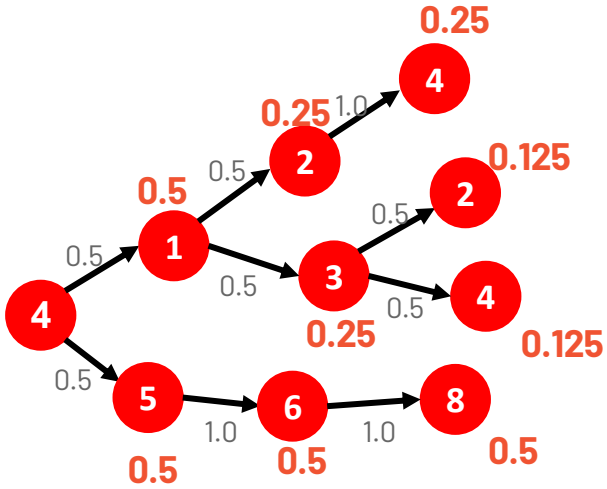


Transition Probability in a Path

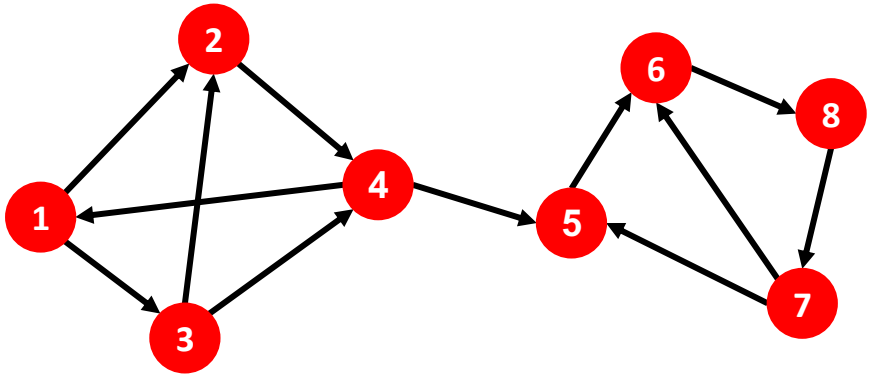
Given a node N_i, assuming to pick a series of K random steps, what is the probability to end up in a specific node N_j?

We compute the joint probability over a path

What if we get tired before taking K steps?



Given K=3



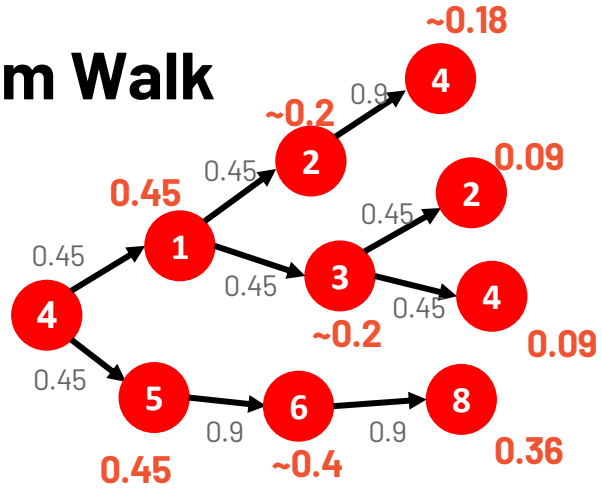
Teleport Probability in a Random Walk

What if we get tired before taking K steps?

During a random walk, we model the probability of interrupting the walk with a parameter α

Transition probability + teleport probability = 1

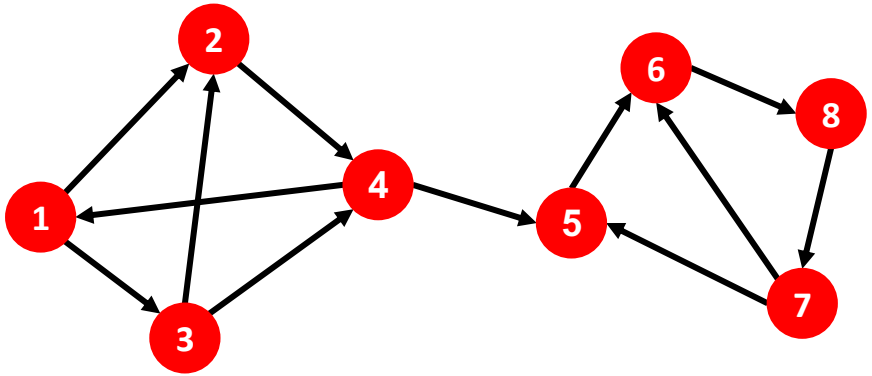
Teleport Probability α **:** *the probability of interrupting the random walk and of "jumping" to any other node at random.*



These numbers are an under-estimation! Why?

After each step the probabilities do not sum to 1

Given Teleport $\alpha = 0.1$

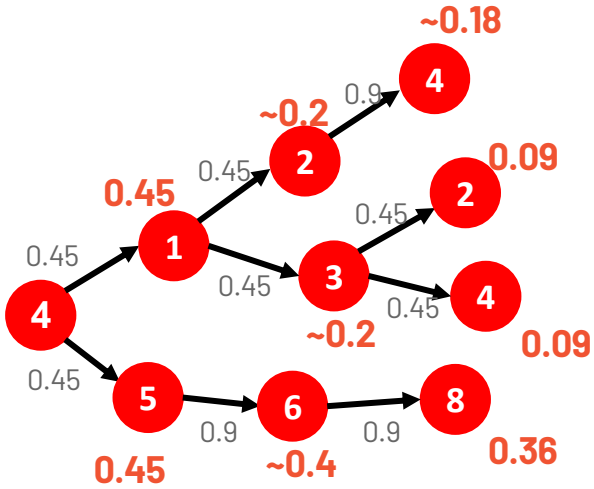


The Random Surfer

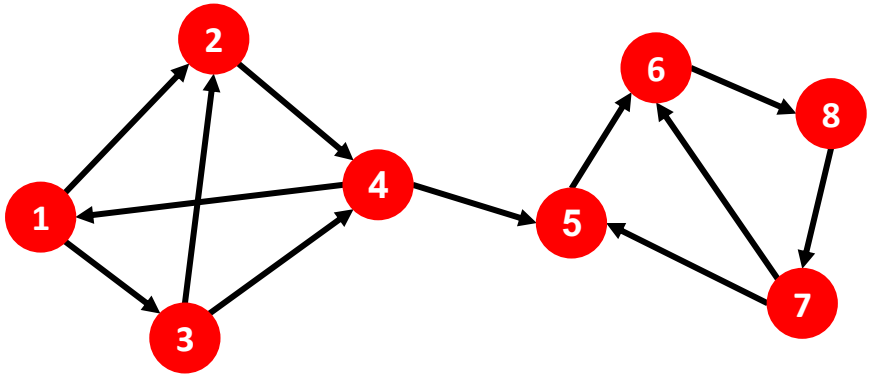
A model for a real use-case

The Random Walk model can represent the behaviour of users surfing the web

Random Surfer : *a web-user that starts from a page, follows links at random, after some clicks decides to start from a random new page*



Given Teleport $\alpha = 0.1$

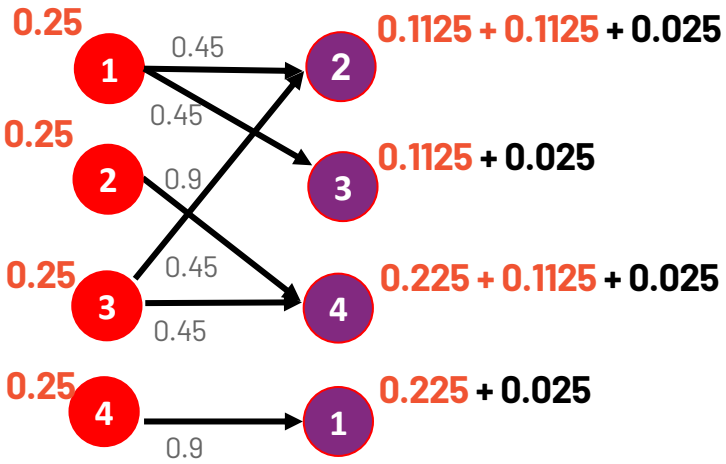


Markov Model

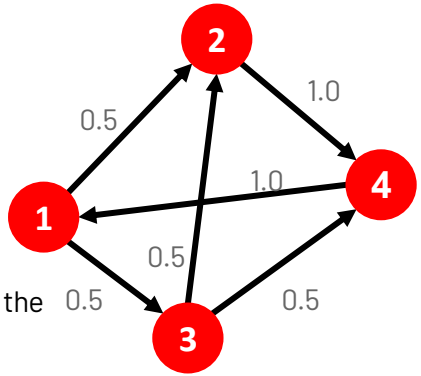
How do we model the random walk for each node

- *Each node is a starting point with equal probability*
- *Each node is connected to the neighbours*
- *We set a teleport probability to end in any other node*
- *Edges have uniform transition probabilities*
- *We sum the transition probabilities of each node and the teleport probability (uniformly distributed)*

Given Teleport $\alpha = 0.1$



From time t_0 to time t_1



Simpler Graph:
Using a simpler graph for now.
Each edge presents the transition probability without accounting for the teleport probability

In this model the teleport probability can be seen also as "special ghost link"

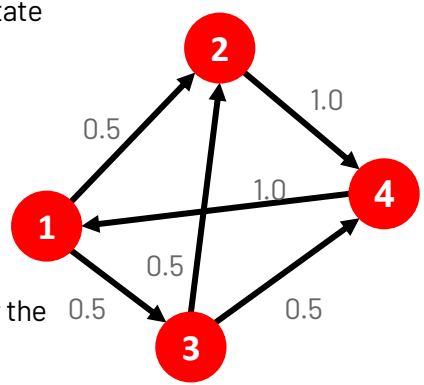
Markov Model

How do we model the random walk for each node

- *We repeat the process iteratively*

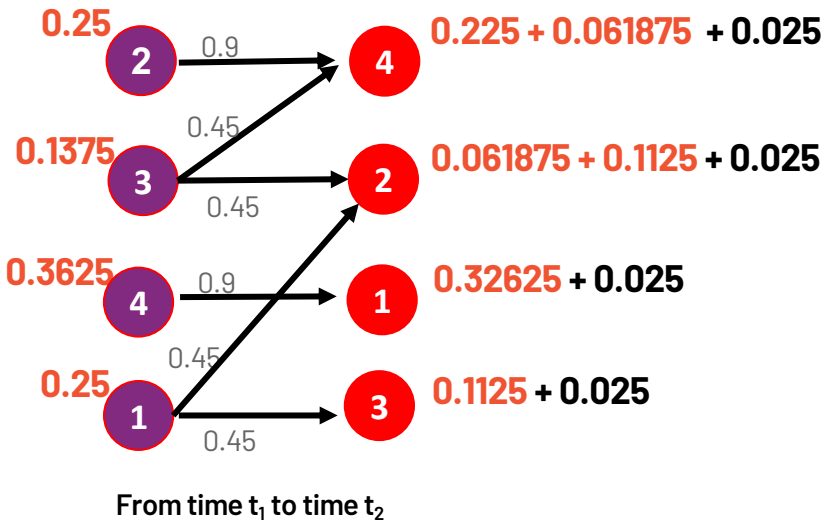
> ***The Markov Property: The future is independent of the past, given the present.***

Markov chain: a Stochastic process. In the Markov chain, each node in the graph is regarded as a state. An edge is a transition, which leads from one state to another state with a certain probability.



Simpler Graph:
Using a simpler graph for now.
Each edge presents the transition probability without accounting for the teleport probability

Given Teleport $\alpha = 0.1$



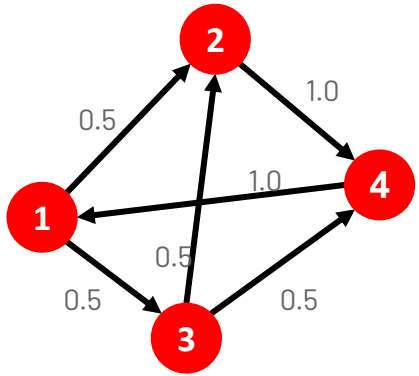
From time t_1 to time t_2

Intuition: Continue the iteration an infinite amount of time until they stop changing, what are the values? I.e., find the "stationary distribution"

Adjacency matrix / Transition Probability Matrix

Adjacency Matrix

- $A = N \times N$ matrix
- $A_{ij} = 1$ if edge between node i and j



A

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

T

	1	2	3	4
1	0	½	½	0
2	0	0	0	1
3	0	½	0	½
4	1	0	0	0

Transition Probability Matrix

- $T = N \times N$ matrix
- $T_{ij} = P_{i,j}$ transition probability from i to j
- **Rows always sum to 1**

The transition probability matrix *is derived from A by normalizing the rows (row-normalized)*

Algebraic Representation of the Markov model

Given the **Transition probability matrix** $\mathbf{T}$ of the graph, assuming no teleport probability $(\alpha = 0)$

Given the initial **vector of probabilities** v of each node

- **1 step of the process from time t_i to time t_{i+1} corresponds to the multiplication: $\mathbf{T}^T \times v_i$**

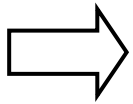
Where $\mathbf{T}^T$ is the transpose matrix and v is column-normalized (sum of column = 1)

	1	2	3	4
1	0	½	½	0
2	0	0	0	1
3	0	½	0	½
4	1	0	0	0

$\mathbf{T}$

	p
1	¼
2	¼
3	¼
4	¼

v_0



	1	2	3	4
1	0	0	0	1
2	½	0	½	0
3	½	0	0	0
4	0	1	½	0

$\mathbf{T}^T$

×

	p
1	¼
2	¼
3	¼
4	¼

v_0

=

	p
1	
2	
3	
4	

v_1

Transpose Matrix $\mathbf{T}^T$: from outgoing to incoming

(it follows that is column normalized) *it is also called the stochastic matrix but only if all columns sum to 1!*

Equivalent to one iteration in the Markov process

Algebraic Representation of the Markov model (2)

Given the **Transition probability matrix** $\mathbf{T}$ & the initial **vector of probabilities** v of each node

We can account for **the teleport probability** α so that

- **1 step of the process from time** t_i **to time** t_{i+1} **corresponds to the multiplication:** $(1\text{-}\alpha)\mathbf{T}^{\mathrm{T}} \times v_i + \alpha \times v_0$

$\alpha = 0.1$

$(1\text{-}\alpha)$

	1	2	3	4
1	0	0	0	1
2	½	0	½	0
3	½	0	0	0
4	0	1	½	0

$\mathbf{T}^{\mathrm{T}}$

$\times$

	p
1	¼
2	¼
3	¼
4	¼

v_0

$+ \quad \alpha$

	p
1	¼
2	¼
3	¼
4	¼

v_0

$=$

	p
1	¼
2	¼
3	29/80
4	11/80

v_1

During iteration the vector for the teleport stays the same we update only the vector multiplying the matrix

Algebraic Representation of the Markov model (3)

Given the **Transition probability matrix** $\mathbf{T}$ & the initial **vector of probabilities** v of each node

We can account for **the teleport probability** α so that

- 1 **step of the process from time** t_i **to time** t_{i+1} **corresponds to the multiplication:**

$$(1-\alpha)\cdot \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix}^T \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}_{t_i} + \alpha\cdot \begin{bmatrix} \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}_{t_{i+1}}$$

We refer to this process as Power Iteration:

When we compute the vector at time t_{i+1} using the multiplication with the vector at time t_i

Page Rank

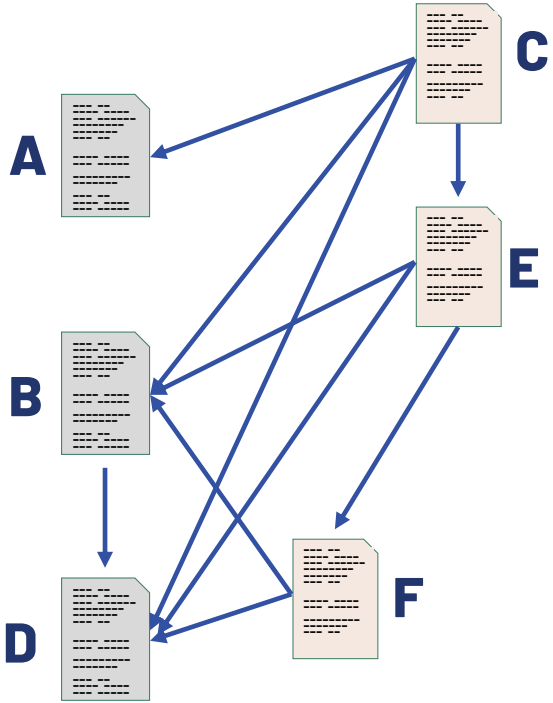
Intuition:

- links from page ***i*** to page ***j*** **is a vote** from ***i*** to ***j***
- Votes from pages with **a lot of votes count more**

In other words

1. Each link's vote is proportional to the **importance** of its source page
2. If page ***j*** with importance r_j has ***n*** out-links, each link gets r_j / n votes
3. Page ***j*** own importance is the sum of the votes on its in-links

$$Rank(B) = Rank(C)/4 + Rank(E)/3 + Rank(F)/2$$



Page Rank: Appropriateness of the Markov Model

Existence and Uniqueness of Stationary Distribution in Markov Processes
The following conditions should hold:

1. *The transition matrix is a stochastic matrix: all columns sum to 1*
2. *The matrix is irreducibile → the graph is strongly connected*
3. *The markov chain is aperiodic → cycles have different length with no greatest common divisor larger than 1*

Change the Transition Matrix to satisfy the Markov Model requirement: Add a link from each page to every page and give each link a small transition probability controlled by a parameter *d*.

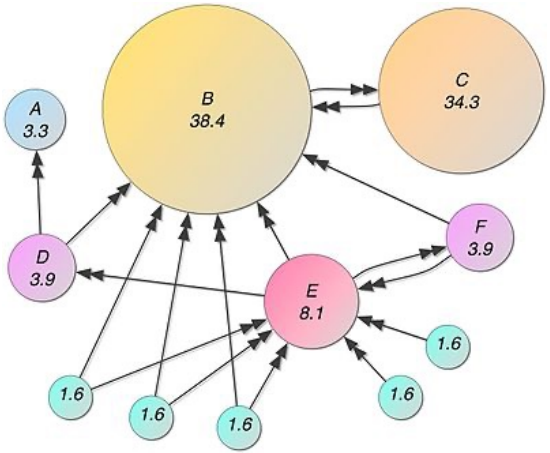
← *The teleport!*

Personalized Page Rank: Topic-Specific PageRank

- **Page Rank measures a "generic" popularity of a page, is no specific for a search query or a topic**
- **Instead of generic popularity, can we measure popularity within a topic?**
- **Goal:** Evaluate Web pages not just according to their popularity, but by how close they are to a particular topic, e.g., "sports" or "history", defined as a specific **subset of pages**
- **Allows search queries to be answered based on interests of the user**
  - **Example:** A programmer looking for "*library for graph traversal*" wants different pages depending on the programming language they use the most

Assume there is a special subset of pages S that we care about

Personalized Page Rank: Topic-Specific PageRank



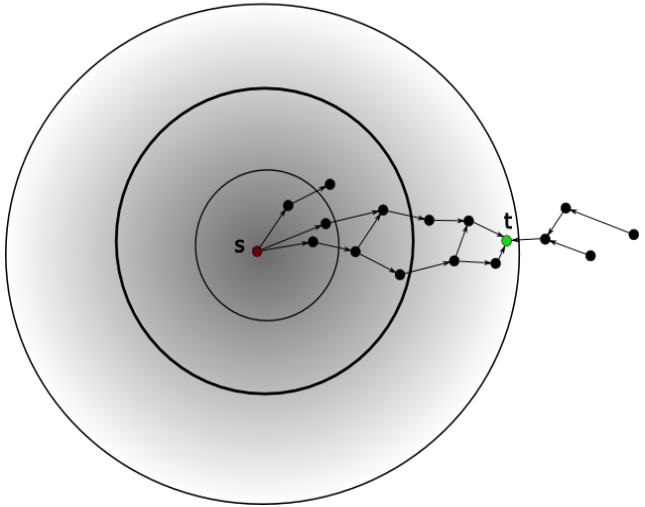
Global Page Rank

Starting from a random node, traversing randomly, **random restart point** anywhere in the graph

The role of the teleport: To avoid dead-end and spider-trap problems

Standard PageRank: Any page with equal probability

Topic Specific PageRank: A topic-specific set of "relevant" pages (teleport set)



Personalized Page Rank

Starting from a **limited set of nodes**, traversing randomly, restart point is one in **the initial set**. Bound not to travel too far

Personalized Page Rank: Topic-Specific PageRank

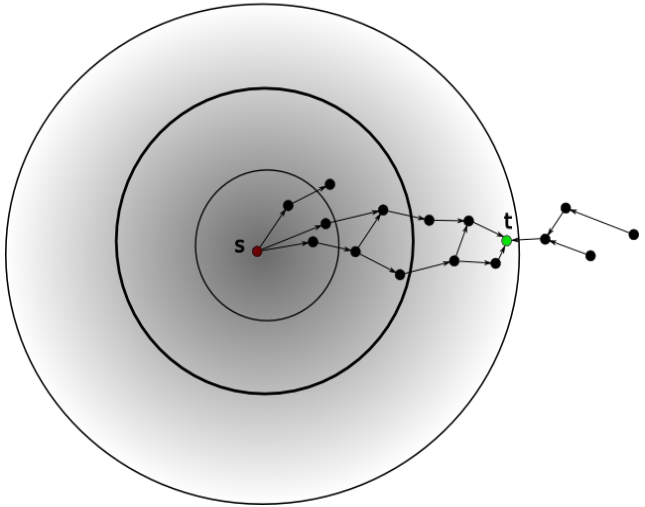
Idea: Bias the random walk

1. When walker teleports, she pick a page from a set ***S***
2. The set ***S*** contains only pages that are relevant to the topic
   E.g., pages with documentation of python libraries
3. For each teleport set ***S***, we get a different vector r_S

$$(1-\alpha)\cdot \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix}^T \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}_{t_i} + \alpha \cdot \begin{bmatrix} 0 \\ \vdots \\ \frac{1}{|S|} \\ 0 \\ \vdots \\ \frac{1}{|S|} \\ \vdots \\ 0 \end{bmatrix}$$

← We change *the teleport* ***Vector!***

When we teleport back to a single node is called: Random Walk with Restart



Personalized Page Rank

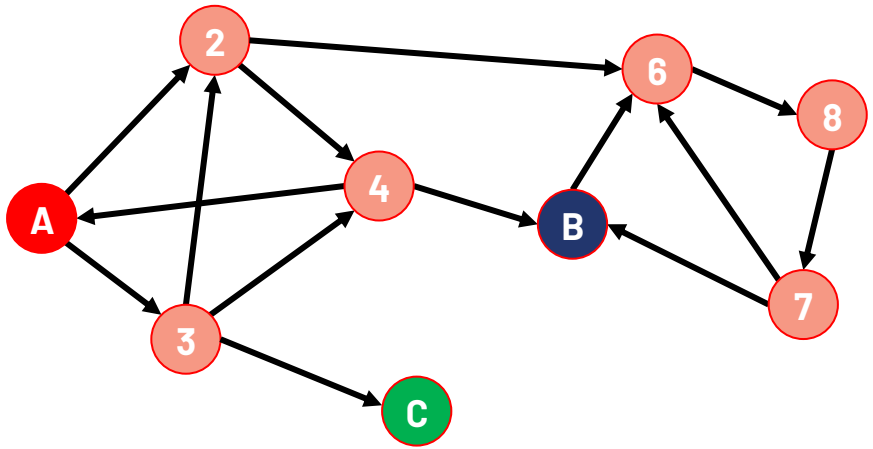
Starting from a **limited set of nodes**, traversing randomly, restart point is one in **the initial set**.
Bound not to travel too far

Personalized Page Rank as a Proximity Measure

What is the probability to reach node B *given that we start from node A? Compared to C?*

What are the most "relevant" nodes for A ranked by "closeness"

a.k.a.: Relevance, 'Relatedness'...



- **Multiple connections**
- **Quality of connection**
  - **Direct & Indirect connections**
  - **Length & "quantity"**

Another Measure: **hitting time**

$\mathbf{h(A \rightarrow B)}$ is the average number of steps to walk from node $\mathbf{A}$ to node $\mathbf{B}$.

Hitting time is asymmetric $\mathbf{h(A \rightarrow B)}$ is not always the same as $\mathbf{h(A \rightarrow B)}$

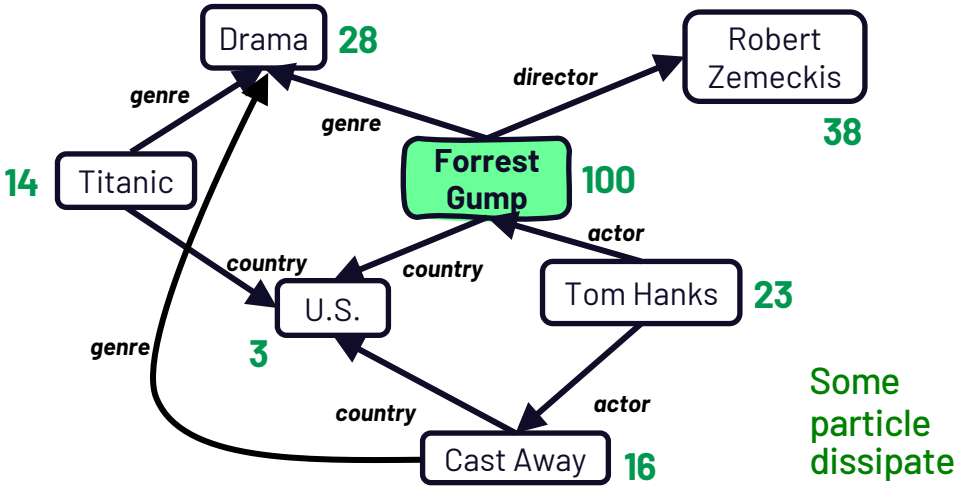
http://www.cs.cornell.edu/courses/cs4850/2009sp/Scribe%20Notes/Lecture%2024%20Friday%20March%2013.pdf

Particle Filtering Approach

Speed up PPR computation

Simulate a set of particles navigating the graphs

Particle spread not-uniformly following edge importance



Edge Importance: ***Director > Genre > Actor > Country***

Edge weights outgoing each node should sum to 1 !

Particles start from the query nodes

Edges are traversed based **on priority**

Particles are **split non-uniformly + dissipation**

Require: Graph G; Query nodes $\mathbf{Q}$
Require: Restart probability $c \in [0,1]$; Threshold $\tau \in [0,1]$
Require: Query value k
Ensure: Ranked Top-K nodes

$\mathbf{p} \leftarrow \{\}$
for each $q_i \in \mathbf{Q}$ **do**
  $\mathbf{p}[q_i] \leftarrow 1/\tau$ ▷ Initialize Particles
while $\exists\ n_i \in \mathbf{p} \mid \mathbf{p}[n_i] \neq 0$ **do**
  $\mathbf{temp} \leftarrow \{\}$
  for each $n_i \in \mathbf{p} \mid \mathbf{p}[n_i] \neq 0$ **do**
    $particles \leftarrow \mathbf{p}[n_i] \times (1-c)$
    for each $e : (n_i \rightarrow n_j) \in G$ **do** ▷ Sorted by Weight
      if $particles \leq \tau$ **then**
        break
      $passing \leftarrow \text{MAX}(particles \times e.\text{weight}(), \tau)$
      $\mathbf{temp}[n_j] \leftarrow \mathbf{temp}[n_j] + passing$
      $particles \leftarrow particles - passing$
  $\mathbf{p} \leftarrow \mathbf{temp}$
  for each $n_i \in \mathbf{p}$ **do**
    $\mathbf{v}[n_i] \leftarrow \mathbf{v}[n_i] + \mathbf{p}[n_i] \times c$ ▷ Update score
return top-k($\mathbf{v}$)

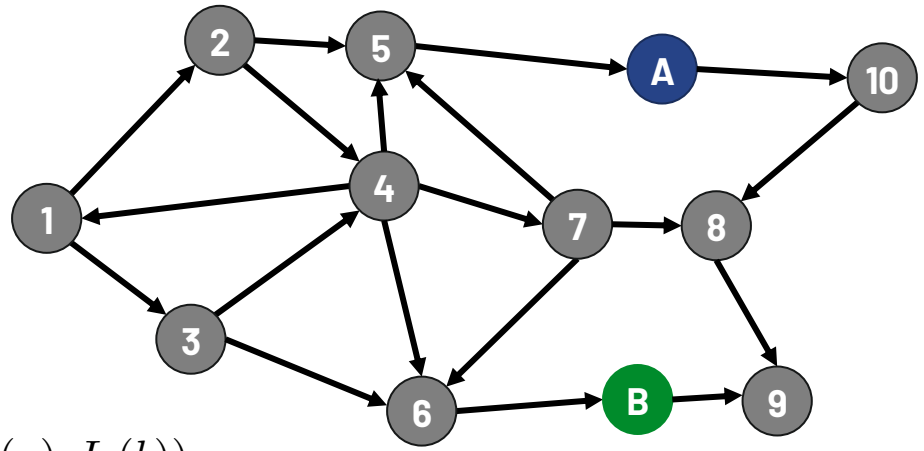
Personalized page rank on knowledge graphs: Particle Filtering is all you need!
Gallo, D., Lissandrini, M., Velegrakis, Y. (*EDBT 2020*)

SimRank: A recursive definition of similarity

Measure the similarity of two objects:

Intuition: *Two objects are similar if they are related to similar objects*

A recursive definition of similarity based on graph structure:



$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s\left(I_i(a), I_j(b)\right)$$

$I(a)$ ← *Incoming nodes to a* $I_i(a)$ ← the i-th *incoming node of a*

Where C is a constant between 0 and 1 When $I(a) = \emptyset$ or $I(b) = \emptyset$ then $s(a,b) = 0$

How similar is the "role" of A and B in this graph?

Outline

1. **Graph Properties**
   - Scale Free Networks
   - Preferential Attachment
   - Small world property
   - Erdös Number
   - Density/Diameter/Eccentricity
   - Clustering Coefficient/ Wiener Index
2. **Centrality Measures**
   - Degree/Closeness
   - Betweenness Centrality
   - Katz Centrality
   - Prestige / H-index
3. **Page Rank**
   - Random Walk & Transition Probability
   - Markov Model
   - Algebraic representation
   - Power Iteration
   - Personalized Page Rank
   - Particle Filtering
   - SimRank

Further References

Davis Shurbert, An Introduction to Graph Homomorphisms
http://buzzard.ups.edu/courses/2013spring/projects/davis-homomorphism-ups-434-2013.pdf

Andreas Schmidt, Iztok Savnik, Conference on Advances in Databases, Knowledge, and Data Applications, Overview of Regular Path Queries in Graphs
https://www.iaria.org/conferences2015/filesDBKDA15/graphsm_overview_of_regular_path_queries_in_graphs.pdf

Jure Leskovec, CS224W: Machine Learning with Graphs | 2019 |
Lecture 3-Motifs and Structural roles in networks
http://snap.stanford.edu/class/cs224w-2019/slides/03-motifs.pdf

Davide Mottin and Emmanuel Müller, Graph Exploration:
Let me Show what is relevant in your graph, KDD tutorial
https://mott.in/slides/KDD2018-Tutorial-Compressed.pdf

Kolaczyk, E.D., Statistical Analysis of Network Data, Chapter 5: Sampling and Estimation in Network Graphs.
https://link.springer.com/chapter/10.1007/978-0-387-88146-1_5

Jure Leskovec, Christos Faloutsos, Sampling from Large Graphs
https://dl.acm.org/doi/pdf/10.1145/1150402.1150479