Graph Learning
2. PageRank

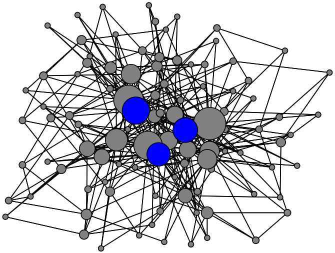
Thomas Bonald

2024 – 2025

INSTITUT
POLYTECHNIQUE
DE PARIS

Motivation

How to identify the most "important" nodes in a graph?



We focus on **PageRank**[1], based on a **random walk** in the graph

Other methods exist (e.g., betweenness centrality)

Betweenness centrality is defined as a measure of how often a node lies on the shortest path between all pairs of nodes in a network. It requires global knowledge of the entire network and assigns centrality values to nodes based on their position in these

[1] Brin & Page (1998)
The anatomy of a large-scale hypertextual Web search engine

Outline

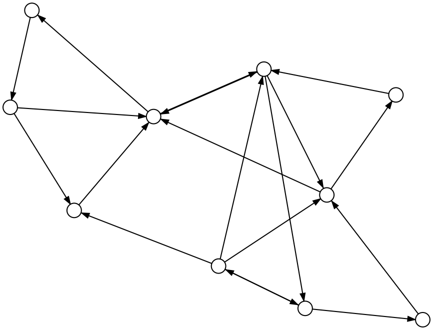
1. Random walk
2. PageRank
3. Personalized PageRank
4. Case of bipartite graphs
5. Applications

Setting

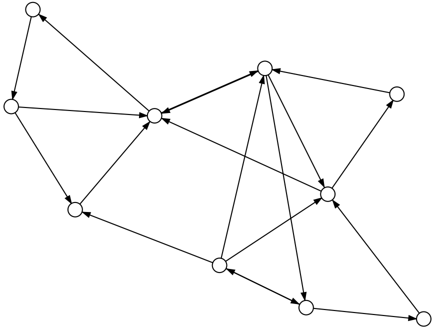
Consider a **directed** graph $G = (V, E)$

Let A be the **adjacency matrix**

Let $\underset{\text{sum of rows}}{d^+} = A1$ and $\underset{\text{sum of columns}}{d^-} = A^T1$ be the **out-degrees** and **in-degrees**



Random walk



Random walk

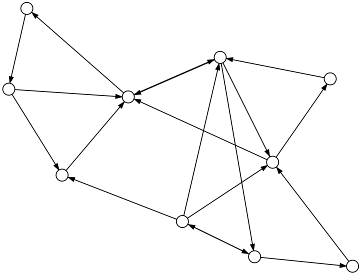
Assume $d^+ > 0$ (no sink) all nodes have outgoing edge

Definition

A **random walk** in the graph is a Markov chain $X_0, X_1, \ldots$ with transition matrix $P = D^{-1}A$ where $D = \mathrm{diag}(d^+)$ start can be random or deterministic

normalize with D^-1

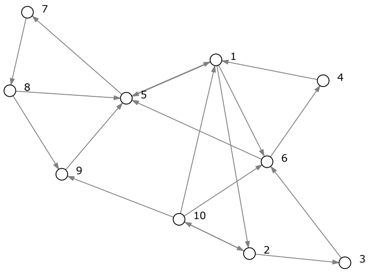
$$\forall i, j \in V, \quad P(X_{t+1} = j | X_t = i) = P_{ij}$$ prob of moving from i to j



Dynamics

Let $\pi(t)$ be the distribution of the random walk at time t:

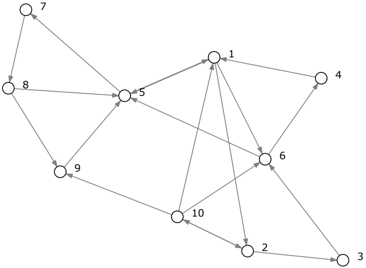
$$\pi(t) = (P(X_t = 1), \ldots, P(X_t = n))$$



Dynamics

Let $\pi(t)$ be the distribution of the random walk at time t:

$$\pi(t) = (P(X_t = 1), \ldots, P(X_t = n))$$



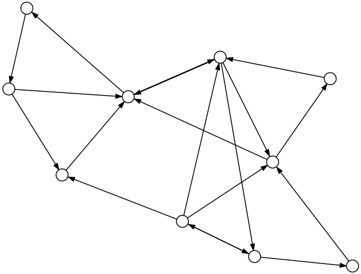
Proposition

The evolution of the random walk is given by

$$\boxed{\pi(t+1) = \pi(t)P}$$

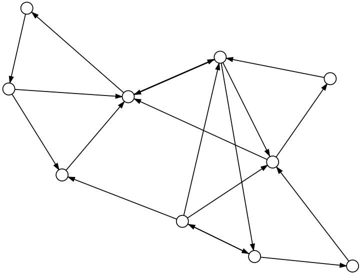
Stationary distribution

Assume that the graph is **strongly connected**



Stationary distribution

Assume that the graph is **strongly connected**



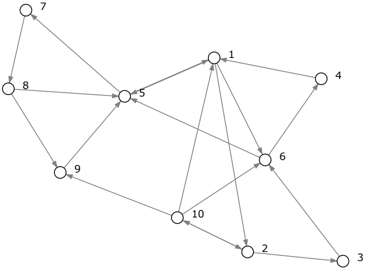
Theorem (Perron-Frobenius)

There is a unique solution π to the balance equations:

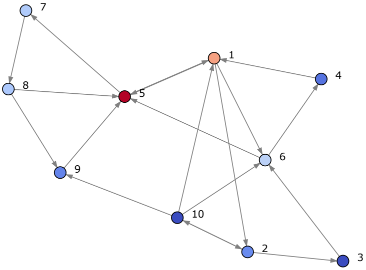
$$\boxed{\pi = \pi P}$$

This is the **frequency of visits** of each node in stationary regime

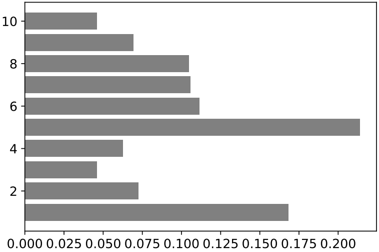
Example



Example



7
1
4
8
5
9
6
10
2
3



10
8
6
4
2
0.000 0.025 0.050 0.075 0.100 0.125 0.150 0.175 0.200

Power iteration

Stationary distribution

Input:
P, transition matrix
K, number of iterations

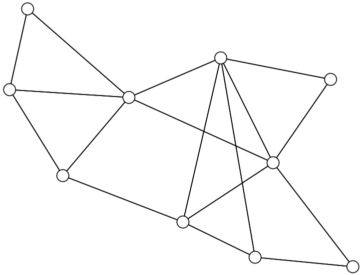
Do:
$\pi \leftarrow \frac{1}{n}(1, \ldots, 1)$
For $t = 1, \ldots, K$, $\pi \leftarrow \pi P$

Output:
π, (approximate) stationary distribution

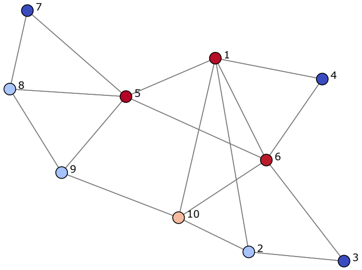
Time complexity: $O(Km)$ for m edges

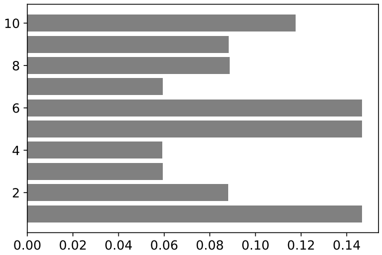
Undirected graphs

We have $d = d^+ = d^-$



Example

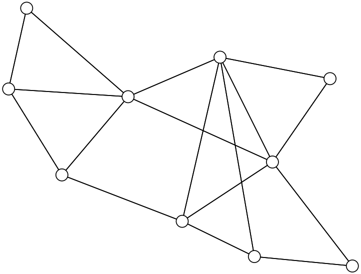




Undirected graphs

Since pi = pi P, prove d satisfies it too

dP = D^{-1} A = 1 A = d
(sum of rows is the indregree)



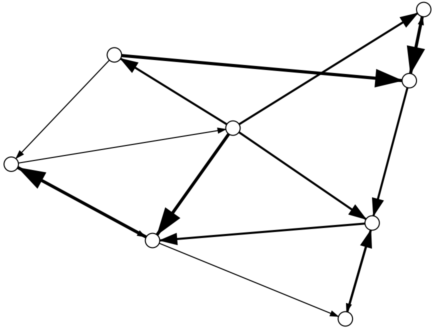
Proposition

If the graph is **undirected** and **connected**, the stationary distribution is proportional to the degrees:

$$\boxed{\pi \propto d}$$

Weighted graphs

Consider a directed, **weighted** graph $G = (V, E)$
Let A be the **weighted** adjacency matrix
Let $w^+ = A1$ and $w^- = A^T 1$ be the **out-weights** and **in-weights**



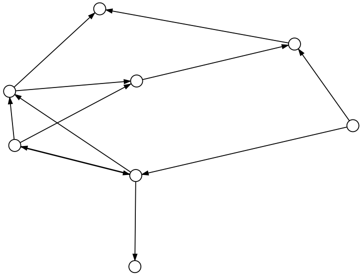
Same results with transition matrix $P = D^{-1}A$ where
$D = \mathrm{diag}(w^+)$

Outline

1. Random walk
2. **PageRank**
3. Personalized PageRank
4. Case of bipartite graphs
5. Applications

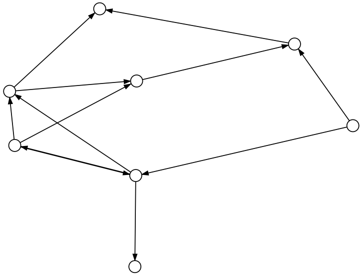
Accounting for sinks

Consider a directed graph $G = (V, E)$



Accounting for sinks

Consider a directed graph $G = (V, E)$



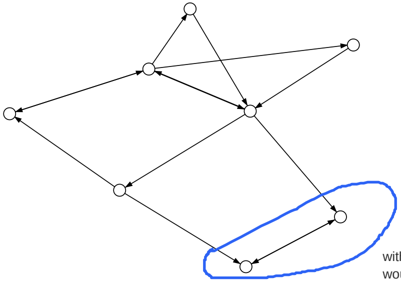
Definition

A random walk with **forced restarts** is a Markov chain with transition matrix:

$$P_{ij} = \begin{cases} \frac{A_{ij}}{d_i^+} & \text{if } d_i^+ > 0 \\ \frac{1}{n} & \text{otherwise} \end{cases}$$

Accounting for traps

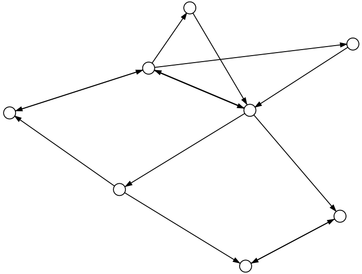
Even in the absence of sinks, the random walk can get **trapped**



with random walk, page rank would say these 2 nodes are the most important, because they trap the walker

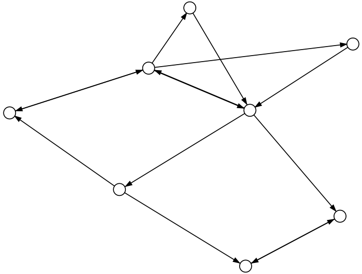
PageRank

Random restarts: walk with probability α, restart otherwise



PageRank

Random restarts: walk with probability α, restart otherwise



Definition

A random walk with **random restarts** is a Markov chain with transition matrix:

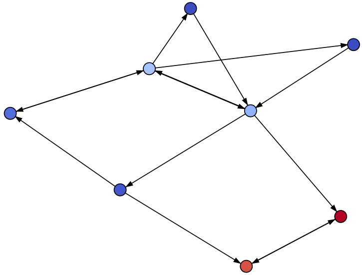
$$P^{(\alpha)} = \alpha P + (1 - \alpha)\frac{11^T}{n}$$

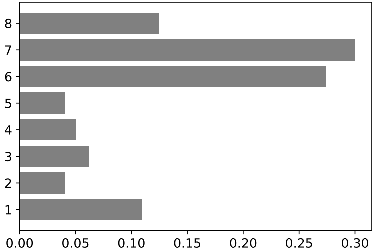
matrix of ones, dense!!!
will not be stored

The stationary distribution $\pi^{(\alpha)}$ is the **PageRank vector**

Example

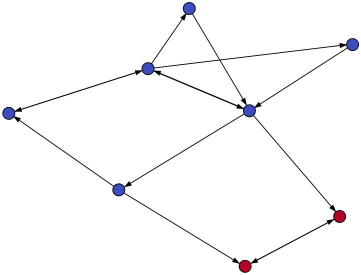
PageRank ($\alpha = 0.85$)

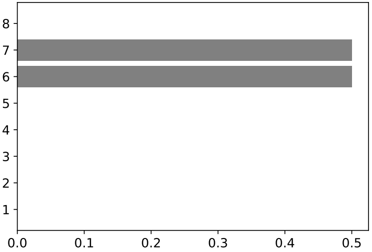




Example

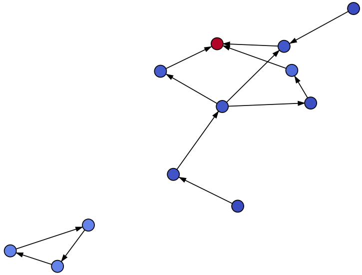
PageRank ($\alpha \to 1$)

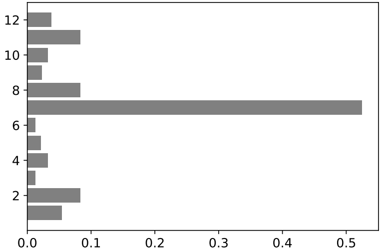




Disconnected graph

PageRank ($\alpha = 0.85$)





Power iteration

PageRank

Input:
P, transition matrix (with forced restarts)
α, damping factor
K, number of iterations

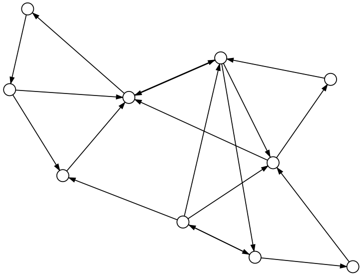
Do:
$\pi \leftarrow \frac{1}{n}(1, \ldots, 1)$
For $t = 1, \ldots, K$, $\pi \leftarrow \alpha \pi P + (1-\alpha)\frac{1}{n}(1, \ldots, 1)$

Output:
π, (approximate) PageRank vector

Time complexity: $O(Km)$ for m edges

Setting the damping factor α

Path length before restart?



Setting the damping factor α

The path length before restart (in the absence of sinks) is **geometric** with parameter $1 - \alpha$

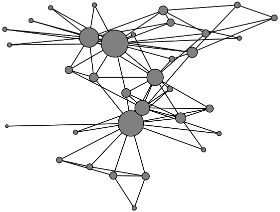
$$\text{Mean path length} \ \rightarrow \ L = \frac{\alpha}{1 - \alpha}$$

Setting the damping factor α

The path length before restart (in the absence of sinks) is **geometric** with parameter $1 - \alpha$

$$\text{Mean path length} \;\rightarrow\; L = \frac{\alpha}{1-\alpha}$$

For $\alpha = 0.85$, we get $L \approx 5.7$, a typical distance between two nodes in real graphs (cf. the **six degrees of separation**)



Expression of the PageRank vector

Proposition

$$\pi^{(\alpha)} = (1-\alpha)\sum_{t=0}^{+\infty} \alpha^t \pi(t) \quad \text{with } \pi(0) \text{ uniform}$$

Expression of the PageRank vector

Proposition

smoothing operation, starting from any node

$$\pi^{(\alpha)} = (1-\alpha)\sum_{t=0}^{+\infty} \alpha^t \pi(t) \quad \text{with } \pi(0) \text{ uniform}$$

Limiting cases

- **No restarts** $(\alpha \to 1)$

$$\pi^{(\alpha)} \to \lim_{t \to +\infty} \pi(t)$$

- **Frequent restarts** $(\alpha \to 0)$

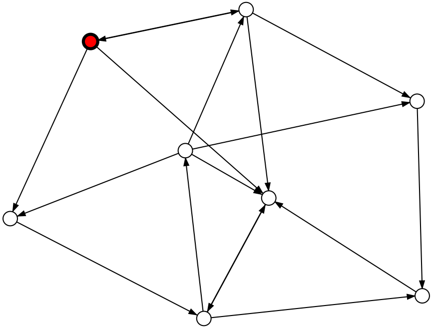
$$\pi^{(\alpha)} = (1-\alpha)\pi(0) + \alpha\pi(1) + o(\alpha)$$

Ranking equivalent to **neighbor sampling**

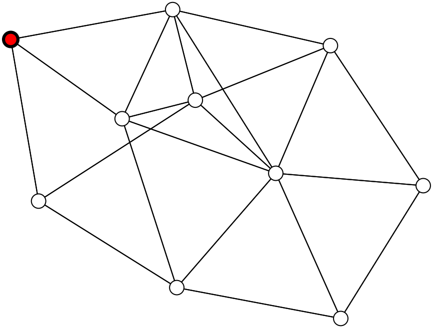
Outline

1. Random walk
2. PageRank
3. **Personalized PageRank**
4. Case of bipartite graphs
5. Applications

Personalization

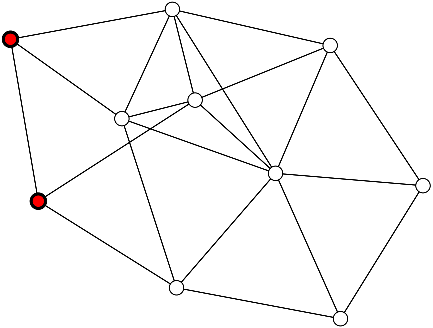


Personalization



Personalization

start form these targets and restart from them as well



Personalized PageRank

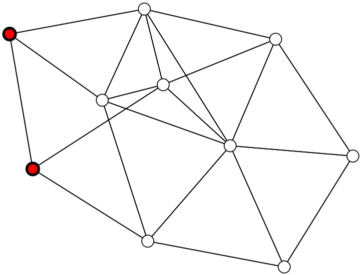
Let μ be some distribution on $S \subset V$ (e.g., uniform)

- Forced restarts:

$$P_{ij} = \begin{cases} \frac{A_{ij}}{d_i^+} & \text{if } d_i^+ > 0 \\ \mu_j & \text{otherwise} \end{cases}$$

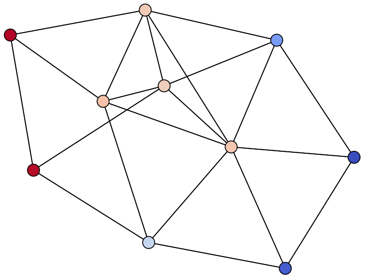
- Random restarts:

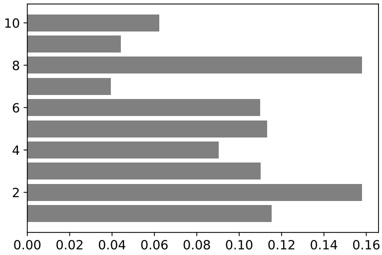
$$P^{(\alpha)} = \alpha P + (1-\alpha)1\mu$$



Example

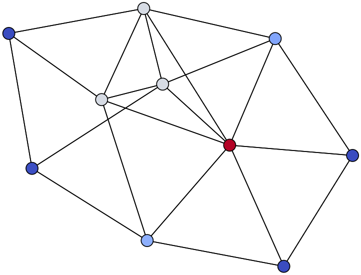
Personalized PageRank

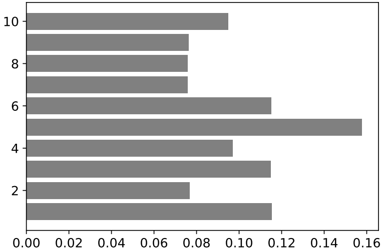




Example

PageRank (not personalized)





Power iteration

Personalized PageRank

Input:
P, transition matrix (with forced restarts)
μ, personalization row vector
α, damping factor
K, number of iterations

Do:
$\pi \leftarrow \mu$
For $t = 1, \ldots, K$, $\pi \leftarrow \alpha\pi P + (1-\alpha)\mu$

Output:
π, (approximate) PageRank vector

Expression of the Personalized PageRank vector

Proposition

$$\pi^{(\alpha)} = (1-\alpha)\sum_{t=0}^{+\infty} \alpha^t \pi(t) \quad \text{with } \pi(0) = \mu$$

Expression of the Personalized PageRank vector

Proposition

$$\pi^{(\alpha)} = (1-\alpha)\sum_{t=0}^{+\infty} \alpha^t \pi(t) \quad \text{with } \pi(0) = \mu$$

Limiting cases

- **No restarts** $(\alpha \to 1)$

$$\pi^{(\alpha)} \to \lim_{t\to+\infty} \pi(t)$$

- **Frequent restarts** $(\alpha \to 0)$

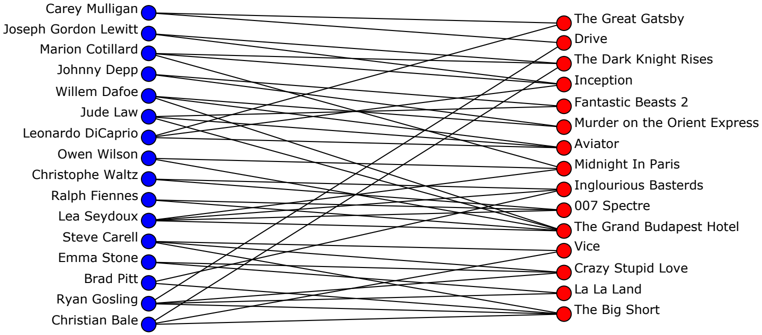
$$\pi^{(\alpha)} = (1-\alpha)\mu + \alpha\pi(1) + o(\alpha)$$

Ranking equivalent to **neighbor sampling**

Outline

Random walk in a bipartite graphs

The random walk (without restarts) has **period 2**



Actors starring in movies

Expression of the PageRank vector

Let $G = (V_1, V_2, E)$ be some connected **bipartite** graph
Assume **starts** and **restarts** in V_1
Let $\pi_1^{(\alpha)}$ and $\pi_2^{(\alpha)}$ be the PageRank vectors on V_1 and V_2

Proposition

$$\pi_1^{(\alpha)} = (1-\alpha) \sum_{t \in 2\mathbb{N}} \alpha^t \pi_1(t), \quad \pi_2^{(\alpha)} = (1-\alpha) \sum_{t \in 2\mathbb{N}+1} \alpha^t \pi_2(t)$$

Expression of the PageRank vector

Let $G = (V_1, V_2, E)$ be some connected **bipartite** graph
Assume **starts** and **restarts** in V_1
Let $\pi_1^{(\alpha)}$ and $\pi_2^{(\alpha)}$ be the PageRank vectors on V_1 and V_2

Proposition

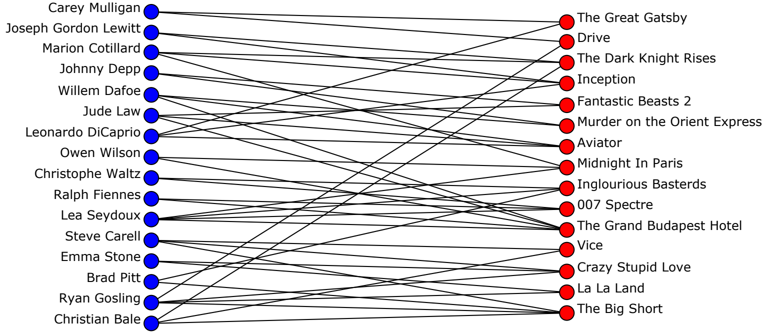
$$\pi_1^{(\alpha)} = (1-\alpha) \sum_{t \in 2\mathbb{N}} \alpha^t \pi_1(t), \quad \pi_2^{(\alpha)} = (1-\alpha) \sum_{t \in 2\mathbb{N}+1} \alpha^t \pi_2(t)$$

Note: The distribution over V_1 and V_2 is constant!

$$1^T \pi_1^{(\alpha)} = \frac{1}{\alpha+1}, \quad 1^T \pi_2^{(\alpha)} = \frac{\alpha}{\alpha+1}$$

$\rightarrow$ Ranking makes sense only **within** sets V_1 and V_2

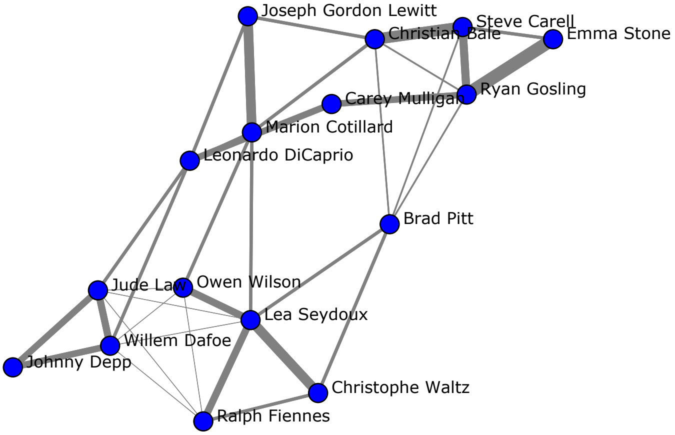
Co-neighbor graph



Actors starring in movies

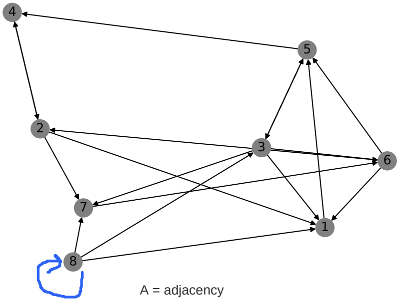
Co-neighbor graph

Graph of actors linked by movies

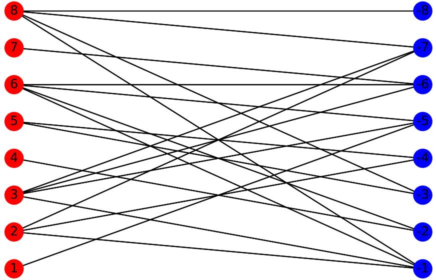


PageRank of actors in the **bipartite graph** with damping factor α
$=$ PageRank in the **co-neighbor graph** with damping factor α^2

Directed graphs as bipartite graphs



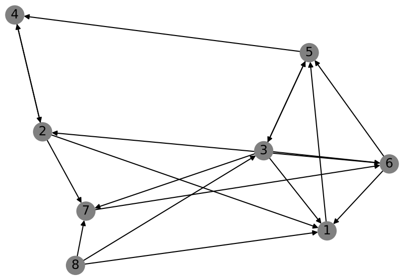
A = adjacency

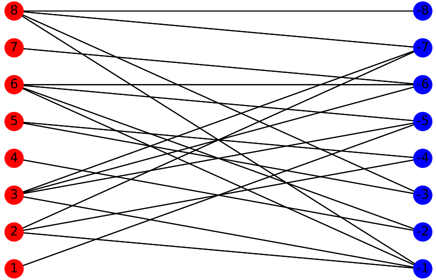


B = bipartite graph

How to build B from A, Bij is the connection between the node on the left and the node on the right

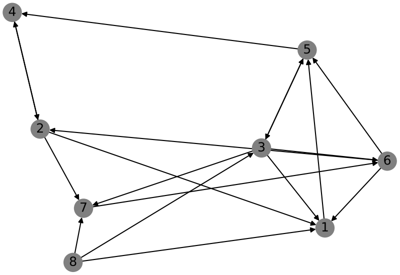
Directed graphs as bipartite graphs

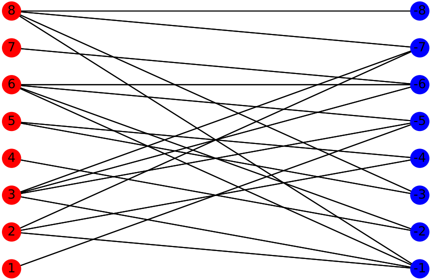




Question: How to interpret PageRank in the bipartite graph?

Directed graphs as bipartite graphs





Question: How to interpret PageRank in the bipartite graph?

Answer: PageRank associated with the **forward-backward** random walk in the directed graph!

Outline

1. Random walk
2. PageRank
3. Personalized PageRank
4. Case of bipartite graphs
5. **Applications**

Recommendation

Consider the **MovieLens** dataset
Rating of movies by users

Which movies to recommend to a target user u?

Recommendation

Consider the **MovieLens** dataset
Rating of movies by users

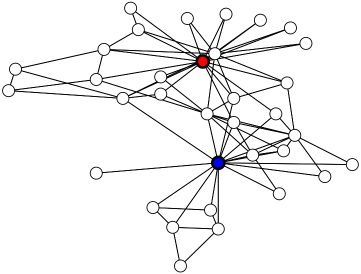
Which movies to recommend to a target user u?

Algorithm

- Construct the **bipartite graph** between users and movies with **positive** reviews (possibly weighted)
- Compute the **Personalized PageRank** vector π starting from the target user u
- **Rank** movies with respect to π

Classification

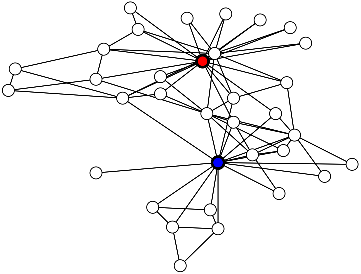
Consider the **Karate Club** graph
You know the class of 2 nodes



How to predict the label of the other nodes?

Classification

Consider the **Karate Club** graph
You know the class of 2 nodes



How to predict the label of the other nodes?

Algorithm

- **Compute** the **Personalized PageRank** vectors $\pi^{(k)}$ starting from each class $k = 1, 2$
- **Predict** $\arg\max_{k=1,2} \pi^{(k)}$ for each node

Clustering

Consider the **Wikivitals** dataset
Links between articles of Wikipedia

How to form k clusters?

Clustering

Consider the **Wikivitals** dataset
Links between articles of Wikipedia

How to form k clusters?

Algorithm

- Select k seeds (e.g., at random)
- **Label** these seeds $1, \ldots, k$
- **Classify** the other nodes by **Personalized PageRank** (one-against-all)

Summary

PageRank

A key tool for graph analysis

- Useful to quantify the **importance** of nodes, possibly relatively to other nodes $\rightarrow$ **Personalized PageRank**
- **Fast** computation through matrix-vector multiplications
- **Applications**: search, ranking, classification, clustering, ...