Graph Mining
SD212
5. PageRank

Thomas Bonald

2017 – 2018

TELECOM ParisTech

Motivation

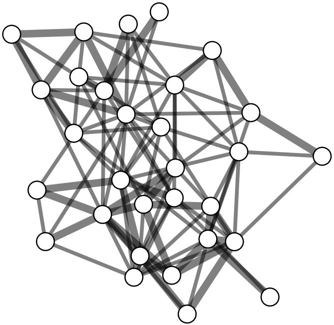
How to identify the most “important” nodes in a graph, either globally or relatively to some other nodes?

Useful for:

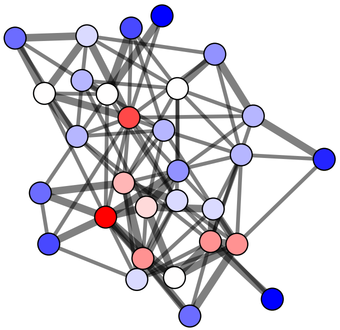
- information retrieval
- content recommendation
- local clustering

We focus on PageRank metrics, originally proposed by Google’s founders in 1999 to rank Web pages: popular pages are typically visited more frequently by a random Web surfer.

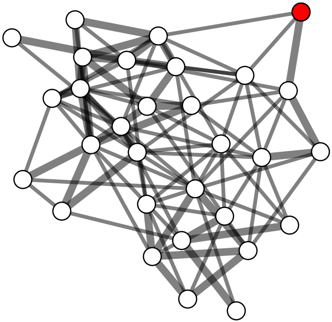
Example



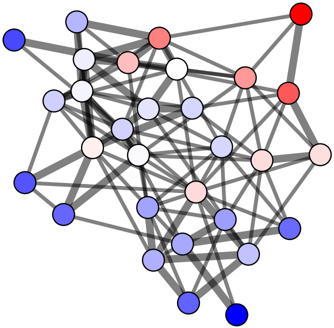
PageRank



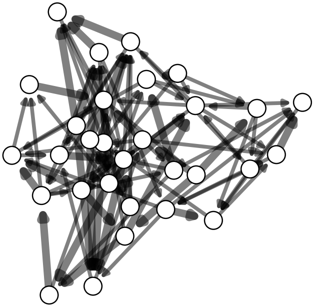
Local ranking



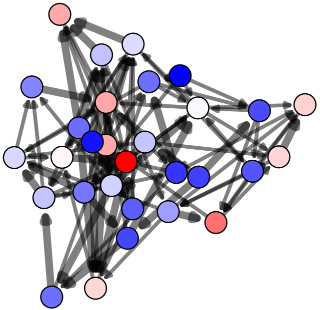
Personalized PageRank



Directed graphs



PageRank



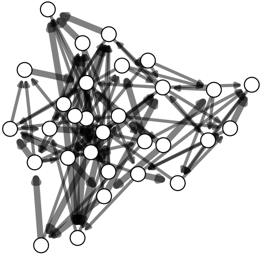
Outline

1. Random walk
2. PageRank
3. Personalized PageRank
4. Forward-Backward PageRank

Notation

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

- $V = \{1, \ldots, n\}$
- A, weighted adjacency matrix
- w^-, w^+, vectors of in, out weights



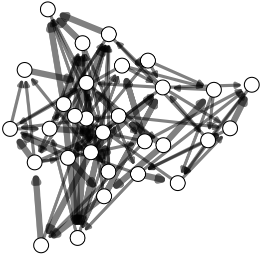
A_{i,j} = w_i,j if (i,j) \ in E
0 elsewise

w^+ = A \indicatrice

Random walk

In the **absence** of sinks:

- $P_{ij} = A_{ij}/w_i^+$, probability of moving from i to j
- A Markov chain $X_0, X_1, X_2, \ldots$ with transition matrix P
- Probability distributions $\pi_0, \pi_1, \pi_2, \ldots$



Computation

Stationary distribution

Input:
P, transition matrix
k, number of iterations

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

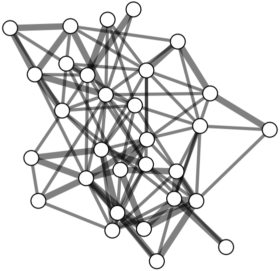
Output:
π, (approximate) stationary distribution

Complexity: $O(km)$ in time, $O(n)$ in memory

The case of undirected graphs

We have:

- $w^- = w^+ = w$
- $P_{ij} = A_{ij}/w_i$, probability of moving from i to j
- $P = D^{-1}A$ with $D = \mathrm{diag}(w)$



Accounting for sinks

Two options:

1. (recursive) pruning
2. (forced) restart, e.g.,

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

Enforcing irreducibility

Random walks with restarts:

- Fix $\alpha \in (0, 1)$
- Walk with probability α, restart (e.g., to a random node) with probability $1 - \alpha$
- An **irreducible** Markov chain with transition matrix:

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

- The stationary distribution $\pi^{(\alpha)}$ satisfies:

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

This is the **PageRank** vector!

Computation

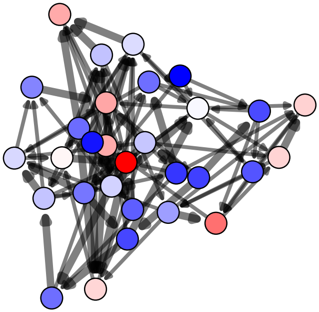
PageRank

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

Example ($\alpha = 0.85$)



Setting the damping factor

- The path length before restart (in the absence of sinks) has a **geometric distribution** with parameter $1 - \alpha$
- Average path length:
$$\frac{\alpha}{1 - \alpha}$$
- For $\alpha = 0.85$, we get about 5.7, a typical distance between two nodes in real graphs (cf. the **small-world** property).

Expression of the PageRank vector

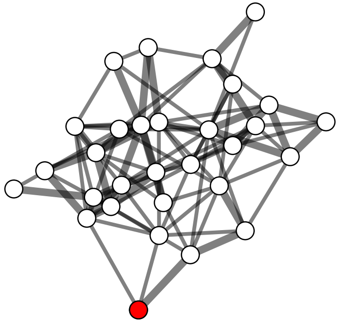
Proposition

$$\pi^{(\alpha)} = (1-\alpha)\sum_{t=0}^{+\infty} \alpha^t \pi_t$$

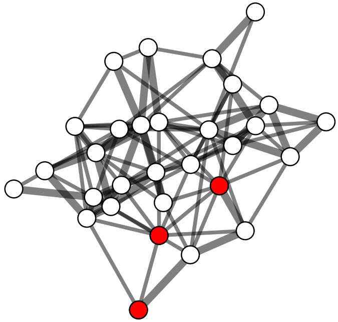
Outline

1. Random walk
2. PageRank
3. **Personalized PageRank**
4. Forward-Backward PageRank

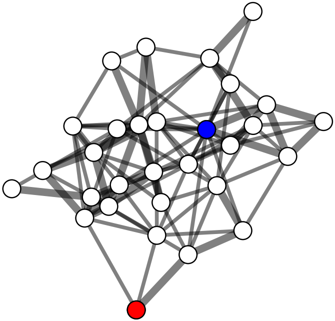
Personalization



Personalization



Local clustering



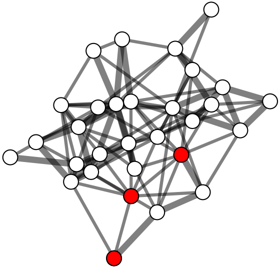
Personalized PageRank

- Restart distribution μ on $S \subset V$
- Restart from sinks:

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

- Damping:

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



Computation

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 $t = 1, \ldots, k$, $\pi \leftarrow \alpha \pi P + (1 - \alpha) 1 \mu$

Output:
π, (approximate) PageRank vector

Expression of the Personalized PageRank vector

Proposition

In the absence of sinks,

$$\pi^{(\alpha)} = \sum_{s \in S} \mu_s \pi_s^{(\alpha)}$$

where $\pi_s^{(\alpha)}$ is the Personalized PageRank vector associated with s

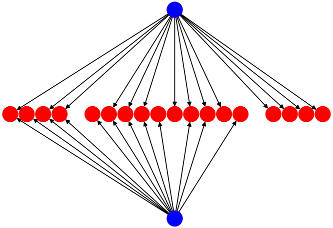
Outline

1. Random walk
2. PageRank
3. Personalized PageRank
4. **Forward-Backward PageRank**

`HITS kleinberg 1999`

Motivation

- In many practical cases, two nodes having a large number of common **successors** (or **predecessors**) are closely related
- For instance, the articles "France" and "Germany" of Wikipedia for Schools have 38 common links: United States, United Kingdom, World War II, Latin, Japan, Italy, Spain, Russia, Time zone, Currency, ...

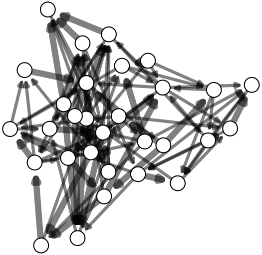


Forward-backward random walk

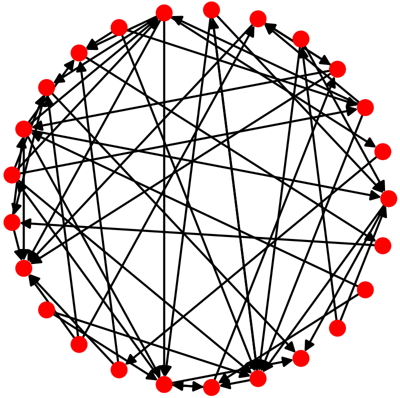
In the **absence** of sources and sinks:

- $P^+_{ik} = A_{ik}/w^+_i$, probability of moving from i to k (original graph)
- $P^-_{kj} = A_{jk}/w^-_k$, probability of moving from k to j (reverse graph)
- A Markov chain $X_0, X_1, X_2, \dots$ with transition matrix $P = P^+P^-$,

$$P_{ij} = \sum_{k \in V:(i,k)\in E} \frac{A_{ik}}{w^+_i}\frac{A_{jk}}{w^-_k}$$



Example



Co-citation graph

- **Weighted, undirected** graph G^{co} associated with G
- Weighted adjacency matrix:

$$A^{\mathrm{co}}_{ij} = \sum_{k \in V} \frac{A_{ik} A_{jk}}{w^-_k}$$

- Weight of node i:

$$w^{\mathrm{co}}_i = \sum_{j \in V} A^{\mathrm{co}}_{ij} = w^+_i$$

- Transition matrix of the random walk:

$$P^{\mathrm{co}}_{ij} = \frac{A^{\mathrm{co}}_{ij}}{w^{\mathrm{co}}_i} = P_{ij}$$

This is the **forward-backward** random walk in G!

Size of the co-citation graph

- Each node k of G forms a clique of d_k^- nodes in G^{co}
- Number of edges in G^{co} possibly as large as:

$$\sum_{k \in V} (d_k^-)^2$$

  May be **huge** for a power-law in-degree distribution!
- In comparison, the size of G is:

$$m = \sum_{k \in V} d_k^-$$

Computation

Assuming neither sinks nor sources:

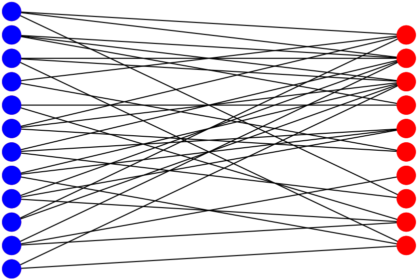
Personalized Forward-Backward PageRank

Input:
P^+ and P^-, forward and backward transition matrices
μ, personalization row vector
α, damping factor
k, number of iterations

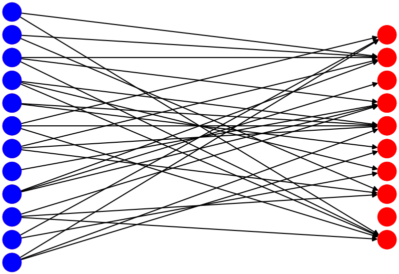
Do:
$\pi \leftarrow \mu$
For $t = 1, \ldots, k$, $\pi \leftarrow \pi P^+$, $\pi \leftarrow \alpha \pi P^- + (1 - \alpha)\mu$

Output:
π, (approximate) Forward-Backward PageRank vector

Bipartite graphs



Bipartite graphs



Summary

PageRank metrics:

- Useful to quantify the “importance” of nodes, relatively to other nodes (through **personalization**)
- **Fast** computation through matrix-vector multiplications using sparse matrix data structure (time complexity in $O(km)$)
- The edge direction is generally better captured by the (Personalized) **Forward-Backward** PageRank
- Applicable ot **bipartite graphs** for local clustering of each part

A **fundamental tool** for graph analysis!