Universitat Pompeu Fabra Barcelona

Lecture 7 · Random Networks I

Networks, Crowds and Markets

Today's Lecture

1. Wrapping-up centrality measures: PageRank and HITS.
2. Random graphs, Erdős–Rényi model.
3. Probability recap: binomial and Poisson distribution.

Recall: PageRank

Note

We define random walk on a directed graph in a natural way. The walk can only follow the direction of arrows.

Algebraically more complicated as A_G is not symmetric and the eigenvalues are complex.

- Web graph = directed network of pages and hyperlinks.
- Eigenvector centrality does not work directly in directed graphs with sinks or disconnected components.
- PageRank modifies the random walk with teleportation:

$$P_\alpha = \alpha P + (1-\alpha)\frac{1}{N}\mathbf{1}\mathbf{1}^T,$$

  where P is the transition matrix of the web, $\alpha \in (0,1)$.
- Stationary distribution of P_α = PageRank vector.

Beyond PageRank: The HITS Algorithm

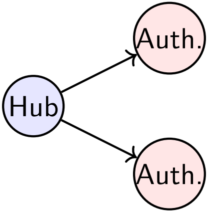
Goal: Identify both *authorities* and *hubs* in a directed network.

- A good hub points to many good authorities.
- A good authority is pointed to by many good hubs.

Beyond PageRank: The HITS Algorithm

Goal: Identify both *authorities* and *hubs* in a directed network.

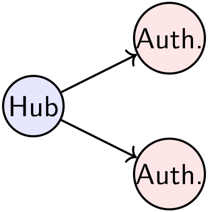
- A good hub points to many good authorities.
- A good authority is pointed to by many good hubs.



Beyond PageRank: The HITS Algorithm

Goal: Identify both *authorities* and *hubs* in a directed network.

- A good hub points to many good authorities.
- A good authority is pointed to by many good hubs.



Context:

- Introduced by Jon Kleinberg (1999).
- Used originally to rank web pages within a *topic query*.
- Query-dependent — unlike PageRank, which is global.

Mathematics of HITS

Let A be the adjacency matrix ($A_{ij} = 1$ if $i \to j$).

Each node i has: authority score a_i, hub score h_i.

Mathematics of HITS

Let A be the adjacency matrix ($A_{ij} = 1$ if $i \to j$).

Each node i has: authority score a_i, hub score h_i.

They satisfy the mutual reinforcement relations:

$$\begin{cases} h \propto Aa, & \text{(hubs get votes from authorities)} \\ a \propto A^\top h, & \text{(authorities get votes from hubs)} \end{cases}$$

Mathematics of HITS

Let A be the adjacency matrix ($A_{ij} = 1$ if $i \to j$).

Each node i has: authority score a_i, hub score h_i.

They satisfy the mutual reinforcement relations:

$$\begin{cases} h \propto Aa, & \text{(hubs get votes from authorities)} \\ a \propto A^\top h, & \text{(authorities get votes from hubs)} \end{cases}$$

Combining gives:

$$a \propto A^\top A a, \qquad h \propto A A^\top h.$$

Mathematics of HITS

Let A be the adjacency matrix ($A_{ij} = 1$ if $i \to j$).

Each node i has: authority score a_i,   hub score h_i.

They satisfy the mutual reinforcement relations:

$$\begin{cases} h \propto Aa, & \text{(hubs get votes from authorities)} \\ a \propto A^\top h, & \text{(authorities get votes from hubs)} \end{cases}$$

Combining gives:

$$a \propto A^\top A a, \qquad h \propto AA^\top h.$$

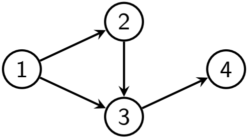
- Take a and h to be **dominant eigenvectors** of $A^\top A$ and $AA^\top$.
- In the iterative HITS algorithm, a and h are renormalized at each step, so the proportionality becomes equality after scaling.
- Equivalent viewpoint: HITS computes the first left and right singular vectors of A.

Example: Hubs and Authorities in a Small Web

Adjacency matrix:

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

Graph representation:

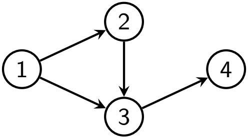


Example: Hubs and Authorities in a Small Web

Adjacency matrix:

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

Graph representation:



Iterative algorithm:

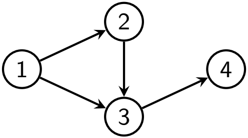
1. Initialize $a_i = h_i = 1$.
2. Repeat $a \leftarrow A^\top h$, normalize; $h \leftarrow Aa$, normalize.

Example: Hubs and Authorities in a Small Web

Adjacency matrix:

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

Graph representation:



Iterative algorithm:

1. Initialize $a_i = h_i = 1$.
2. Repeat $a \leftarrow A^\top h$, normalize; $h \leftarrow Aa$, normalize.

Python demo:

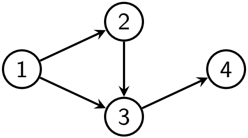
```
import networkx as nx
G = nx.DiGraph()
G.add_edges_from([(1,2),(1,3),(2,3),(3,4)])
hubs, auth = nx.hits(G)
```

Example: Hubs and Authorities in a Small Web

Adjacency matrix:

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

Graph representation:



Iterative algorithm:

1. Initialize $a_i = h_i = 1$.
2. Repeat $a \leftarrow A^\top h$, normalize; $h \leftarrow Aa$, normalize.

Python demo:

```
import networkx as nx
G = nx.DiGraph()
G.add_edges_from([(1,2),(1,3),(2,3),(3,4)])
hubs, auth = nx.hits(G)
```

Interpretation:

- Node 1 $\rightarrow$ strong hub (points to many).
- Node 4 $\rightarrow$ strong authority (pointed to by many).

Random graphs and Erdős–Rényi model

Why random graphs?

Real networks (social, economic, financial) are noisy and constantly evolving. We need a simple *baseline model* to compare against.

Definition (Erdős–Rényi (ER) model)

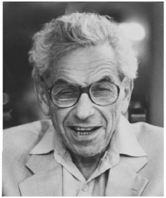
$G(N, p)$: a random graph on N nodes where each of the $\binom{N}{2}$ possible edges appears independently with probability p.

Why random graphs?

Real networks (social, economic, financial) are noisy and constantly evolving. We need a simple *baseline model* to compare against.

Definition (Erdős–Rényi (ER) model)

$G(N, p)$: a random graph on N nodes where each of the $\binom{N}{2}$ possible edges appears independently with probability p.

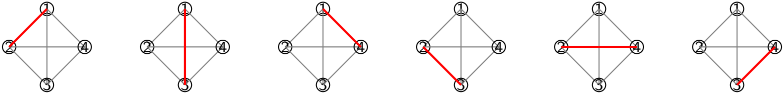



Paul Erdős (1913 - 1996)   Alfréd Rényi (1921-1970)

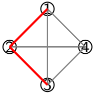
Erdős and Rényi (1959–60) launched the probabilistic study of graphs.

$G(N, p)$ Model

Take $N = 4$ then the graph can have up to six edges. Each with distribution $\mathrm{Bern}(p)$:



12 13 14 23 24 34

$\mathbb{P}($  $) = p^2(1-p)^4$

If $p = \frac{1}{2}$, each graph appears with the same probability $\frac{1}{2^6} = \frac{1}{64}$.

Probability recap: Binomial

Definition

If $X \sim \mathrm{Bin}(n, p)$ then

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad \mathbb{E}[X] = np, \quad \mathrm{Var}(X) = np(1-p).$$

Useful characterization: $X = \sum_{i=1}^{n} Z_i$ with independent $Z_i \sim \mathrm{Bern}(p)$.

In the ER graph $G(N, p)$:

- Number of edges:

$$L \sim \mathrm{Bin}\left(\binom{N}{2}, p\right).$$

- Degree of a fixed vertex v:

$$\deg(v) \sim \mathrm{Bin}(N-1, p).$$

Probability recap: Binomial

Definition

If $X \sim \mathrm{Bin}(n, p)$ then

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad \mathbb{E}[X] = np, \quad \mathrm{Var}(X) = np(1-p).$$

Useful characterization: $X = \sum_{i=1}^{n} Z_i$ with independent $Z_i \sim \mathrm{Bern}(p)$.

In the ER graph $G(N, p)$:

- Number of edges:

$$L \sim \mathrm{Bin}\left(\binom{N}{2}, p\right).$$

- Degree of a fixed vertex v:

$$\deg(v) \sim \mathrm{Bin}(N-1, p).$$

Probability recap: Poisson (as Binomial limit)

Theorem

If $X_n \sim \mathrm{Bin}(n, p_n)$ with $n \to \infty$ and $np_n \to \lambda > 0$, then

$$X_n \;\longrightarrow\; X \sim \mathrm{Pois}(\lambda), \qquad \mathbb{P}(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}.$$

The approximation $\mathrm{Bin}(n, p) \approx \mathrm{Poiss}(\lambda)$ for $\lambda = pn$ is particularly good if p is small.

Example (Quick check)

For $n = 2000$, $p = 0.003$, $\lambda = np = 6$. Compare $\mathbb{P}(X = 0)$: Binomial $= (1 - p)^{2000} \approx 0.00245$ vs. Poisson $e^{-6} \approx 0.00248$ (very close).

Degree distribution in $G(N, p)$

If $p = \lambda/(N-1)$, then, for any $v \in V$,

$$\deg(v) \ \sim \ \mathrm{Bin}(N-1, p) \ \approx \ \mathrm{Pois}(\lambda).$$

- Mean degree: $\mathbb{E}[\deg(v)] = (N-1)p$.
- $\mathbb{P}\left(\deg(v) = k\right) \approx \dfrac{\lambda^k}{k!} e^{-\lambda}$.

Note

This gives closed forms for expectations; Poisson is a great approximation when N is large and p small.