Ranking Nodes from an Adjacency Matrix

Eigenvectors, PageRank, and SVD-based Methods

Sergio Peignier

Network Analysis / Applied Linear Algebra

Motivation

Ranking nodes in a network is a fundamental task in data science and network analysis.

- Web search and PageRank
- Citation and collaboration networks
- Social networks and influence analysis
- Recommendation systems

Key idea: Node importance should be inferred from the structure of the graph itself.

Adjacency Matrix

Consider a directed graph with n nodes. Its adjacency matrix $A \in \mathbb{R}^{n \times n}$ is defined by

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge from node } j \text{ to node } i, \\ 0 & \text{otherwise.} \end{cases}$$

- Columns correspond to outgoing edges
- Rows correspond to incoming edges

This convention is convenient for influence propagation.

Degree-Based Ranking

A simple ranking measure is the *in-degree*:

$$d_i = \sum_{j=1}^{n} A_{ij}.$$

- Nodes with many incoming edges are considered important
- Easy to compute and intuitive

Limitation: All incoming edges are treated equally, regardless of where they come from.

Weighted Degree Ranking

A refinement:

A node is important if it is pointed to by other important nodes.

Let x_j denote the importance of node j. Then

$$x_i \propto \sum_{j=1}^{n} A_{ij} x_j .$$

In vector form:

$$\boldsymbol{x} \propto A\boldsymbol{x}.$$

This leads naturally to eigenvector-based ranking.

Eigenvector Centrality

We seek a nonzero vector $\boldsymbol{x}$ such that

$$A\boldsymbol{x} = \lambda \boldsymbol{x}.$$

- $\boldsymbol{x}$: ranking vector
- λ: eigenvalue

To obtain a proper ranking:

$$\boldsymbol{x} \leftarrow \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_1}.$$

This is known as **eigenvector centrality**.

Perron–Frobenius Theorem

For graphs with nonnegative connections:

- A unique largest (dominant) eigenvalue λ
- A unique eigenvector with strictly positive entries
- Convergence of iterative methods

This guarantees that eigenvector-based ranking is:

- Well-defined
- Stable
- Interpretable

Power Iteration Algorithm

The dominant eigenvector can be computed efficiently using **power iteration**:

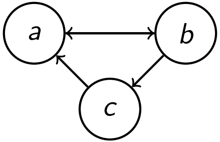
1. Initialize x_0 (e.g. uniform vector)
2. Iterate:

$$x_{k+1} = \frac{Ax_k}{\|Ax_k\|}$$

3. Stop when x_k converges

This method scales well to very large, sparse graphs.

Example Graph



Directed network

- Node a receives edges from B and C
- Node b receives an edge from a
- Node c receives an edge from b

Adjacency matrix

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

Eigenvector centrality

$$Av = \lambda v$$

$$v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad \Rightarrow \quad \text{score of node } i = |v_i|$$

Power Iteration Result

Applying power iteration yields

$$\boldsymbol{x} = (0.743,\ 0.557,\ 0.371)^{\top}.$$

- Node 1 is most important
- It is referenced by multiple nodes, including an influential one

This illustrates recursive importance propagation.

From Adjacency to Transition Matrix

We often normalize A into a column-stochastic matrix:

$$P_{ij} = \frac{A_{ij}}{\sum_k A_{kj}}.$$

- Each column sums to 1
- Importance is evenly distributed among outgoing edges

The ranking vector satisfies

$$\boldsymbol{x} = P\boldsymbol{x}.$$

Random Walk Interpretation

- P defines a random walk on the graph
- $\boldsymbol{x}$ is the stationary distribution

Interpretation: Ranking corresponds to the long-term visiting frequency of a random surfer.

PageRank

PageRank stabilizes the random walk:

$$P' = \alpha P + (1-\alpha)\frac{1}{n}\mathbf{1}\mathbf{1}^\top, \qquad \alpha \in [0.85, 0.95].$$

The ranking vector satisfies

$$\boldsymbol{x} = P'\boldsymbol{x}.$$

- Avoids dead ends
- Guarantees a unique positive solution

Graph Laplacian

Let D be the out-degree matrix:

$$D_{ii} = \sum_j A_{ij}.$$

The (unnormalized) Laplacian:

$$L = D - A.$$

The symmetric normalized Laplacian:

$$L_{\mathrm{sym}} = I - D^{-1/2} A D^{-1/2}.$$

Why Normalize?

Normalization:

- Compensates for heterogeneous degrees
- Prevents domination by high-degree nodes
- Emphasizes relative connectivity

Often A is symmetrized first:

$$A_{\text{sym}} = \frac{1}{2}(A + A^\top).$$

Singular Value Decomposition

The SVD of A is

$$A = U\Sigma V^\top,$$

where:

- $\Sigma = \mathrm{diag}(\sigma_1 \geq \sigma_2 \geq \cdots \geq 0)$
- Singular values capture dominant connectivity patterns

Undirected Networks

If A is symmetric:

$$U = V.$$

The leading singular vector u_1 gives:

$$\text{score}(i) = |u_1(i)|.$$

This coincides with eigenvector centrality.

Directed Networks

For directed graphs:

- Right singular vectors (V): authority-like behavior
- Left singular vectors (U): hub-like behavior

$$v_1 = \arg\max_{\|v\|=1} \|Av\|, \qquad u_1 = \arg\max_{\|u\|=1} \|A^\top u\|.$$

Low-Rank SVD Approximation

Many networks have strong low-dimensional structure.

$$A \approx U_k \Sigma_k V_k^\top .$$

- Captures communities and global patterns
- Filters noise and local fluctuations

SVD-Based Ranking

A multi-dimensional importance score:

$$\mathrm{score}(i) = \sqrt{\sum_{j=1}^{k} \sigma_j^2 u_{ij}^2}.$$

- Emphasizes dominant structures
- Robust to noise
- Interpretable as latent-space importance

Summary

Key Takeaways

- Ranking is based on recursive importance propagation
- Eigenvector centrality formalizes this idea
- Power iteration scales to massive graphs
- PageRank stabilizes random-walk rankings
- SVD generalizes ranking to multiple latent patterns
- Low-rank methods yield robust, noise-resistant rankings

Thank You

Questions?