Network Analysis with matrices

For us a Network is an undirected, unweighted graph G with N nodes.
Usually represented through a symmetric adjacency matrix $A \in \mathbb{R}^{N \times N}$

Many different centrality measures

- $deg(i) = \sum_{j=1}^{N} a_{ij} = (Ae)_i$ is the degree of node i
- eigenvector centrality $f_i = \frac{1}{\lambda_1} \sum_{j=1}^{N} a_{ij} f_j = \left(\frac{1}{\lambda_1} A f \right)_i$, where λ_1 and f is the Perron-Frobenius eigenpair.

Centrality measures

For any positive integer k, $A^k(i,j)$ counts the number of walks of length k in G that connect node i to node j.

A walk is an ordered list of nodes such that successive nodes in the list are connected. The nodes need not to be distinct.

The length of a walk is the number of edges that form the walk.

Centrality measures

Katz measure

$$k_i = \sum_{j=1}^{N} \sum_{k=1}^{\infty} \alpha^k (A)_{ij}^k = ((I - \alpha A)^{-1} - I)e)_i$$

We can introduce another centrality measure

$$c(i) = (exp(A))_{ii}$$

where the matrix function $exp(A)$ is defined as

$$exp(A) = I + A + \frac{1}{2}A^2 + \frac{1}{3!}A^3 + \frac{1}{4!}A^4 + \cdots$$

$c(i)$ accounts for the number of walks of any length from i to i, penalizing long walks respect to shorter ones.

Communicability and Betweenness

Communicability :The idea of counting walks can be extended to the case of a pair of distinct nodes, i and j.

$$C(i,j) = (exp(A))_{ij}$$

Betweenness : How does the overall communicability change when a node is removed?
Let $A - E(r)$ the adjacency matrix of the network with node r removed

$$B(r) = \frac{1}{(N-1)^2 - (N-1)} \sum_{i \neq j, i \neq r, j \neq r} \frac{\exp(A)_{ij} - exp(A - E(r))_{ij}}{(\exp(A))_{ij}}$$

f-centrality

We can extend the concept of centrality/communucability to $c(i) = \sum_{k=1}^{\infty} c_k (A^k)_{ii}$. Adding the coefficient c_0 if the series is convergent for any adjacency matrix A, taking

$$f(x) = \sum_{k=0}^{\infty} c_k x^k, c_k \geq 0$$

we can define

- f-centrality as $c(i) = f(A)_{ii}$
- f-communicability as $C(i,j) = f(A)_{ij}$

f-centrality

We can express A in terms of its spectrum $(\lambda_1 \geq \lambda_2 \leq \cdots \geq \lambda_N$
$A = \sum_{k=1}^{N} \lambda_k x_k x_k^T$ so we have

- f-centrality
$$c(i) = \sum_{k=1}^{N} f(\lambda_k)(x_k(i))^2,$$
- f-communicability
$$C(i,j) = \sum_{k=1}^{N} f(\lambda_k) x_k(i) x_k(j).$$

We can for example take the function

$$r(x) = \left(1 - \frac{x}{N-1}\right)^{-1}$$

In the case of large and sparse networks, $\lambda_k \in [-(N+2), N-2]$, and

$$c(i) = \sum_{k=1}^{N} \frac{N-1}{N-1-\lambda_k} x_k(i)^2,$$

Graph Laplacian and Spectral clustering

Problem : partition nodes into two groups so that we have high intra-connection and low inter-connections

Let $x \in \mathbb{R}^N$ be an indicator vector $x_i = 1/2$ if i belongs to the first cluster, $x_i = -1/2$ if i otherwise.

$$\sum_{i=1}^{N} \sum_{j=1}^{N} (x_i - x_j)^2 a_{ij}$$

counts the number of edges through the cut.
Relax the problem

$$\min_{x \in \mathbb{R}^N : \|x\|_2 = 1 \sum_i x_i = 0} \sum_{j=1}^{N} (x_i - x_j)^2 a_{ij}$$

Let $D = diag(deg(i))$, we have

$$\min_{x\in\mathbb{R}^N:\|x\|_2=1\ \sum_i x_i=0} x^T(D-A)x.$$

The matrix $D - A$ is colled the Graph Laplacian

- $(D - A)e = 0$ so 0 is eigenvalue and the corresponding eigenvector is e
- $D - A$ has nonegative eigenvalues, and the algebric multiplitity of $\mu_1 = 0$ is the number of connected components of the graph
- if the graph is connected $0 = \mu_1 < \mu_2 \leq \cdots \leq \mu_N$ with eigenvectors $e = v_1, v_2, \ldots v_N$, the v_2 solves the optimization problem

$$v_2 = \operatorname*{argmin}_{x\in\mathbb{R}^N:\|x\|_2=1\ \sum_i x_i=0} x^T(D-A)x.$$

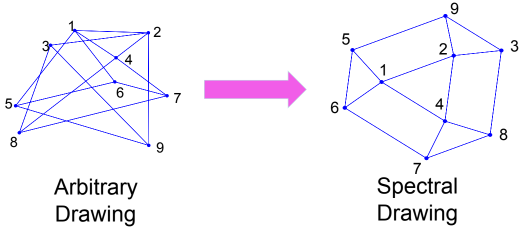
v_2 is called the Fiedler vector of the graph.

Fiedler vector

The Fiedler vector can be the used to

- cluster nodes into two sets, $v_2(i)v_2(j) > 0$, i, j belongs to the same cluster.
- reordering nodes in such a way $i \leq j \implies v_2(i) \leq v_2(j)$
- μ_2 is big iff G has not good clusters
- μ_2 is smal iff G has good clusters

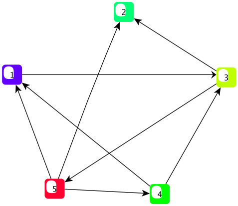
Graph drawing: use spectral coordinates $(v_2(i), v_3(i))$ to draw the graph



Web Graph

The Web is seen as a directed graph:

- Each page is a node
- Each hyperlink is an edge



$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

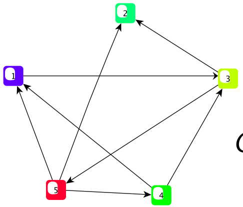
Google's PageRank

- Is a static ranking schema
- At query time relevant pages are retrieved
- The ranking of pages is based on the PageRank of pages which is precomputed
- A page is important if is voted by important pages
- The vote is expressed by a link

PageRank

- A page distribute its importance equally to its neighbours
- The importance of a page is the sum of the importances of pages which points to it

$$\pi_j = \sum_{i \in \mathcal{I}(j)} \frac{\pi_i}{\text{outdeg}(i)}$$



$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 \end{bmatrix}$$

P is row stochastic, $\sum_{j=1}^{N} p_{ij} = 1$.

It is called Random surfer model

The web surfer jumps from page to page following hyperlinks. The probability of jumping to a page depends of the number of links in that page.

Starting with a vector $\pi^{(0)}$, compute

$$\pi_j^{(k)} = \sum_{i \in \mathcal{I}(j)} \pi_i^{(k-1)} p_{ij}, \quad p_{ij} = \frac{1}{\text{outdeg}(i)}$$

It is called Random surfer model

The web surfer jumps from page to page following hyperlinks. The probability of jumping to a page depends of the number of links in that page.

Starting with a vector $\pi^{(0)}$, compute

$$\pi_j^{(k)} = \sum_{i \in \mathcal{I}(j)} \pi_i^{(k-1)} p_{ij}, \quad p_{ij} = \frac{1}{\text{outdeg}(i)}$$

Equivalent to compute the stationary distribution of the Markov chain with transition matrix P.

It is called Random surfer model

The web surfer jumps from page to page following hyperlinks. The probability of jumping to a page depends of the number of links in that page.

Starting with a vector $\pi^{(0)}$, compute

$$\pi_j^{(k)} = \sum_{i \in \mathcal{I}(j)} \pi_i^{(k-1)} p_{ij}, \quad p_{ij} = \frac{1}{\text{outdeg}(i)}$$

Equivalent to compute the stationary distribution of the Markov chain with transition matrix P.

Equivalent to compute the left eigenvector of P corresponding to eigenvalue 1.

PageRank

Two problems:

- Presence of dangling nodes
  - P cannot be stochastic
  - P might not have the eigenvalue 1
- Presence of cycles
  - The random surfer get trapped
  - more than an eigenvalue equal to the spectral radius

Perron-Frobenius Theorem

Let $A \geq 0$ be an irreducible matrix

- there exists an eigenvector equal to the spectral radius of A, with algebraic multiplicity 1
- there exists an eigenvector $\mathbf{x} > \mathbf{0}$ such that $A\mathbf{x} = \rho(A)\mathbf{x}$.
- if $A > 0$, then $\rho(A)$ is the unique eigenvalue with maximum modulo.

The same as the ergoodic theorem for Markov chians

PageRank

Presence of dangling nodes

$$\bar{P} = P + \mathbf{d}\mathbf{v}^T$$

$$d_i = \begin{cases} 1 & \text{if page } i \text{ is dangling} \\ 0 & \text{otherwise} \end{cases} \qquad v_i = 1/n;$$

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 \end{bmatrix} \bar{P} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 \end{bmatrix}$$

PageRank

Presence of cycles

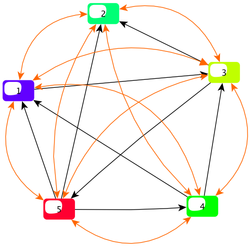
Force irreducibility by adding artificial arcs chosen by the random surfer with “small probability” α.

$$\hat{P} = (1-\alpha)\bar{P} + \alpha \mathbf{e}\mathbf{v}^T,$$

$$\hat{P} = (1-\alpha)\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 \end{bmatrix} + \alpha \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix}.$$

Typical values of α is 0.15.

A toy eample



$$\hat{P} = \begin{bmatrix} 0.05 & 0.05 & 0.8 & 0.05 & 0.05 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.05 & 0.425 & 0.05 & 0.05 & 0.425 \\ 0.425 & 0.05 & 0.425 & 0.05 & 0.05 \\ 0.3 & 0.3 & 0.05 & 0.3 & 0.05 \end{bmatrix}$$

Computing the largest left eigenvector of $\hat{P}$ we get

$$\pi^T \approx [0.39, 0.51, 0.59, 0.29, 0.40],$$

which corresponds to the following order of importance of pages

$$[3, 2, 5, 1, 4].$$

PageRank

- P is sparse, $\hat{P}$ is full.
- The vector $y^T = x^T \hat{P}$, for $x \geq 0$, such that $\|x\|_1 = 1$ can be computed as follows

$$\begin{aligned} y^T &= (1-\alpha) x^T P \\ \gamma &= \|x\|_1 - \|y\|_1 = 1 - \|y\|_1, \\ y &= y + \gamma v. \end{aligned}$$

- The eigenvalues of $\bar{P}$ and $\hat{P}$ are related:

$$\lambda_1(\bar{P}) = \lambda_1(\hat{P}) = 1, \quad \lambda_j(\hat{P}) = (1-\alpha)\,\lambda_j(\bar{P}), j > 1.$$

- For the web graph $|\lambda_2(\hat{P})| \leq (1-\alpha)$, $\lambda_2(\hat{P}) = (1-\alpha)$ if the graph has at least two strongly connected components

Generally solved by the power method: rate of convergence $|\lambda_2|/|\lambda_1|$.