Matrix methods in the analysis of complex networks

Second-order random walks

Dario Fasino

Rome, Univ. "Tor Vergata", November 22–24, 2022

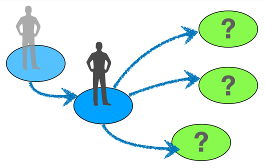
Introduction

Classical random walks suffer from a few drawbacks: localization phenomena, limited expressiveness, slow mixing rates...

Extend the random walk idea by accounting for an earlier step in navigation:

$$\mathbb{P}(X_{t+1} = i | X_t = j, X_{t-1} = k) = \mathbf{P}_{ijk}$$

Second-order random walk



- $Y(t)$: joint probability matrix,.

  $Y(t)_{ij} = \mathbb{P}(X_t = i, X_{t-1} = j)$
- $\mathbf{P}$: stochastic transition tensor, $\sum_i \mathbf{P}_{ijk} = 1$
- $x(t)$: probability vector, $x(t)_i = \sum_j Y(t)_{ij}$.

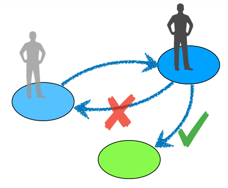
Introduction

Classical random walks suffer from a few drawbacks: localization phenomena, limited expressiveness, slow mixing rates...

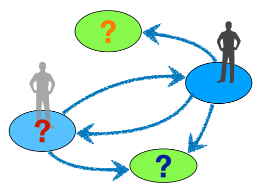
Extend the random walk idea by accounting for an earlier step in navigation:

$$\mathbb{P}(X_{t+1} = i | X_t = j, X_{t-1} = k) = \mathbf{P}_{ijk}$$

non-backtracking rw



`node2vec` **algorithm**



Introduction

Classical random walks suffer from a few drawbacks: localization phenomena, limited expressiveness, slow mixing rates...

Extend the random walk idea by accounting for an earlier step in navigation:

$$\mathbb{P}(X_{t+1} = i | X_t = j, X_{t-1} = k) = \mathbf{P}_{ijk}$$

But then we need extra info:

$$\mathbb{P}(X_1 = j | X_0 = i) = p'_{ij}.$$

Together with x_0, this yields the initial condition for a two-phase iteration:

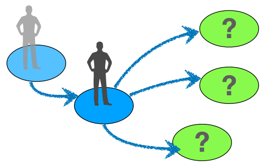
$$Y(1)_{ij} = \mathbb{P}(X_1 = i, X_0 = j), \qquad \begin{cases} Y(t+1)_{ij} = \sum_k \mathbf{P}_{ijk} Y(t)_{jk} \\ x(t+1) = Y(t+1)e. \end{cases}$$

Second-order random walks

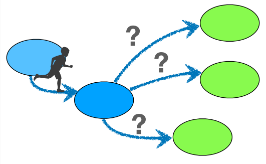
The "lifting" idea: Introduce a walker moving from edge to edge,

$$\mathbf{P}_{ijk} = \mathbb{P}(W_{t+1} = (j,i) | W_t = (k,j))$$

This converts the second-order rw on V . . .



. . . to a first-order rw on E



Second-order rw.s on $\mathcal{G} = (V, E)$ correspond to first-order rw.s on $\widehat{\mathcal{G}} = (E, \widehat{E})$, the (directed) line graph of $\mathcal{G}$.

Directed line graph

The (directed) line graph $\mathcal{L}(\mathcal{G})$ of a digraph $\mathcal{G}$ is the graph defined as follows:

- The vertex set of $\mathcal{L}(\mathcal{G})$ is the edge set of $\mathcal{G}$
- (e_1, e_2) is an edge in $\mathcal{L}(\mathcal{G})$ iff $e_1 = (i, j)$ and $e_2 = (j, k)$ in $\mathcal{G}$.

Notation: $|V| = n$, $|E| = m$.

Define the source matrix $S \in \mathbb{R}^{n \times m}$ and the target matrix $T \in \mathbb{R}^{m \times n}$,

$$S_{ie} = \begin{cases} 1 & e = (i, *) \\ 0 & \text{otherwise,} \end{cases} \qquad T_{ej} = \begin{cases} 1 & e = (*, j) \\ 0 & \text{otherwise.} \end{cases}$$

Then $A = ST$ is the adj. matrix of $\mathcal{G}$ and $B = TS$ is the adj. matrix of $\mathcal{L}(\mathcal{G})$. Basically, $R = T^{\mathrm{T}}$.

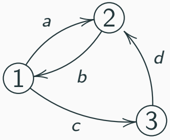
Directed line graph

The (directed) line graph $\mathcal{L}(\mathcal{G})$ of a digraph $\mathcal{G}$ is the graph defined as follows:

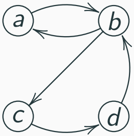
- The vertex set of $\mathcal{L}(\mathcal{G})$ is the edge set of $\mathcal{G}$
- (e_1, e_2) is an edge in $\mathcal{L}(\mathcal{G})$ iff $e_1 = (i, j)$ and $e_2 = (j, k)$ in $\mathcal{G}$.

Example:

$\mathcal{G}$:



$\mathcal{L}(\mathcal{G})$:



$$S = \begin{array}{@{}c@{}l} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \begin{array}{l} 1 \\ 2 \\ 3 \end{array} \end{array} \qquad T = \begin{array}{@{}c@{}l} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} & \begin{array}{l} a \\ b \\ c \\ d \end{array} \end{array}$$

Knowing the adjacency matrix $B = TS$, it is then possible to define the random walks on the line graph. However, attention must be paid to some pitfalls, e.g., sink nodes in $\mathcal{L}(\mathcal{G})$.

Example: The non-backtracking random walk cannot be defined in this graph:

$\mathcal{G}:$

The Hashimoto graph and matrix

The non-backtracking random walk can be associated to the RW on the Hashimoto graph, whose adjacency matrix is the Hashimoto matrix

$$H = B - B \circ B^{\mathrm{T}} ,$$

where $\circ$ is the Hadamard (entrywise) matrix product.

Second-order RWs - construction

Knowing the adjacency matrix $B = TS$, it is then possible to define the random walks on the line graph. However, attention must be paid to some pitfalls, e.g., sink nodes in $\mathcal{L}(\mathcal{G})$.

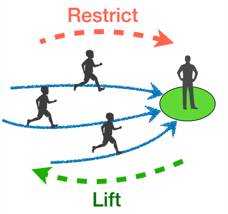
The factorization $B = ST$ helps in devising efficient algorithms for matrix-vector products in $\mathcal{L}(\mathcal{G})$. As a result, if graphs and matrices are sparse then the computational overhead of 2nd-order RWs is irrelevant.

After defining a sound RW on $\mathcal{L}(\mathcal{G})$, the quantities of interest need to be transferred back to the original graph: stationary density, hitting times, return times...

Second-order RWs - stationary density

- $L : V \mapsto E$, the "lifting matrix"
- $R : E \mapsto V$, the "restriction matrix"

$$\{W_t\} \underset{L}{\overset{R}{\rightleftarrows}} \{X_t\}$$

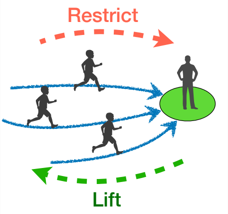


- The lifting matrix L apportions $\mathbb{P}(X_t = i)$ among the incoming links and determines $\mathbb{P}(W_t = e)$ for any edge $e = (*, i)$;
- the restriction matrix collects in each node $i \in V$ the occupancy probabilities of incoming links, so that $\mathbb{P}(X_t = i) = \sum_{e=(*,i)} \mathbb{P}(W_t = e)$. Basically, $R = T^{\mathrm{T}}$.

Second-order RWs - stationary density

- $L : V \mapsto E$, the "lifting matrix"
- $R : E \mapsto V$, the "restriction matrix"

$$\{W_t\} \underset{L}{\overset{R}{\rightleftarrows}} \{X_t\}$$



Theorem

Let $\widehat{P}$ be the transition matrix of $\{W_t\}$. If $\widehat{P}$ is ergodic with stationary density $\widehat{\pi}$ then the pullback matrix $P = R\widehat{P}L$ is irreducible, column stochastic, and

$$P_{ij} = \mathbb{P}(X_{t+1} = i | X_t = j), \qquad \forall t \geq 1.$$

The stationary density of P is $\pi = R\widehat{\pi}$.

Hitting times and return times for 2nd-order RWs

Pseudo-theorem

The second-order hitting times matrix for $\{X_t\}$ can be obtained by an appropriate "squeezing" of the hitting times matrix for $\{W_t\}$. (Rigorous statement and explicit formulas in reference.)

Theorem

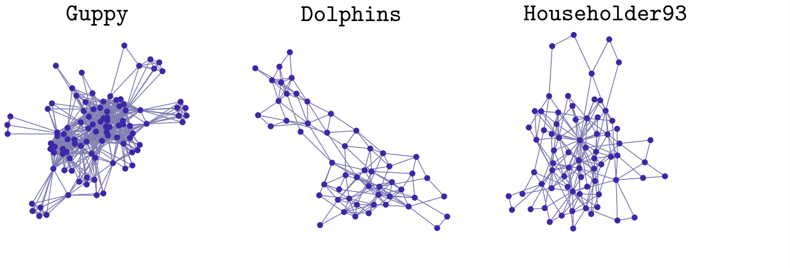
Let $\widetilde{\rho}_i$ be the 2nd-order return time to $i \in V$ of $\{X_t\}$. Then

$$\widetilde{\rho}_i = 1/\pi_i,$$

where $\pi = (\pi_1, \ldots, \pi_n)^{\mathrm{T}}$ is the stationary density of the pullback matrix $P = L\widehat{P}R$.

Examples

network	nodes	edges	diam.
Guppy	98	725	5
Dolphins	53	150	7
Householder93	73	180	5



Examples

network	nodes	edges	diam.
`Guppy`	98	725	5
`Dolphins`	53	150	7
`Householder93`	73	180	5

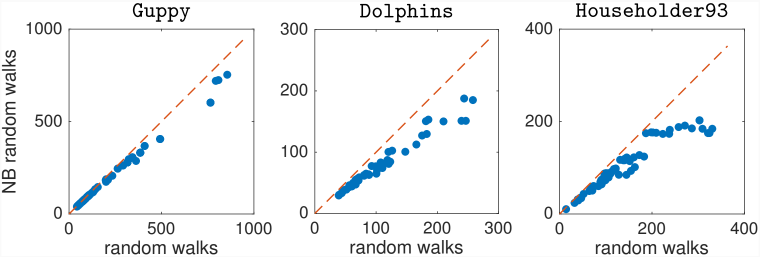


Figure 1: The mean hitting time $(\sum_{j=1}^{n} \tau_{i\to j})/n$ computed from classical (x-axis) or non-backtracking (y-axis) random walks.
Red dotted line: the $y = x$ line.

Examples

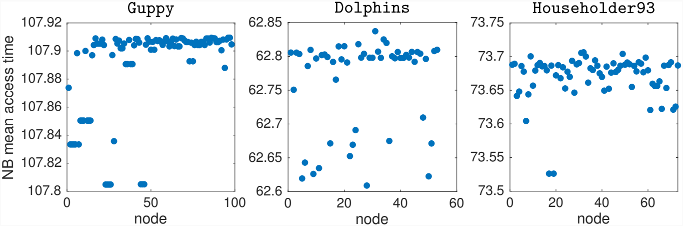


Figure 2: The non-backtracking mean access time $m_i = \sum_{j=1}^{n} \pi_j \tau_{i \to j}$.

In classical (first-order) rw.s we have $m_i = \kappa$, Kemeny's constant.

network	Guppy	Dolphins	Householder93
Kemeny	119.03	84.524	97.697

Three-mode stochastic tensors

Three-mode tensor: $\mathbf{P} = (\mathbf{P}_{ijk}) \in \mathbb{R}^{n \times n \times n}$.

Tensor-vector-vector product: $z = \mathbf{P}xy$ is the vector

$$z_i = \sum_{j,k} \mathbf{P}_{ijk} x_j y_k.$$

Z-eigenvector: A vector $x \neq 0$ such that $\mathbf{P}xx = \lambda x$ for some $\lambda \in \mathbb{R}$.

Let $\mathbf{P}$ be a stochastic tensor, $\mathbf{P}_{ijk} \geq 0$ and $\sum_i P_{ijk} = 1$. If $x, y \in \mathcal{S}$ and $z = \mathbf{P}xy$ then $z \in \mathcal{S}$.

The equation $x = \mathbf{P}xx$ admits at least one solution $x \in \mathcal{S}$. But there can also be Z-eigenvectors with $|\lambda| \geq 1$.

A simplified approach

Previously introduced notation:

- $Y(t)$: joint probability matrix, $Y(t)_{ij} = \mathbb{P}(X_t = i, X_{t-1} = j)$
- $\mathbf{P}$: stochastic transition tensor, $\mathbf{P}_{ijk} = \mathbb{P}(X_{t+1} = i | X_t = j, X_{t-1} = k)$
- $x(t)$: probability vector, $x(t)_i = \mathbb{P}(X_t = i)$.

$$\begin{cases} Y(t+1)_{ij} = \sum_k \mathbf{P}_{ijk} Y(t)_{jk} \\ x(t+1) = Y(t+1)e. \end{cases}$$

Although exact, this approach requires the time-consuming computation of the sequence of joint probability matrices $\{Y(t)\}$.

Idea

Approximate the exact model by condensing the information in the $n \times n$ matrix $Y(t)$ into one or two probability n-vectors, e.g., replace $Y(t)$ by $x(t)x(t-1)^{\mathrm{T}}$.

Structured, 2nd-order stochastic process

- 2nd-order process $x_{k+1} = \mathbf{P} x_k x_{k-1}$ [Li, Ng 2014], [Wu, Chu 2017]
- Bilinear (shifted) power method [Kolda, Mayo 2011]

$$x_{k+1} = \alpha \mathbf{P} x_k x_k + (1-\alpha) x_k$$

- Nonlinear PageRank [Gleich, Lim,Yu 2015]

$$x_{k+1} = \alpha \mathbf{P} x_k x_{k-1} + (1-\alpha) v, \qquad \alpha \in (0,1), \quad v \in \mathcal{S}.$$

- Spacey random walk [Benson, Gleich, Lim 2017]

$$\begin{cases} x_{k+1} = \mathbf{P} x_k y_k \\ y_{k+1} = c_k x_k + (1-c_k) y_k. \end{cases}$$

Dobrushin-type coefficients for 3-mode tensors

Let $\mathcal{S} = \{x \geq 0, e^{\mathrm{T}} x = 1\}$ be the set of probability n-vectors

The Dobrushin coefficient

$$\mathcal{T}(\mathbf{P}) = \max_{x \in \mathcal{S}} \sup_{y : e^{\mathrm{T}} y = 0} \frac{\|\mathbf{P}xy + \mathbf{P}yx\|_1}{\|y\|_1}.$$

Explicit formula:

$$\mathcal{T}(\mathbf{P}) = \frac{1}{2} \max_{j, k_1, k_2} \sum_i |\mathbf{P}_{ijk_1} - \mathbf{P}_{ijk_2} + \mathbf{P}_{ik_1j} - \mathbf{P}_{ik_2j}|$$

Properties:

- $0 \leq \mathcal{T}(\mathbf{P}) \leq 2$
- $\mathcal{T}(\mathbf{P}) = 0 \iff \mathbf{P}_{ijk} = v_i$ for some $v \in \mathcal{S}$.
- If $x = \mathbf{P}xx$ and $x' = \mathbf{P}'x'x'$ (probability vectors) then $\|x - x'\|_1 \leq \|\mathbf{P} - \mathbf{P}'\|_1 / (1 - \mathcal{T}(\mathbf{P}))$.

Applications: nonlinear PageRank

Let $v \in \mathcal{S}$. The nonlinear PageRank vector

$$x = \alpha \mathbf{P} x x + (1-\alpha) v \qquad x \in \mathcal{S},$$

corresponds to the stationary solution of

$$\begin{aligned} x_{k+1} &= \alpha \mathbf{P} x_k x_k + (1-\alpha) v \\ &= (\alpha \mathbf{P} + (1-\alpha)\mathbf{V}) x_k x_k \end{aligned}$$

where $\mathbf{V}_{ijk} = v_i$ and $x_0 \in \mathcal{S}$. Note: $\mathcal{T}(\mathbf{V}) = 0$.

Theorem

If $\alpha \mathcal{T}(\mathbf{P}) < 1$ (e.g., $\alpha < 1/2$) then the solution is unique.
Moreover, $\forall x_0 \in \mathcal{S}$ the iteration converges to x and

$$\|x_k - x\|_1 \leq (\alpha \mathcal{T}(\mathbf{P}))^k \|x_0 - x\|_1.$$

References

D. F., F. Tudisco. Ergodicity coefficients for higher-order stochastic processes. *SIAM J. Math. Data Sci.*, 2 (2020), 740–769.

D. F., A. Tonetto, F. Tudisco. Hitting times for second-order random walks. *Europ. J. Appl. Math.* 2023, in press.