Remember where you came from:
Hitting times for second-order random walks

Dario Fasino
Univ. Udine

Napoli, Feb. 15, 2022
At a glance

Second-order random walks are increasingly popular. How can we define (and compute) their hitting/return times? And what properties do they have?

D. F., A. Tonetto, F. Tudisco
Hitting times for second-order random walks.
Intro

Random walk on a graph $G = (V, E)$

- Start from a node chosen $i \in V$ with some probability
  - Pick one of the outgoing edges
  - Move to the destination of the edge
  - Repeat.

Applications

Random walks are widely used in network science to

- model user navigation and diffusive processes
- quantify node centrality and accessibility
- reveal network communities and core-periphery structures.
Random walk on a graph $G = (V, E)$

- Start from a node chosen $i \in V$ with some probability
  - Pick one of the outgoing edges
  - Move to the destination of the edge
  - Repeat.

First-order random walk

- $X_t$: node visited at time $t = 0, 1, \ldots$
- $x_t$: probability vector of $X_t$
- $P$: (row stochastic) transition matrix $P_{ij} = \mathbb{P}(X_{t+1} = j \mid X_t = i)$

\[ x_{t+1}^T = x_t^T P. \]
Random walk on a graph $\mathcal{G} = (V, E)$

- Start from a node chosen $i \in V$ with some probability
  - Pick one of the outgoing edges
  - Move to the destination of the edge
  - Repeat.

Second-order random walk

- $Y_t$: joint probability matrix
  $$(Y_t)_{ij} = \mathbb{P}(X_t = j, X_{t-1} = i)$$
- $\mathbf{P}$: transition tensor
  $$\mathbf{P}_{ijk} = \mathbb{P}(X_{t+1} = k|X_t = j, X_{t-1} = i)$$
  $$\left\{ \begin{array}{l}
  (Y_{t+1})_{jk} = \sum_i \mathbf{P}_{ijk}(Y_t)_{ij} \\
  x_{t+1} = \mathbf{1}^T Y_{t+1}
  \end{array} \right.$$
Let $S \subset V$, with $G = (V, E)$ strongly connected.

- $\tau_{i \to S}$: hitting time to $S$ starting from $i$.
  \[
  \tau_{i \to S} = \begin{cases} 
  0 & i \in S \\
  1 + \sum_{k=1}^{n} P_{ik} \tau_{k \to S} & i \notin S.
  \end{cases}
  \]

- $\tau_{i \to S}^+$: return time to $S$ starting from $i$.
  \[
  \tau_{i \to S}^+ = \begin{cases} 
  1 + \sum_{k=1}^{n} P_{ik} \tau_{k \to S} & i \in S \\
  0 & i \notin S.
  \end{cases}
  \]

- $\pi = (\pi_1, \ldots, \pi_n)^T$: the stationary probability vector,
  \[
  \pi > 0, \quad \pi^T = \pi^T P, \quad 1^T \pi = 1.
  \]
For a fixed $S \subset V$, let $t^S$ and $b^S$ be the vectors with the hitting times and return times to $S$, respectively:

$$t^S_i = \begin{cases} 
0 & i \in S \\
\tau_{i \rightarrow S} & i \notin S 
\end{cases}$$

$$b^S_i = \begin{cases} 
\tau_{i \rightarrow S}^+ & i \in S \\
0 & i \notin S 
\end{cases}$$

The vector $t^S$ solves the singular equation $(I - P)t^S = 1 - b^S$.

**Hitting time matrix**

The hitting time matrix $T = (\tau_{i \rightarrow j})$ solves the equation

$$(I - P)T = 11^T - \text{Diag}(\rho_1, \ldots, \rho_n)$$

where $\rho_i = \tau_{i \rightarrow i}^+$ is the return time to node $i$. 
For a fixed $S \subset V$, let $t^S$ and $b^S$ be the vectors with the hitting times and return times to $S$, respectively:

$$
t^S_i = \begin{cases} 
0 & i \in S \\
\tau_{i \rightarrow S} & i \notin S 
\end{cases}
$$

$$
b^S_i = \begin{cases} 
\tau_{i \rightarrow S} & i \in S \\
0 & i \notin S.
\end{cases}
$$

The vector $t^S$ solves the singular equation $(I - P)t^S = 1 - b^S$.

**Kac’s lemma**

Let $\rho_S = \sum_{i \in S} \pi_i \tau_{i \rightarrow S}^+$ be the average return time to $S$. Then,

$$
\rho_S = (\sum_{i \in S} \pi_i)^{-1}.
$$

In particular, $\rho_i = 1/\pi_i$. 

From first- to second-order rw.s

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

Extend 1-order random walk idea by accounting for an earlier step in navigation:

\[ P(X_{n+1} = k \mid X_n = j, X_{n-1} = i) = p_{ijk} \]
Classical random walks suffer from a few drawbacks: localization, limited expressiveness, slow mixing rates. . .

Extend 1-order random walk idea by accounting for an earlier step in navigation:

\[
P(X_{n+1} = k | X_n = j, X_{n-1} = i) = p_{ijk}
\]

non-backtracking rw
Classical random walks suffer from a few drawbacks: localization, limited expressiveness, slow mixing rates. . .

Extend 1-order random walk idea by accounting for an earlier step in navigation:

$$P(X_{n+1} = k | X_n = j, X_{n-1} = i) = p_{ijk}$$

non-backtracking rw node2vec algorithm
From second- to first-order random walks

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

\[ p_{ijk} = P(W_{n+1} = (j, k) | W_n = (i, j)) \]

This converts the second-order rw on \( V \) ... to a first-order rw on \( E \).
From second- to first-order random walks

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

\[ p_{ijk} = \mathbb{P}(W_{n+1} = (j, k) | W_n = (i, j)) \]

This converts the second-order rw on \( V \) …

…to a first-order rw on \( E \)

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

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

$$\begin{align*}
\{ W_t \} & \overset{L}{\underset{R}{\leftrightarrow}} \{ X_t \}
\end{align*}$$
Main results

- \( L : V \leftrightarrow E \), the “lifting matrix”
- \( R : E \leftrightarrow V \), the “restriction matrix”

\[ \{W_t\} \xrightarrow{R} \{X_t\} \xleftarrow{L} \{W_t\} \]

**Theorem**

Let \( \hat{P} \), \( \hat{\pi} \) be the transition matrix and stationary density of \( \{W_t\} \). The matrix \( P = L\hat{P}R \) is irreducible, row stochastic, and

\[
P_{ij} = \mathbb{P}(X_{t+1} = j \mid X_t = i), \quad t \geq 1.
\]

The stationary density of \( P \) is \( \pi^T = \hat{\pi}^T R \).

We call \( P \) the pullback of \( \hat{P} \).
Main results

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

\[
\begin{align*}
\{W_t\} & \xrightarrow{R} \{X_t\} \\
\{X_t\} & \xrightarrow{L} \{W_t\}
\end{align*}
\]

Theorem

Let $\tilde{T} = (\tilde{\tau}_{i \to j})$ be the second-order hitting times matrix for $\{X_t\}$ and $\hat{T} = (\hat{\tau}_{e \to f})$ the hitting time matrix for $\{W_t\}$. Then,

\[
\tilde{T} = L\hat{T}\text{Diag}(a)R - \mathbb{1}b^T,
\]

for some (explicitly known) vectors $a, b$. 
Main results

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

\[
\{ W_t \} \xleftrightarrow{L} \{ X_t \} \xleftrightarrow{R}
\]

Corollary

Let $X$ be any solution of $(I - \hat{P})X = 11^T - \text{Diag}(\hat{\pi})^{-1}$. Then $\tilde{T} = (\tilde{\tau}_{i \to j})$ is given by

\[
\tilde{T} = LX \text{Diag}(a)R - 1b^T,
\]

where $b$ is chosen so that $\tilde{T}_{ii} = 0$. 
Main results - A 2nd-order Kac’s-type result

Corollary

Let $\tilde{\rho}_S$ be the average 2nd-order average return time to $S \subset V$ of \( \{X_t\} \). Then

$$\tilde{\rho}_S = 1/(\sum_{i \in S} \pi_i),$$

where $\pi = (\pi_1, \ldots, \pi_n)^T$ is the stationary density of the pullback matrix $P$.

Proof. Let $\mathcal{I} = \bigcup_{i \in S} \{(j, i) \in E\}$. By Kac’s lemma,

$$\tilde{\rho}_S = \hat{\rho}_\mathcal{I} = \left( \sum_{(j,i) \in \mathcal{I}} \hat{\pi}(j,i) \right)^{-1} = \left( \sum_{i \in S} (\hat{\pi}^T R)_i \right)^{-1} = (\sum_{i \in S} \pi_i)^{-1}.$$
### Examples

<table>
<thead>
<tr>
<th>network</th>
<th>nodes</th>
<th>edges</th>
<th>diam.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guppy</td>
<td>98</td>
<td>725</td>
<td>5</td>
</tr>
<tr>
<td>Dolphins</td>
<td>53</td>
<td>150</td>
<td>7</td>
</tr>
<tr>
<td>Householder93</td>
<td>73</td>
<td>180</td>
<td>5</td>
</tr>
</tbody>
</table>

![Network Diagrams]

![Guppy](image1.png)  ![Dolphins](image2.png)  ![Householder93](image3.png)
### Examples

<table>
<thead>
<tr>
<th>network</th>
<th>nodes</th>
<th>edges</th>
<th>diam.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guppy</td>
<td>98</td>
<td>725</td>
<td>5</td>
</tr>
<tr>
<td>Dolphins</td>
<td>53</td>
<td>150</td>
<td>7</td>
</tr>
<tr>
<td>Householder93</td>
<td>73</td>
<td>180</td>
<td>5</td>
</tr>
</tbody>
</table>

**Figure:** The mean hitting time $c_i = \left( \sum_{j=1}^{n} \tau_{i\rightarrow j} \right) / n$ computed from 1-order ($x$-axis) and non-backtracking ($y$-axis) random walks. Red dotted line: the $y = x$ line.
Experiments with a second-order rw depending on $\alpha \in [0, 1]$ that interpolates between classical ($\alpha = 1$) and non-backtracking ($\alpha = 0$) random walks.

**Figure:** Mean hitting times $c_i = (\sum_{j=1}^{n} \tau_{i \to j})/n$ normalized to the $\alpha = 1$ case. Solid lines: maximum, average, and minimum values as $\alpha \in [0, 1]$. 
Kemeny’s (almost) constant

Corollary

Let $\mathcal{G} = (V, E)$ be such that, for every pair of edges $(j, i), (k, i) \in E$ there exists a graph automorphism $\varphi : V \mapsto V$ such that $\varphi(j) = k$ and $\varphi(k) = j$. Then

$$\sum_{j=1}^{n} \pi_j \tilde{\tau}_{i \rightarrow j} = \kappa'$$

for some constant $\kappa' < \kappa$, where $\pi$ is the stationary density of the pullback and $\kappa$ the Kemeny’s constant of the rw on $\mathcal{G}$. 
**Figure:** The non-backtracking mean access time $m_i = \sum_{j=1}^{n} \pi_j \tau_{i \rightarrow j}$.

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

<table>
<thead>
<tr>
<th>network</th>
<th>Guppy</th>
<th>Dolphins</th>
<th>Householder93</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kemeny</td>
<td>119.03</td>
<td>84.524</td>
<td>97.697</td>
</tr>
</tbody>
</table>
Thank you for your attention.