A Higher-Order Temporal H-Index for Evolving Networks

Lutz Oettershagen[1] Nils M. Kriege[2] Petra Mutzel[3]

October 24, 2024

1) University of Liverpool 2) University of Vienna 3) University of Bonn

Static H-Index

- The H-index was proposed for measuring the productivity and impact of scientists:

  (J. E. Hirsch, 2005)

  The maximum value of h such that the author has published at least h papers that have each been cited at least h times

- Has been used as centrality measure for ranking nodes according to spreading ability

Static H-Index

- The H-index was proposed for measuring the productivity and impact of scientists:

  (J. E. Hirsch, 2005)

  The maximum value of h such that the author has published at least h papers that have each been cited at least h times

- Has been used as centrality measure for ranking nodes according to spreading ability

Static H-Index

- The H-index was proposed for measuring the productivity and impact of scientists:

  (J. E. Hirsch, 2005)

  The maximum value of h such that the author has published at least h papers that have each been cited at least h times

- Has been used as centrality measure for ranking nodes according to spreading ability

Static n-th Order H-Index (Lü et al., 2016)

Let $\mathcal{M}$ be the set of finite multisets of integers.

- Function $\mathcal{H}\colon \mathcal{M} \to \mathbb{N}_0$ returns for a finite multiset of integers $S \subseteq \{\!\{ j \mid j \in \mathbb{N}_0 \}\!\}$ the maximum integer i such that there are at least i elements j in S with $j \geq i$.

n-th order H-index

n-th order H-index $s_u^{(n)}$ of a node $u \in V$ in a static graph $G = (V, E)$:
Let $s_u^{(0)} = \delta(u)$ the degree of node u, then

$$s_u^{(n)} = \mathcal{H}\left(\{\!\{ s_v^{(n-1)} \mid v \in V \text{ and } v \text{ is neighbor of } u \}\!\}\right)$$

$s_u^{(1)}$ corresponds to the H-index of u

Problem: Only considers static structure of network!

Static n-th Order H-Index (Lü et al., 2016)

Let $\mathcal{M}$ be the set of finite multisets of integers.

- Function $\mathcal{H} \colon \mathcal{M} \to \mathbb{N}_0$ returns for a finite multiset of integers $S \subseteq \{\!\{j \mid j \in \mathbb{N}_0\}\!\}$ the maximum integer i such that there are at least i elements j in S with $j \geq i$.

n-th order H-index

n-th order H-index $s_u^{(n)}$ of a node $u \in V$ in a static graph $G = (V, E)$:
Let $s_u^{(0)} = \delta(u)$ the degree of node u, then

$$s_u^{(n)} = \mathcal{H}\left(\{\!\{s_v^{(n-1)} \mid v \in V \text{ and } v \text{ is neighbor of } u\}\!\}\right)$$

$s_u^{(1)}$ corresponds to the H-index of u

Problem: Only considers static structure of network!

Static n-th Order H-Index (Lü et al., 2016)

Let $\mathcal{M}$ be the set of finite multisets of integers.

- Function $\mathcal{H}\colon \mathcal{M} \to \mathbb{N}_0$ returns for a finite multiset of integers $S \subseteq \{\!\{ j \mid j \in \mathbb{N}_0 \}\!\}$ the maximum integer i such that there are at least i elements j in S with $j \geq i$.

n-th order H-index

n-th order H-index $s_u^{(n)}$ of a node $u \in V$ in a static graph $G = (V, E)$:
Let $s_u^{(0)} = \delta(u)$ the degree of node u, then

$$s_u^{(n)} = \mathcal{H}\left(\{\!\{ s_v^{(n-1)} \mid v \in V \text{ and } v \text{ is neighbor of } u \}\!\}\right)$$

$s_u^{(1)}$ corresponds to the H-index of u

Problem: Only considers static structure of network!

Static n-th Order H-Index (Lü et al., 2016)

Let $\mathcal{M}$ be the set of finite multisets of integers.

- Function $\mathcal{H}\colon \mathcal{M} \to \mathbb{N}_0$ returns for a finite multiset of integers $S \subseteq \{\!\{ j \mid j \in \mathbb{N}_0 \}\!\}$ the maximum integer i such that there are at least i elements j in S with $j \geq i$.

n-th order H-index

n-th order H-index $s_u^{(n)}$ of a node $u \in V$ in a static graph $G = (V, E)$:
Let $s_u^{(0)} = \delta(u)$ the degree of node u, then

$$s_u^{(n)} = \mathcal{H}\left(\{\!\{ s_v^{(n-1)} \mid v \in V \text{ and } v \text{ is neighbor of } u \}\!\}\right)$$

$s_u^{(1)}$ corresponds to the H-index of u

Problem: Only considers static structure of network!

Temporal Networks

Many networks change over time:

- Prominent examples: Social and online communication networks
- Highly dynamic networks
- Information spreads over time
- H-index should take the dynamics into account

We introduce n-**th Order Temporal H-Index:**

- **Ranking nodes:** Which nodes are important or central? Who can influence others well?
- **Core-like decomposition:** Find strongly connected subgraphs or communities?

Temporal Networks

Many networks change over time:

- Prominent examples: Social and online communication networks
- Highly dynamic networks
- Information spreads over time
- H-index should take the dynamics into account

We introduce *n*-**th Order Temporal H-Index:**

- **Ranking nodes:** Which nodes are important or central? Who can influence others well?
- **Core-like decomposition:** Find strongly connected subgraphs or communities?

Temporal Networks

Many networks change over time:

- Prominent examples: Social and online communication networks
- Highly dynamic networks
- Information spreads over time
- H-index should take the dynamics into account

We introduce *n*-**th Order Temporal H-Index:**

- **Ranking nodes:** Which nodes are important or central? Who can influence others well?
- **Core-like decomposition:** Find strongly connected subgraphs or communities?

Many networks change over time:

- Prominent examples: Social and online communication networks
- Highly dynamic networks
- Information spreads over time
- H-index should take the dynamics into account

We introduce *n*-**th Order Temporal H-Index:**

- **Ranking nodes:** Which nodes are important or central? Who can influence others well?
- **Core-like decomposition:** Find strongly connected subgraphs or communities?

Temporal Networks

Many networks change over time:

- Prominent examples: Social and online communication networks
- Highly dynamic networks
- Information spreads over time
- H-index should take the dynamics into account

We introduce *n*-**th Order Temporal H-Index:**

- **Ranking nodes:** Which nodes are important or central? Who can influence others well?
- **Core-like decomposition:** Find strongly connected subgraphs or communities?

Temporal Networks

Many networks change over time:

- Prominent examples: Social and online communication networks
- Highly dynamic networks
- Information spreads over time
- H-index should take the dynamics into account

We introduce *n*-**th Order Temporal H-Index:**

- **Ranking nodes:** Which nodes are important or central? Who can influence others well?
- **Core-like decomposition:** Find strongly connected subgraphs or communities?

Temporal Networks

Many networks change over time:

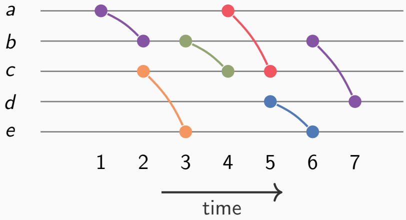
- Prominent examples: Social and online communication networks
- Highly dynamic networks
- Information spreads over time
- H-index should take the dynamics into account

We introduce *n*-**th Order Temporal H-Index:**

- **Ranking nodes:** Which nodes are important or central? Who can influence others well?
- **Core-like decomposition:** Find strongly connected subgraphs or communities?

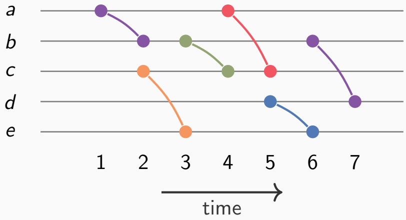
Temporal Network Model

- A **temporal network** is represented as $\mathcal{G} = (V, \mathcal{E})$
  - with (static) set of nodes V, and
  - set of temporal edges $\mathcal{E} = \{(u, v, t, \lambda)\}$, with $u, v \in V$ and $t, \lambda \in \mathbb{N}$
  - transition time λ equals time required to traverse the edge



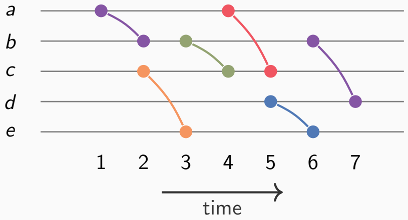
Temporal Network Model

- A **temporal network** is represented as $\mathcal{G} = (V, \mathcal{E})$
  - with (static) set of nodes V, and
  - set of temporal edges $\mathcal{E} = \{(u, v, t, \lambda)\}$, with $u, v \in V$ and $t, \lambda \in \mathbb{N}$
  - transition time λ equals time required to traverse the edge



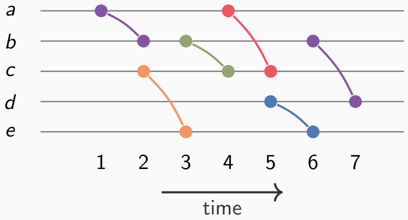
Temporal Network Model

- A **temporal network** is represented as $\mathcal{G} = (V, \mathcal{E})$
  - with (static) set of nodes V, and
  - set of temporal edges $\mathcal{E} = \{(u, v, t, \lambda)\}$, with $u, v \in V$ and $t, \lambda \in \mathbb{N}$
  - transition time λ equals time required to traverse the edge



Temporal Network Model

- A **temporal network** is represented as $\mathcal{G} = (V, \mathcal{E})$
  - with (static) set of nodes V, and
  - set of temporal edges $\mathcal{E} = \{(u, v, t, \lambda)\}$, with $u, v \in V$ and $t, \lambda \in \mathbb{N}$
  - transition time λ equals time required to traverse the edge



n-th Order Temporal H-Index

What is a useful adaption of the *n*-th order H-index for temporal networks?

- Goal is to rank according to influence and ability to spread information
- Information spreads along temporal walks
  - sequence of temporal edges, such that
  - consecutive edges share a common node, and
  - time stamps of temporal edges are increasing

n-th Order Temporal H-Index

What is a useful adaption of the *n*-th order H-index for temporal networks?

- Goal is to rank according to influence and ability to spread information
- Information spreads along temporal walks
  - sequence of temporal edges, such that
  - consecutive edges share a common node, and
  - time stamps of temporal edges are increasing

n-th Order Temporal H-Index

What is a useful adaption of the *n*-th order H-index for temporal networks?

- Goal is to rank according to influence and ability to spread information
- Information spreads along temporal walks
  - sequence of temporal edges, such that
  - consecutive edges share a common node, and
  - time stamps of temporal edges are increasing

n-th Order Temporal H-Index

What is a useful adaption of the *n*-th order H-index for temporal networks?

- Goal is to rank according to influence and ability to spread information
- Information spreads along temporal walks
  - sequence of temporal edges, such that
  - consecutive edges share a common node, and
  - time stamps of temporal edges are increasing

n-th Order Temporal H-Index

What is a useful adaption of the *n*-th order H-index for temporal networks?

- Goal is to rank according to influence and ability to spread information
- Information spreads along temporal walks
  - sequence of temporal edges, such that
  - consecutive edges share a common node, and
  - time stamps of temporal edges are increasing

n-th Order Temporal H-Index

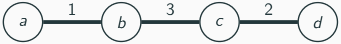
What is a useful adaption of the *n*-th order H-index for temporal networks?

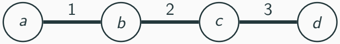
- Goal is to rank according to influence and ability to spread information
- Information spreads along temporal walks
  - sequence of temporal edges, such that
  - consecutive edges share a common node, and
  - time stamps of temporal edges are increasing

n-th Order Temporal H-Index

What is a useful adaption of the *n*-th order H-index for temporal networks?

- Goal is to rank according to influence and ability to spread information
- Information spreads along temporal walks
  - sequence of temporal edges, such that
  - consecutive edges share a common node, and
  - time stamps of temporal edges are increasing

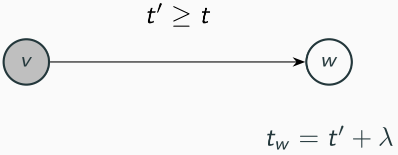




n-th Order Temporal H-Index

Temporal Neighborhood

The multiset $\mathcal{N}(v, t)$ contains all pairs of nodes and times (w, t_w) such that there is a temporal edge from v to w leaving at time $t' \geq t$ and arriving at time t_w.



n-th Order Temporal H-Index

The *n*-th order temporal H-index of a node $v \in V$ is defined as $h_v^{(n)} = h_{v,0}^{(n)}$ with

$$h_{v,t}^{(n)} = \mathcal{H}\left(\left\{\!\!\left\{ h_{w,t_w}^{(n-1)} \;\middle|\; (w, t_w) \in \mathcal{N}(v,t) \right\}\!\!\right\}\right).$$

We define $h_{v,t}^{(0)} = |\mathcal{N}(v,t)|$.

- Captures node importance in terms of temporal reachability
- Nodes with high reachability are ranked high

n-th Order Temporal H-Index

n-th Order Temporal H-Index

The n-th order temporal H-index of a node $v \in V$ is defined as $h_v^{(n)} = h_{v,0}^{(n)}$ with

$$h_{v,t}^{(n)} = \mathcal{H}\left(\left\{\!\!\left\{ h_{w,t_w}^{(n-1)} \;\middle|\; (w, t_w) \in \mathcal{N}(v,t) \right\}\!\!\right\}\right).$$

We define $h_{v,t}^{(0)} = |\mathcal{N}(v,t)|$.

- Captures node importance in terms of temporal reachability
- Nodes with high reachability are ranked high

n-th Order Temporal H-Index

n-th Order Temporal H-Index

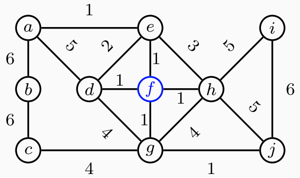
The n-th order temporal H-index of a node $v \in V$ is defined as $h_v^{(n)} = h_{v,0}^{(n)}$ with

$$h_{v,t}^{(n)} = \mathcal{H}\left(\left\{\!\!\left\{ h_{w,t_w}^{(n-1)} \;\middle|\; (w, t_w) \in \mathcal{N}(v, t) \right\}\!\!\right\}\right).$$

We define $h_{v,t}^{(0)} = |\mathcal{N}(v, t)|$.

- Captures node importance in terms of temporal reachability
- Nodes with high reachability are ranked high

Example

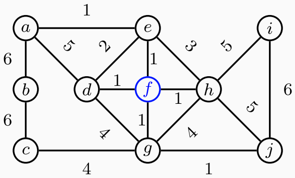


(a) $\mathcal{G}$ with $\lambda = 1$ for all edges.

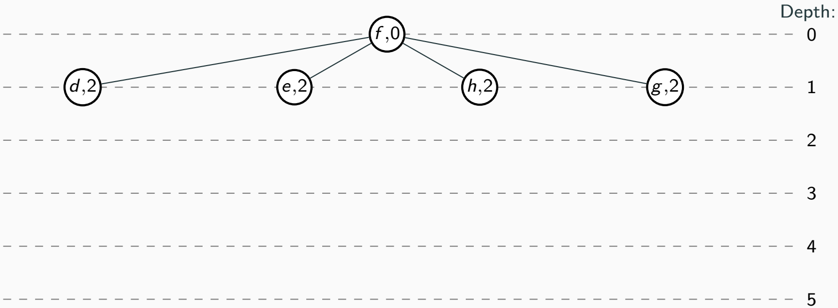
(b) The reachability tree $\Gamma(f)$ for vertex f in $\mathcal{G}$.

$$h^{(1)}_{f,0} =$$

Example



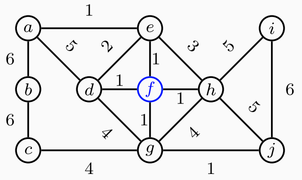
(a) $\mathcal{G}$ with $\lambda = 1$ for all edges.



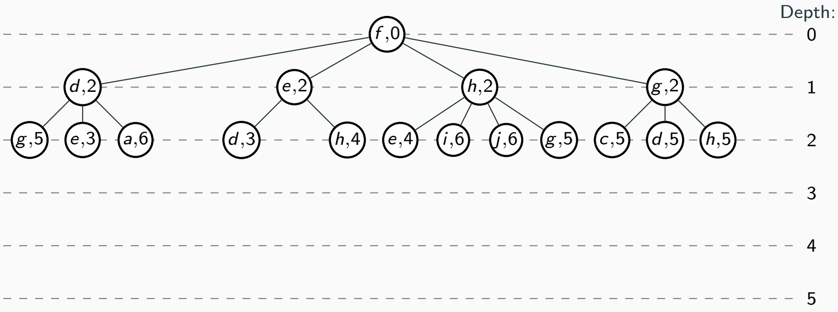
(b) The reachability tree $\Gamma(f)$ for vertex f in $\mathcal{G}$.

$$h^{(1)}_{f,0} = \mathcal{H}(\{\!\{ h^{(0)}_{d,2}, h^{(0)}_{e,2}, h^{(0)}_{h,2}, h^{(0)}_{g,2} \}\!\})$$

Example



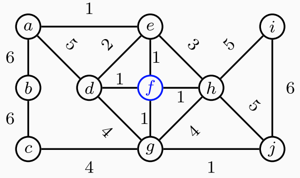
(a) $\mathcal{G}$ with $\lambda = 1$ for all edges.



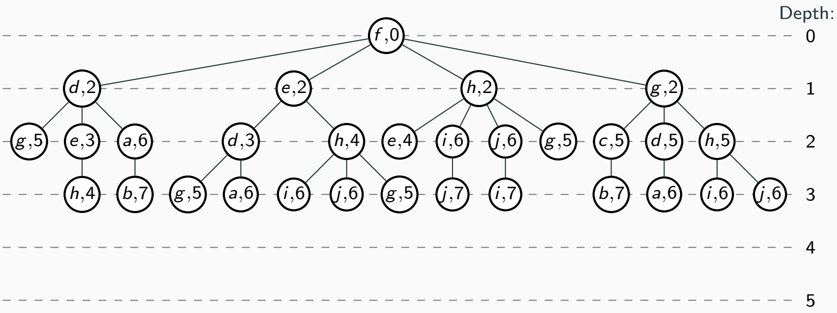
(b) The reachability tree $\Gamma(f)$ for vertex f in $\mathcal{G}$.

$$h_{f,0}^{(1)} = \mathcal{H}(\{\!\{ h_{d,2}^{(0)}, h_{e,2}^{(0)}, h_{h,2}^{(0)}, h_{g,2}^{(0)} \}\!\}) = \mathcal{H}(\{\!\{3, 2, 4, 3\}\!\}) = 3$$

Example



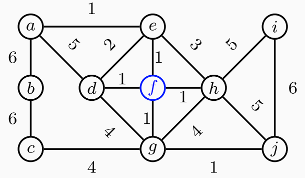
(a) $\mathcal{G}$ with $\lambda = 1$ for all edges.



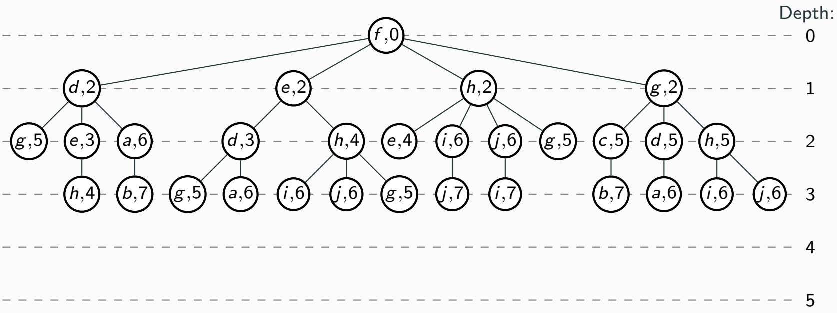
(b) The reachability tree $\Gamma(f)$ for vertex f in $\mathcal{G}$.

$$
\begin{aligned}
h^{(2)}_{f,0} &= \mathcal{H}(\{\!\{ h^{(1)}_{d,2}, h^{(1)}_{e,2}, h^{(1)}_{h,2}, h^{(1)}_{g,2} \}\!\}) \\
&= \mathcal{H}(\{\!\{ \mathcal{H}(\{\!\{ h^{(0)}_{g,5}, h^{(0)}_{e,3}, h^{(0)}_{a,6} \}\!\}), \mathcal{H}(\{\!\{ h^{(0)}_{d,3}, h^{(0)}_{h,4} \}\!\}), \mathcal{H}(\{\!\{ h^{(0)}_{e,4}, h^{(0)}_{i,6}, h^{(0)}_{j,6}, h^{(0)}_{g,5} \}\!\}), \mathcal{H}(\{\!\{ h^{(0)}_{c,5}, h^{(0)}_{d,5}, h^{(0)}_{h,5} \}\!\}) \}\!\}) \\
&= \mathcal{H}(\{\!\{ \mathcal{H}(\{\!\{0,1,1\}\!\}), \mathcal{H}(\{\!\{2,3\}\!\}), \mathcal{H}(\{\!\{0,1,1,0\}\!\}), \mathcal{H}(\{\!\{1,1,2\}\!\}) \}\!\}) = \mathcal{H}(\{\!\{1,2,1,1\}\!\}) = 1
\end{aligned}
$$

Example



(a) $\mathcal{G}$ with $\lambda = 1$ for all edges.



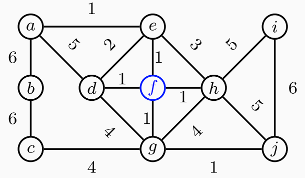
(b) The reachability tree $\Gamma(f)$ for vertex f in $\mathcal{G}$.

$$h^{(2)}_{f,0} = \mathcal{H}(\{\!\{h^{(1)}_{d,2}, h^{(1)}_{e,2}, h^{(1)}_{h,2}, h^{(1)}_{g,2}\}\!\})$$

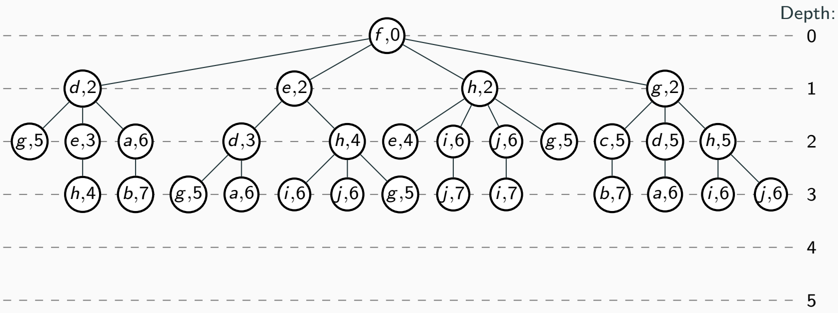
$$= \mathcal{H}(\{\!\{\mathcal{H}(\{\!\{h^{(0)}_{g,5}, h^{(0)}_{e,3}, h^{(0)}_{a,6}\}\!\}), \mathcal{H}(\{\!\{h^{(0)}_{d,3}, h^{(0)}_{h,4}\}\!\}), \mathcal{H}(\{\!\{h^{(0)}_{e,4}, h^{(0)}_{i,6}, h^{(0)}_{j,6}, h^{(0)}_{g,5}\}\!\}), \mathcal{H}(\{\!\{h^{(0)}_{c,5}, h^{(0)}_{d,5}, h^{(0)}_{h,5}\}\!\})\}\!\})$$

$$= \mathcal{H}(\{\!\{\mathcal{H}(\{\!\{0,1,1\}\!\}), \mathcal{H}(\{\!\{2,3\}\!\}), \mathcal{H}(\{\!\{0,1,1,0\}\!\}), \mathcal{H}(\{\!\{1,1,2\}\!\})\}\!\}) = \mathcal{H}(\{\!\{1,2,1,1\}\!\}) = 1$$

Example



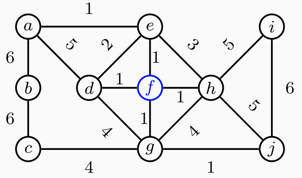
(a) $\mathcal{G}$ with $\lambda = 1$ for all edges.



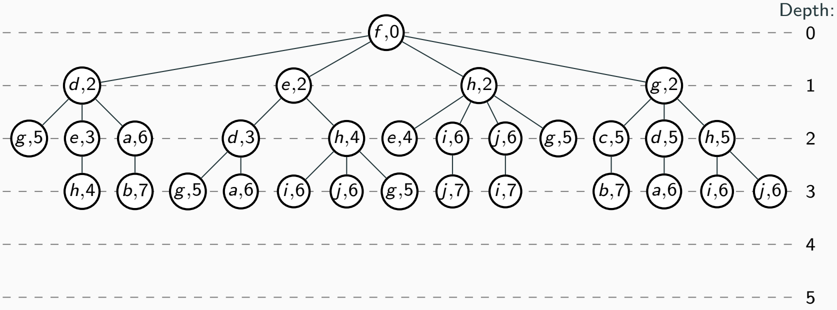
(b) The reachability tree $\Gamma(f)$ for vertex f in $\mathcal{G}$.

$$
\begin{aligned}
h^{(2)}_{f,0} &= \mathcal{H}(\{\!\{ h^{(1)}_{d,2}, h^{(1)}_{e,2}, h^{(1)}_{h,2}, h^{(1)}_{g,2} \}\!\}) \\
&= \mathcal{H}(\{\!\{ \mathcal{H}(\{\!\{ h^{(0)}_{g,5}, h^{(0)}_{e,3}, h^{(0)}_{a,6} \}\!\}), \mathcal{H}(\{\!\{ h^{(0)}_{d,3}, h^{(0)}_{h,4} \}\!\}), \mathcal{H}(\{\!\{ h^{(0)}_{e,4}, h^{(0)}_{i,6}, h^{(0)}_{j,6}, h^{(0)}_{g,5} \}\!\}), \mathcal{H}(\{\!\{ h^{(0)}_{c,5}, h^{(0)}_{d,5}, h^{(0)}_{h,5} \}\!\}) \}\!\}) \\
&= \mathcal{H}(\{\!\{ \mathcal{H}(\{\!\{0,1,1\}\!\}), \mathcal{H}(\{\!\{2,3\}\!\}), \mathcal{H}(\{\!\{0,1,1,0\}\!\}), \mathcal{H}(\{\!\{1,1,2\}\!\}) \}\!\}) = \mathcal{H}(\{\!\{1,2,1,1\}\!\}) = 1
\end{aligned}
$$

Example



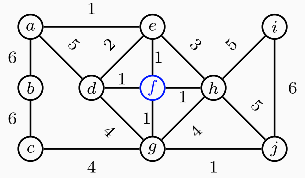
(a) $\mathcal{G}$ with $\lambda = 1$ for all edges.



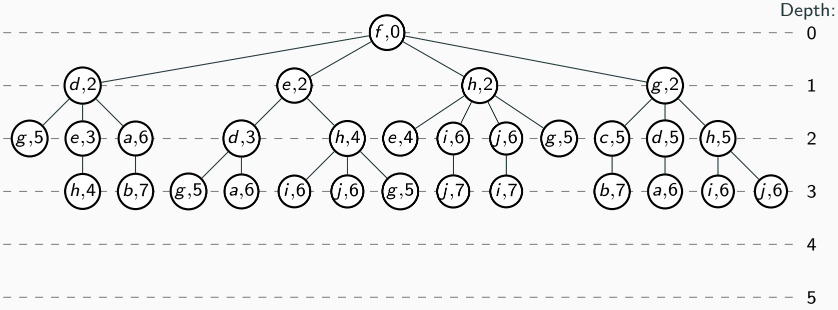
(b) The reachability tree $\Gamma(f)$ for vertex f in $\mathcal{G}$.

$$
\begin{aligned}
h_{f,0}^{(2)} &= \mathcal{H}(\{\!\{ h_{d,2}^{(1)}, h_{e,2}^{(1)}, h_{h,2}^{(1)}, h_{g,2}^{(1)} \}\!\}) \\
&= \mathcal{H}(\{\!\{ \mathcal{H}(\{\!\{ h_{g,5}^{(0)}, h_{e,3}^{(0)}, h_{a,6}^{(0)} \}\!\}), \mathcal{H}(\{\!\{ h_{d,3}^{(0)}, h_{h,4}^{(0)} \}\!\}), \mathcal{H}(\{\!\{ h_{e,4}^{(0)}, h_{i,6}^{(0)}, h_{j,6}^{(0)}, h_{g,5}^{(0)} \}\!\}), \mathcal{H}(\{\!\{ h_{c,5}^{(0)}, h_{d,5}^{(0)}, h_{h,5}^{(0)} \}\!\}) \}\!\}) \\
&= \mathcal{H}(\{\!\{ \mathcal{H}(\{\!\{0,1,1\}\!\}), \mathcal{H}(\{\!\{2,3\}\!\}), \mathcal{H}(\{\!\{0,1,1,0\}\!\}), \mathcal{H}(\{\!\{1,1,2\}\!\}) \}\!\}) = \mathcal{H}(\{\!\{1,2,1,1\}\!\}) = 1
\end{aligned}
$$

Example

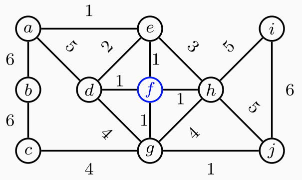


(a) $\mathcal{G}$ with $\lambda = 1$ for all edges.

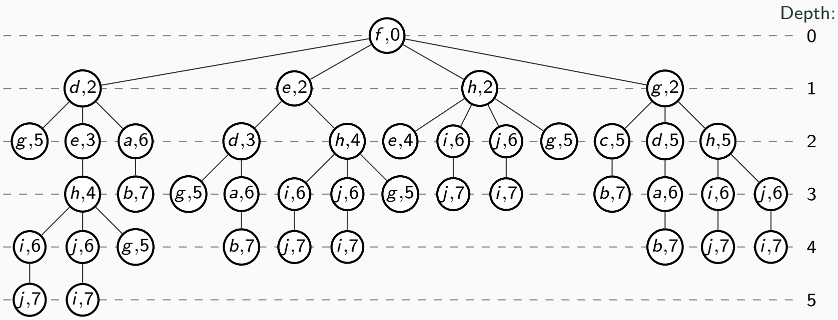


(b) The reachability tree $\Gamma(f)$ for vertex f in $\mathcal{G}$.

$$
\begin{aligned}
h^{(2)}_{f,0} &= \mathcal{H}(\{\!\{ h^{(1)}_{d,2}, h^{(1)}_{e,2}, h^{(1)}_{h,2}, h^{(1)}_{g,2} \}\!\}) \\
&= \mathcal{H}(\{\!\{ \mathcal{H}(\{\!\{ h^{(0)}_{g,5}, h^{(0)}_{e,3}, h^{(0)}_{a,6} \}\!\}), \mathcal{H}(\{\!\{ h^{(0)}_{d,3}, h^{(0)}_{h,4} \}\!\}), \mathcal{H}(\{\!\{ h^{(0)}_{e,4}, h^{(0)}_{i,6}, h^{(0)}_{j,6}, h^{(0)}_{g,5} \}\!\}), \mathcal{H}(\{\!\{ h^{(0)}_{c,5}, h^{(0)}_{d,5}, h^{(0)}_{h,5} \}\!\}) \}\!\}) \\
&= \mathcal{H}(\{\!\{ \mathcal{H}(\{\!\{0,1,1\}\!\}), \mathcal{H}(\{\!\{2,3\}\!\}), \mathcal{H}(\{\!\{0,1,1,0\}\!\}), \mathcal{H}(\{\!\{1,1,2\}\!\}) \}\!\}) = \mathcal{H}(\{\!\{1,2,1,1\}\!\}) = 1
\end{aligned}
$$



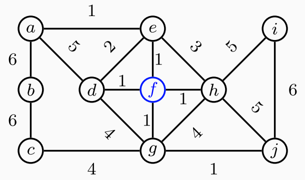
(a) $\mathcal{G}$ with $\lambda = 1$ for all edges.



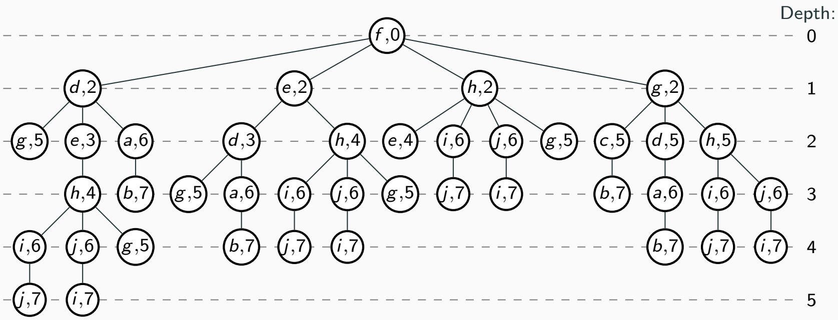
(b) The reachability tree $\Gamma(f)$ for vertex f in $\mathcal{G}$.

For $h_v^{(n)} = k > 1$, there are at least $\frac{k^{(n+2)} - k}{k-1}$ descendants u of the root r in $\Gamma(v)$ with $d(u) \leq n$.

Properties



(a) $\mathcal{G}$ with $\lambda = 1$ for all edges.



(b) The reachability tree $\Gamma(f)$ for vertex f in $\mathcal{G}$.

It holds that $h_v^{(n)} = 0$ for all $n > \Delta(\mathcal{G})$ with $\Delta(\mathcal{G})$ being the max. temporal walk length.

Overview

So far:

- Adapted n-th order H-index for temporal networks
- Rank nodes according to spreading capabilities

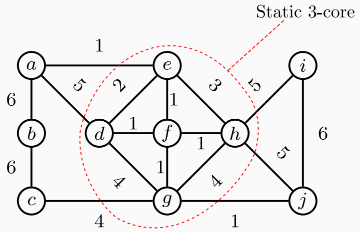
What's next?

- For increasing n, the static n-th order H-index converges to the core number of u
  - *k-core* is a max. subgraph G_k of G, s.t. every node in G_k has at least k neighbors in G_k
  - Node u has *core number* $c(u) = k$ if u belongs to a k-core but not the $k+1$-core
- **We define decomposition for the temporal variant**

Overview

So far:

- Adapted n-th order H-index for temporal networks
- Rank nodes according to spreading capabilities



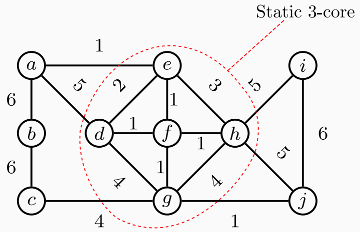
What's next?

- For increasing n, the static n-th order H-index converges to the core number of u
  - *k-core* is a max. subgraph G_k of G, s.t. every node in G_k has at least k neighbors in G_k
  - Node u has *core number* $c(u) = k$ if u belongs to a k-core but not the $k+1$-core
- **We define decomposition for the temporal variant**

Overview

So far:

- Adapted n-th order H-index for temporal networks
- Rank nodes according to spreading capabilities



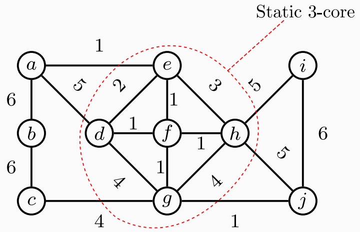
What's next?

- For increasing n, the static n-th order H-index converges to the core number of u
  - k-*core* is a max. subgraph G_k of G, s.t. every node in G_k has at least k neighbors in G_k
  - Node u has *core number* $c(u) = k$ if u belongs to a k-core but not the $k+1$-core
- **We define decomposition for the temporal variant**

Overview

So far:

- Adapted n-th order H-index for temporal networks
- Rank nodes according to spreading capabilities



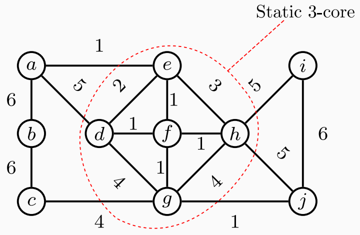
What's next?

- For increasing n, the static n-th order H-index converges to the core number of u
  - *k-core* is a max. subgraph G_k of G, s.t. every node in G_k has at least k neighbors in G_k
  - Node u has *core number* $c(u) = k$ if u belongs to a k-core but not the $k+1$-core
- We define decomposition for the temporal variant

Overview

So far:

- Adapted n-th order H-index for temporal networks
- Rank nodes according to spreading capabilities



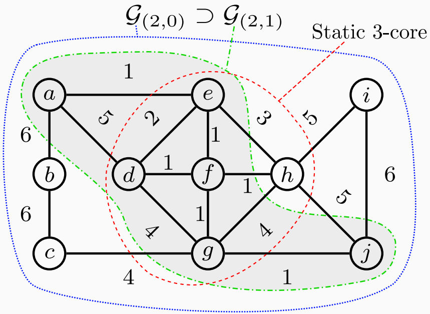
What's next?

- For increasing n, the static n-th order H-index converges to the core number of u
  - *k-core* is a max. subgraph G_k of G, s.t. every node in G_k has at least k neighbors in G_k
  - Node u has *core number* $c(u) = k$ if u belongs to a k-core but not the $k+1$-core
- **We define decomposition for the temporal variant**

Temporal (n,k)-Pseudocore

Let $k, n \in \mathbb{N}$. The temporal (n,k)-pseudocore of $\mathcal{G}$ is a maximal induced temporal subgraph $\mathcal{G}_{(n,k)}$ of $\mathcal{G}$ such that for all $v \in V(\mathcal{G}_{(n,k)})$ the n-th order temporal H-index $h_v^{(n)} \geq k$ in $\mathcal{G}$.

- (n,k)-pseudocore: Temporal subgraph containing nodes with similar temporal activity and importance in the network $\mathcal{G}$
- For a node v in a (n,k)-pseudocore $\mathcal{G}_{(n,k)}$, the inequality $h_v^{(n)} \geq k$ does not hold necessarily with respect to $\mathcal{G}_{(n,k)}$

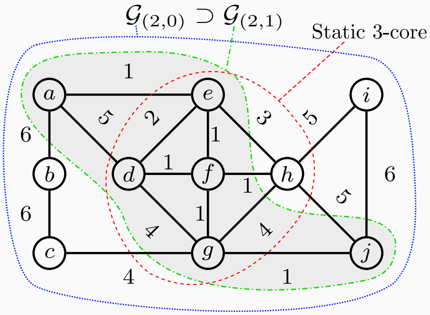


Core-like Decomposition

Temporal (n,k)-Pseudocore

Let $k, n \in \mathbb{N}$. The temporal (n,k)-pseudocore of $\mathcal{G}$ is a maximal induced temporal subgraph $\mathcal{G}_{(n,k)}$ of $\mathcal{G}$ such that for all $v \in V(\mathcal{G}_{(n,k)})$ the n-th order temporal H-index $h_v^{(n)} \geq k$ in $\mathcal{G}$.

- (n,k)-pseudocore: Temporal subgraph containing nodes with similar temporal activity and importance in the network $\mathcal{G}$
- For a node v in a (n,k)-pseudocore $\mathcal{G}_{(n,k)}$, the inequality $h_v^{(n)} \geq k$ does not hold necessarily with respect to $\mathcal{G}_{(n,k)}$

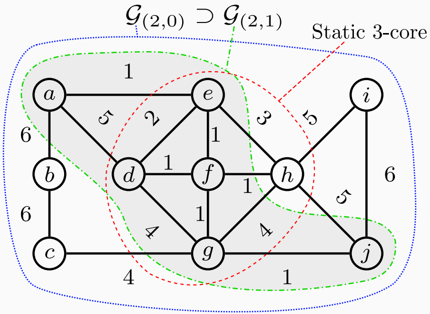


Core-like Decomposition

Temporal (n, k)-Pseudocore

Let $k, n \in \mathbb{N}$. The temporal (n, k)-pseudocore of $\mathcal{G}$ is a maximal induced temporal subgraph $\mathcal{G}_{(n,k)}$ of $\mathcal{G}$ such that for all $v \in V(\mathcal{G}_{(n,k)})$ the n-th order temporal H-index $h_v^{(n)} \geq k$ in $\mathcal{G}$.

- (n, k)-pseudocore: Temporal subgraph containing nodes with similar temporal activity and importance in the network $\mathcal{G}$
- For a node v in a (n, k)-pseudocore $\mathcal{G}_{(n,k)}$, the inequality $h_v^{(n)} \geq k$ does not hold necessarily with respect to $\mathcal{G}_{(n,k)}$



Computation

Two algorithms

- Naive recursive algorithm
- Streaming algorithm
  - Single pass over edges in reverse chronological order
  - Computes for each node i-th order H-indices for $0 \leq i \leq n$

Algorithm	Running Time	Space	Edge Trans. Times	Results for $\forall i \in [n]$
RECURS	$\mathcal{O}(\lvert V \rvert n(\delta_{\max})^2)$	$\mathcal{O}(\lvert V \rvert n\delta_{\max})$	individual	✗
STREAM	$\mathcal{O}(\lvert \mathcal{E} \rvert n\delta_{\max})$	$\mathcal{O}(\lvert V \rvert n\delta_{\max})$	uniform	✓

Computation

Two algorithms

- Naive recursive algorithm
- Streaming algorithm
  - Single pass over edges in reverse chronological order
  - Computes for each node i-th order H-indices for $0 \leq i \leq n$

Algorithm	Running Time	Space	Edge Trans. Times	Results for $\forall i \in [n]$
RECURS	$\mathcal{O}(\lvert V \rvert n(\delta_{\max})^2)$	$\mathcal{O}(\lvert V \rvert n\delta_{\max})$	individual	✗
STREAM	$\mathcal{O}(\lvert \mathcal{E} \rvert n\delta_{\max})$	$\mathcal{O}(\lvert V \rvert n\delta_{\max})$	uniform	✓

Computation

Two algorithms

- Naive recursive algorithm
- Streaming algorithm
  - Single pass over edges in reverse chronological order
  - Computes for each node i-th order H-indices for $0 \leq i \leq n$

Algorithm	Running Time	Space	Edge Trans. Times	Results for $\forall i \in [n]$
RECURS	$\mathcal{O}(\lvert V \rvert n(\delta_{\max})^2)$	$\mathcal{O}(\lvert V \rvert n\delta_{\max})$	individual	✗
STREAM	$\mathcal{O}(\lvert \mathcal{E} \rvert n\delta_{\max})$	$\mathcal{O}(\lvert V \rvert n\delta_{\max})$	uniform	✓

Streaming Algorithm

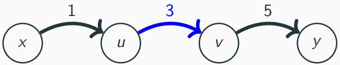
- Streaming algorithm of the temporal edges in reverse chronological order of time steps
- Manages for each $v \in V$ and $1 \leq i \leq n$ a multiset of i-th order H-indices of the neighbors
- When edge $(u, v, t) \in \mathcal{E}$ is processed
  1. Update degree $|\mathcal{N}(u, t)|$
  2. Append at u the $(i+1)$-th order H-index of the multisets of i-th order H-indices of v
- After processing all edges, return the i-th order H-indices for each $v \in V$ and $1 \leq i \leq n$

Streaming Algorithm

- Streaming algorithm of the temporal edges in reverse chronological order of time steps
- Manages for each $v \in V$ and $1 \leq i \leq n$ a multiset of i-th order H-indices of the neighbors
- When edge $(u, v, t) \in \mathcal{E}$ is processed
  1. Update degree $|\mathcal{N}(u, t)|$
  2. Append at u the $(i+1)$-th order H-index of the multisets of i-th order H-indices of v
- After processing all edges, return the i-th order H-indices for each $v \in V$ and $1 \leq i \leq n$

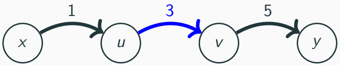
Streaming Algorithm

- Streaming algorithm of the temporal edges in reverse chronological order of time steps
- Manages for each $v \in V$ and $1 \leq i \leq n$ a multiset of i-th order H-indices of the neighbors
- When edge $(u, v, t) \in \mathcal{E}$ is processed
    1. Update degree $|\mathcal{N}(u, t)|$
    2. Append at u the $(i+1)$-th order H-index of the multisets of i-th order H-indices of v
- After processing all edges, return the i-th order H-indices for each $v \in V$ and $1 \leq i \leq n$



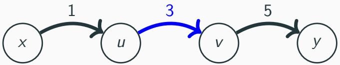
Streaming Algorithm

- Streaming algorithm of the temporal edges in reverse chronological order of time steps
- Manages for each $v \in V$ and $1 \leq i \leq n$ a multiset of i-th order H-indices of the neighbors
- When edge $(u, v, t) \in \mathcal{E}$ is processed
    1. Update degree $|\mathcal{N}(u, t)|$
    2. Append at u the $(i+1)$-th order H-index of the multisets of i-th order H-indices of v
- After processing all edges, return the i-th order H-indices for each $v \in V$ and $1 \leq i \leq n$



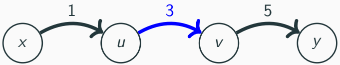
Streaming Algorithm

- Streaming algorithm of the temporal edges in reverse chronological order of time steps
- Manages for each $v \in V$ and $1 \leq i \leq n$ a multiset of i-th order H-indices of the neighbors
- When edge $(u, v, t) \in \mathcal{E}$ is processed
  1. Update degree $|\mathcal{N}(u, t)|$
  2. Append at u the $(i + 1)$-th order H-index of the multisets of i-th order H-indices of v
- After processing all edges, return the i-th order H-indices for each $v \in V$ and $1 \leq i \leq n$



Streaming Algorithm

- Streaming algorithm of the temporal edges in reverse chronological order of time steps
- Manages for each $v \in V$ and $1 \leq i \leq n$ a multiset of i-th order H-indices of the neighbors
- When edge $(u, v, t) \in \mathcal{E}$ is processed
    1. Update degree $|\mathcal{N}(u, t)|$
    2. Append at u the $(i + 1)$-th order H-index of the multisets of i-th order H-indices of v
- After processing all edges, return the i-th order H-indices for each $v \in V$ and $1 \leq i \leq n$



1. Running Times

Running times in seconds (s). OOT: out of time (time limit 12h).

Data set	Graph size		$n = 8$		$n = 16$		$n = 32$		$n = 64$	
	$\lvert V\rvert$	$\lvert\mathcal{E}\rvert$	Recurs	Stream	Recurs	Stream	Recurs	Stream	Recurs	Stream
FacebookMsg	1 899	59 798	2.56	0.08	5.52	0.15	11.94	0.31	26.70	0.64
Infectious	10 972	415 912	18.81	1.02	39.87	1.98	76.73	4.19	144.48	8.51
FacebookWall	63 731	817 035	31.11	3.48	69.01	6.03	135.03	11.31	310.44	22.49
Enron	86 806	1 133 968	411.59	4.89	866.53	11.03	1882.11	24.45	4226.25	52.33
AskUbuntu	134 035	257 305	1.23	0.21	2.50	0.38	5.31	0.72	13.15	1.44
Digg	279 630	1 731 652	62.80	3.33	120.30	6.84	229.93	13.83	364.88	27.61
Wikipedia	1 870 709	39 953 145	4863.44	117.03	10332.81	230.70	21998.44	457.65	OOT	861.95
Flickr	2 302 925	33 140 016	870.88	168.92	1767.10	332.29	3323.15	640.81	5373.19	1282.84

1. Running Times

Running times in seconds (s). OOT: out of time (time limit 12h).

Data set	Graph size		$n = 8$		$n = 16$		$n = 32$		$n = 64$	
	$\lvert V \rvert$	$\lvert \mathcal{E} \rvert$	Recurs	Stream	Recurs	Stream	Recurs	Stream	Recurs	Stream
FacebookMsg	1 899	59 798	2.56	0.08	5.52	0.15	11.94	0.31	26.70	0.64
Infectious	10 972	415 912	18.81	1.02	39.87	1.98	76.73	4.19	144.48	8.51
FacebookWall	63 731	817 035	31.11	3.48	69.01	6.03	135.03	11.31	310.44	22.49
Enron	86 806	1 133 968	411.59	4.89	866.53	11.03	1882.11	24.45	4226.25	52.33
AskUbuntu	134 035	257 305	1.23	0.21	2.50	0.38	5.31	0.72	13.15	1.44
Digg	279 630	1 731 652	62.80	3.33	120.30	6.84	229.93	13.83	364.88	27.61
Wikipedia	1 870 709	39 953 145	4863.44	117.03	10332.81	230.70	21998.44	457.65	OOT	861.95
Flickr	2 302 925	33 140 016	870.88	168.92	1767.10	332.29	3323.15	640.81	5373.19	1282.84

1. Running Times

Running times in seconds (s). OOT: out of time (time limit 12h).

Data set	Graph size		$n = 8$		$n = 16$		$n = 32$		$n = 64$	
	$\|V\|$	$\|\mathcal{E}\|$	RECURS	STREAM	RECURS	STREAM	RECURS	STREAM	RECURS	STREAM
FacebookMsg	1 899	59 798	2.56	0.08	5.52	0.15	11.94	0.31	26.70	0.64
Infectious	10 972	415 912	18.81	1.02	39.87	1.98	76.73	4.19	144.48	8.51
FacebookWall	63 731	817 035	31.11	3.48	69.01	6.03	135.03	11.31	310.44	22.49
Enron	86 806	1 133 968	411.59	4.89	866.53	11.03	1882.11	24.45	4226.25	52.33
AskUbuntu	134 035	257 305	1.23	0.21	2.50	0.38	5.31	0.72	13.15	1.44
Digg	279 630	1 731 652	62.80	3.33	120.30	6.84	229.93	13.83	364.88	27.61
Wikipedia	1 870 709	39 953 145	4863.44	117.03	10332.81	230.70	21998.44	457.65	OOT	861.95
Flickr	2 302 925	33 140 016	870.88	168.92	1767.10	332.29	3323.15	640.81	5373.19	1282.84

Experiments

2. Comparison of reachability scores

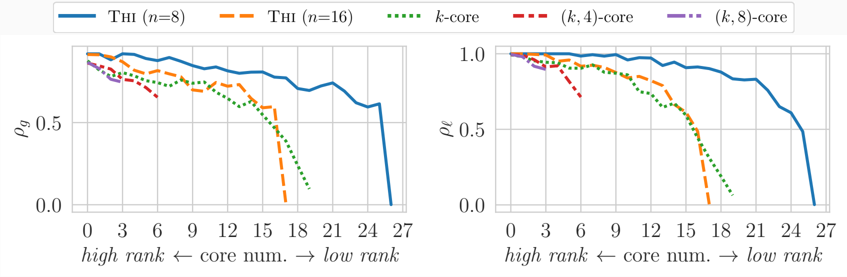
Let $r\colon V \times V \to \{0,1\}$ the indicator function for temporal reachability, i.e., $r(u,v) = 1$ iff u can reach v via a temporal walk. For pseudocore $\mathcal{G}_{(n,k)} = (V', \mathcal{E}')$:

- global reachability score: $\rho_g = \frac{\sum_{u \in V', v \in V} r(u,v)}{|V'| \cdot |V|}$
- local score: $\rho_\ell = \frac{\sum_{u,v \in V'} r(u,v)}{|V'|^2}$

2. Comparison of reachability scores

Let $r\colon V \times V \to \{0,1\}$ the indicator function for temporal reachability, i.e., $r(u,v) = 1$ iff u can reach v via a temporal walk. For pseudocore $\mathcal{G}_{(n,k)} = (V', \mathcal{E}')$:

- global reachability score: $\rho_g = \frac{\sum_{u \in V', v \in V} r(u,v)}{|V'| \cdot |V|}$
- local score: $\rho_\ell = \frac{\sum_{u,v \in V'} r(u,v)}{|V'|^2}$



(a) *FacebookMsg – global* **(b)** *FacebookMsg – local*

3. Use Case: Influential Spreader Identification

- We computed for different infection probabilities β the mean node influence R_u over 1000 independent SIR simulations leading to the SIR node rankings
- We compare the SIR rankings with those obtained by the centrality measures using the Kendall τ_b rank correlation measure

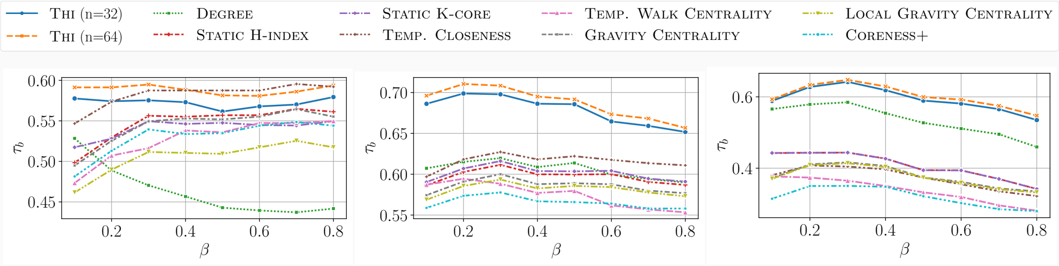
3. Use Case: Influential Spreader Identification

- We computed for different infection probabilities β the mean node influence R_u over 1000 independent SIR simulations leading to the SIR node rankings
- We compare the SIR rankings with those obtained by the centrality measures using the Kendall τ_b rank correlation measure

Experiments

3. Use Case: Influential Spreader Identification

- We computed for different infection probabilities β the mean node influence R_u over 1000 independent SIR simulations leading to the SIR node rankings
- We compare the SIR rankings with those obtained by the centrality measures using the Kendall τ_b rank correlation measure



(a) *Malawi* (b) *FacebookMsg* (c) *Email*

Conclusion

- We introduced the **n-th order temporal H-index**
- Obtained inward and outward variants based on incoming and outgoing temporal walks
- Highly scalable streaming algorithm
- Effective pseudocore decomposition
- We showed that the n-th order temporal H-index can be a successful heuristic for identifying possible super-spreaders

Link to our paper

Conclusion

- We introduced the ***n*-th order temporal H-index**
- Obtained inward and outward variants based on incoming and outgoing temporal walks
- Highly scalable streaming algorithm
- Effective pseudocore decomposition
- We showed that the *n*-th order temporal H-index can be a successful heuristic for identifying possible super-spreaders

Link to our paper

Conclusion

- We introduced the ***n*-th order temporal H-index**
- Obtained inward and outward variants based on incoming and outgoing temporal walks
- Highly scalable streaming algorithm
- Effective pseudocore decomposition
- We showed that the *n*-th order temporal H-index can be a successful heuristic for identifying possible super-spreaders

Link to our paper

Conclusion

- We introduced the ***n*-th order temporal H-index**
- Obtained inward and outward variants based on incoming and outgoing temporal walks
- Highly scalable streaming algorithm
- Effective pseudocore decomposition
- We showed that the *n*-th order temporal H-index can be a successful heuristic for identifying possible super-spreaders

Link to our paper

Conclusion

- We introduced the ***n*-th order temporal H-index**
- Obtained inward and outward variants based on incoming and outgoing temporal walks
- Highly scalable streaming algorithm
- Effective pseudocore decomposition
- We showed that the *n*-th order temporal H-index can be a successful heuristic for identifying possible super-spreaders

Link to our paper