A Comparitive Analysis of Source Identification Algorithms

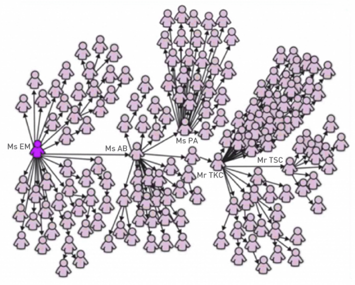
Pablo A. Curiel, Richard C. Tillquist

CSCSU

March 2023

Introduction

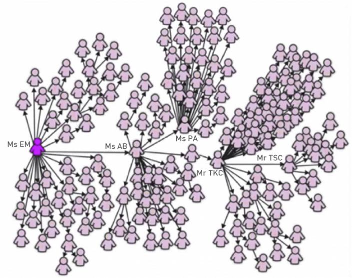
- Networks capture important features necessary for spread



Graphic from *Network Science*

Introduction

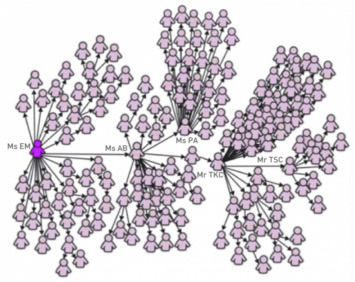
- Networks capture important features necessary for spread
- Applications of modeling spread using networks
  - Diseases in a physical-contact network
  - Electricity in a power grid network
  - Malware attacks in a computer-system network



Graphic from *Network Science*

Introduction

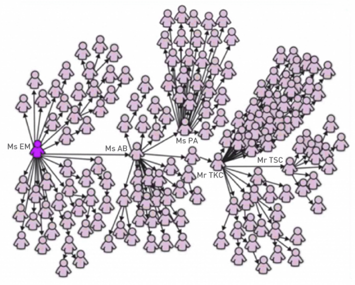
- Networks capture important features necessary for spread
- Applications of modeling spread using networks
  - Diseases in a physical-contact network
  - Electricity in a power grid network
  - Malware attacks in a computer-system network
- Often important to identify source of a spread
  - Contact tracing
  - Optimizing the allocation of immunization resources
  - Mitigating spread of rumors or misinformation
  - Defending against malware attacks



Graphic from *Network Science*

Introduction

- Networks capture important features necessary for spread
- Applications of modeling spread using networks
  - Diseases in a physical-contact network
  - Electricity in a power grid network
  - Malware attacks in a computer-system network
- Often important to identify source of a spread
  - Contact tracing
  - Optimizing the allocation of immunization resources
  - Mitigating spread of rumors or misinformation
  - Defending against malware attacks
- Focus: comparing four state-of-the-art methods of identifying the source
  - Rumor centrality (2011)
  - Jordan centrality (2017)
  - NETSLEUTH (2014)
  - LISN (2019)



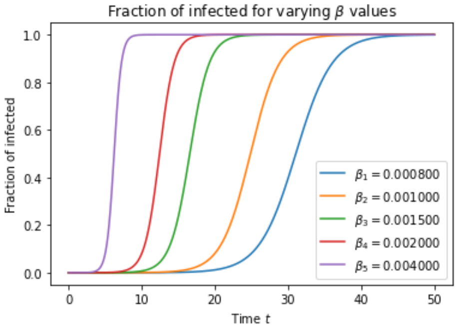
Graphic from *Network Science*

- Susceptible-Infected (SI) model
  - Relies on two assumptions
    1. Compartmentalization
    2. Homogeneous mixing
  - Homogeneous mixing disregards the true contact network

$$\underbrace{\frac{dS}{dt}}_{\textit{Susceptible}} = \underbrace{-\beta SI}_{\text{Susceptible become infected}}$$

$$\underbrace{\frac{dI}{dt}}_{\textit{Infected}} = \underbrace{\beta SI}_{\text{Susceptible become infected}}$$

$\Longrightarrow$



Modeling Spread on Networks

- Networks take into account each individual's direct contacts

Graphic from Medium "Social Network Analytics"

Modeling Spread on Networks

- Networks take into account each individual's direct contacts
- SI-model on a network:
  - Use discrete time-steps and compartmentalization
  - Single-source node chosen at time $t = 0$
  - Infected nodes can infect susceptible neighbors with probability β

Graphic from Medium "Social Network Analytics"

Modeling Spread on Networks

- Networks take into account each individual's direct contacts
- SI-model on a network:
  - Use discrete time-steps and compartmentalization
  - Single-source node chosen at time $t = 0$
  - Infected nodes can infect susceptible neighbors with probability β
- Given a connected network G and $\beta > 0$, $|V_I| \to |V|$ as $t \to \infty$
  - β is the infection rate
  - V_I is the set of infected nodes
  - V is the set of all nodes

Graphic from Medium "Social Network Analytics"

Modeling Spread on Networks

- Networks take into account each individual's direct contacts
- SI-model on a network:
  - Use discrete time-steps and compartmentalization
  - Single-source node chosen at time $t = 0$
  - Infected nodes can infect susceptible neighbors with probability β
- Given a connected network G and $\beta > 0$, $|V_I| \to |V|$ as $t \to \infty$
  - β is the infection rate
  - V_I is the set of infected nodes
  - V is the set of all nodes
- Transmission rate (or speed of the spread) $\beta\langle k\rangle$
  - $\langle k\rangle$ is the average degree of the network

Graphic from Medium "Social Network Analytics"

Modeling Spread on Networks

- Networks take into account each individual's direct contacts
- SI-model on a network:
  - Use discrete time-steps and compartmentalization
  - Single-source node chosen at time $t = 0$
  - Infected nodes can infect susceptible neighbors with probability β
- Given a connected network G and $\beta > 0$, $|V_I| \to |V|$ as $t \to \infty$
  - β is the infection rate
  - V_I is the set of infected nodes
  - V is the set of all nodes
- Transmission rate (or speed of the spread) $\beta\langle k\rangle$
  - $\langle k\rangle$ is the average degree of the network
- Transmission rate is analogous to basic reproductive rate R_0
  - $R_0 = \frac{\beta\langle k\rangle}{\mu}$, where μ is a recovery/removal rate
  - $R_0 > 1 \implies$ spread, $R_0 < 1 \implies$ spread dies out

Graphic from Medium "Social Network Analytics"

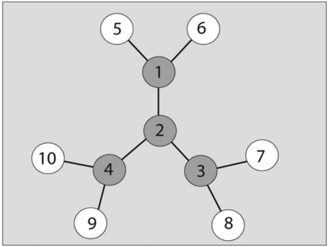
Rumor Centrality (2011)

- First paper to analytically study source identification
- Rumor centrality $R(v, G_I)$ is the number of permitted permutations of a graph G_I centered at some node v
  - Permitted permutation describes a possible ordering of nodes that led to G_I

Graphic from Shah and Zaman (2011)

Rumor Centrality (2011)

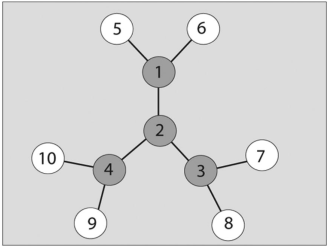
- First paper to analytically study source identification
- Rumor centrality $R(v, G_I)$ is the number of permitted permutations of a graph G_I centered at some node v
  - Permitted permutation describes a possible ordering of nodes that led to G_I
- Assuming source node 1:
  - Permitted permutation: $\{1, 2, 3, 4\}$
  - Unpermitted permutation: $\{1, 3, 2, 4\}$



Graphic from Shah and Zaman (2011)

Rumor Centrality (2011)

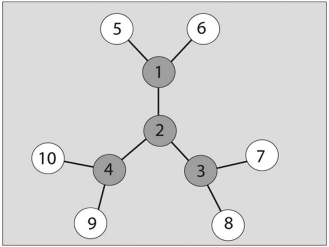
- First paper to analytically study source identification
- Rumor centrality $R(v, G_I)$ is the number of permitted permutations of a graph G_I centered at some node v
  - Permitted permutation describes a possible ordering of nodes that led to G_I
- Assuming source node 1:
  - Permitted permutation: $\{1, 2, 3, 4\}$
  - Unpermitted permutation: $\{1, 3, 2, 4\}$
- Source is estimated to be v^* such that $R(v^*, G_I) \geq R(v, G_I)\ \forall v \in G_I$



Graphic from Shah and Zaman (2011)

Rumor Centrality (2011)

- First paper to analytically study source identification
- Rumor centrality $R(v, G_I)$ is the number of permitted permutations of a graph G_I centered at some node v
  - Permitted permutation describes a possible ordering of nodes that led to G_I
- Assuming source node 1:
  - Permitted permutation: $\{1, 2, 3, 4\}$
  - Unpermitted permutation: $\{1, 3, 2, 4\}$
- Source is estimated to be v^* such that $R(v^*, G_I) \geq R(v, G_I)\ \forall v \in G_I$
- Method is designed for tree-structured data
- For general graphs, method uses a BFS-tree rooted at each node
- Complexity: $O(|V_I|)$ for tree graphs, $O(|V_I|^2)$ for general graphs
  - V_I is the set of infected nodes



Graphic from Shah and Zaman (2011)

Jordan Centrality (2017)

- Measures a node's eccentricity in the graph
  - Eccentricity: max hop-distance to all other nodes
  - $J(v, I) = \max_{u \in I} d(v, u)$, where $d(v, u)$ is the hop-distance between nodes v, u

Graphic from Shah and Zaman (2011)

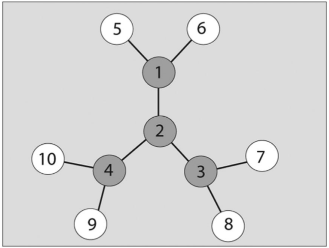
Jordan Centrality (2017)

- Measures a node's eccentricity in the graph
  - Eccentricity: max hop-distance to all other nodes
  - $J(v, I) = \max_{u \in I} d(v, u)$, where $d(v, u)$ is the hop-distance between nodes v, u
- Jordan infection center: infected node with minimum Jordan centrality
  - Estimated to be source node
  - For a Jordan infection center v^*, $v^* = \arg\min_{v \in I} J(v, I)$

Graphic from Shah and Zaman (2011)

Jordan Centrality (2017)

- Measures a node's eccentricity in the graph
  - Eccentricity: max hop-distance to all other nodes
  - $J(v, I) = \max_{u \in I} d(v, u)$, where $d(v, u)$ is the hop-distance between nodes v, u
- Jordan infection center: infected node with minimum Jordan centrality
  - Estimated to be source node
  - For a Jordan infection center v^*, $v^* = \arg\min_{v \in I} J(v, I)$
- Calculating Jordan centralities:
  - $J(1, I) = J(3, I) = J(4, I) = 2$, $J(2, I) = 1$
  - Node 2 would be estimated to be source
- Complexity: $O(|V_I||E_I|)$, where E_I is the set of edges in the infected subgraph



Graphic from Shah and Zaman (2011)

NETSLEUTH (2014)

- Makes use of minimum description length principle
    - Referred to as minimal infection description

NETSLEUTH (2014)

- Makes use of minimum description length principle
  - Referred to as minimal infection description
- Goal is to minimize total description length $\mathcal{L}(G_I, S, R)$
  - G_I is the infected subgraph
  - S is the set of seed nodes (possible sources)
  - R is a valid spread propagation ripple

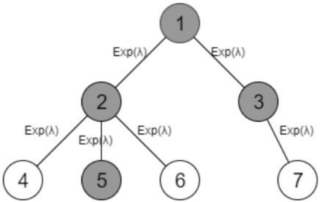
NETSLEUTH (2014)

- Makes use of minimum description length principle
  - Referred to as minimal infection description
- Goal is to minimize total description length $\mathcal{L}(G_I, S, R)$
  - G_I is the infected subgraph
  - S is the set of seed nodes (possible sources)
  - R is a valid spread propagation ripple
- $\mathcal{L}(G_I, S, R) = \mathcal{L}(S) + \mathcal{L}(R|S)$
  - $\mathcal{L}(S)$ is the encoded length of the seed set S
  - $\mathcal{L}(R|S)$ is the encoded length of a ripple R starting at a seed set S

NETSLEUTH (2014)

- Makes use of minimum description length principle
  - Referred to as minimal infection description
- Goal is to minimize total description length $\mathcal{L}(G_I, S, R)$
  - G_I is the infected subgraph
  - S is the set of seed nodes (possible sources)
  - R is a valid spread propagation ripple
- $\mathcal{L}(G_I, S, R) = \mathcal{L}(S) + \mathcal{L}(R|S)$
  - $\mathcal{L}(S)$ is the encoded length of the seed set S
  - $\mathcal{L}(R|S)$ is the encoded length of a ripple R starting at a seed set S
- Able to estimate multiple sources
- Complexity: $O(|E_I| + |E_F| + |V_I|)$
  - E_F is the set of edges connecting susceptible nodes to infected nodes

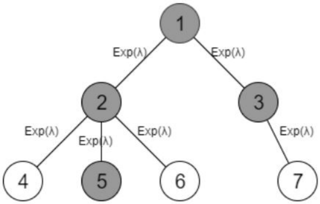
- Assumes time for spread to propagate is exponentially distributed



Graphic from Nie and Quinn (2019)

LISN (2019)

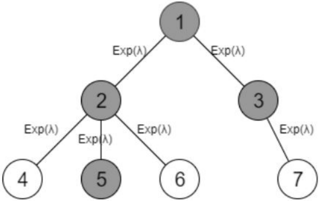
- Assumes time for spread to propagate is exponentially distributed
- Method sums exponential random variables
  - Results in a gamma distrubtion



Graphic from Nie and Quinn (2019)

LISN (2019)

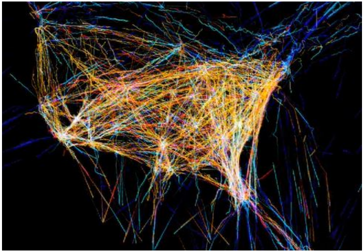
- Assumes time for spread to propagate is exponentially distributed
- Method sums exponential random variables
  - Results in a gamma distrubtion
- Cumulative distribution function (CDF) used to update probabilities
- Node with max probability is estimated to be source
- Complexity: $O(|V_I|(|V| + |E|))$



Graphic from Nie and Quinn (2019)

Methods

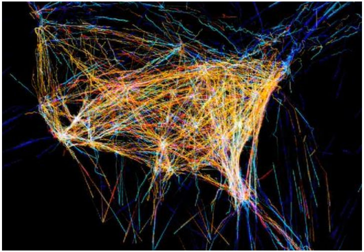
- Used cosasi Python package to test the four methods
  - Simulates spreads on NetworkX graph
  - Implements various source identification algorithms



Graphic from *Network Science*

Methods

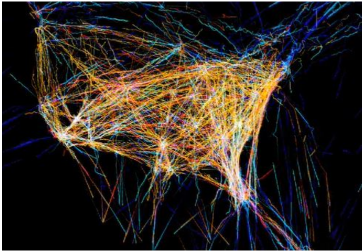
- Used cosasi Python package to test the four methods
  - Simulates spreads on NetworkX graph
  - Implements various source identification algorithms
- Simulated spread for 100 time-steps
  - Applied source algorithms for $t \in \{10, 20, 30\}$



Graphic from *Network Science*

Methods

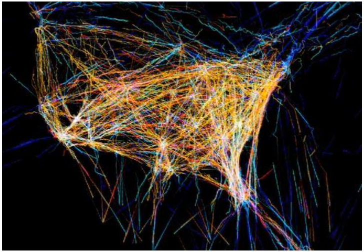
- Used cosasi Python package to test the four methods
  - Simulates spreads on NetworkX graph
  - Implements various source identification algorithms
- Simulated spread for 100 time-steps
  - Applied source algorithms for $t \in \{10, 20, 30\}$
- Graph types:
  - Barabási-Albert random graphs ($m = 1$ and $m = 3$)
  - Erdős-Rényi random graphs ($p = \frac{\ln(N)+1}{N}$)
  - Watts-Strogatz random graphs ($p = 0.01, k = 4$)
  - Real-world U.S. airport network ($N = 1,572$, $|E| = 17,214$)



Graphic from *Network Science*

Methods

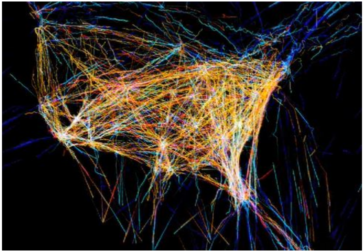
- Used cosasi Python package to test the four methods
  - Simulates spreads on NetworkX graph
  - Implements various source identification algorithms
- Simulated spread for 100 time-steps
  - Applied source algorithms for $t \in \{10, 20, 30\}$
- Graph types:
  - Barabási-Albert random graphs ($m = 1$ and $m = 3$)
  - Erdős-Rényi random graphs ($p = \frac{\ln(N)+1}{N}$)
  - Watts-Strogatz random graphs ($p = 0.01, k = 4$)
  - Real-world U.S. airport network ($N = 1,572$, $|E| = 17,214$)
- Graph sizes: $N \in \{100, 250, 500\}$
- Infection rates: $\beta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$



Graphic from *Network Science*

Methods

- Used cosasi Python package to test the four methods
  - Simulates spreads on NetworkX graph
  - Implements various source identification algorithms
- Simulated spread for 100 time-steps
  - Applied source algorithms for $t \in \{10, 20, 30\}$
- Graph types:
  - Barabási-Albert random graphs ($m = 1$ and $m = 3$)
  - Erdős-Rényi random graphs ($p = \frac{\ln(N)+1}{N}$)
  - Watts-Strogatz random graphs ($p = 0.01, k = 4$)
  - Real-world U.S. airport network ($N = 1,572$, $|E| = 17,214$)
- Graph sizes: $N \in \{100, 250, 500\}$
- Infection rates: $\beta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$
- $6,000$ experiments for random graphs
  - $1,500$ for each graph type
  - 100 for each set of: graph type, graph size, β-value
- 100 experiments for airport network (20 per β)



Graphic from *Network Science*

Results and Discussion

- Metrics:
  - Estimated source's distance from true source (number of hops)
  - Average distance from estimated source to true source
  - Rank of the true source
  - Average of the true source's rank

- Metrics:
  - Estimated source's distance from true source (number of hops)
  - Average distance from estimated source to true source
  - Rank of the true source
  - Average of the true source's rank
- Most important factors:
  - Observation time t
  - Infection rate β
  - Graph structure $G(V, E)$

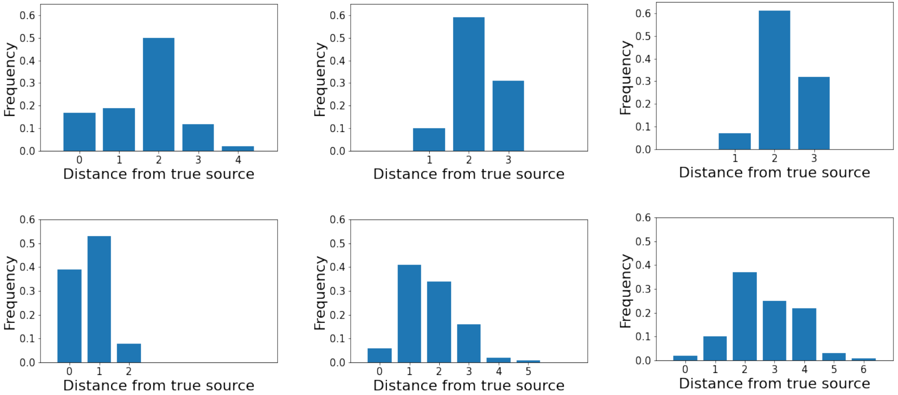
Results and Discussion

- Metrics:
  - Estimated source's distance from true source (number of hops)
  - Average distance from estimated source to true source
  - Rank of the true source
  - Average of the true source's rank
- Most important factors:
  - Observation time t
  - Infection rate β
  - Graph structure $G(V, E)$
- Substantial improvements on Barabási-Albert trees ($m = 1$)
- Watts-Strogatz random graphs performed exceptionally well

Results and Discussion

- Metrics:
  - Estimated source's distance from true source (number of hops)
  - Average distance from estimated source to true source
  - Rank of the true source
  - Average of the true source's rank
- Most important factors:
  - Observation time t
  - Infection rate β
  - Graph structure $G(V, E)$
- Substantial improvements on Barabási-Albert trees ($m = 1$)
- Watts-Strogatz random graphs performed exceptionally well
- Minor changes when lowering only infection rate on:
  - Barabási-Albert random graphs ($m = 3$)
  - Erdős-Rényi random graphs
  - U.S. airport network
- Results obtained reflect those in literature

Results and Discussion (cont.)



- Top figures: LISN, Barabási-Albert ($m = 3$), $N = 500$, $\beta = 0.1$, $t \in \{10, 20, 30\}$
- Bottom figures: Rumor centrality, Barabási-Albert ($m = 1$), $N = 500$, $t = 10$, $\beta \in \{0.1, 0.3, 0.5\}$

Results and Discussion (cont.)

Graph Type	Method	a.) $\beta = 0.5, t = 30$		b.) $\beta = 0.1, t = 10$	
		$\frac{\langle R \rangle}{N}$	$\frac{\langle d \rangle}{\langle diam(G) \rangle}$	$\frac{\langle R \rangle}{N}$	$\frac{\langle d \rangle}{\langle diam(G) \rangle}$
Barabási-Albert ($m = 1$)	Rumor	0.50	0.25	**0.02**	0.06
	Jordan	0.51	0.25	**0.02**	0.10
	NETSLEUTH	**0.48**	0.29	**0.02**	0.13
	LISN	0.49	0.25	**0.02**	0.10
Barabási-Albert ($m = 3$)	Rumor	0.50	0.36	0.16	0.36
	Jordan	0.51	0.36	0.19	0.36
	NETSLEUTH	0.51	0.36	**0.15**	0.36
	LISN	**0.48**	0.36	0.17	0.36
Erdős-Rényi ($p = \frac{\ln(N)+1}{N}$)	Rumor	0.55	0.48	**0.06**	0.32
	Jordan	0.50	0.40	0.08	0.32
	NETSLEUTH	**0.45**	0.48	0.14	0.40
	LISN	0.53	0.40	0.09	0.32
Watts-Strogatz ($p = 0.01$, $k = 4$)	Rumor	0.12	0.15	**0.01**	0.02
	Jordan	**0.04**	0.12	**0.01**	0.02
	NETSLEUTH	0.26	0.19	0.02	0.02
	LISN	0.11	0.15	**0.01**	0.02
US Airport Network	Rumor	**0.43**	0.25	0.32	0.25
	Jordan	0.44	0.25	0.36	0.25
	NETSLEUTH	0.47	0.25	0.47	0.25
	LISN	0.46	0.25	**0.28**	0.13

Future Work and Conclusion

- Future work
  - Experimenting on more source identification algorithms
  - Applying algorithms to more network structures
  - Extracting metrics in a statistical manner
  - More variation in parameters
  - Applying deep learning methods

Future Work and Conclusion

- Future work
  - Experimenting on more source identification algorithms
  - Applying algorithms to more network structures
  - Extracting metrics in a statistical manner
  - More variation in parameters
  - Applying deep learning methods
- Over $6,000$ experiments across several:
  - Graph types
  - Graph sizes
  - Infection rates
  - Observation times

Future Work and Conclusion

- Future work
  - Experimenting on more source identification algorithms
  - Applying algorithms to more network structures
  - Extracting metrics in a statistical manner
  - More variation in parameters
  - Applying deep learning methods
- Over $6,000$ experiments across several:
  - Graph types
  - Graph sizes
  - Infection rates
  - Observation times
- Algorithm performance shown to depend on:
  - Observation time
  - Infection rate
  - Graph structure
  - Graph size

Acknowledgements

- Thank you to the organizers of CSCSU

References

- Barabási A, Pósfai M. Network Science. Cambridge University Press, 2016. ISBN 9781107076266. URL https://books.google.com/books?id=iLtGDQAAQBAJ
- Kunegis J. Konect: The Koblenz Network Collection. In Proceedings of the 22nd International Conference on World Wide Web. 2013; 1343–1350. URL http://konect.cc/networks/opsahl-usairport/
- Murray JD. Mathematical Biology: I. An Introduction. Springer, 2002.
- Shah D, Zaman T. Rumors in a network: Who's the culprit? IEEE Transactions on Information Theory 2011;57(8):5163–5181.
- Ying L, Zhu K. Diffusion source localization in large networks. Synthesis Lectures on Communication Networks 2018;11(1):1–95.
- Luo W, Tay WP, Leng M. On the universality of jordan centers for estimating infection sources in tree networks. IEEE Transactions on Information Theory 2017;63(7):4634–4657.
- Prakash BA, Vreeken J, Faloutsos C. Efficiently spotting the starting points of an epidemic in a large graph. Knowledge and Information Systems 2014;38(1):35–59.
- Nie G, Quinn C. Localizing the information source in a network. In TrueFact 2019: KDD 2019 Workshop on Truth Discovery and Fact Checking: Theory and Practice. 2019.
- McCabe LH. cosasi: Graph diffusion source inference in python. Journal of Open Source Software 2022;7(80):4894.
- Hagberg A, Swart P, S Chult D. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab (LANL), Los Alamos, NM (United States), 2008.
- Barabási A, Albert R. Emergence of scaling in random networks. science 1999;286(5439):509–512.
- Erdős P, Rényi A. On random graphs i. Publicationes mathematicae 1959;6(1):290–297.
- Choi J. Epidemic source detection over dynamic networks. Electronics 2020;9(6). ISSN 2079-9292. URL https://www.mdpi.com/2079-9292/9/6/1018
- Shah C, Dehmamy N, Perra N, Chinazzi M, Barabási A, Vespignani A, Yu R. Finding patient zero: Learning contagion source with graph neural networks. CoRR 2020;abs/2006.11913. URL https://arxiv.org/abs/2006.11913

Appendix: LISN

- $F(t; n, \beta) = \frac{\gamma(n, \beta t)}{\Gamma(n)}$ is CDF of the gamma distribution
  - $\Gamma(n)$ is the standard gamma function
  - $\gamma(n, \beta t)$ is the lower-incomplete gamma function
  - n is the shortest distance between the two nodes
  - t is the observation time
  - β is the infection rate

Algorithm 1: A source detection algorithm

```
Input: G (network graph), I (infected nodes), T (total
       propagation time)
Output: rumor source estimate
initialization;
p ← {};
source ← v ∈ I;
forall v ∈ I do
    p(v) ← 1;
    forall u ∈ G.nodes do
        n ← ShortestPath(v, u);
        if u ∈ I then
            p(v) ← p(v) * F(T; n, λ);
            /* F is the cdf of gamma distribution */
        else
            p(v) ← p(v) * (1 − F(T; n, λ));
        end
    end
end
return source ← arg max_{v∈I} P(v)
```

Appendix: Full table

Graph Type	Method	a.) $\beta = 0.5, t = 30$		b.) $\beta = 0.1, t = 30$		c.) $\beta = 0.1, t = 10$	
		$\frac{\langle R \rangle}{N}$	$\frac{\langle d \rangle}{\langle diam(G) \rangle}$	$\frac{\langle R \rangle}{N}$	$\frac{\langle d \rangle}{\langle diam(G) \rangle}$	$\frac{\langle R \rangle}{N}$	$\frac{\langle d \rangle}{\langle diam(G) \rangle}$
Barabási-Albert ($m = 1$)	Rumor	0.50	0.25	0.09	0.13	**0.02**	0.06
	Jordan	0.51	0.25	0.08	0.14	**0.02**	0.10
	NETSLEUTH	**0.48**	0.29	0.10	0.16	**0.02**	0.13
	LISN	0.49	0.25	**0.07**	0.16	**0.02**	0.10
Barabási-Albert ($m = 3$)	Rumor	0.50	0.36	0.52	0.36	0.16	0.36
	Jordan	0.51	0.36	0.54	0.36	0.19	0.36
	NETSLEUTH	0.51	0.36	**0.44**	0.36	**0.15**	0.36
	LISN	**0.48**	0.36	0.54	0.36	0.17	0.36
Erdős-Rényi ($p = \frac{\ln(N)+1}{N}$)	Rumor	0.55	0.48	0.50	0.37	**0.06**	0.32
	Jordan	0.50	0.40	0.50	0.32	0.08	0.322
	NETSLEUTH	**0.45**	0.48	**0.48**	0.40	0.14	0.40
	LISN	0.53	0.40	**0.48**	0.40	0.09	0.32
Watts-Strogatz ($p = 0.01$, $k = 4$)	Rumor	0.12	0.15	**0.02**	0.05	**0.01**	0.02
	Jordan	**0.04**	0.12	**0.02**	0.05	**0.01**	0.02
	NETSLEUTH	0.26	0.19	0.04	0.07	0.02	0.02
	LISN	0.11	0.15	**0.02**	0.03	**0.01**	0.02
US Airport Network	Rumor	**0.43**	0.25	0.40	0.25	0.32	0.25
	Jordan	0.44	0.25	**0.39**	0.25	0.36	0.25
	NETSLEUTH	0.47	0.25	0.47	0.25	0.47	0.25
	LISN	0.46	0.25	**0.39**	0.25	**0.28**	0.13