Statistical Estimation of Diffusion Network Topologies

Keqi Han[†] Yuan Tian[‡] Yunjia Zhang[§] Ling Han[†] Hao Huang[†] Yunjun Gao[#]

[†]*School of Computer Science, Wuhan University, China*

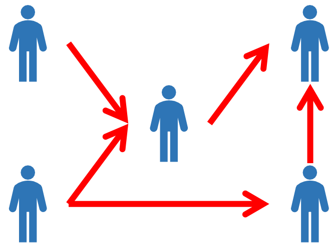
[‡]*School of Mathematics and Statistics, Wuhan University, China*

[§]*Department of Computer Sciences, University of Wisconsin-Madison, USA*

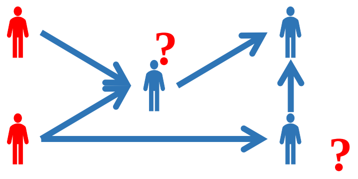
[#]*College of Computer Science and Technology, Zhejiang University, China*

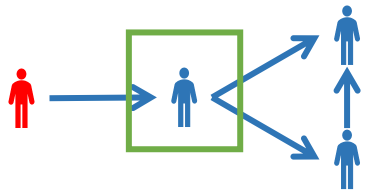
Background

- What is diffusion network?
  - Diffusion network is a directed graph that represent the diffusion relation between nodes (usually people and users)



- What can diffusion network represent?
  - Epidemic spread-out network (like COVID-19)
  - Social network
- How can diffusion network be used?
  - Prediction of number of cases
  - Precise quarantine

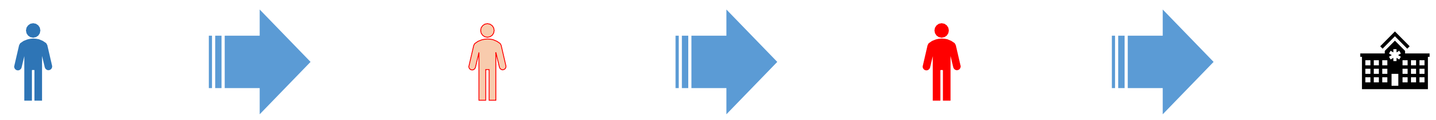




- Diffusion network reconstruction aims at recovering these influence relations based on diffusion results observed from historical diffusion processes.

Motivation

- Traditional methods rely on accurate timestamps
  - Assumption: shorter infection time intervals indicate more likely two nodes are connected
- Accurate infection timestamps: hard to get and sometimes misleading
  - Monitoring real-world diffusion processes so that obtain temporal information is often expensive and is not always feasible
  - The observed timestamps do not usually reflect the exact occurrence time of each infection



Healthy

Incubation at time t^*

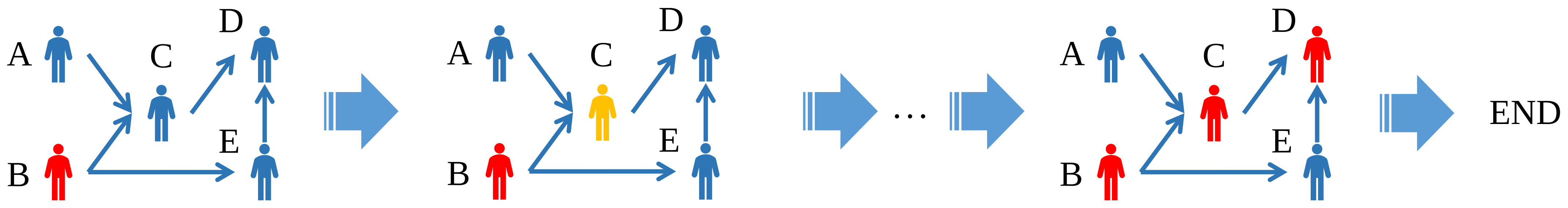
Has symptoms at time t_1

Goes to hospital and gets recorded at time t_2

Vary from person to person

Problem Statement

- Diffusion process:



Start with some initially infected nodes

Try to infect children with probability for only once

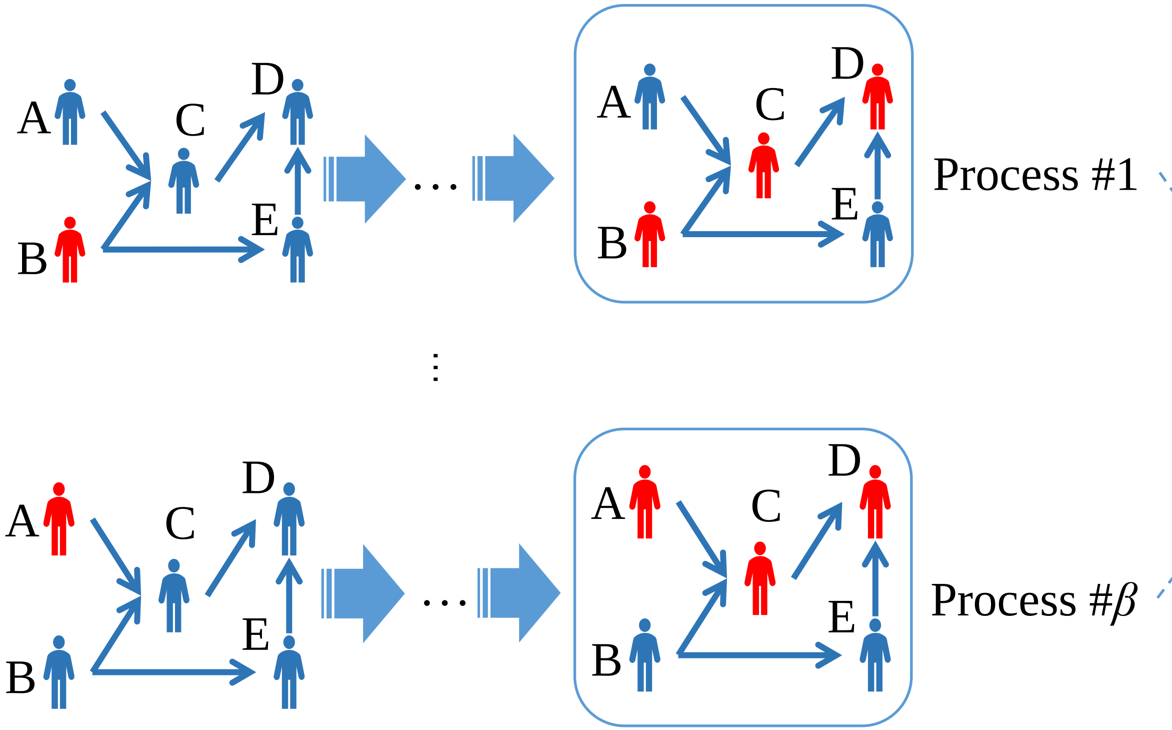
Until there are no newly infected nodes

- Assumptions:
  - All diffusion processes are independent to each other
  - All diffusion processes are following the same network topology

Problem Statement

- *Given* diffusion records:

  We are given : a set $S = \{S^1, \ldots, S^\beta\}$ of infection *status (0/1)* results observed on a diffusion network G in β diffusion processes



Infection Status Records	*A*	*B*	*C*	*D*	*E*
Process S^1	0	1	1	1	0
⋮	⋮	⋮	⋮	⋮	⋮
Process S^β	1	0	1	1	0

Problem Statement

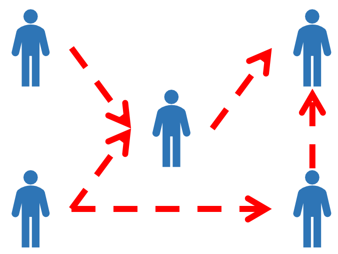
- *Given* diffusion records:

  We are given : a set $S = \{S^1, ..., S^\beta\}$ of infection *status (0/1)* results observed on a diffusion network G in β diffusion processes

- *Infer* network topology:

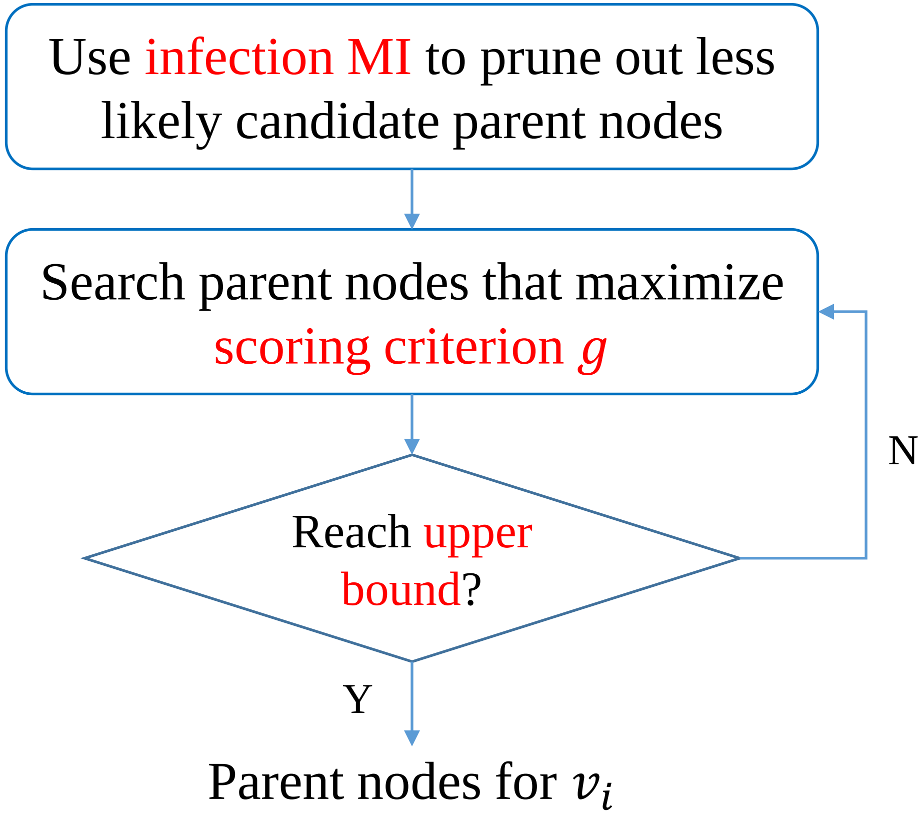
  Edge set E of the diffusion network G (the parent node set F_i of each node v_i)

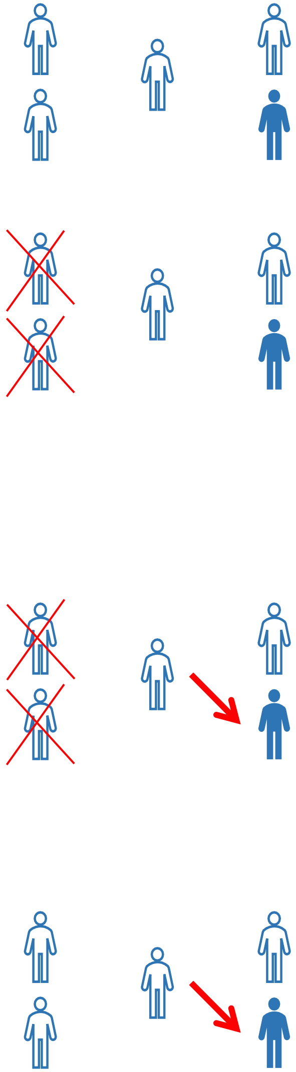
Infection Status Records	*A*	*B*	*C*	*D*	*E*
Process #1	0	1	1	1	0
⋮	⋮	⋮	⋮	⋮	⋮
Process #β	1	0	1	1	0



TENDS Algorithm: Overview

For each node v_i in the graph

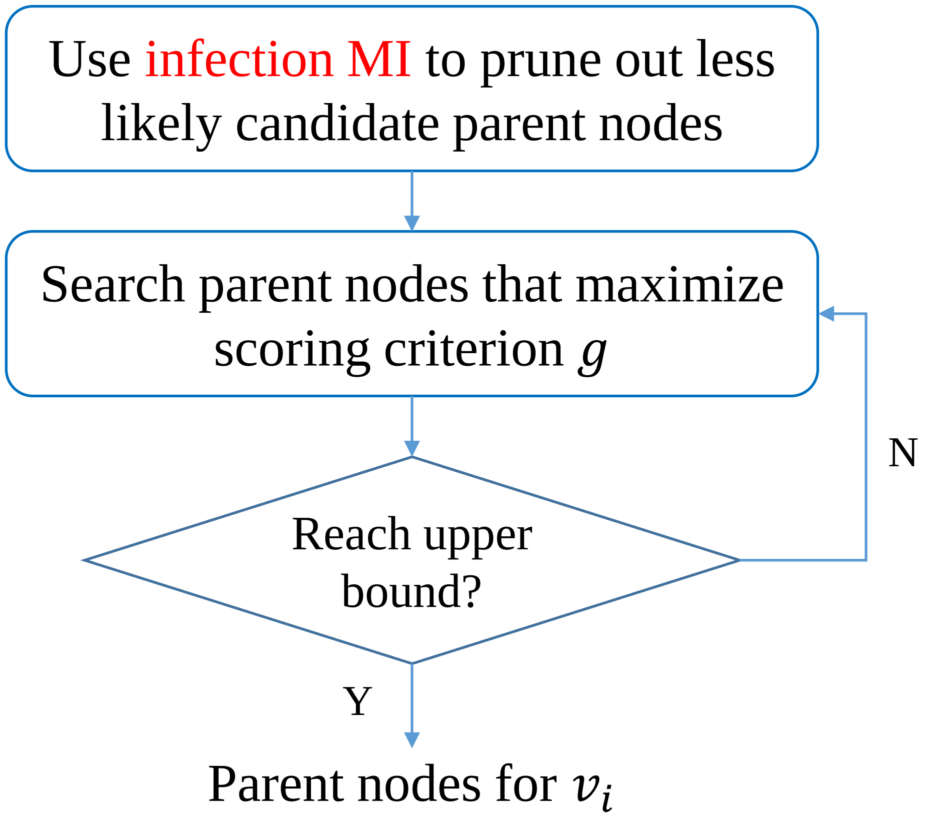




- **Pruning candidate parent nodes**:
  calculate infection *MI* value for each node pair and performs K-means to select candidate parent nodes
- **Greedy search for the parent node set F_i of each node v_i**:
  calculate corresponding scores, and then continuously expand the parent node set F_i with the highest scored parent node sets until reaching the upper bound or no candidate parent node left.

TENDS Algorithm: Details

For each node v_i in the graph



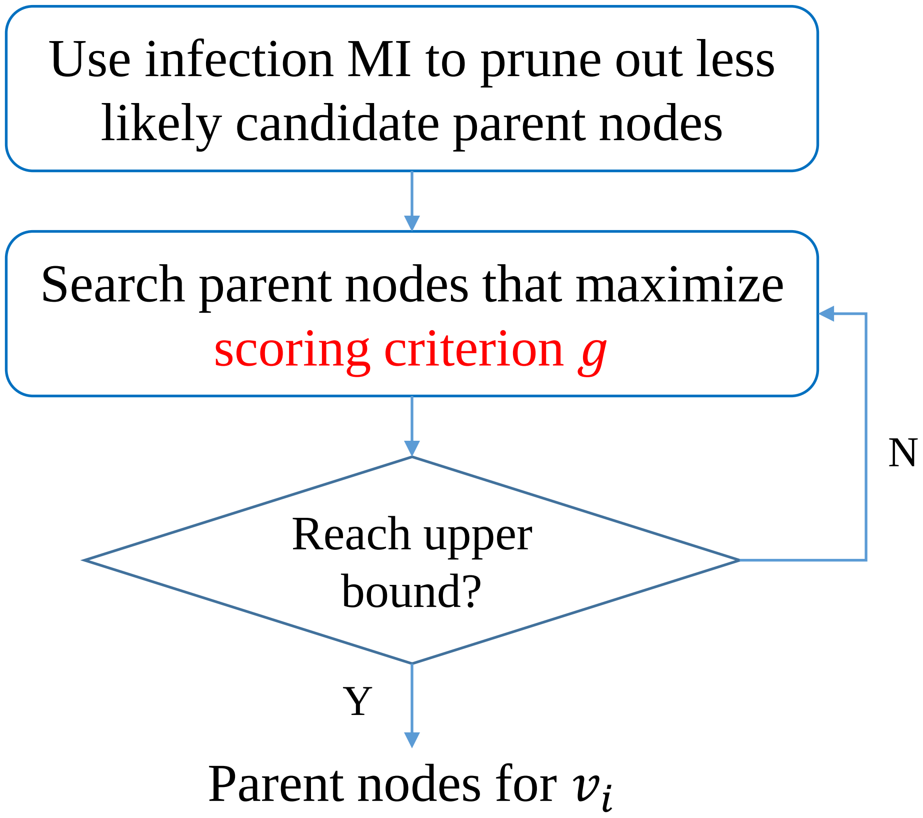
(1) Squeeze search space

- We screen out the insignificant candidate parent nodes whose infections have low correlations
- We modify the original MI metric as a new version called infection MI to better measure the positive correlation:

$$\boxed{\begin{aligned} IMI(X_i, X_j) = {} & MI(X_i = 1, X_j = 1) + MI(X_i = 0, X_j = 0) \\ & - |MI(X_i = 1, X_j = 0)| - |MI(X_i = 0, X_j = 1)| \end{aligned}}$$

TENDS Algorithm: Details

For each node v_i in the graph



(2) Scoring criterion

- We then use a scoring criterion to select the parent node set for the node;
- The scoring criterion:

$$\boxed{g(v_i, F_i) = \log L(v_i, F_i) - \lambda \, \mathrm{pen}\left(F_i\right)}$$

- Likelihood:

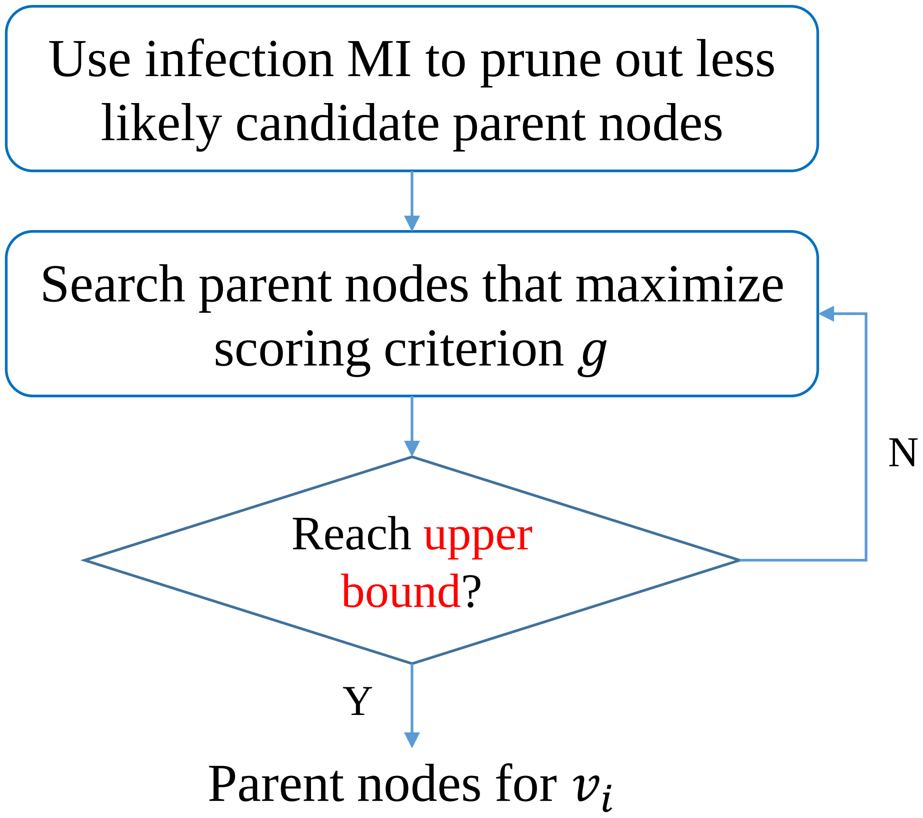
$$\log L(v_i, F_i) = \sum_{j=1}^{2^{|F_i|}} \sum_{k=1}^{2} N_{ijk} \log\left(\frac{N_{ijk}}{N_{ij}}\right)$$

- Penalty term:

$$\lambda \, \mathrm{pen}\left(F_i\right) = \frac{1}{2} \sum_{j=1}^{2^{|F_i|}} \log(N_{ij} + 1)$$

TENDS Algorithm: Details

For each node v_i in the graph



(3) Upper bound on number of parent nodes

- From naïve constraints on the scoring criterion:

$$g(v_i, F_i) \geq g(v_i, \varnothing)$$

we can derive an upper bound on the number of parent nodes for each node

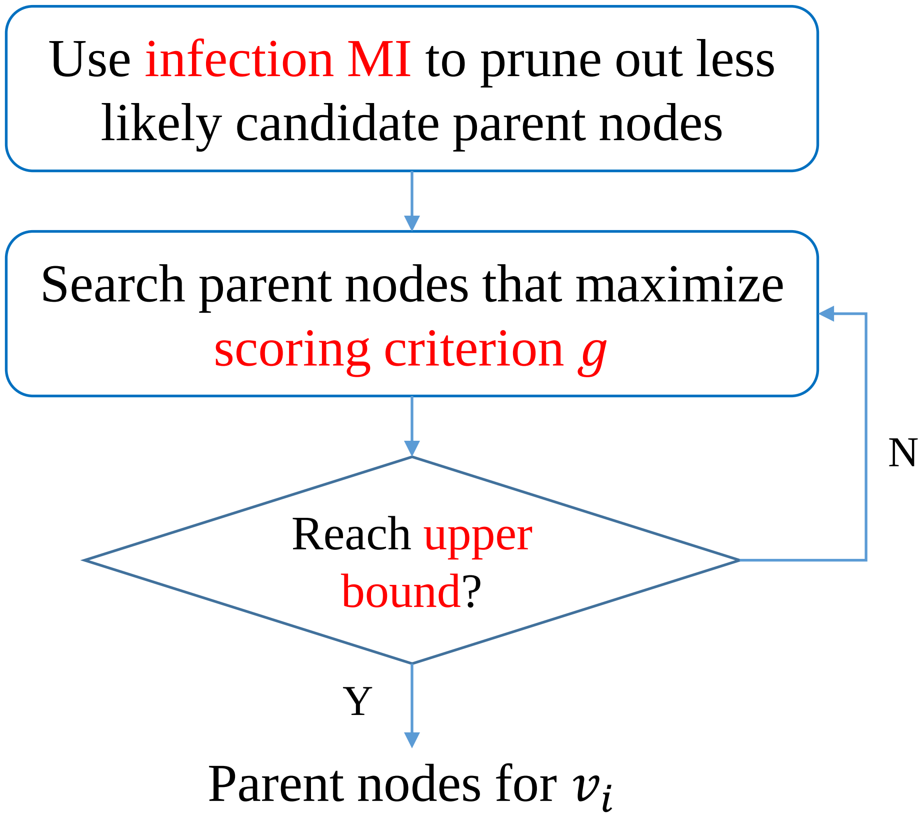
$$\boxed{|F_i| \leq \log(\phi_{F_i} + \delta_i)}$$

where

$$\delta_i = 2N_1 \log\frac{\beta}{N_1} + 2N_2 \log\frac{\beta}{N_2} + \log(\beta + 1)$$

TENDS Algorithm: Analysis

For each node v_i in the graph



- Complexity:
  - For infection mutual information: quadratic
  - For subgraph structure scan: ~linear
  - Total: quadratic
- Insights and key idea:
  - We are keep finding local optimal structures: find optimal parent nodes
  - Infection mutual information is very useful to roughly measure the infection relations
  - To find the directed edges, we use asymmetric likelihood as scoring criterion

Experiment Settings

- **Network**: Three series of LFR benchmark graphs are generated as synthetic networks. In addition, we adopt two real-world networks: NetSci and DUNF.
  - LFR: we use the directed version
  - NetSci and DUNF: we convert the undirected edges into pairs of directed edges

- **Infection Data**: The infection status results S can be obtained by *simulating β times of diffusion processes on each network* with randomly selected initial infected nodes.
  - For those baselines using temporal data, we input the original timestamps
  - For those baselines using non-temporal data, we convert the timestamps to status record

Experiment Settings

- **Performance Criterion**: *F*-score of inferred directed edges is used to evaluate the accuracy performance of algorithms.

- **Benchmark Algorithms**:

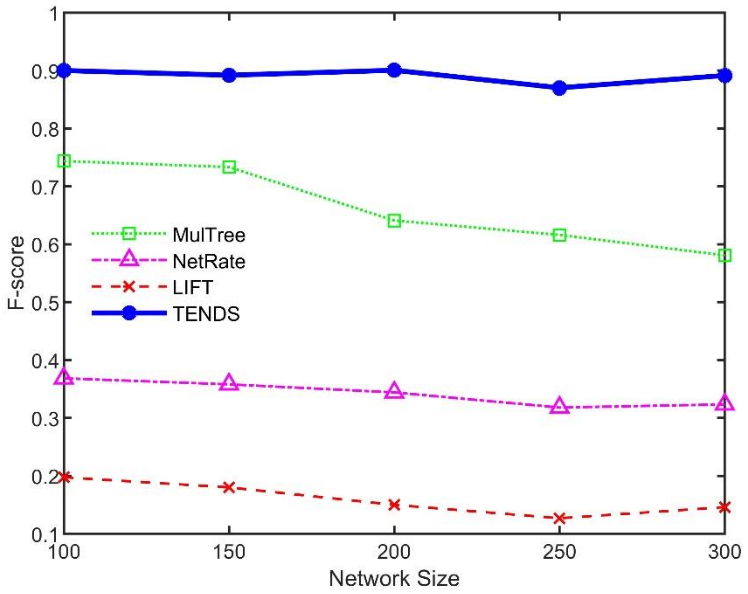
  (1) sub modularity-based approach ***MulTree***: consider all propagation tree supported by diffusion processes

  (2) convex programming-based approach ***NetRate***: convex optimization method to find optimal topology method and infers the edge weights as well.

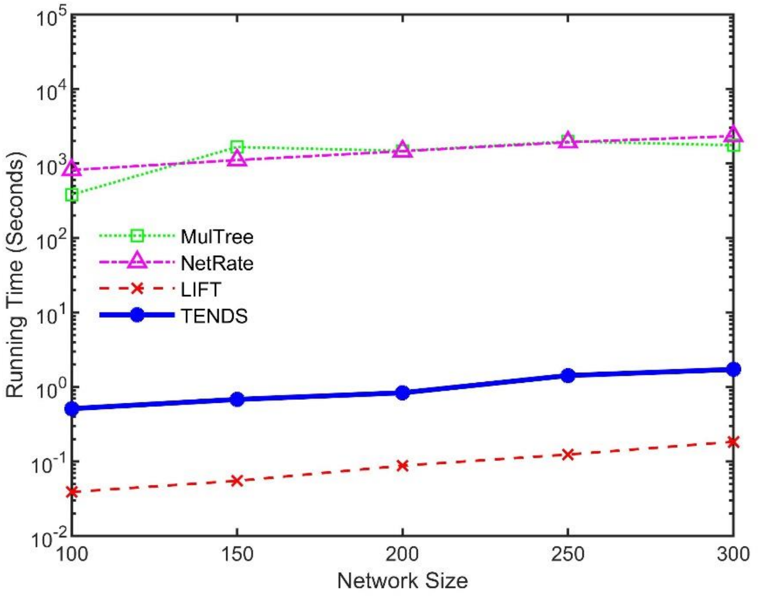
  (3) infection timestamp-free approach ***LIFT***: a non-temporal method but requires diffusion sources

Experimental Evaluation

- **Effect of Diffusion Network Size**: we adopt five synthetic networks, of which the sizes vary from 100 to 300. We simulate 150 times of diffusion processes on each network. In each simulation, 0.15*n* nodes are randomly selected as the initial infected nodes.



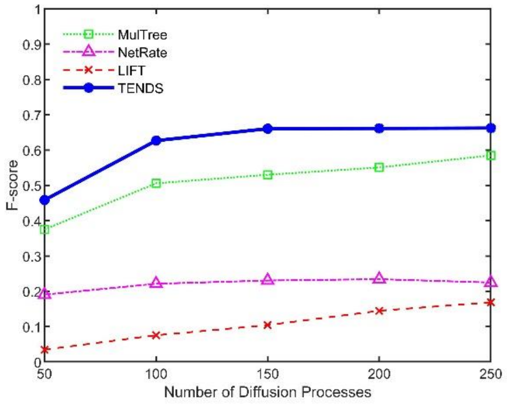
(a) *F-score*



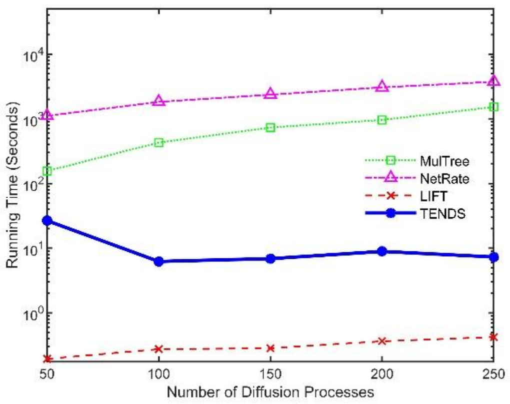
(b) *Running Time*

Experimental Evaluation

➢ **Effect of Number of Diffusion Records**: we test the algorithms on NetSci and DUNF with different number β of diffusion processes (β varies from 50 to 250). In each diffusion process, we randomly select $0.15n$ nodes as the initial infection nodes.

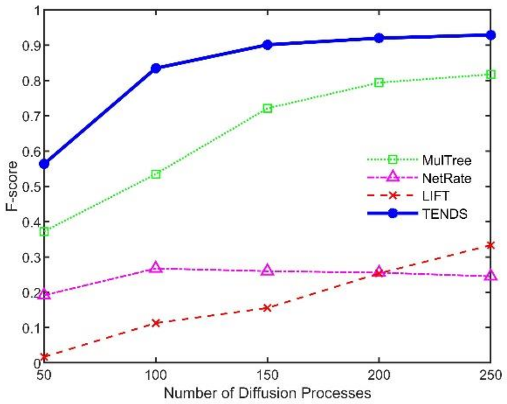


(a) *F-score*

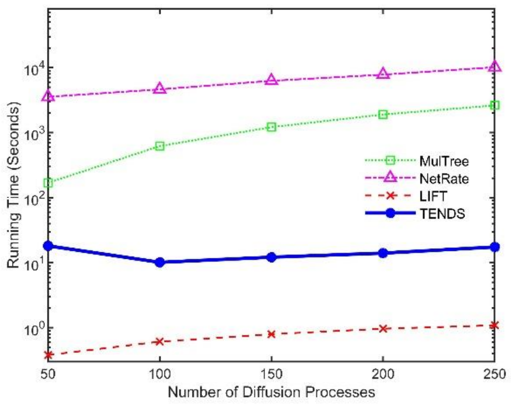


(b) *Running Time*

Effect of number of diffusion processes on NetSci



(a) *F-score*

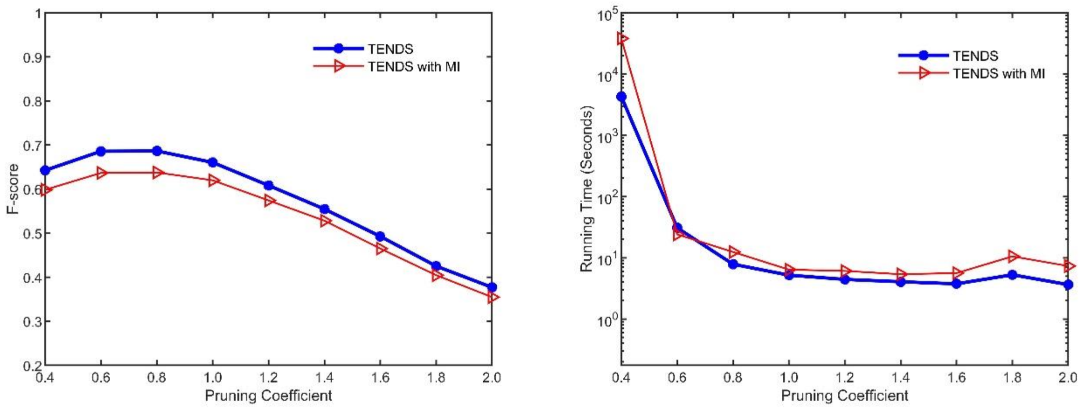


(b) *Running Time*

Effect of number of diffusion processes on DUNF

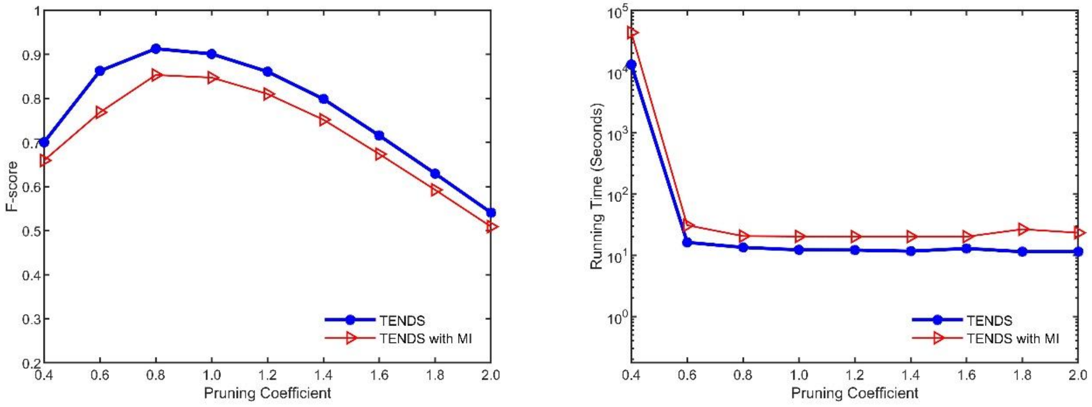
Experimental Evaluation

- **Effect of Infection MI-based Pruning Method**: we test the algorithms on NetSci and DUNF with different pruning threshold, varying from 0.4tau to 2tau, and for each MI threshold, we simulate 150 diffusion processes on each network. In each diffusion process, we randomly select 0.15*n* nodes as the initial infection nodes.



(a) *F-score* (b) *Running Time*

Effect of Infection MI-based Pruning Method on NetSci



(a) *F-score* (b) *Running Time*

Effect of Infection MI-based Pruning Method on DUNF

Takeaway

- Contribution: we proposed a diffusion network topology reconstruction method, using a scoring criterion and the upper bound of parent node size, with the help of a pruning method using infection mutual information.

- Exact timestamps in diffusion records are hard to get and misleading; we do not necessarily need them.

- Experiments showed that our method is robust a wide range of network settings.

Thanks!