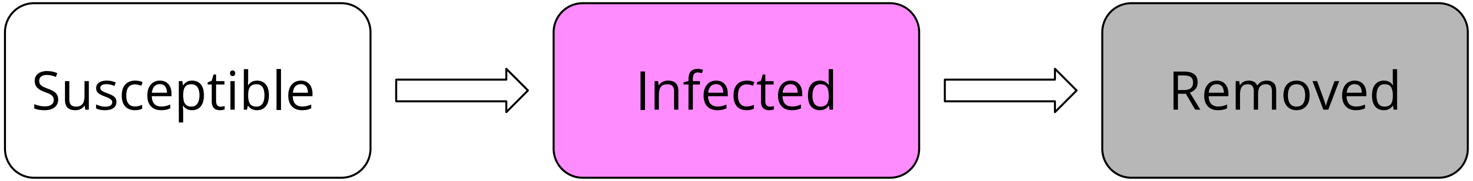
Epidemics on Graphs under Uncertainty

Jessica Hoffmann

Google DeepMind

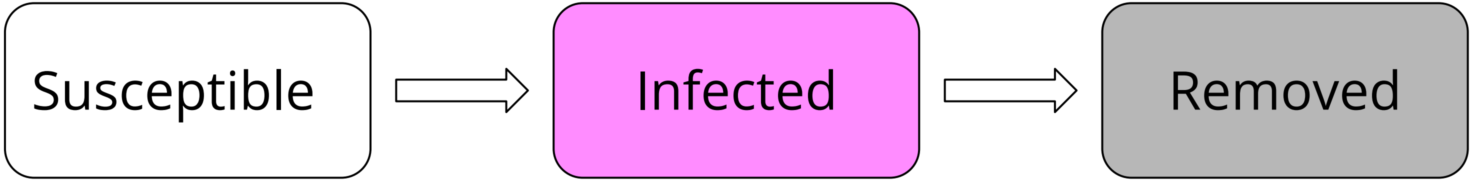
SAMO
April 2025

SIR, SIS, SIRS…



SIR

- Diseases ending with immunization (chicken pox [1]) or death (bubonic plague [2])
- Spread of rumors, viral videos or news [3,4,5] on social networks



→ SIR epidemics end (relatively) fast, with a fraction of the population still susceptible.

[1] J.A. Yorke, W.P. London, *Recurrent outbreak of measles, chickenpox and mumps: II. Systematic differences in contact rates and stochastic effects.*

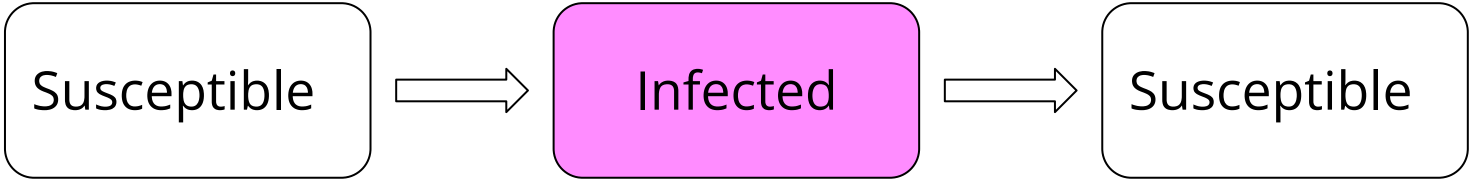
[2] M. J. Keeling and C. A. Gilligan, *Bubonic plague: a metapopulation model of a zoonosis*

[3] Adrien Friggeri, Lada A Adamic, Dean Eckles, and Justin Cheng. *Rumor Cascades.*

[4] C. Bauckhage, F. Hadiji and K. Kersting. *How viral are viral videos?*

[5] Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. *Epidemiological modeling of news and rumors on twitter.*

SIS



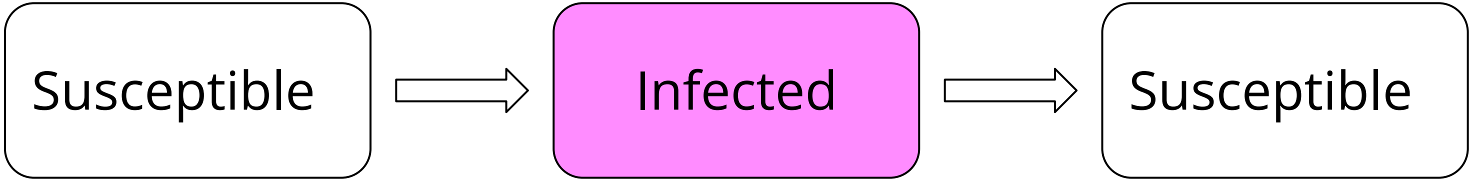
Susceptible

Infected

Susceptible

SIS

- Diseases which mutate too fast (flu [1])
- Malwares [2]

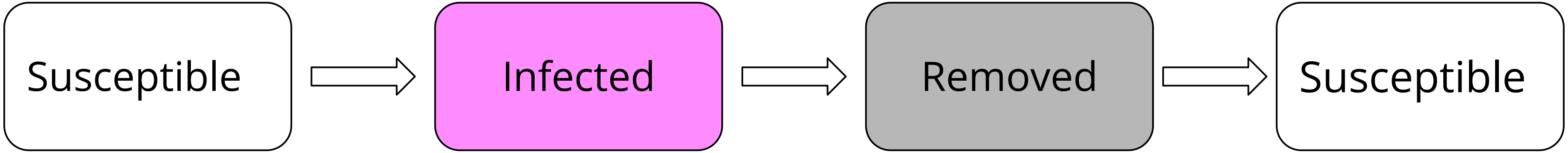


→ SIS epidemics can continue **forever**.

[1] I. Abouelkheir, M. Rachik, O. Zakary and I. Elmouki. *A Multi-regions SIS Discrete Influenza Pandemic Model with a Travel-blocking Vicinity Optimal Control Approach on Cells.*
[2] S. Peng, S. Yu and A. Yang. *Smartphone malware and its propagation modeling: A survey*.

SIRS

- Diseases with temporary immunization (cold [1])
- Memes on social networks [2]
- Information in the brain [3]



[1] A. Webera, M. Weber and P. Milligan. *Modeling epidemics caused by respiratory syncytial virus (RSV).*
[2] C. Bauckhage. *Insights into Internet Memes.*
[3] L. Acedo and J. A. Morano. *Brain oscillations in a random neural network.*

Epidemics on Graphs

- Epidemic estimation
- Epidemic control
- Community detection/clustering
- Edge/link prediction on time-evolving networks
- Network estimation from epidemic
- Source(s) identification/obfuscation

Epidemics on Graphs

- **Epidemic estimation**
- **Epidemic control**
- Community detection/clustering
- Edge/link prediction on time-evolving networks
- **Network estimation from epidemic**
- Source(s) identification/obfuscation

Why uncertainty?

- Most of the previous work has assumed perfect observation to some degree

- For some applications, this is an unreasonable assumption:
  e.g. for COVID-19, data is scarce, delayed, and/or imprecise

- Previous algorithms are not robust to adding back noise. And as we show, neither are the results.

Plan

I. Uncertainty about who is infected/not infected

II. Uncertainty about when people are infected

III. Uncertainty about what infected people

Plan

I. **Uncertainty about who is infected/not infected**

II. Uncertainty about when people are infected

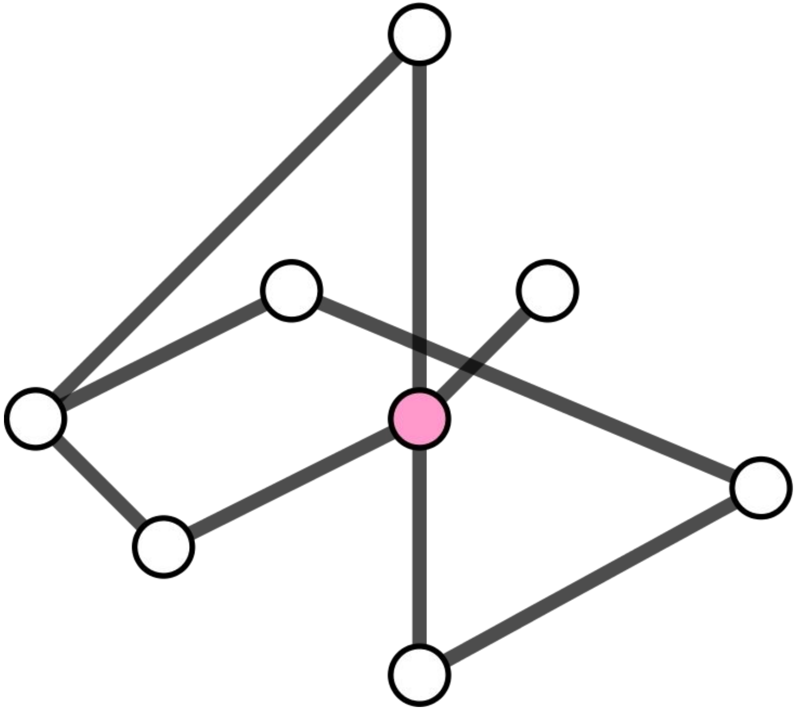
III. Uncertainty about what infected people

The Cost of Uncertainty in Curing Epidemics

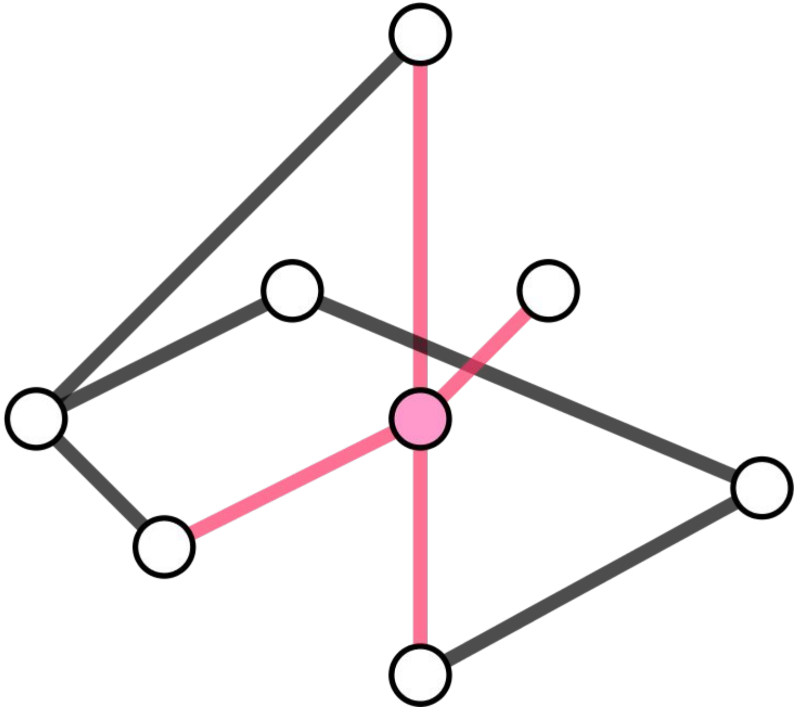
Jessica Hoffmann
Constantine Caramanis

SIGMETRICS 2018

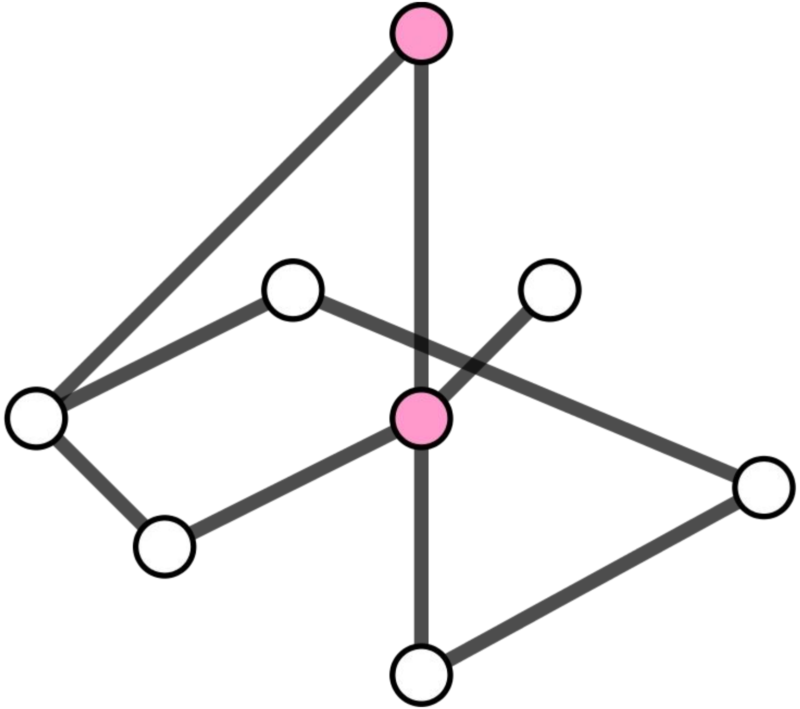
Setting: Controlled SIS



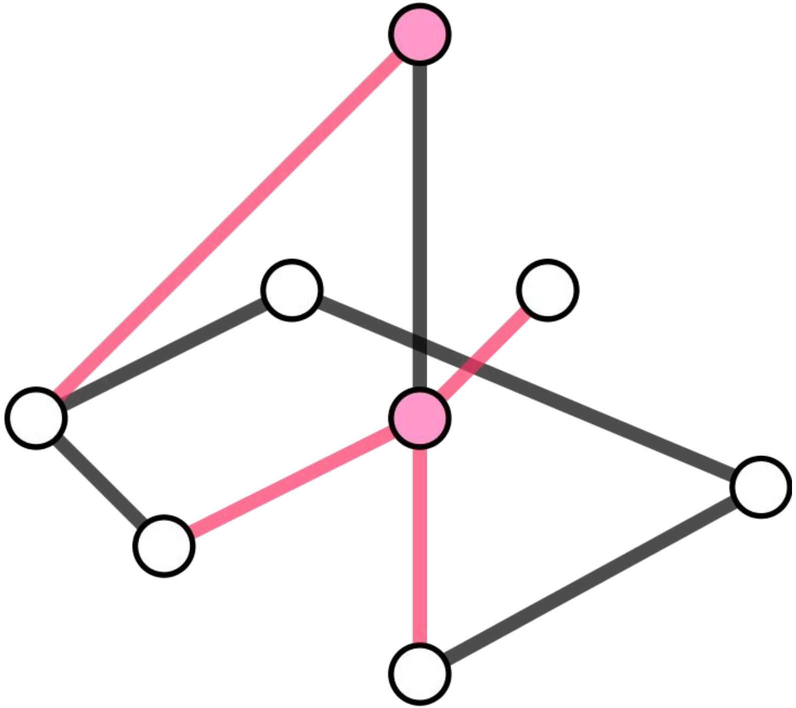
Setting: Controlled SIS



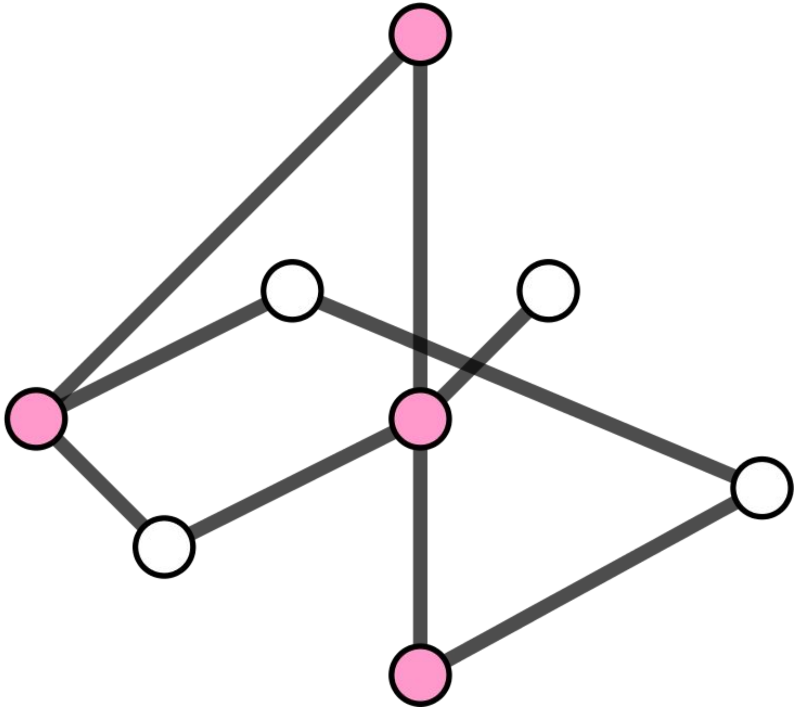
Setting: Controlled SIS



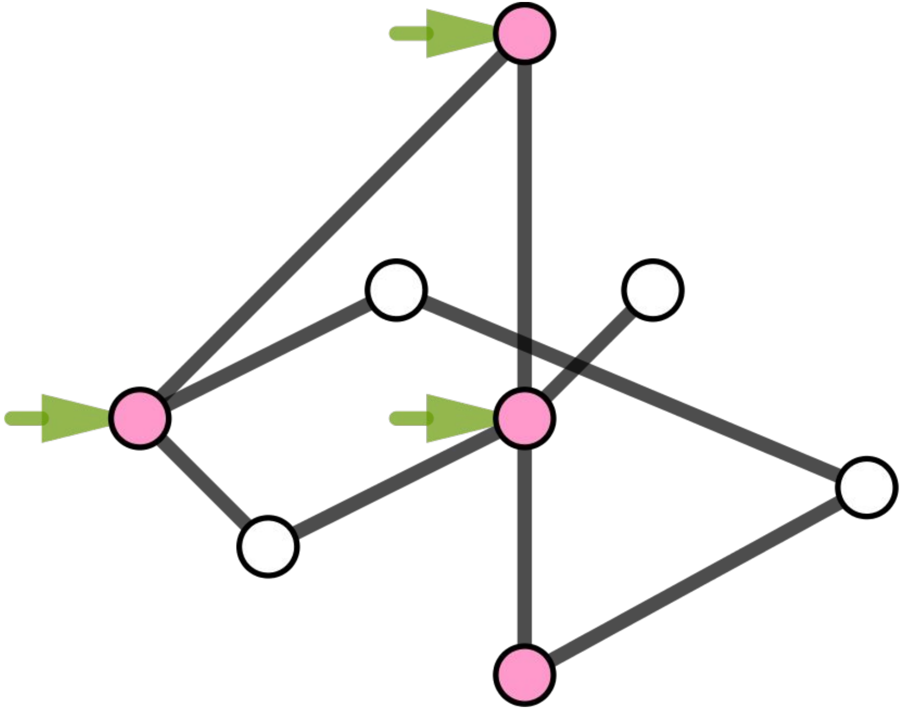
Setting: Controlled SIS



Setting: Controlled SIS

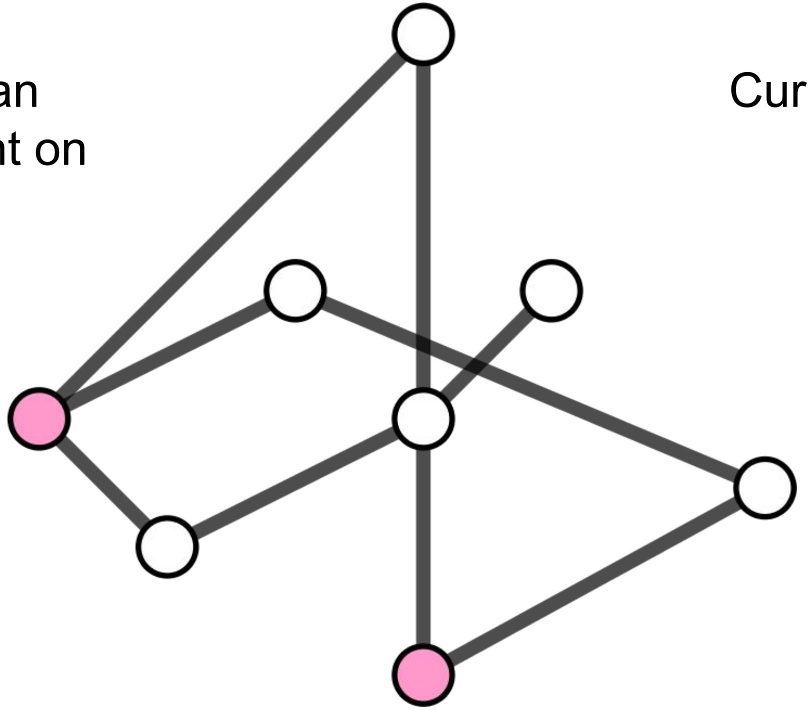


Setting: Controlled SIS



Setting: Controlled SIS

Only infected nodes can be cured. Budget spent on susceptible nodes is wasted.



Cured nodes can become reinfected.

Epidemics on graphs - goals

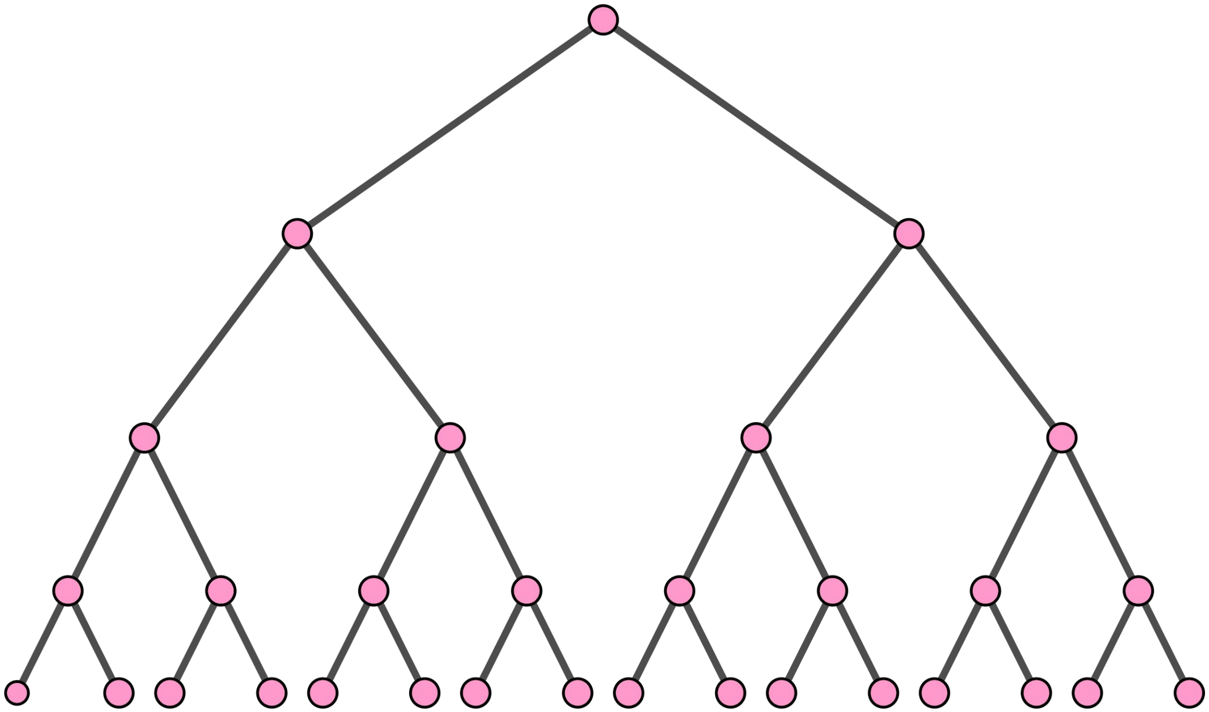
- We start with a fully infected graph

- Our budget is limited

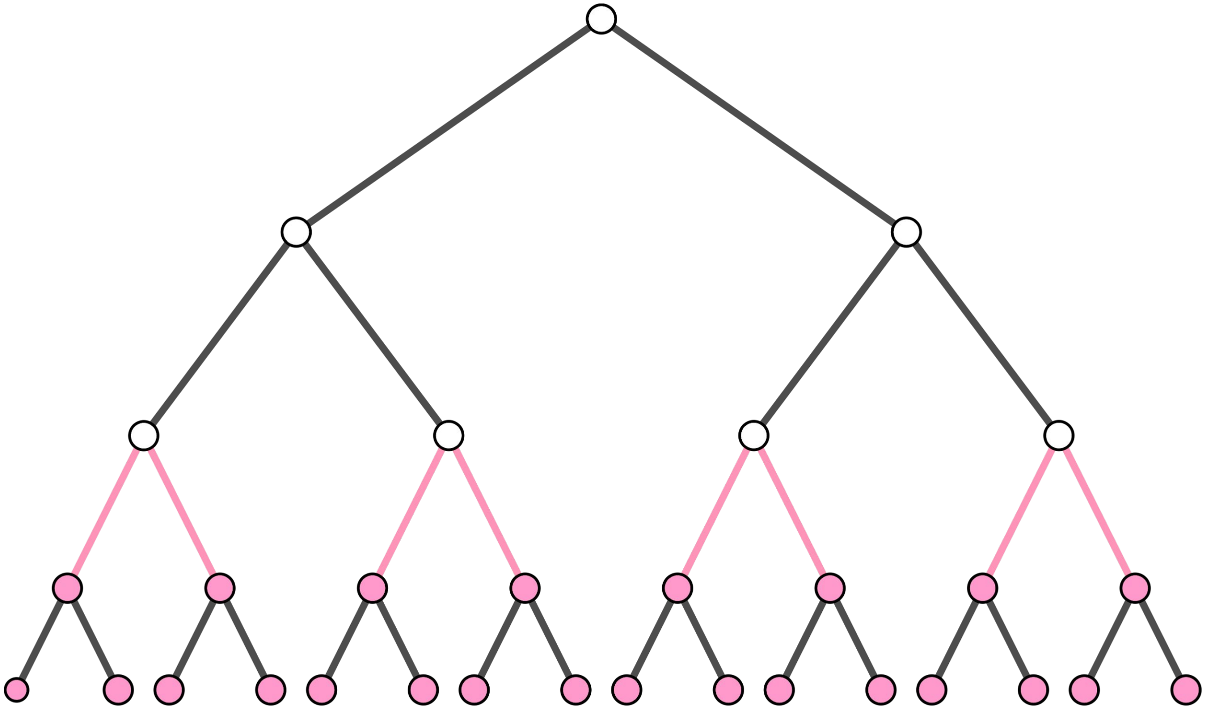
- We can choose which nodes to cure

- The goal is to eradicate the epidemic

Curing the binary tree - 1st try

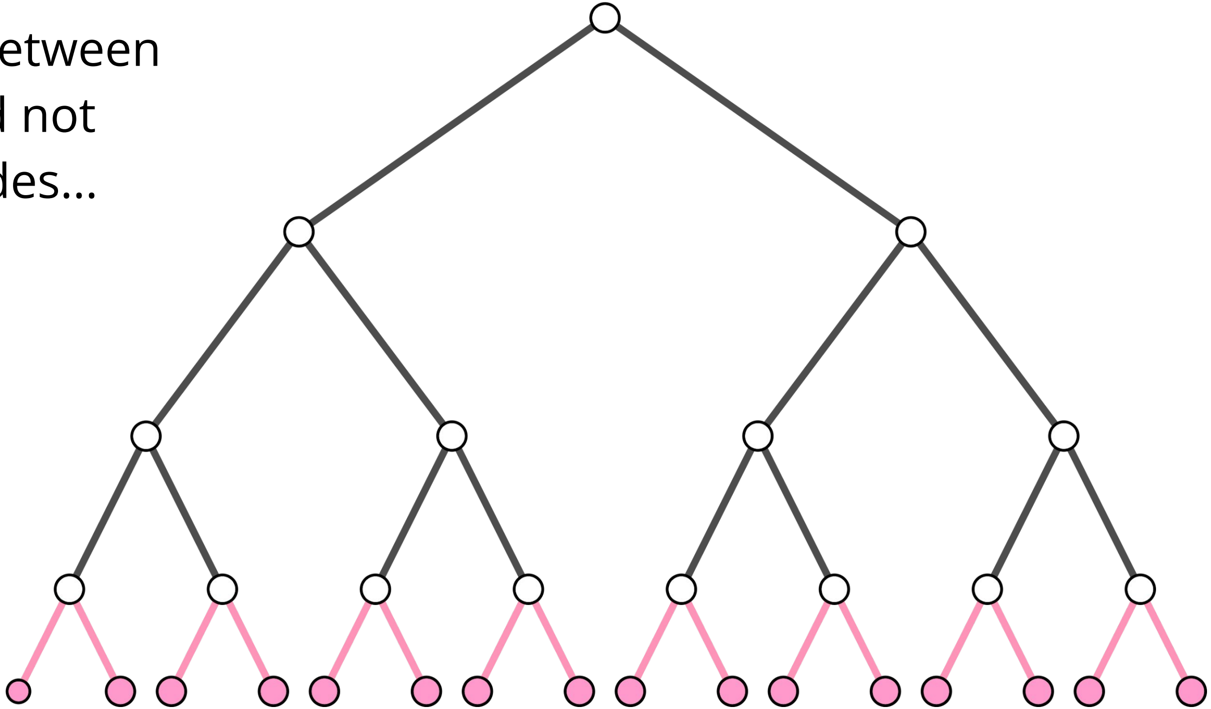


Curing the binary tree - 1st try

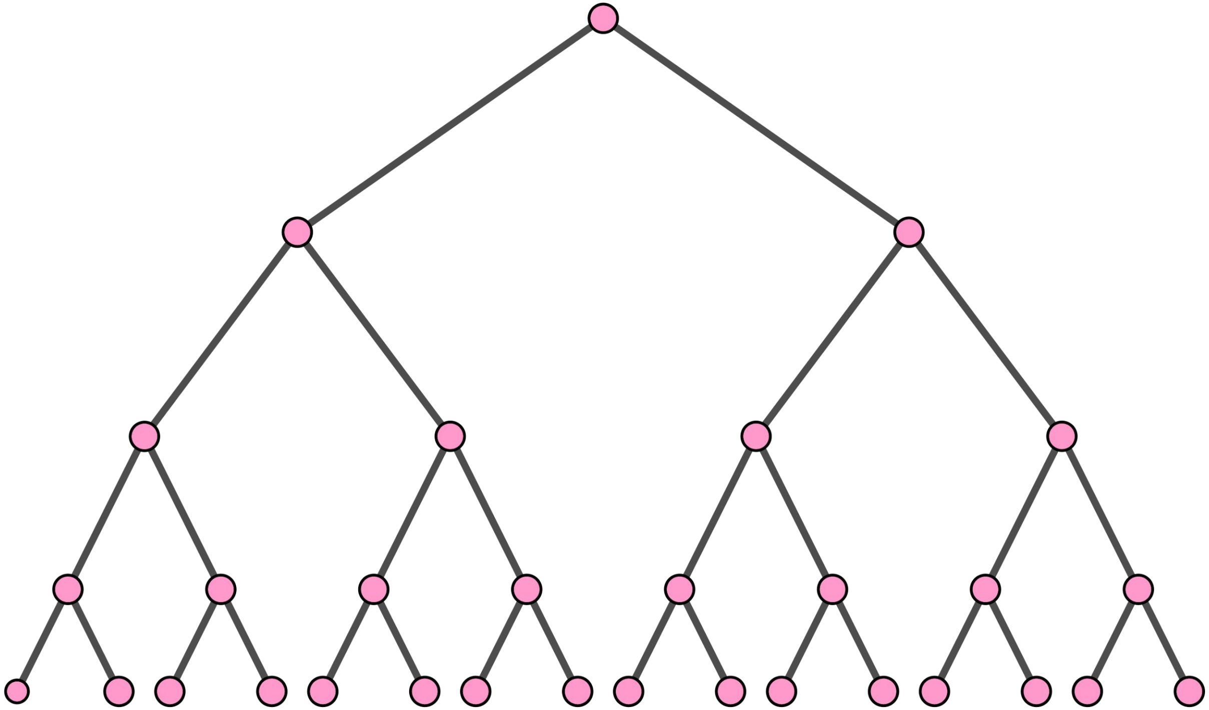


Curing the binary tree - 1st try

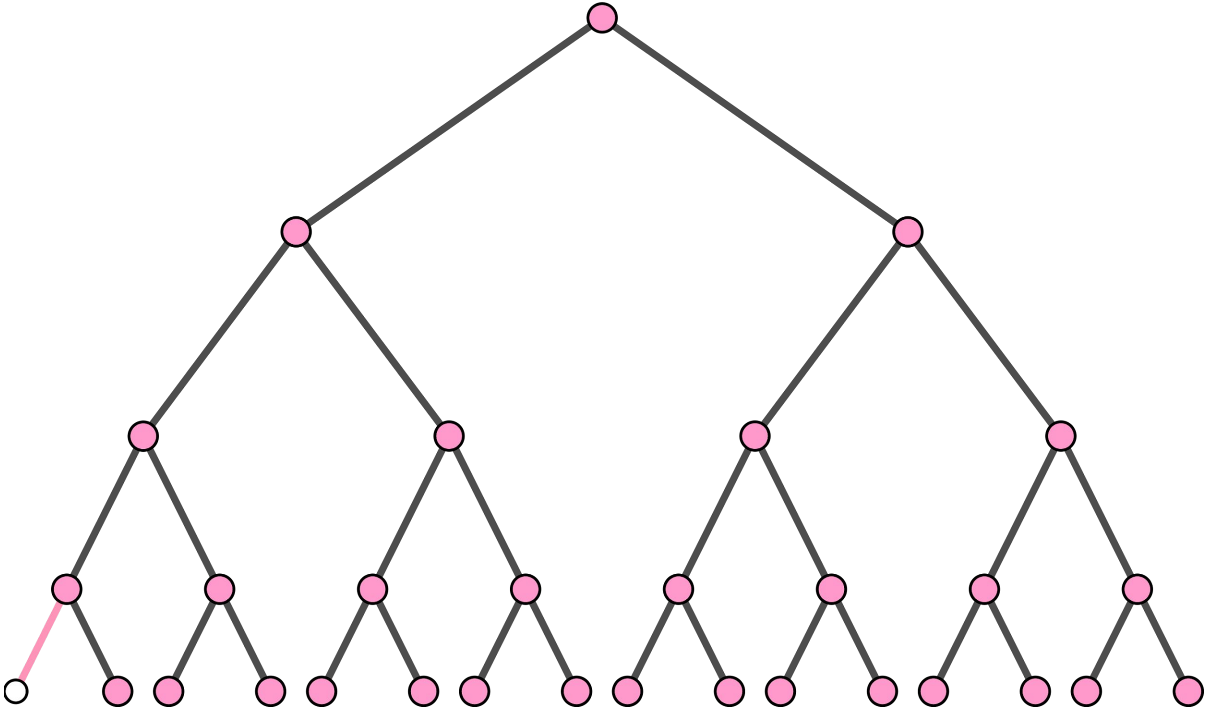
N/2 edges between infected and not infected nodes...



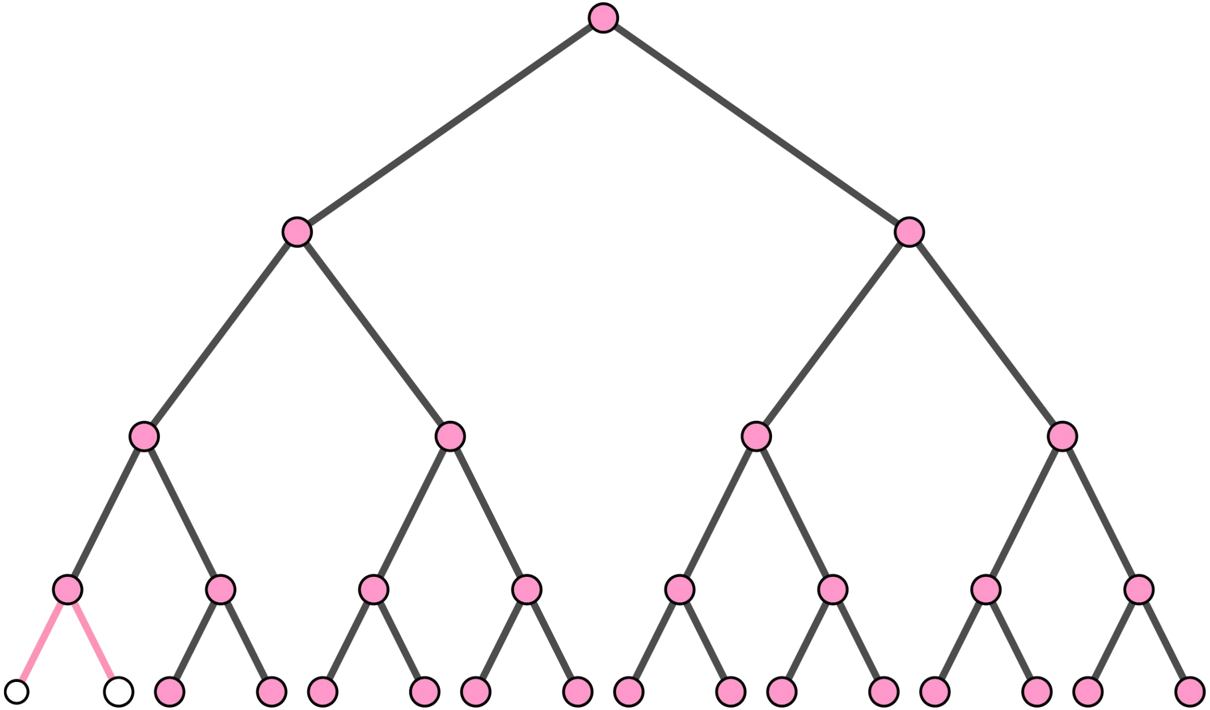
Curing the binary tree - CutWidth



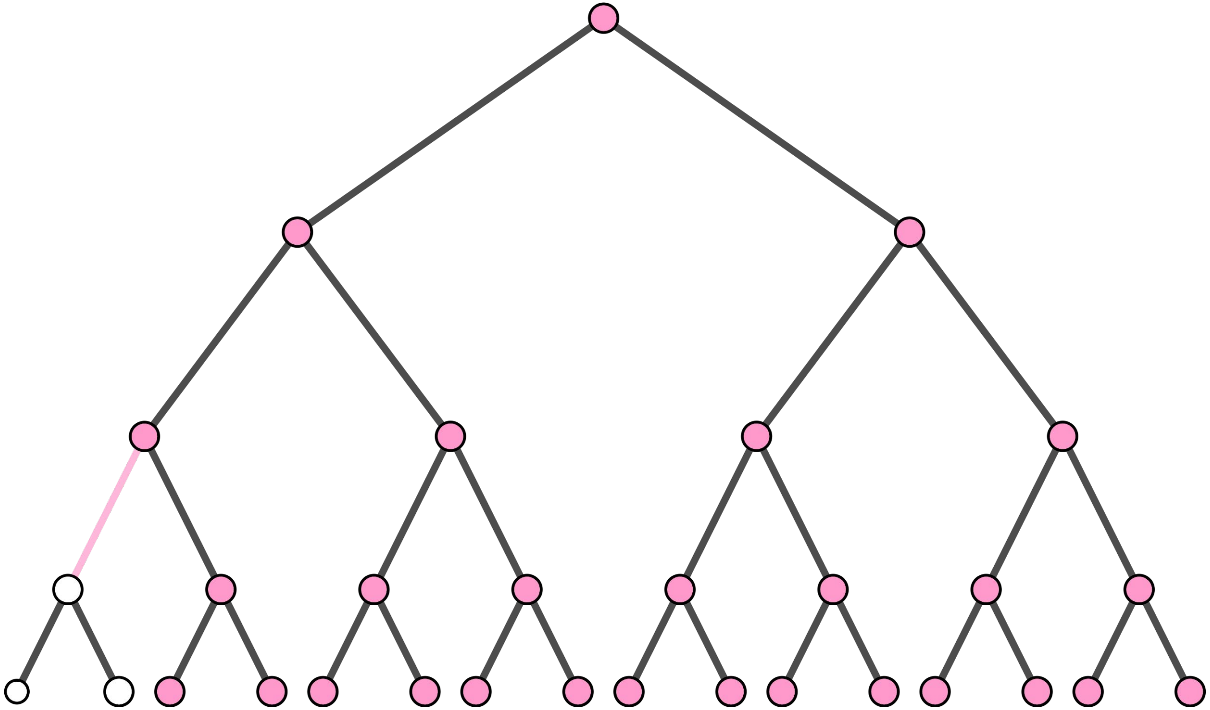
Curing the binary tree - CutWidth



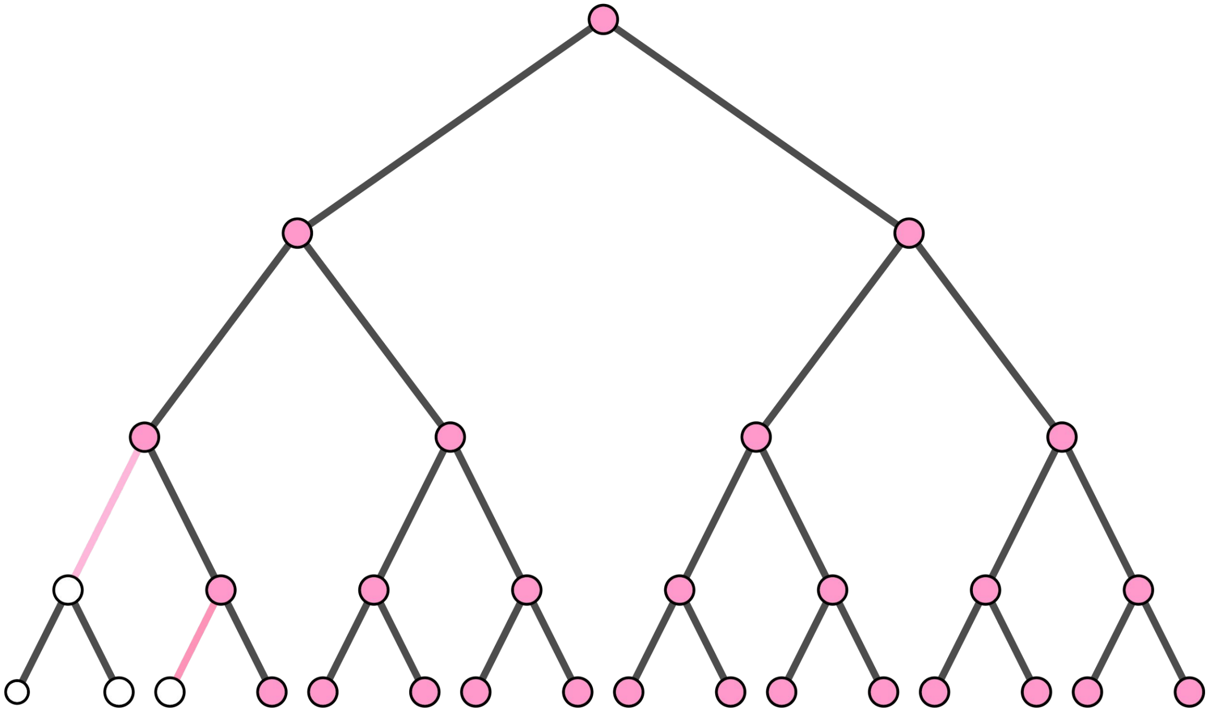
Curing the binary tree - CutWidth



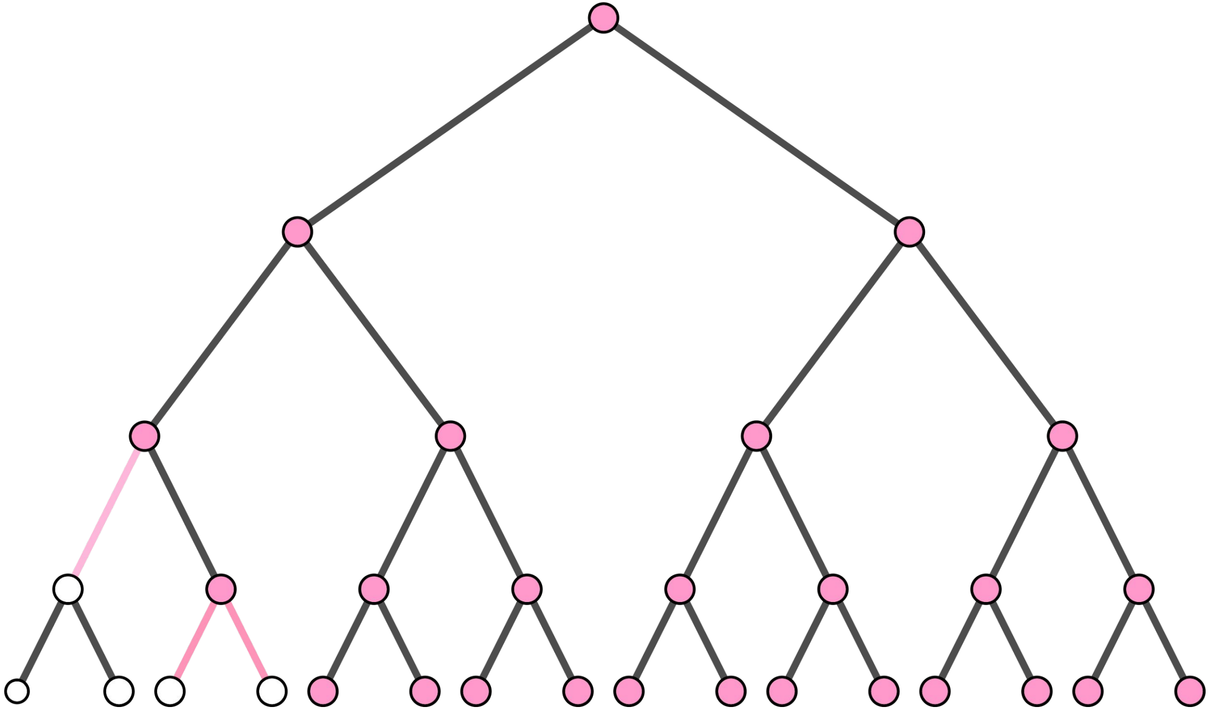
Curing the binary tree - CutWidth



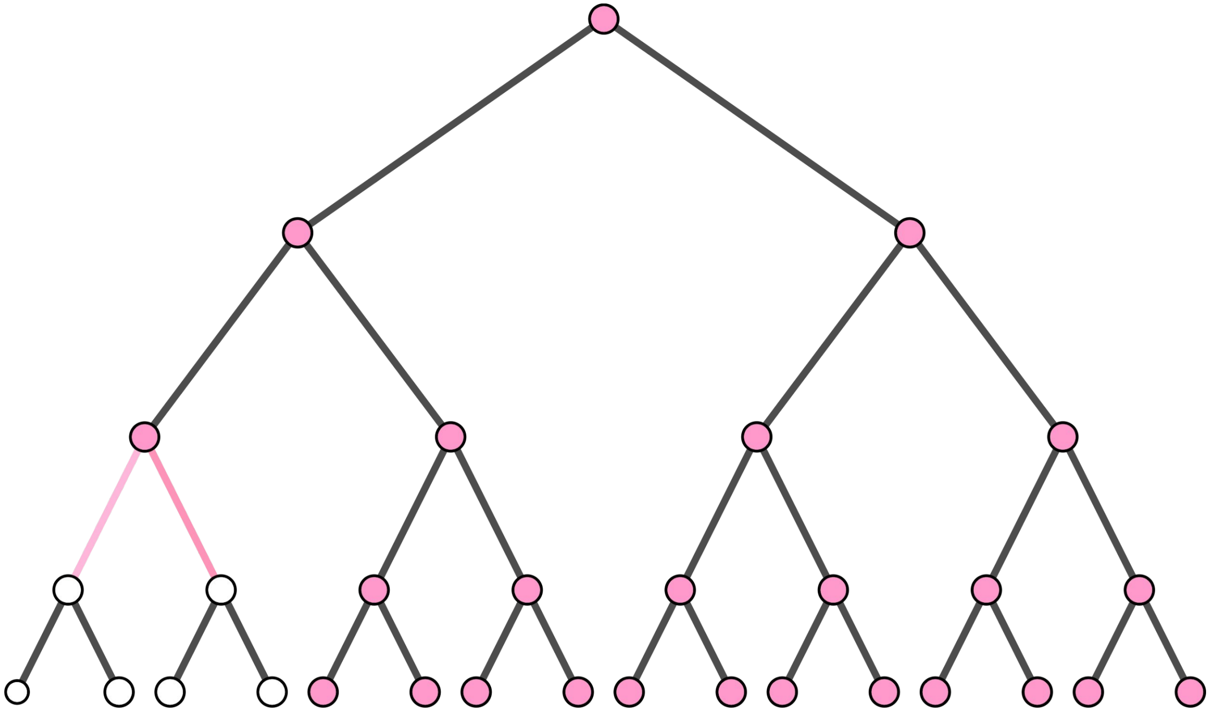
Curing the binary tree - CutWidth



Curing the binary tree - CutWidth

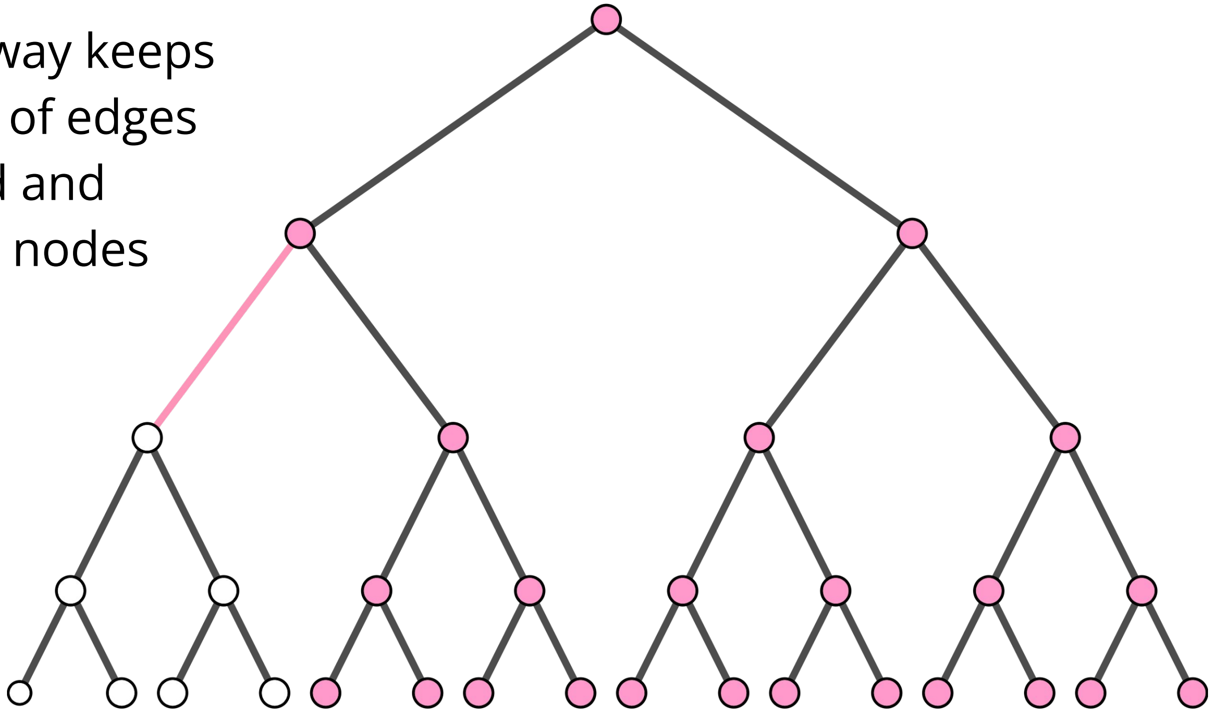


Curing the binary tree - CutWidth



Curing the binary tree - CutWidth

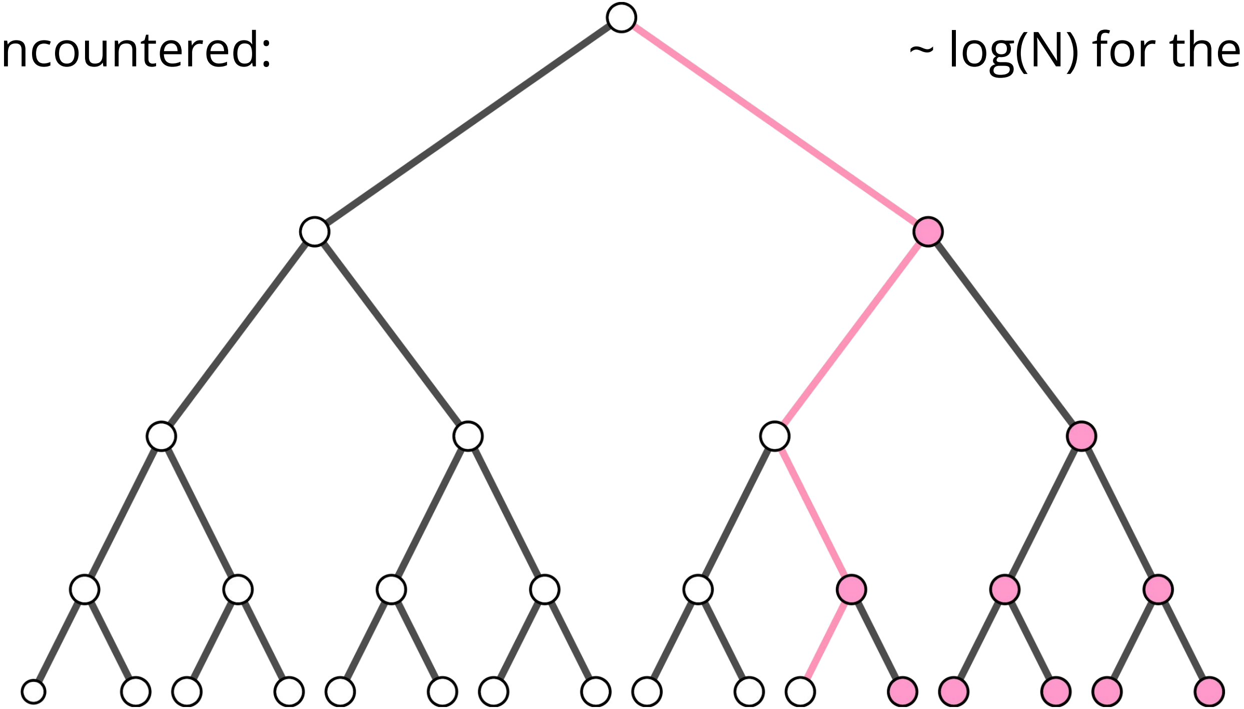
Curing this way keeps the number of edges b/w infected and not infected nodes low.



Curing the binary tree - CutWidth

Worst cut encountered:
Cutwidth

~ log(N) for the binary tree



Cutwidth - definition

- Crusade: set of sets, $\{S_0, \ldots, S_k\}$ where $S_0 = V,\ S_k = \emptyset$
  $S_0 \supset S_1 \supset \cdots \supset S_k, \quad |S_i \setminus S_{i+1}| = 1$

- Rate of a crusade: $\max_i \ CUT(S_i, V \setminus S_i)$

- Cutwidth: $\min_{crusade\ \mathcal{C}} \ RATE(\mathcal{C})$

State-of-the-Art before this paper

- Mitigating/eradicating epidemics is still an ongoing research topic [1, 2]

- 2015: K. Drakopoulos, A. Ozdaglar, and J. N. Tsitsiklis, establishes that there exists a combinatorics property of graphs, called the CutWidth, which plays a crucial role in curing graphs.

- If budget ≤ (1 - ε) × CutWidth, curing takes at least exponential time (in the number of nodes) in expectation [1].

- **If budget ≥ (1 + ε) × CutWidth, curing is easy** and takes linear time [2].

- Their results hold if we **know exactly which nodes are infected, at each time**.

[1] Lars Lorch, Abir De, Samir Bhatt, William Trouleau, Utkarsh Upadhyay, Manuel Gomez-Rodriguez. *Stochastic Optimal Control of Epidemic Processes in Networks*
[2] Han-Ching Ou, Arunesh Sinha, Sze-Chuan Suen, Andrew Perrault, Milind Tambe. *Who and When to Screen: Multi-Round Active Screening for Recurrent Infectious Diseases Under Uncertainty*
[3] Kimon Drakopoulos, Asuman Ozdaglar, and John N. Tsitsiklis. *A lower bound on the performance of dynamic curing policies for epidemics on graphs*.
[4] Kimon Drakopoulos, Asuman Ozdaglar, and John N. Tsitsiklis. *An efficient curing policy for epidemics on graphs.*

Uncertainty about the states of the nodes

- In practice, no one gets tested as soon as there are infected

- False positive/negative when tested

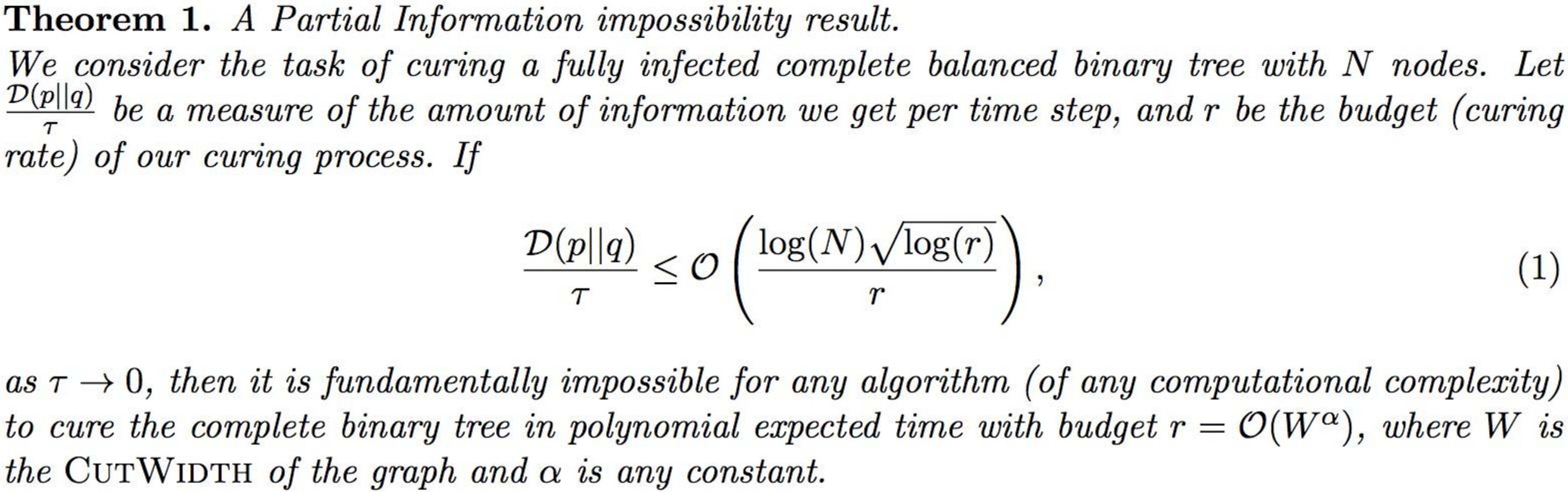
Can we extend the results to the uncertain setting?

Is curing with uncertainty always possible?

No.

(We will show a counter-example)

Our theorem

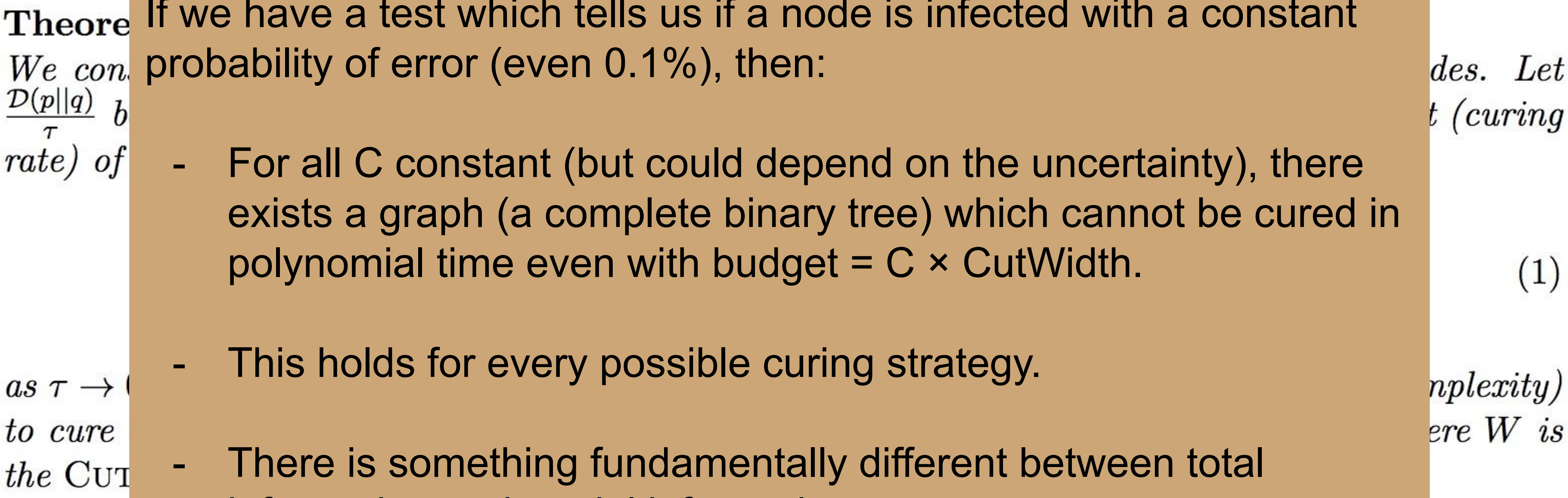


Theorem 1. *A Partial Information impossibility result.*
We consider the task of curing a fully infected complete balanced binary tree with N nodes. Let $\frac{\mathcal{D}(p||q)}{\tau}$ be a measure of the amount of information we get per time step, and r be the budget (curing rate) of our curing process. If

$$\frac{\mathcal{D}(p||q)}{\tau} \leq \mathcal{O}\left(\frac{\log(N)\sqrt{\log(r)}}{r}\right), \tag{1}$$

as $\tau \rightarrow 0$, then it is fundamentally impossible for any algorithm (of any computational complexity) to cure the complete binary tree in polynomial expected time with budget $r = \mathcal{O}(W^{\alpha})$, where W is the CutWidth *of the graph and α is any constant.*

Our theorem - what it means



Theore *We con* *des. Let*

$\frac{\mathcal{D}(p||q)}{\tau}$ *b* *t (curing*

rate) of

(1)

as $\tau \rightarrow$ *nplexity)*

to cure *ere W is*

the CUT

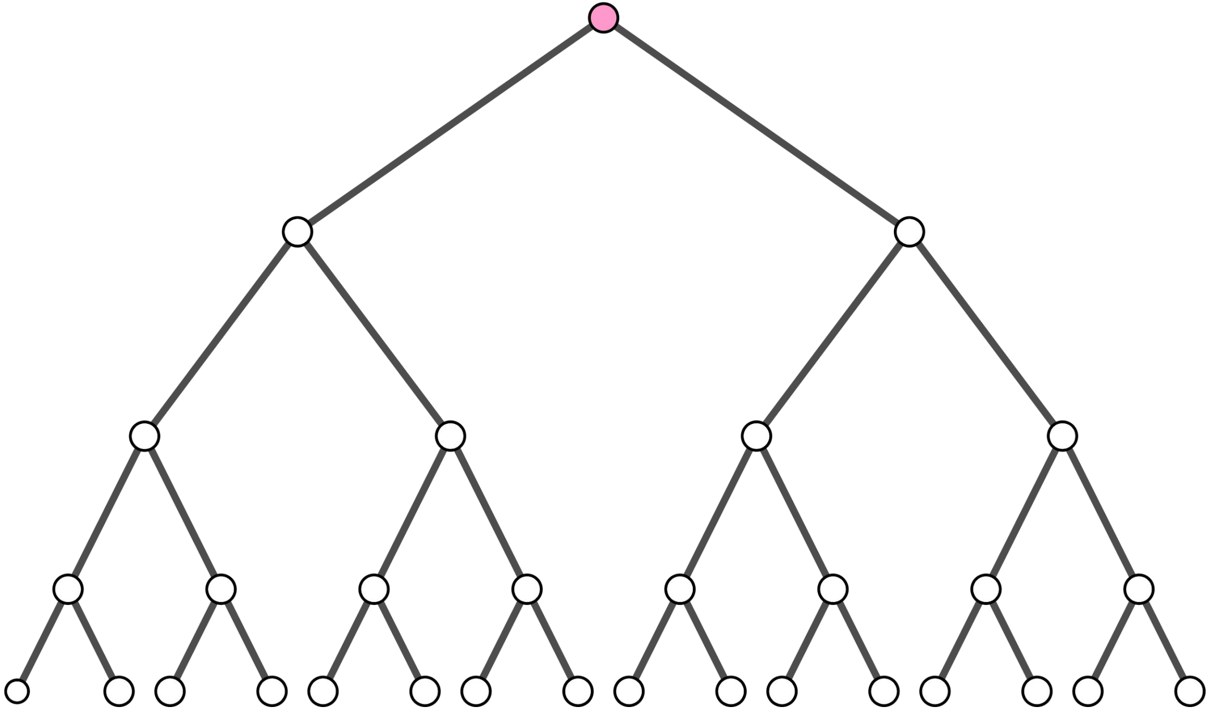
If we have a test which tells us if a node is infected with a constant probability of error (even 0.1%), then:

- For all C constant (but could depend on the uncertainty), there exists a graph (a complete binary tree) which cannot be cured in polynomial time even with budget = C × CutWidth.

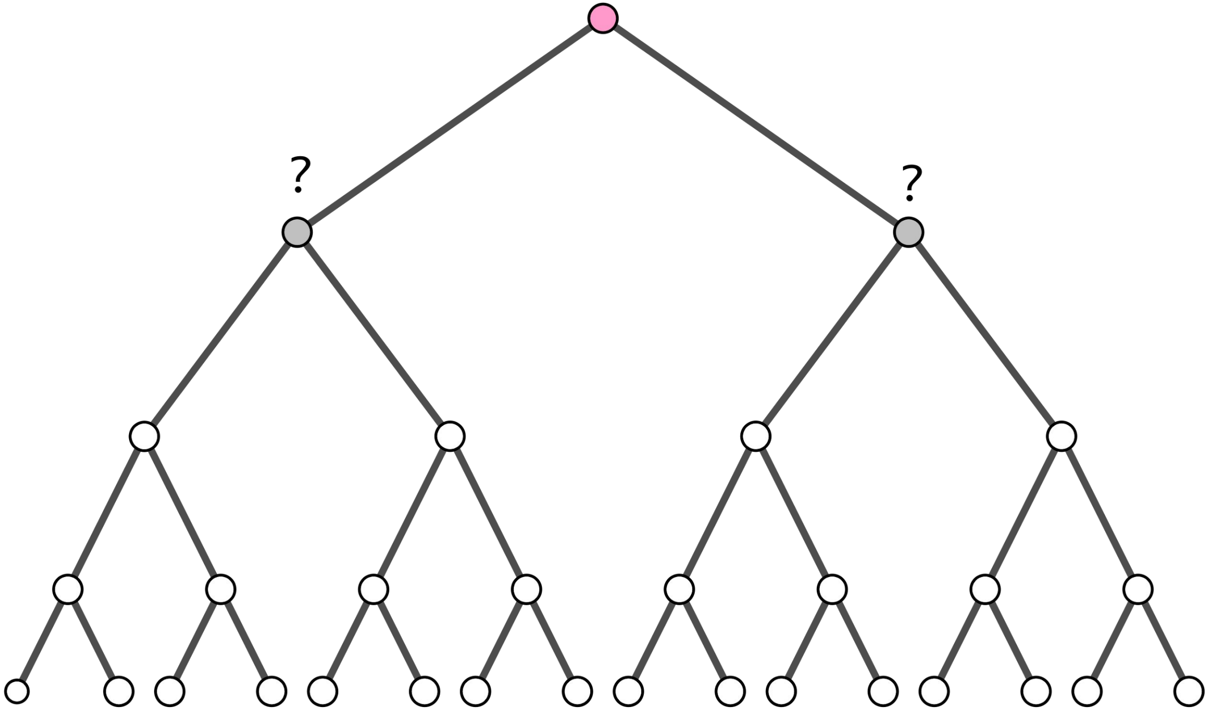
- This holds for every possible curing strategy.

- There is something fundamentally different between total information and partial information.

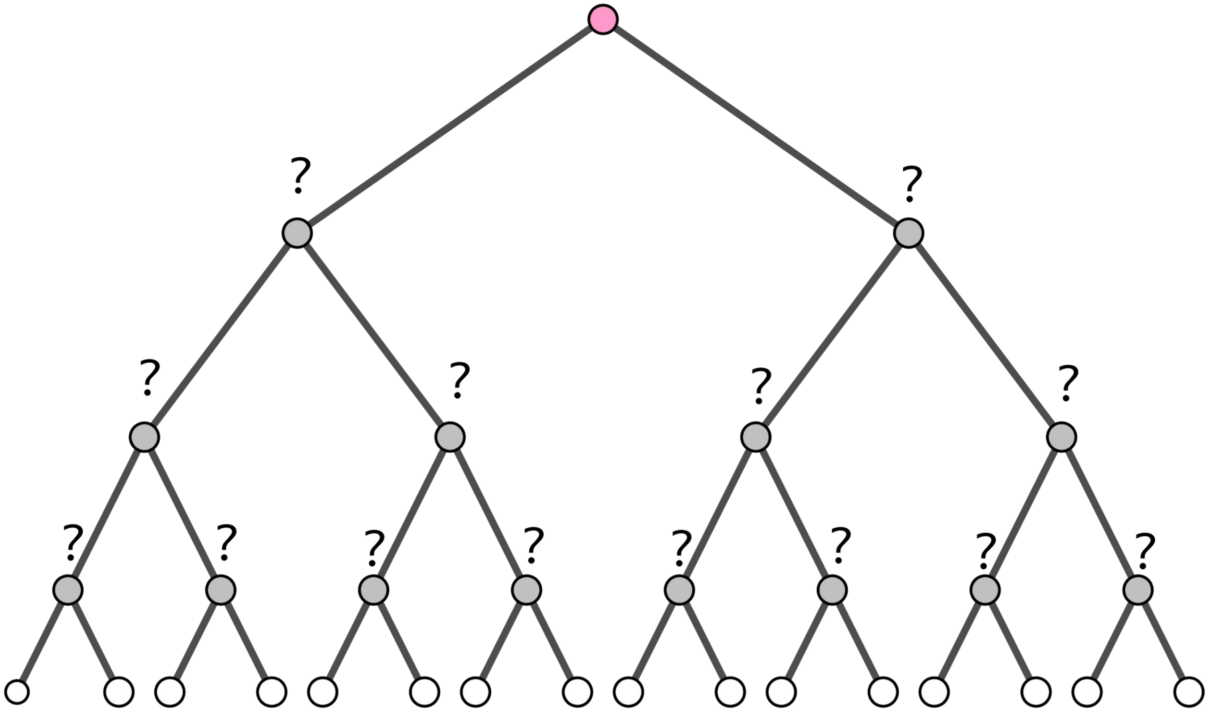
Why is curing with uncertainty so different?



Why is curing with uncertainty so different?

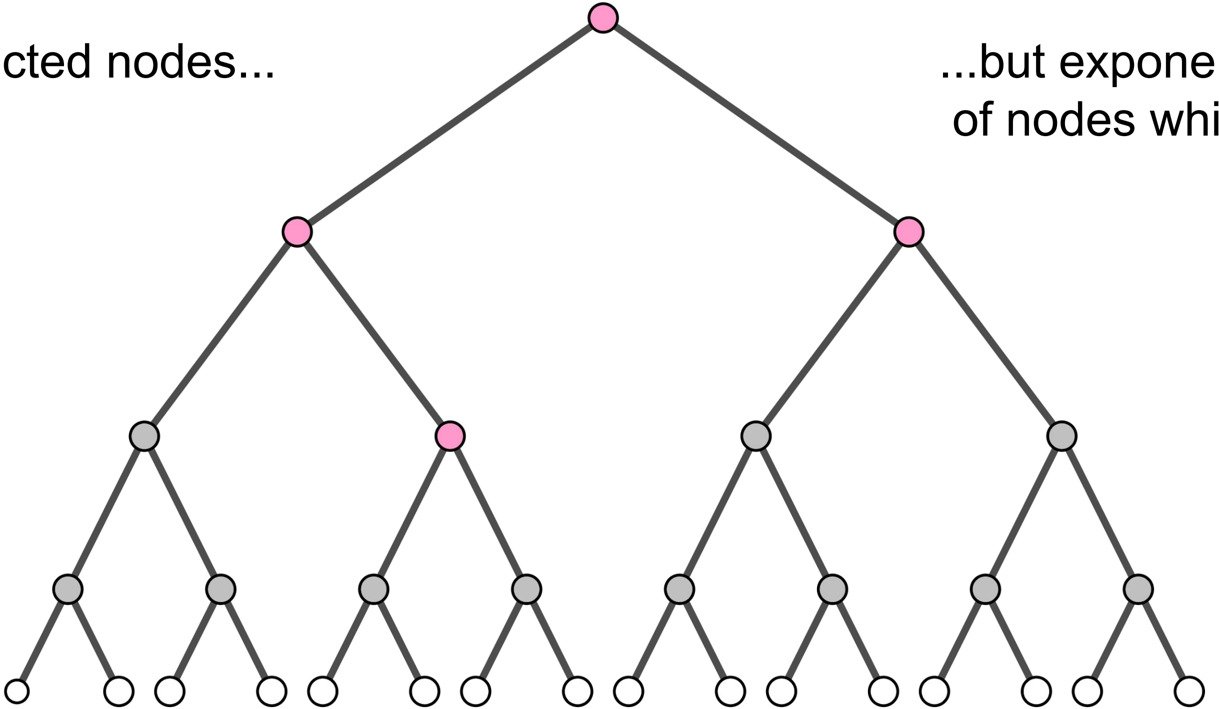


Why is curing with uncertainty so different?



Why is curing with uncertainty so different?

Very few infected nodes...



...but exponential number of nodes which **could** be infected.

Conclusions

- The binary tree cannot be cured in polynomial time within reasonable budget.

- We identified bottlenecks which would happen under **any** curing strategy.

- With partial information, we have to take into account all the nodes which could **potentially** be reinfected. This can be exponentially bigger than the number of nodes actually infected.

- Uncertainty completely changes the results!

Plan

I. Uncertainty about who is infected/not infected

II. **Uncertainty about when people are infected**

III. Uncertainty about what infected people

Learning Graphs from Noisy Epidemic Cascades

Jessica Hoffmann

Constantine Caramanis

SIGMETRICS 2019

2nd place at INFORMS George Nicholson student paper competition

Differences with the previous problem

- Inverse problem: we now aim to reconstruct graphs from epidemic cascades.

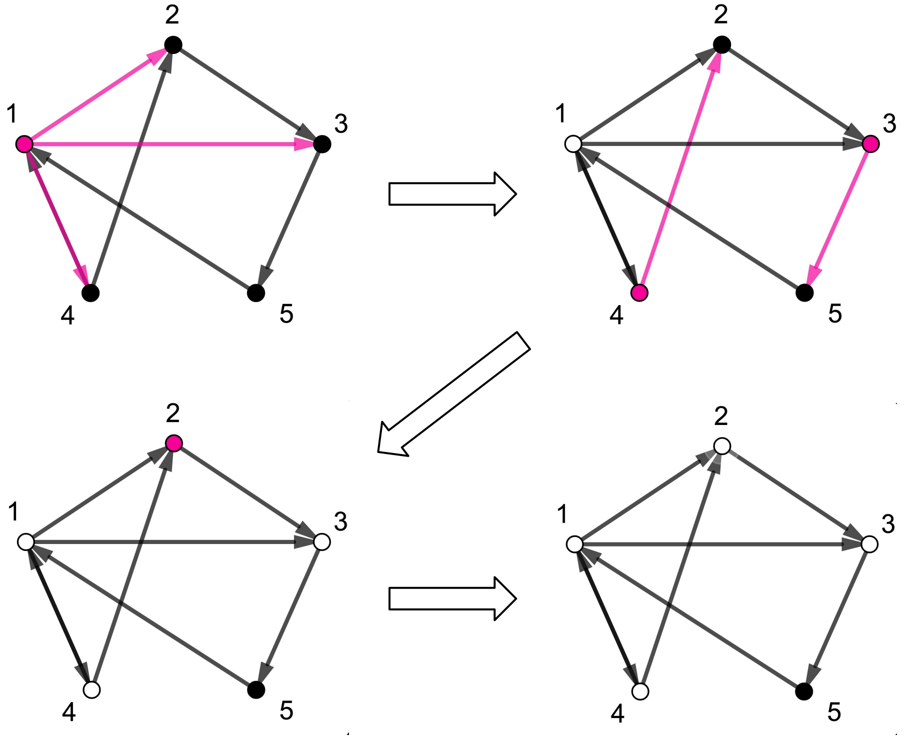
- Propagation model: we are now in a SIR model (nodes can be infected only once, cascades die out spontaneously)

- Observation model: we know exactly who was infected, but we are not sure when (noisy times of infection)

Times of infection as samples

$$\begin{array}{ll} \text{Node 1} & \\ \text{Node 2} & \\ \text{Node 3} & \\ \text{Node 4} & \\ \text{Node 5} & \end{array} \begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \\ \infty \end{bmatrix}$$

One sample



Times of infection as samples

$$\begin{array}{l} \text{Node 1} \\ \text{Node 2} \\ \text{Node 3} \\ \text{Node 4} \\ \text{Node 5} \end{array} \quad \begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \\ \infty \end{bmatrix}$$

↑

One sample

These samples can be used to reconstruct the exact weights of every edge, for any graph [1].

Rich literature on network inference in a variety of settings [2,3,4, ...]

[1] Praneeth Netrapalli and Sujay Sanghavi. 2012. *Learning the graph of epidemic cascades.*

[2] Bruno Abrahao, Flavio Chierichetti, Robert Kleinberg, and Alessandro Panconesi. *Trace complexity of network inference*.

[3] Hadi Daneshmand, Manuel Gomez-Rodriguez, Le Song, Bernhard Schoelkopf. *Estimating Diffusion Network Structures: Recovery Conditions, Sample Complexity & Soft-thresholding Algorithm.*

[4] Ali Zarezade, Ali Khodadadi, Mehrdad Farajtabar, Hamid R Rabiee, and Hongyuan Zha. *Correlated Cascades : Compete or Cooperate*

Noisy times of infection as samples

$$\begin{array}{l}\text{Node 1}\\\text{Node 2}\\\text{Node 3}\\\text{Node 4}\\\text{Node 5}\end{array}\quad \underset{\substack{\Big\uparrow\\ \text{Cascade}}}{\begin{bmatrix}0\\2\\1\\1\\\infty\end{bmatrix}} + \underset{\substack{\Big\uparrow\\ \text{Noise}}}{\begin{bmatrix}2\\1\\2\\0\\2\end{bmatrix}} = \underset{\substack{\Big\uparrow\\ \text{One sample}}}{\begin{bmatrix}2\\3\\3\\1\\\infty\end{bmatrix}}$$

Noisy times of infection as samples

$$\begin{array}{l} \text{Node 1} \\ \text{Node 2} \\ \text{Node 3} \\ \text{Node 4} \\ \text{Node 5} \end{array} \quad \underset{\substack{\Big\uparrow \\ \text{Cascade}}}{\begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \\ \infty \end{bmatrix}} + \underset{\substack{\Big\uparrow \\ \text{Noise}}}{\begin{bmatrix} 2 \\ 1 \\ 2 \\ 0 \\ 2 \end{bmatrix}} = \underset{\substack{\Big\uparrow \\ \text{One sample}}}{\begin{bmatrix} 2 \\ 3 \\ 3 \\ 1 \\ \infty \end{bmatrix}}$$

Noise could represent:

- time it takes for someone to visit a doctor

- hibernation (latent phase) of disease (HIV, COVID-19)

Noisy times of infection as samples

$$\begin{array}{l} \text{Node 1}\\ \text{Node 2}\\ \text{Node 3}\\ \text{Node 4}\\ \text{Node 5} \end{array} \quad \underset{\substack{\Big\uparrow\\ \text{Cascade}}}{\begin{bmatrix}0\\2\\1\\1\\\infty\end{bmatrix}} + \underset{\substack{\Big\uparrow\\ \text{Noise}}}{\begin{bmatrix}2\\1\\2\\0\\2\end{bmatrix}} = \underset{\substack{\Big\uparrow\\ \text{One sample}}}{\begin{bmatrix}2\\3\\3\\1\\\infty\end{bmatrix}}$$

Noise assumptions:

- i.i.d.

- does not take infinite values

Noisy times of infection as samples

$$\begin{array}{l}\text{Node 1}\\ \text{Node 2}\\ \text{Node 3}\\ \text{Node 4}\\ \text{Node 5}\end{array} \quad \underset{\substack{\uparrow\\ \text{Cascade}}}{\begin{bmatrix}0\\2\\1\\1\\\infty\end{bmatrix}} + \underset{\substack{\uparrow\\ \text{Noise}}}{\begin{bmatrix}2\\1\\2\\0\\2\end{bmatrix}} = \underset{\substack{\uparrow\\ \text{One sample}}}{\begin{bmatrix}2\\3\\3\\1\\\infty\end{bmatrix}}$$

Noisy times of infection as samples

$$\begin{array}{l} \text{Node 1} \\ \text{Node 2} \\ \text{Node 3} \\ \text{Node 4} \\ \text{Node 5} \end{array} \quad \underset{\substack{\Uparrow \\ \text{Cascade}}}{\begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \\ \infty \end{bmatrix}} + \underset{\substack{\Uparrow \\ \text{Noise}}}{\begin{bmatrix} 2 \\ 1 \\ 2 \\ 0 \\ 2 \end{bmatrix}} = \underset{\substack{\Uparrow \\ \text{One sample}}}{\begin{bmatrix} 2 \\ 3 \\ 3 \\ 1 \\ \infty \end{bmatrix}}$$

Limited-noise model

Noisy times of infection as samples

$$\begin{array}{l}\text{Node 1}\\ \text{Node 2}\\ \text{Node 3}\\ \text{Node 4}\\ \text{Node 5}\end{array} \quad \underset{\substack{\Big\uparrow\\ \text{Cascade}}}{\begin{bmatrix}0\\2\\1\\1\\\infty\end{bmatrix}} + \underset{\substack{\Big\uparrow\\ \text{Noise}}}{\begin{bmatrix}?\\?\\?\\?\\?\end{bmatrix}} = \underset{\substack{\Big\uparrow\\ \text{One sample}}}{\begin{bmatrix}0\\0\\0\\0\\\infty\end{bmatrix}}$$

Extreme-noise model

- imprecision due to frequency of reports

Noise models

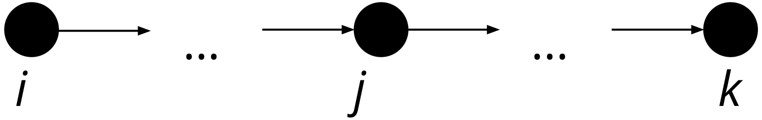
- Limited noise:
    - For each cascade, know noisy estimate of the times of infection.
    - We learn the weights of all edges up to precision ϵ

- Extreme noise:
    - For each cascade, we only know which nodes were infected.
    - We learn the presence/absence of edges

Extreme-noise setting - case of tree

1) Uniqueness of paths in trees

2) Therefore, if there is a path as below, *i* and *j* will be co-infected more often than *i* and *k*.



3) We can order all pairs of nodes by decreasing order of co-infections, and keep any edge that does not form a cycle.

Limited-noise setting - case of tree

- We define estimators:

$$\hat{f}_{i<j} = Fraction\ of\ infections\ for\ which\ i\ \ and\ j\ \ got\ infected,\ and\ i\ \ reported\ before\ j\ .$$
$$\hat{g}_{i,\not{j}} = Fraction\ of\ infections\ for\ which\ i\ \ got\ infected,\ but\ j\ \ did\ not.$$

- Complex expectation in general

- If *i* and *j* share an edge (which we can learn using the method above), we can express the limit of these estimators in a simple way:

$$f_{i<j} = \mathcal{P}_{\not{j}}(\rightarrow i) \cdot p_{ij} \cdot s_0 + \mathcal{P}_{\not{i}}(\rightarrow j) \cdot p_{ji} \cdot s_2$$
$$g_{i,\not{j}} = \mathcal{P}_{\not{j}}(\rightarrow i) \cdot (1 - p_{ij}).$$

Theorems: sample complexity

	No noise [1]	Limited-noise	Extreme-noise
Trees	$\mathcal{O}(N\log(N))$	$\mathcal{O}(N\log(N))$	$\mathcal{O}(N\log(N))$
degree ≤ d, $p_{max} \sim \frac{1}{d}$	$\mathcal{O}(d^2 N\log(N))$	$\mathcal{O}(dN\log(N))$	$\mathcal{O}(dN\log(N))$
General graphs	$\mathcal{O}(N^3\log(N))$	$e^{\mathcal{O}(N)}$	$e^{\mathcal{O}(N)}$

[1] Praneeth Netrapalli and Sujay Sanghavi. 2012. *Learning the graph of epidemic cascades*.

Conclusions

- We can learn the edge **weights** of trees and bounded-degree graphs from **noisy** epidemic cascade with **optimal sample complexity** (up to log factors)

- We proved learning general graphs is possible

- We believe our result can be extended to any discrete-time spreading model, with multiple sources of infection

Plan

I. Uncertainty about who is infected/not infected

II. Uncertainty about when people are infected

III. **Uncertainty about what infected people**

Learning Mixture of Graphs from Epidemic Cascades

Jessica Hoffmann, Soumya Basu
Surbhi Goel, Constantine Caramanis

ICML 2020

Motivation

Why is the problem important?

- Mixtures are everywhere. For instance:
  - multiple strains of diseases
  - someone tweeting about both politics and football, writing “We won!”

- From a theory perspective, new and exciting: even learning mixture of two Gaussians/mixed regression is a hard problem with recent progress [1,2]

[1] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. List-decodable robust mean estimation and learning mixtures of spherical gaussians.

[2] Xinyang Yi, Constantine Caramanis, and Sujay Sanghavi. Alternating minimization for mixed linear regression

Motivation

Why is it hard?

- If we only had one graph, weight p_{ij} between nodes *i* and *j*:

$$\hat{p}_{ij} = \frac{\#\text{i infected j}}{\#\text{i could have infected j}} \rightarrow \frac{\mathbb{P}(\text{i could have infected j}) \cdot p_{ij}}{\mathbb{P}(\text{i could have infected j})} = p_{ij}$$

Motivation

Why is it hard?

- If we only had one graph, weight p_{ij} between nodes *i* and *j*:

$$\hat{p}_{ij} = \frac{\#\text{i infected j}}{\#\text{i could have infected j}} \rightarrow \frac{\cancel{\mathbb{P}(\text{i could have infected j})} \cdot p_{ij}}{\cancel{\mathbb{P}(\text{i could have infected j})}} = p_{ij}$$

Motivation

Why is it hard?

- With a mixture, weight p_{ij} in graph 1 and q_{ij} in graph 2:

$$\frac{\#\text{i infected j}}{\#\text{i could have infected j}} \to \frac{\mathbb{P}(\text{i could have infected j} \mid \text{graph 1}) \cdot p_{ij} + \mathbb{P}(\text{i could have infected j} \mid \text{graph 2}) \cdot q_{ij}}{\mathbb{P}(\text{i could have infected j} \mid \text{graph 1}) + \mathbb{P}(\text{i could have infected j} \mid \text{graph 2})}$$

Motivation

Why is it hard?

- With a mixture, weight p_{ij} in graph 1 and q_{ij} in graph 2:

$$\frac{\#\text{i infected j}}{\#\text{i could have infected j}} \rightarrow \frac{\overbrace{\mathbb{P}(\text{i could have infected j} \mid \text{graph 1})}\cdot p_{ij}+\overbrace{\mathbb{P}(\text{i could have infected j} \mid \text{graph 2})}\cdot q_{ij}}{\underbrace{\mathbb{P}(\text{i could have infected j} \mid \text{graph 1})+\mathbb{P}(\text{i could have infected j} \mid \text{graph 2})}}$$

Unknown and don’t cancel out

Motivation

Why is it hard?

- With a mixture, weight p_{ij} in graph 1 and q_{ij} in graph 2:

$$\frac{\#\text{i infected j}}{\#\text{i could have infected j}} \to \frac{\mathbb{P}(\text{i could have infected j} \mid \text{graph 1}) \cdot p_{ij} + \mathbb{P}(\text{i could have infected j} \mid \text{graph 2}) \cdot q_{ij}}{\mathbb{P}(\text{i could have infected j} \mid \text{graph 1}) + \mathbb{P}(\text{i could have infected j} \mid \text{graph 2})}$$

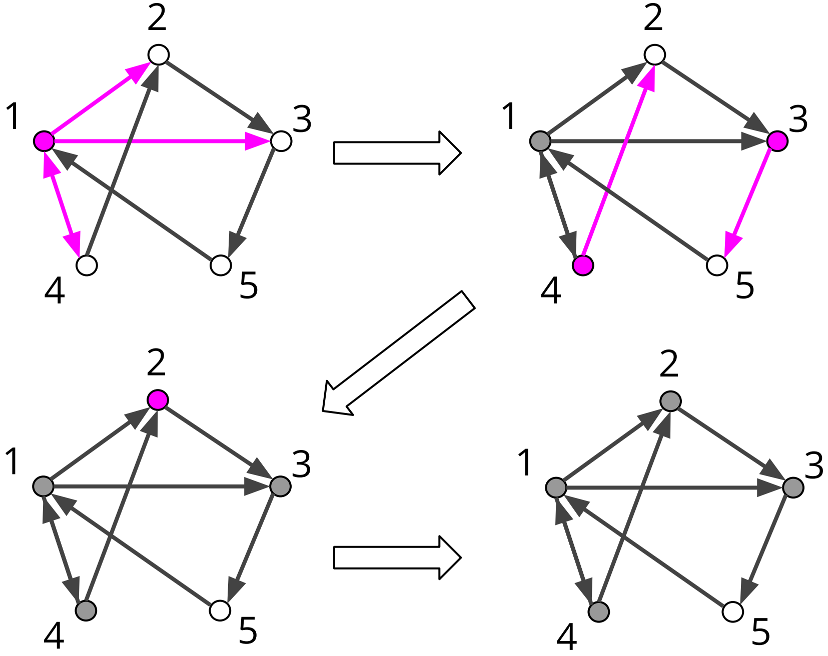
→ **No simple estimator.**

Lists of infections as samples

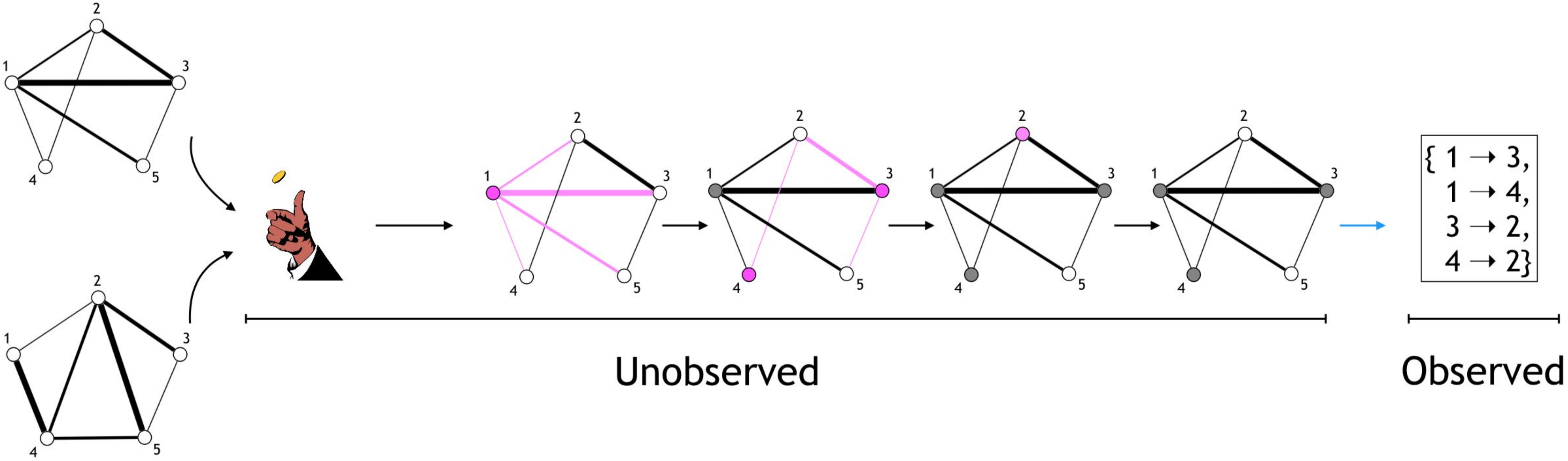
$$\begin{bmatrix} 1 \rightarrow 3 \\ 1 \rightarrow 4 \\ 4 \rightarrow 2 \end{bmatrix}$$

↑

One sample



What if we have two graphs?



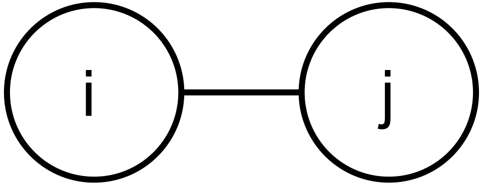
{ 1 → 3,
1 → 4,
3 → 2,
4 → 2}

Unobserved

Observed

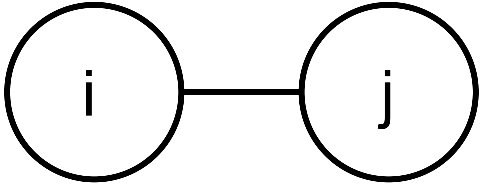
?

Some examples



Some examples

$$p_{ij} = \beta,\ q_{ij} = 1 - \beta$$



$$\mathbb{P}(i) = \frac{1}{2} \cdot \frac{1}{2} \cdot (1 - p_{ij}) + \frac{1}{2} \cdot \frac{1}{2} \cdot (1 - q_{ij}) = 1/4$$

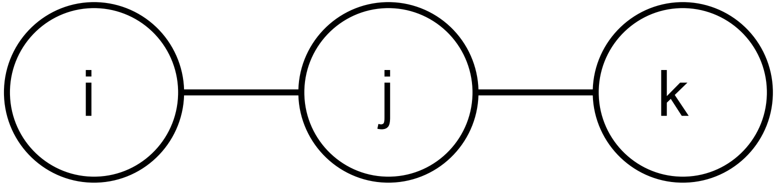
$$\mathbb{P}(j) = \frac{1}{2} \cdot \frac{1}{2} \cdot (1 - p_{ji}) + \frac{1}{2} \cdot \frac{1}{2} \cdot (1 - q_{ji}) = 1/4$$

$$\mathbb{P}(i \to j) = \frac{1}{2} \cdot \frac{1}{2} \cdot p_{ij} + \frac{1}{2} \cdot \frac{1}{2} \cdot q_{ij} = 1/4$$

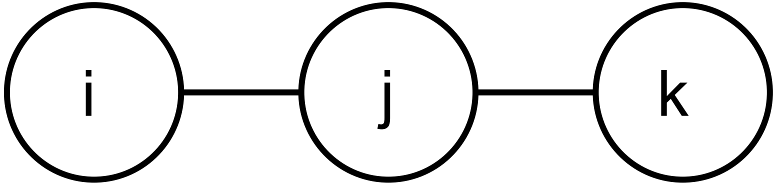
$$\mathbb{P}(j \to i) = \frac{1}{2} \cdot \frac{1}{2} \cdot p_{ji} + \frac{1}{2} \cdot \frac{1}{2} \cdot q_{ji} = 1/4$$

unsolvable

Some examples

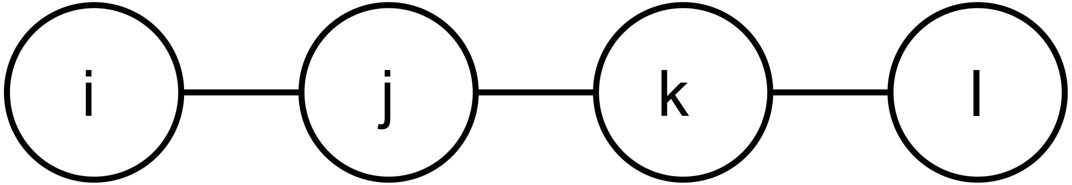


Some examples

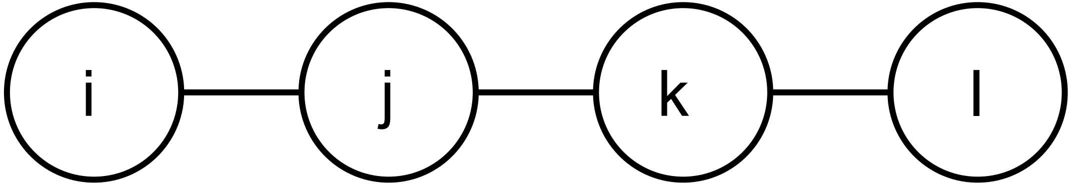


Still unsolvable

Some examples

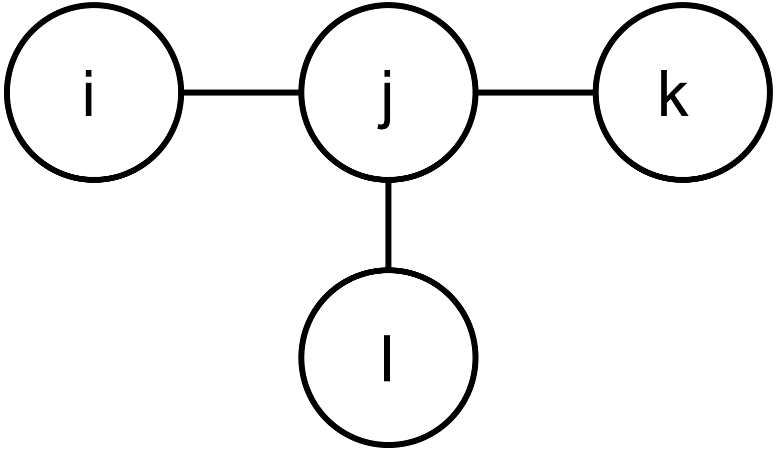


Some examples

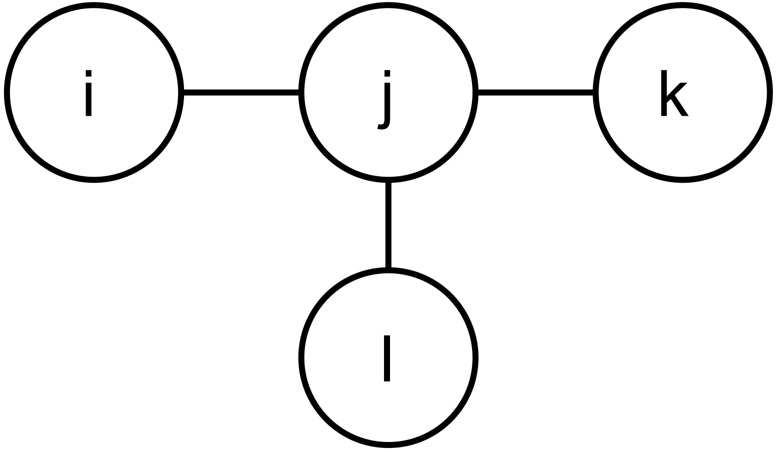


Solvable!

Some examples



Some examples



Also solvable!

Which mixtures are learnable?

We can learn all the edges up to precision ϵ in polynomial time **if and only if**:

1. The union of edges is connected

2. At least 3 edges

3. $\exists \Delta > 0, \forall i, j \in E_1 \cup E_2, \ |p_{ij} - q_{ij}| > \Delta$

Mixture *vs* one graph

We know the list of infections $\{i \to j,\ i \to k,\ j \to l,\ \ldots\}$

Let I_t^m (resp. S_t^m) be the set of infected (resp. susceptible) nodes during cascade *m* at time *t*.

- One graph:

$$\hat{p}_{ua} = \frac{\sum_{m=1}^{M} \mathbf{1}_{u \to a}}{\sum_{m=1}^{M} \sum_{t=0}^{N} \mathbf{1}_{u \in I_t^m,\ a \in S_t^m}} \to_{M \to \infty} \frac{\sum_{t=0}^{N} \mathbb{P}(u \in I_t^m,\ a \in S_t^m) \cdot p_{ua}}{\sum_{t=0}^{N} \mathbb{P}(u \in I_t^m,\ a \in S_t^m)} = p_{ua}$$

- Two graphs:

$$\to_{M \to \infty} \frac{\sum_{t=0}^{N} \mathbb{P}(u \in I_t^m,\ a \in S_t^m | E_1) \cdot p_{ua} + \sum_{t=0}^{N} \mathbb{P}(u \in I_t^m,\ a \in S_t^m | E_2) \cdot q_{ua}}{\sum_{t=0}^{N} \mathbb{P}(u \in I_t^m,\ a \in S_t^m | E_1) + \sum_{t=0}^{N} \mathbb{P}(u \in I_t^m,\ a \in S_t^m | E_2)}$$

Mixture *vs* one graph

- Simple estimator of edges weights does NOT work
- Complex terms do not cancel out anymore
- Computing probability of u being infected while a is susceptible is almost as hard as solving the mixture problem
- All estimators involve BOTH p_{ua} and q_{ua}
- Two graphs: $\rightarrow_{M\rightarrow\infty} \dfrac{\sum_{t=0}^{N}\mathbb{P}(u\in I_t^m,\, a\in S_t^m|E_1)\cdot p_{ua}+\sum_{t=0}^{N}\mathbb{P}(u\in I_t^m,\, a\in S_t^m|E_2)\cdot q_{ua}}{\sum_{t=0}^{N}\mathbb{P}(u\in I_t^m,\, a\in S_t^m|E_1)+\sum_{t=0}^{N}\mathbb{P}(u\in I_t^m,\, a\in S_t^m|E_2)}$

Mixture *vs* one graph

Now, what?

Learning edges of $E_1 \cup E_2$

First, we learn the edges of the **union of the mixtures**:

- For each pair of nodes *u* and *a*, we calculate the fraction of times *u* infected *a* knowing ***u* was the source** of the cascade:

$$\hat{X}_{ua} := \frac{\sum_{m=1}^{M} 1_{u \to a, u \in I_0^m}}{\sum_{m=1}^{M} 1_{u \in I_0^m}}$$

- *u* is the source with probability 1/N in **both** mixtures, so it cancels out.

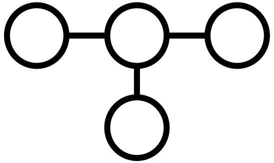
$$\hat{X}_{ua} \to_{M \to \infty} \mathbf{Pr}(u \to a \mid u \in I_0) = \frac{p_{ua} + q_{ua}}{2} \geq \frac{p_{min}}{2}$$

- Simple test $\hat{X}_{ua} > \frac{p_{min}}{4}$ can decide which edges are in the union.

General algorithm

1. We can find the edges of the union of the mixture

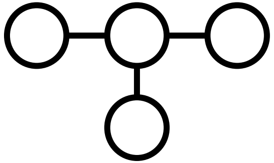
2. We can calculate the edge weights for nodes of degree > 2



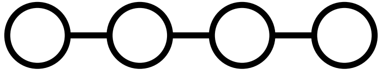
General algorithm

1. We can find the edges of the union of the mixture

2. We can calculate the edge weights for nodes of degree > 2



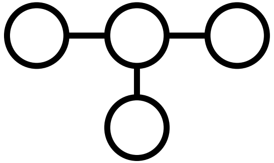
3. Similarly, we can calculate the edge weights for nodes of degree 2



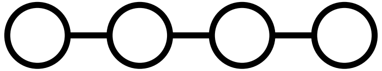
General algorithm

1. We can find the edges of the union of the mixture.

2. We can calculate the edge weights for nodes of degree > 2.



3. Similarly, we can calculate the edge weights for nodes of degree 2.



4. Edges are already learned for nodes of degree 1.

→ We can add the nodes one by one

Solution for the "star" structure, 1/2

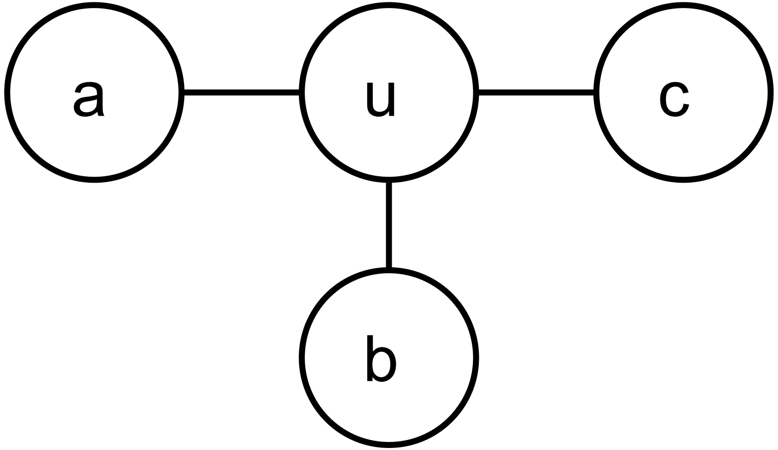
We find *u* with degree > 2.

We use **second moment**:

$$\hat{Y}_{ua,ub} = \frac{\sum_{m=1}^{M} \mathbf{1}_{u\to a,\, u\to b}}{\sum_{m=1}^{M} \mathbf{1}_{u\in I_0^m}}$$

$$\to_{M\to\infty} \Pr(u \to a, u \to b \mid u \in I_0)$$

$$= \frac{p_{ua}p_{ub} + q_{ua}q_{ub}}{2}$$



Solution for the “star” structure, 2/2

We have six unknowns: $p_{ua}, p_{ub}, p_{uc}, q_{ua}, q_{ub}, q_{uc}$

And six 1st and 2nd moment estimators: $\hat{X}_{ua}, \hat{X}_{ub}, \hat{X}_{uc}, \hat{Y}_{ua,\, ub}, \hat{Y}_{ua,\, uc}, \hat{Y}_{ub,\, uc}$

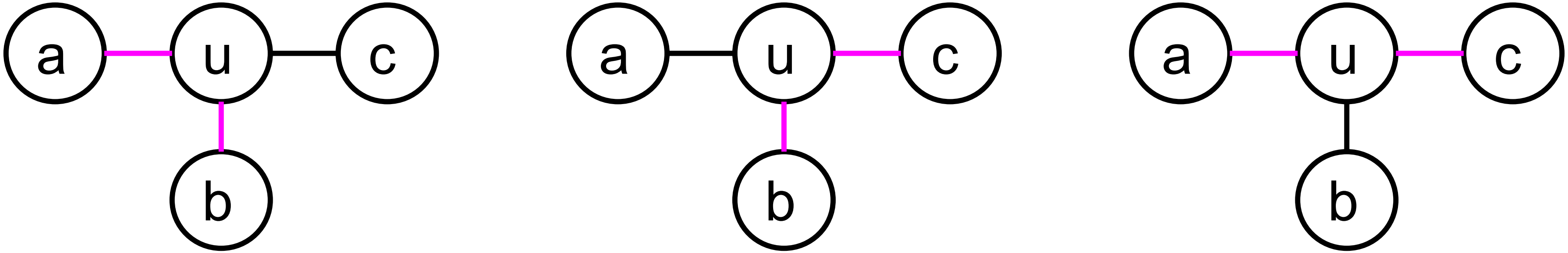
Systems of polynomial equations are hard to solve in general. Here, we find a closed-form solution:

$$p_{ua} = X_{ua} + s_{ua}\sqrt{\frac{(Y_{ua,ub} - X_{ua}X_{ub})(Y_{ua,uc} - X_{ua}X_{uc})}{Y_{ub,uc} - X_{ub}X_{uc}}},$$

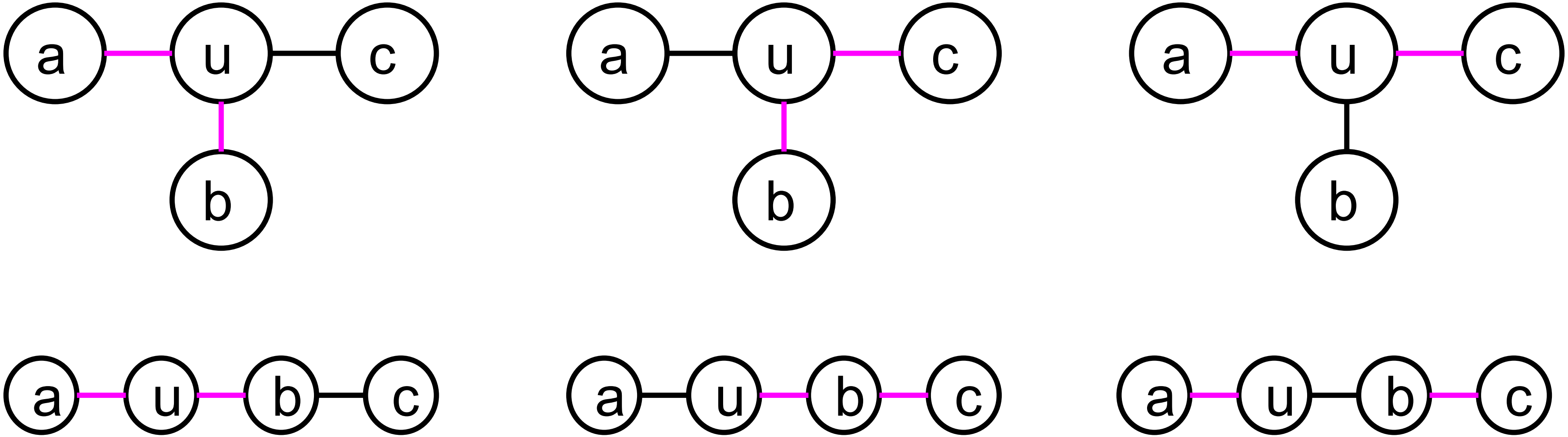
$$q_{ua} = X_{ua} - s_{ua}\sqrt{\frac{(Y_{ua,ub} - X_{ua}X_{ub})(Y_{ua,uc} - X_{ua}X_{uc})}{Y_{ub,uc} - X_{ub}X_{uc}}}.$$

$$s_{ua}s_{ub} = \mathsf{sgn}(Y_{ua,ub} - X_{ua}X_{ub})$$

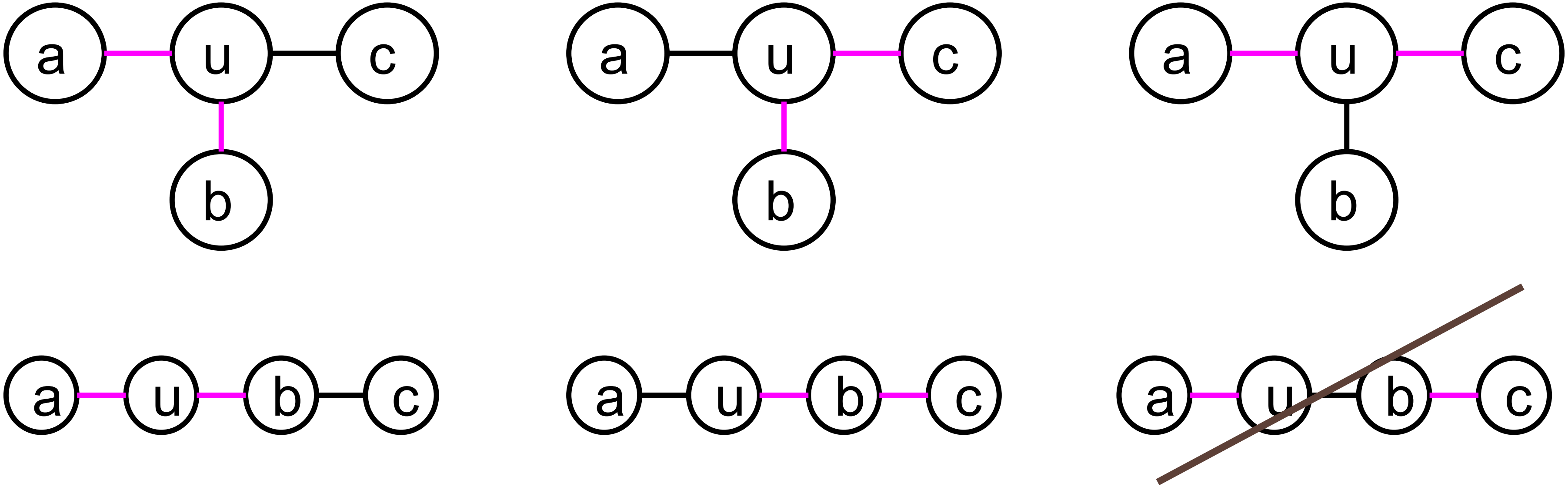
Issue with line graph



Issue with line graph



Issue with line graph



Impossible

Line graph: solution

We use 3rd moment: $Z_{ua,ub,bc}$

$$\boldsymbol{p_{ua} = X_{ua} + s_{ua}}\sqrt{\frac{(Y_{ua,ub}-X_{ua}X_{ub})\left(X_{ua}X_{bc}+\frac{Z_{ua,ub,bc}-X_{ua}Y_{ub,bc}-X_{bc}Y_{ua,ub}}{X_{ub}}\right)}{Y_{ub,bc}-X_{ub}X_{bc}}}$$

$$\boldsymbol{q_{ua} = X_{ua} - s_{ua}}\sqrt{\frac{(Y_{ua,ub}-X_{ua}X_{ub})\left(X_{ua}X_{bc}+\frac{Z_{ua,ub,bc}-X_{ua}Y_{ub,bc}-X_{bc}Y_{ua,ub}}{X_{ub}}\right)}{Y_{ub,bc}-X_{ub}X_{bc}}}$$

$s_{ua}s_{ub} = \mathsf{sgn}(Y_{ua,ub} - X_{ua}X_{ub})$

Sample complexity and optimal bounds

	Our algorithm	Lower bound
Undirected graphs	$O\left(\frac{N}{\epsilon^2 \cdot \Delta^4} \log(\frac{N}{\delta})\right)$	$\Omega\left(\frac{N}{\Delta^2}\right)$
Directed graph, min-degree > 2	$O\left(\frac{N}{\epsilon^2 \cdot \Delta^2} \log(\frac{N}{\delta})\right)$	$\Omega\left(N \log(N) + \frac{N \log\log(N)}{\Delta^2}\right)$

Conclusion

- We provided **necessary and sufficient conditions** for learning mixtures of two graphs up to any precision.

- Our algorithm is **sample-optimal** (up to log factors).

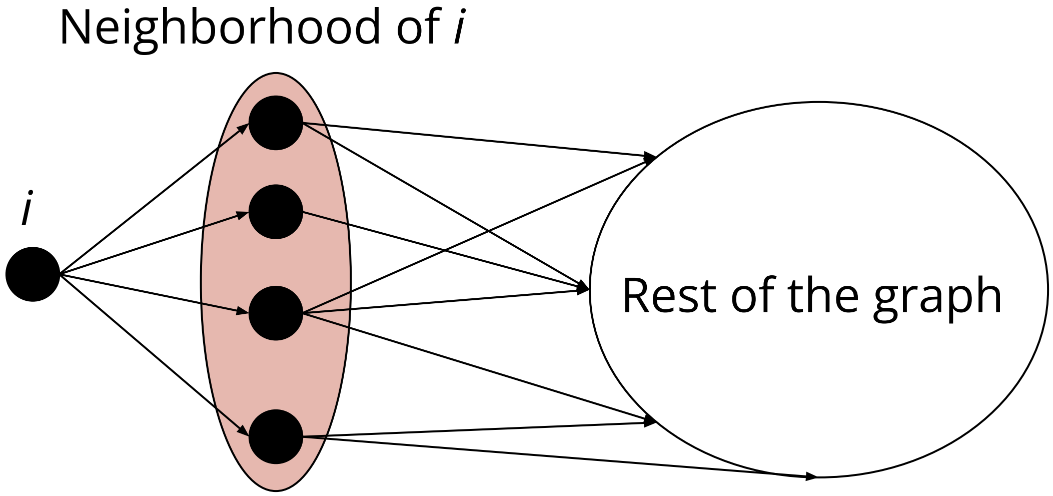
- Our results can be **extended to directed graphs** if min-degree > 2, and unbalanced mixtures.

- Easily **parallelizable**.

Thank you!

Extreme-noise setting - case of bounded degree

1) Co-infection between a node *i* and a set *S*



2) The neighborhood of *i* is the set *S* of largest co-infection, and smallest size