Data science for networked data

Po-Ling Loh

University of Wisconsin-Madison
Department of Statistics

AISTATS
Okinawa, Japan
April 16, 2019

Joint work with:

Justin Khim (UPenn), Varun Jog (UW-Madison), Ashley Hou (UW-Madison), Wen Yan (Southeast University), and Muni Pydi (UW-Madison)

Key problems in network modeling

1. Given data from a network, how do we estimate the network?
2. How do we model dynamic processes over a network?
3. How do we perform efficient search over a network?

Key problems in network modeling

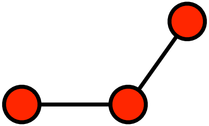
1. Given data from a network, how do we estimate the network?
2. How do we model dynamic processes over a network?
3. How do we perform efficient search over a network?

Key problems in network modeling

1. Given data from a network, how do we estimate the network?
2. How do we model dynamic processes over a network?
3. How do we perform efficient search over a network?

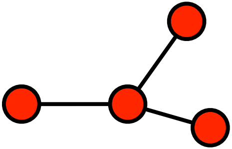
Key problems in network modeling

1. Given data from a network, how do we estimate the network?
2. How do we model dynamic processes over a network?
3. How do we perform efficient search over a network?



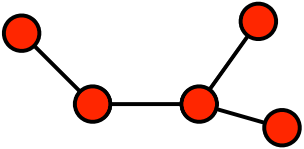
Key problems in network modeling

1. Given data from a network, how do we estimate the network?
2. How do we model dynamic processes over a network?
3. How do we perform efficient search over a network?



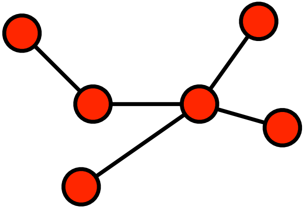
Key problems in network modeling

1. Given data from a network, how do we estimate the network?
2. How do we model dynamic processes over a network?
3. How do we perform efficient search over a network?



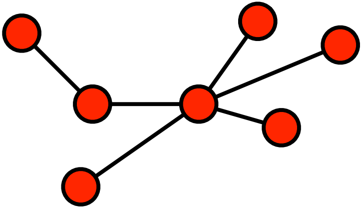
Key problems in network modeling

1. Given data from a network, how do we estimate the network?
2. How do we model dynamic processes over a network?
3. How do we perform efficient search over a network?



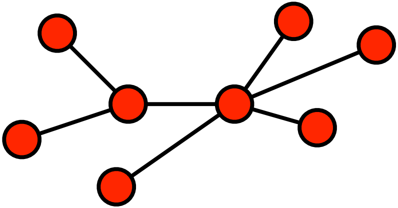
Key problems in network modeling

1. Given data from a network, how do we estimate the network?
2. How do we model dynamic processes over a network?
3. How do we perform efficient search over a network?



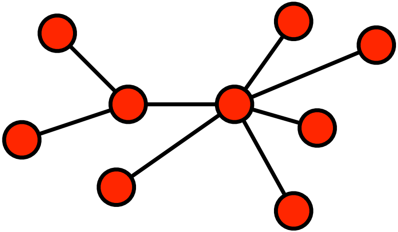
Key problems in network modeling

1. Given data from a network, how do we estimate the network?
2. How do we model dynamic processes over a network?
3. How do we perform efficient search over a network?



Key problems in network modeling

1. Given data from a network, how do we estimate the network?
2. How do we model dynamic processes over a network?
3. How do we perform efficient search over a network?



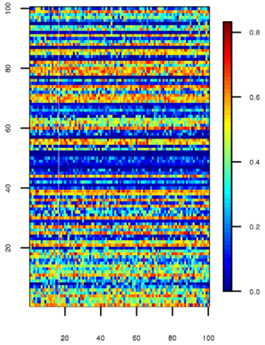
Prelude: Network estimation

Graphical models

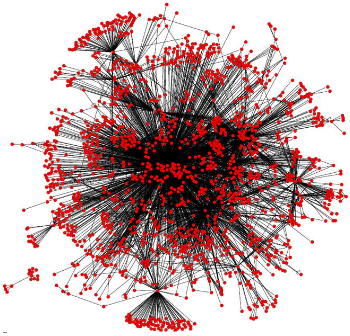
- Method for constructing connectivity network from matrix of data

Graphical models

- Method for constructing connectivity network from matrix of data



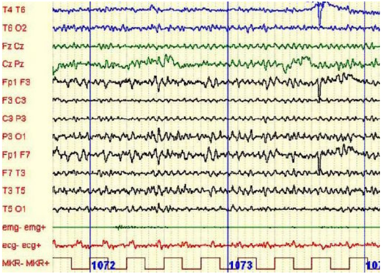
gene expression (mRNA) data



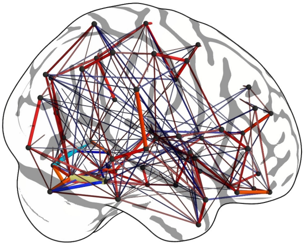
E. coli network

Graphical models

- Method for constructing connectivity network from matrix of data



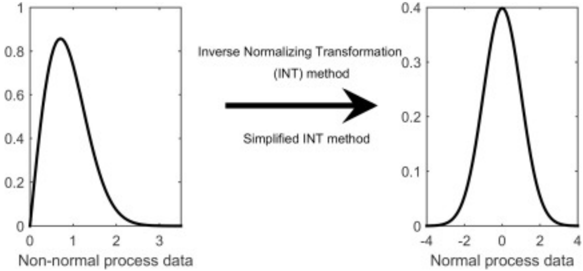
fMRI/EEG readings



"functional connectivity" network

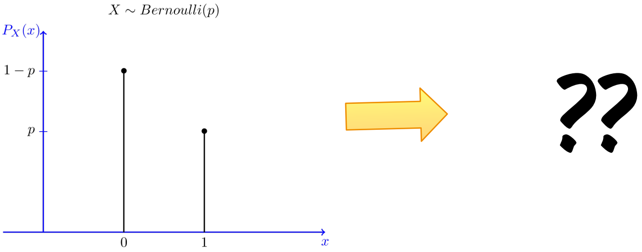
Graphical models

- Mathematical analysis derived for Gaussian data



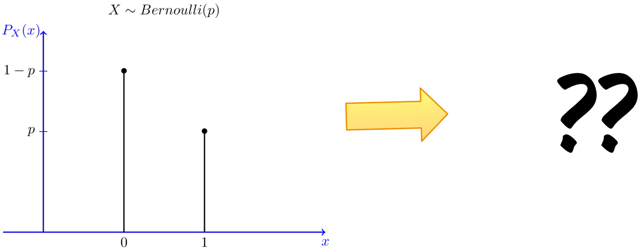
- In practice, transform data to Gaussian before applying algorithm

- But not all data are transformable!



Graphical models

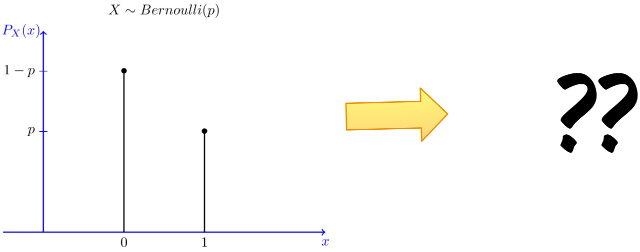
- But not all data are transformable!



- We have developed new methods for estimating graphical models for discrete (count) data

Graphical models

- But not all data are transformable!



- We have developed new methods for estimating graphical models for discrete (count) data
- **However, life is more than network estimation. . .**

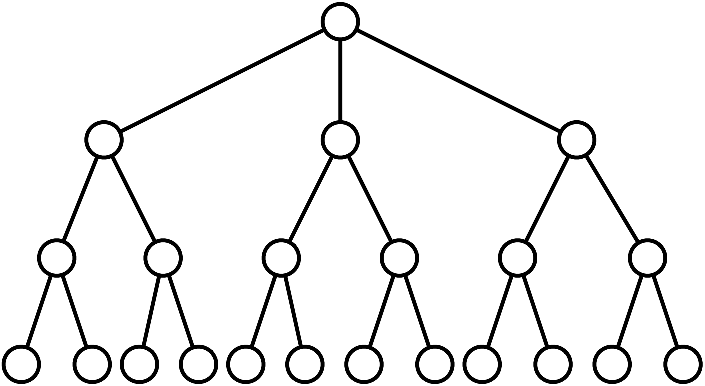
Outline

1. Statistical inference
   - Confidence sets for source estimation
   - Graph hypothesis testing
2. Resource allocation
   - Influence maximization
   - Budget allocation
   - Network immunization
3. Local algorithms

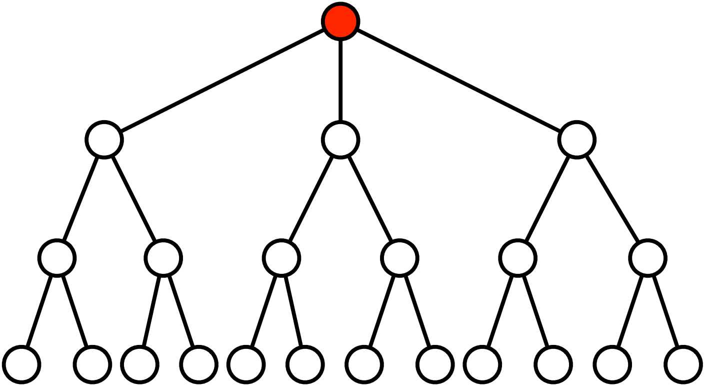
Statistical inference

Justin Khim
(UPenn)

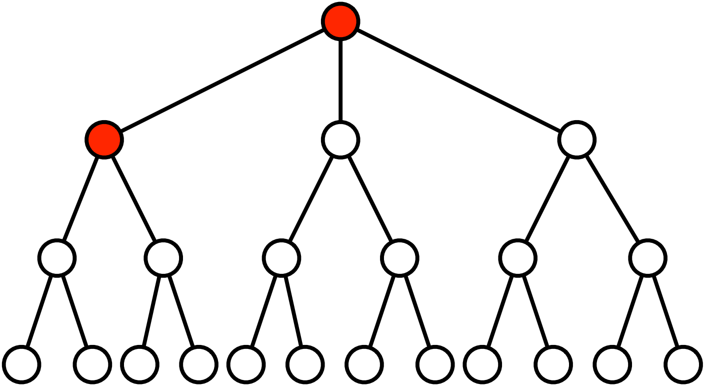
Source estimation



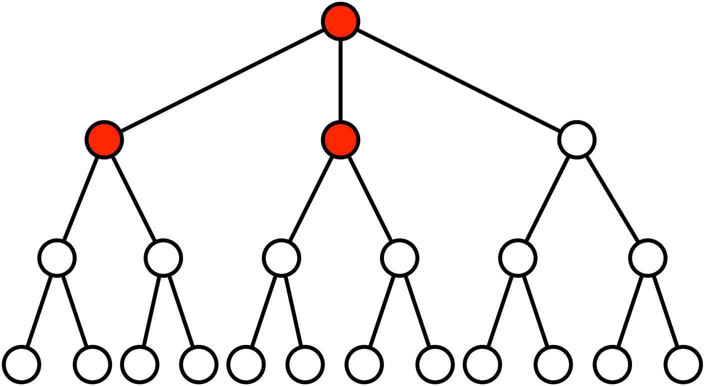
Source estimation



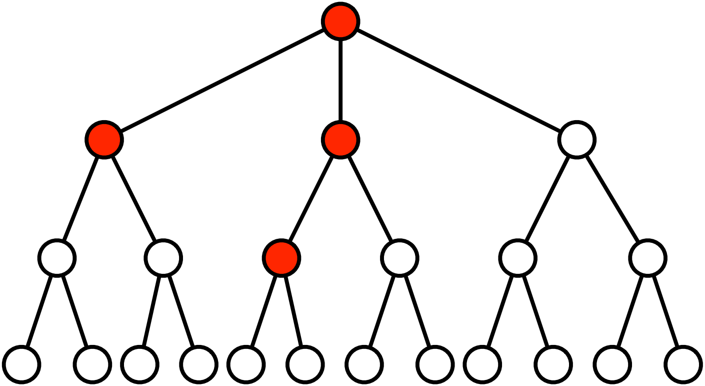
Source estimation



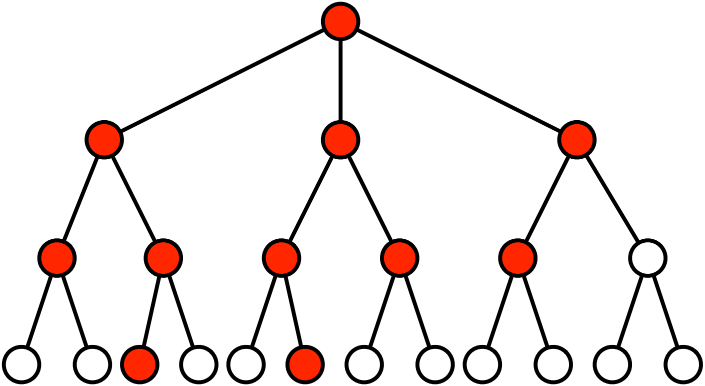
Source estimation



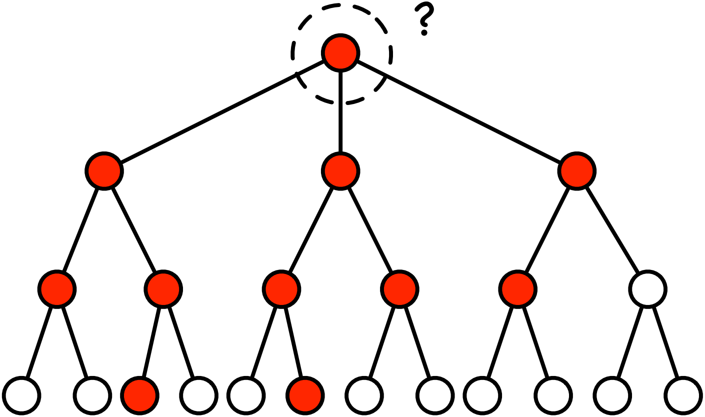
Source estimation



Source estimation

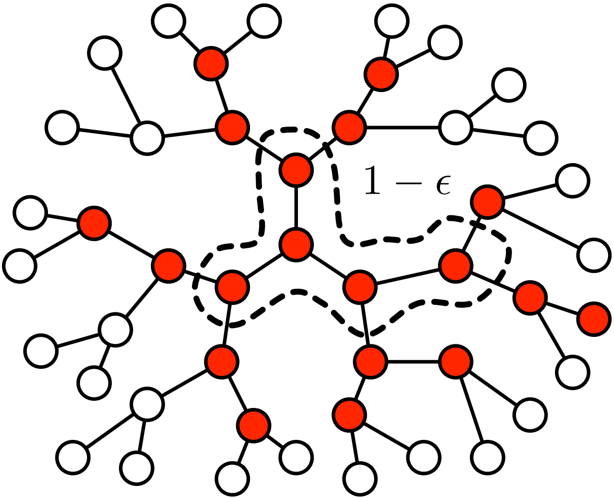


Source estimation



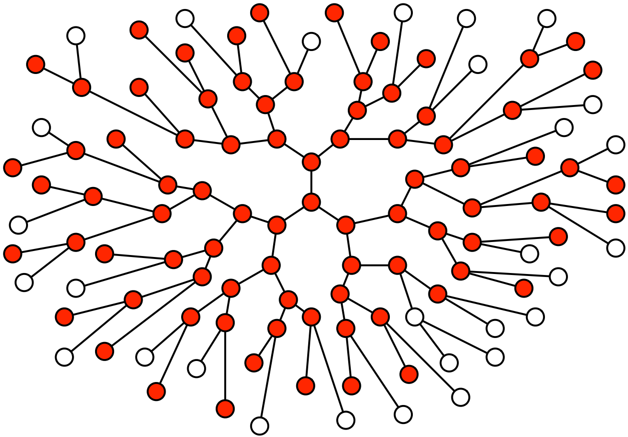
Confidence sets

- **Instead:** Find a *confidence set* that includes root node with probability at least $1 - \epsilon$



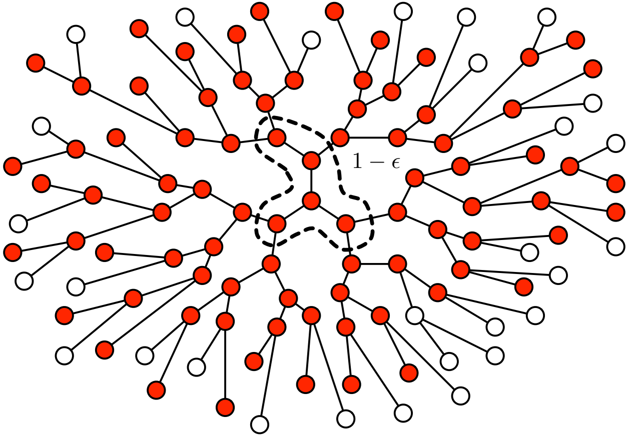
Confidence sets

- **Question:** How does size of confidence set grow with number of infected nodes n?



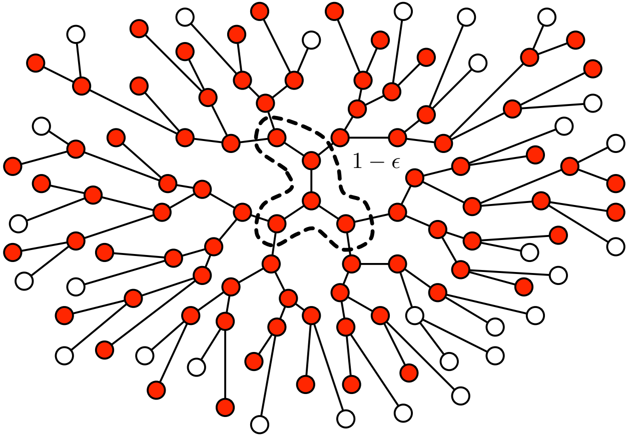
Confidence sets

- **It doesn't!**



Confidence sets

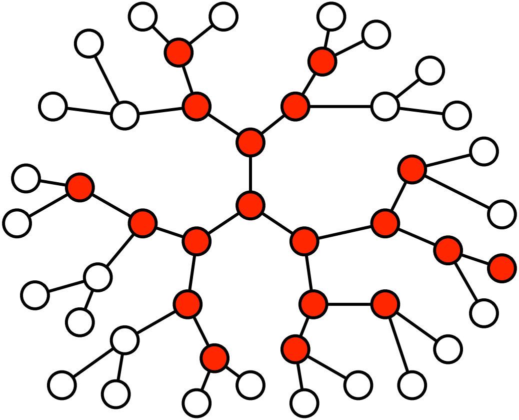
- **It doesn't!**



- **Rough interpretation:** No "information loss" about source as disease spreads

Inference algorithm

- Select nodes that are most "central" to network of infected individuals

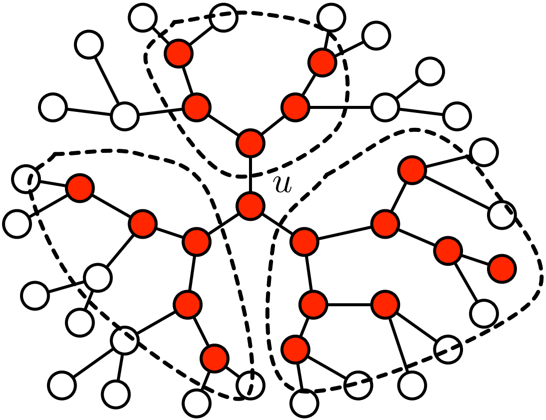


Inference algorithm

- For each node, compute "min-max subtree size"

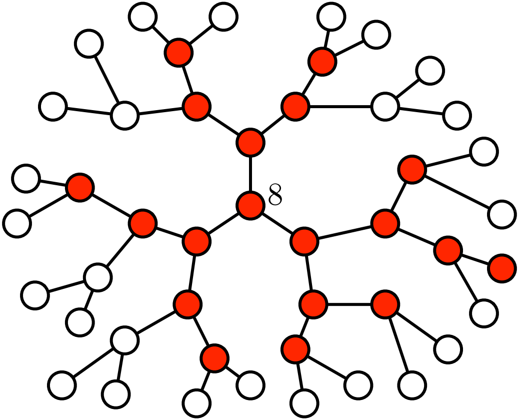
Inference algorithm

- For each node, compute "min-max subtree size"



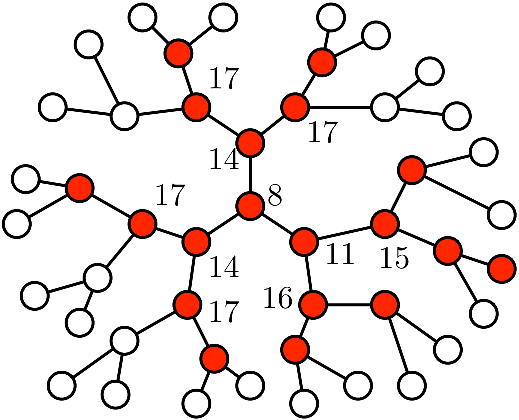
Inference algorithm

- For each node, compute "min-max subtree size"



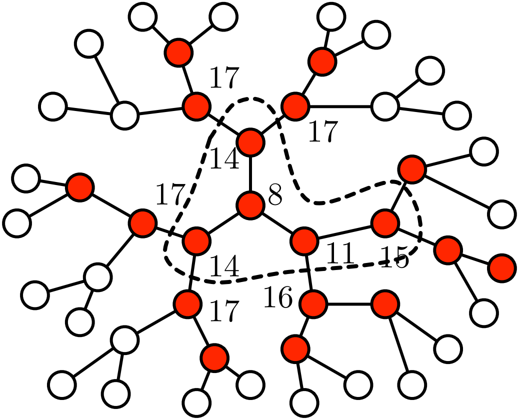
Inference algorithm

- For each node, compute “min-max subtree size”



Inference algorithm

- Select $K(\epsilon)$ nodes with smallest values



Theory for confidence sets

Theorem

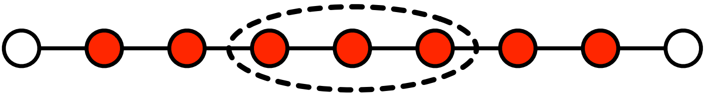
Suppose $d \geq 3$. Then the min-max subtree estimator with $K_\psi(\epsilon) = \frac{C(d)}{\epsilon}$ yields a $1 - \epsilon$ confidence set for the root.

Theory for confidence sets

Theorem

Suppose $d \geq 3$. Then the min-max subtree estimator with $K_\psi(\epsilon) = \frac{C(d)}{\epsilon}$ yields a $1 - \epsilon$ confidence set for the root.

- **Note:** Cannot construct finite confidence set for $d = 2$; need set of size $K = \Theta(\sqrt{n})$



Extensions and open directions

- Similar result holds for broader class of “regular” trees
- **Robustness:** Confidence set eventually settles down after finitely many steps

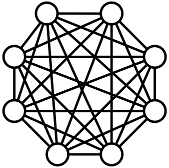
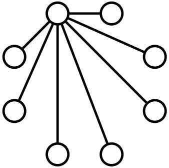
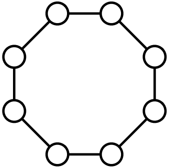
Extensions and open directions

- Similar result holds for broader class of "regular" trees
- **Robustness:** Confidence set eventually settles down after finitely many steps

Open directions:

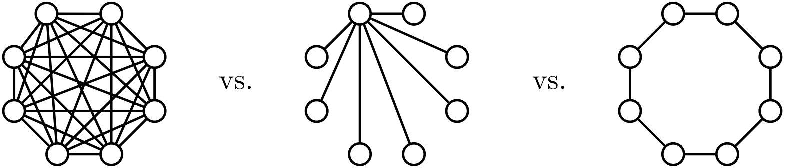
- What if underlying graph is not a tree?
- What if network is asymmetric?
- What if nodes can heal?

Graph testing

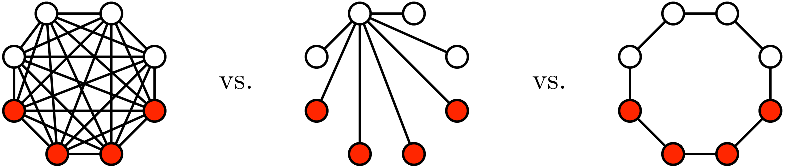
 vs.  vs. 

- **Question:** Can we use epidemic data to infer network structure?

Graph testing

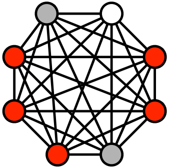


- **Question:** Can we use epidemic data to infer network structure?

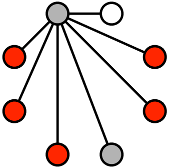


Graph testing

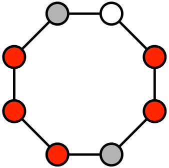
- **Observations:** Infection status of n nodes in graph
  - k infected nodes (1)
  - c censored (nonreporting) nodes ($\star$)
  - $n - k - c$ uninfected nodes (0)



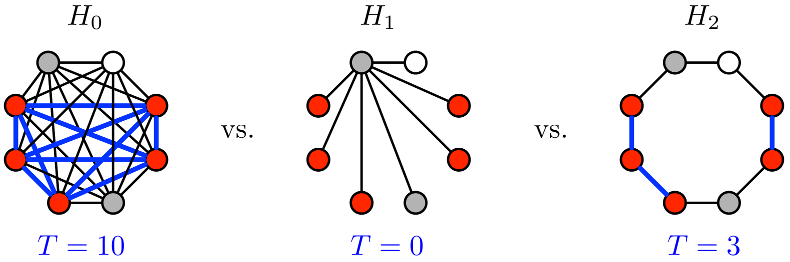
vs.



vs.



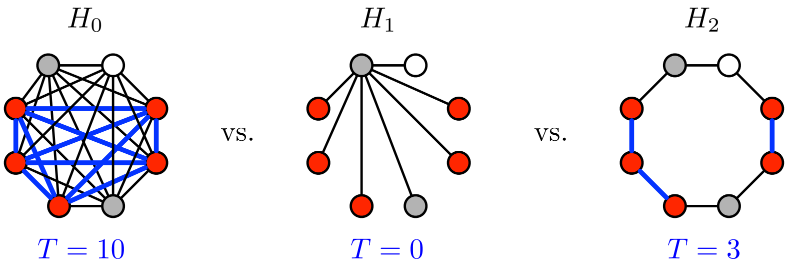
Graph testing



- Compute test statistic

$$T = \# \text{ edges between infected nodes}$$

Graph testing



- Compute test statistic

$$T = \#\text{ edges between infected nodes}$$

- Need to construct proper rejection rule based on T, derive validity of hypothesis test

Infection model

- Parameters λ, η
  - For each node v, generate $T_v \sim Exp(\lambda)$
  - For each edge (u, v), generate $T_{uv} \sim Exp(\eta)$
- Infection time of any vertex v is $t_v = \min_{u \in N(v)}\{t_u + T_{uv}\} \wedge T_v$

Infection model

- Parameters λ, η
  - For each node v, generate $T_v \sim Exp(\lambda)$
  - For each edge (u, v), generate $T_{uv} \sim Exp(\eta)$
- Infection time of any vertex v is $t_v = \min_{u \in N(v)} \{t_u + T_{uv}\} \wedge T_v$
- Observation vector corresponds to infection states at a certain time
- Subset of censored nodes chosen uniformly at random

Permutation test

- **Goal:** For $\alpha \in (0,1)$, construct rejection rule such that

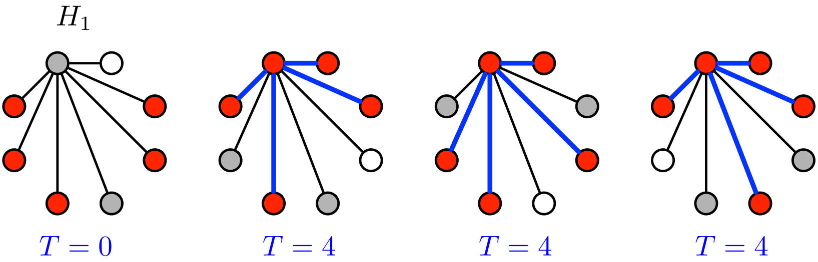
$$P(\text{reject} \mid H_0 \text{ is true}) \leq \alpha$$

Permutation test

- **Goal:** For $\alpha \in (0,1)$, construct rejection rule such that

$$P(\text{reject} \mid H_0 \text{ is true}) \leq \alpha$$

- Use *permutation test* that computes T for $\binom{n}{k, c, n-k-c}$ reassignments of infected/nonreporting/uninfected nodes

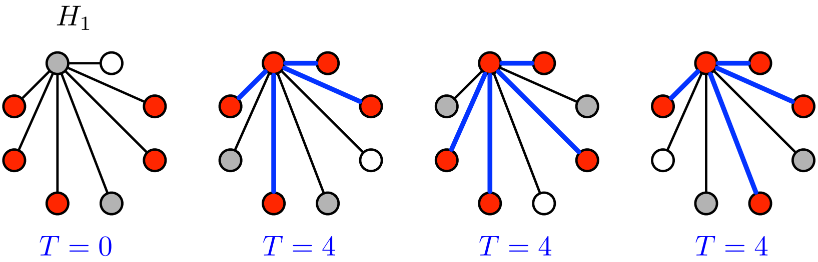


Permutation test

- **Goal:** For $\alpha \in (0, 1)$, construct rejection rule such that

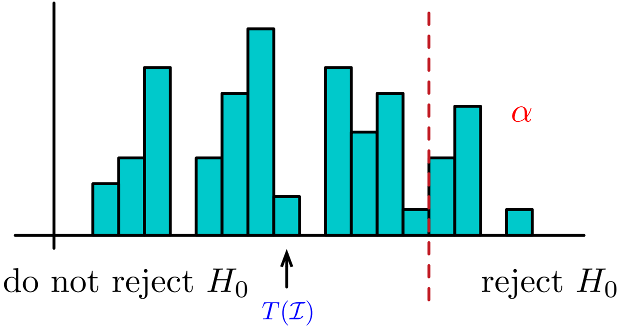
$$P(\text{reject} \mid H_0 \text{ is true}) \le \alpha$$

- Use *permutation test* that computes T for $\binom{n}{k, c, n-k-c}$ reassignments of infected/nonreporting/uninfected nodes

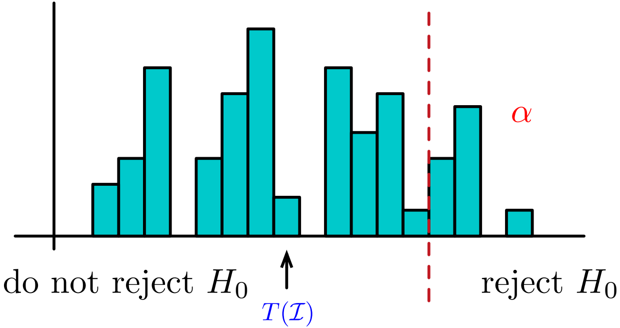


- Based on (randomly chosen) permutations, compute p-value/rejection region and reject H_0 if (p-value of T) $\le \alpha$

Permutation test



Permutation test



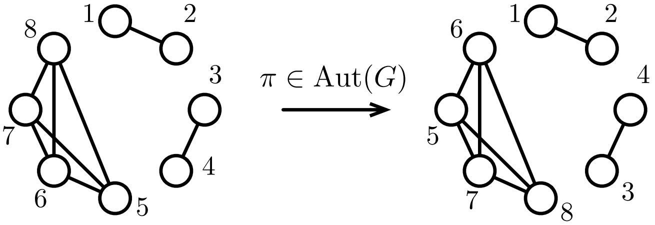
- In practice, sufficient to compute empirical distribution from large number of random permutations

Theory for permutation test

- Success depends on *symmetries* of underlying networks rather than parameters λ, η
- Consider $\Pi_0 = \mathrm{Aut}(G_0)$ and $\Pi_1 = \mathrm{Aut}(G_1)$, subsets of S_n

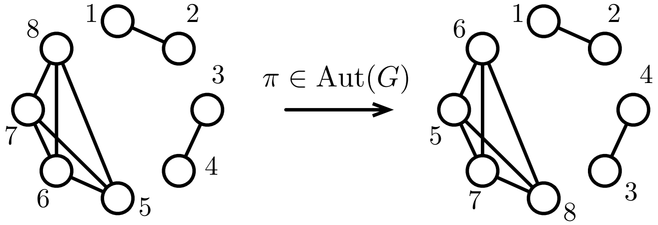
Theory for permutation test

- Success depends on *symmetries* of underlying networks rather than parameters λ, η
- Consider $\Pi_0 = \mathrm{Aut}(G_0)$ and $\Pi_1 = \mathrm{Aut}(G_1)$, subsets of S_n



Theory for permutation test

- Success depends on *symmetries* of underlying networks rather than parameters λ, η
- Consider $\Pi_0 = \mathrm{Aut}(G_0)$ and $\Pi_1 = \mathrm{Aut}(G_1)$, subsets of S_n



Theorem

Let π be drawn uniformly from S_n. If $\Pi_1 \Pi_0 = S_n$, the permutation test controls Type I error at level α.

Extensions and open directions

- Characterization of condition $\Pi_1 \Pi_0 = S_n$ for various graph families
- Bounds on Type II error for specific graphs
- Conditioning on identity of censored nodes

Extensions and open directions

- Characterization of condition $\Pi_1 \Pi_0 = S_n$ for various graph families
- Bounds on Type II error for specific graphs
- Conditioning on identity of censored nodes

Open directions:

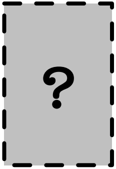
- How to identify which graphs to use as null/alternative hypotheses?
- Inhomogeneous λ and η?
- Confidence sets for underlying network?

Resource allocation

Justin Khim
(UPenn)

Varun Jog
(UW-Madison)

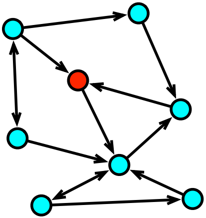
Ashley Hou
(UW-Madison)



Wen Yan
(Southeast University)

Influence maximization (with Justin Khim and Varun Jog)

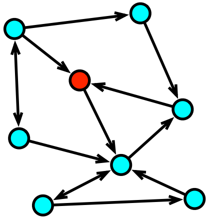
- **New goal:** Seed a network to "infect" as many nodes as possible
- Useful for information dissemination, marketing, etc.



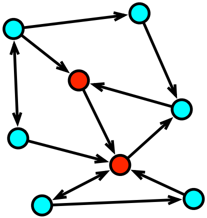
t = 0

Influence maximization (with Justin Khim and Varun Jog)

- **New goal:** Seed a network to "infect" as many nodes as possible
- Useful for information dissemination, marketing, etc.



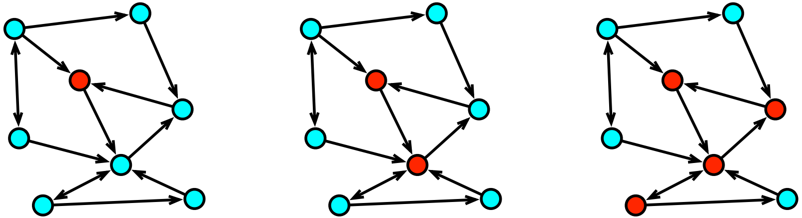
$t = 0$



$t = 1$

Influence maximization (with Justin Khim and Varun Jog)

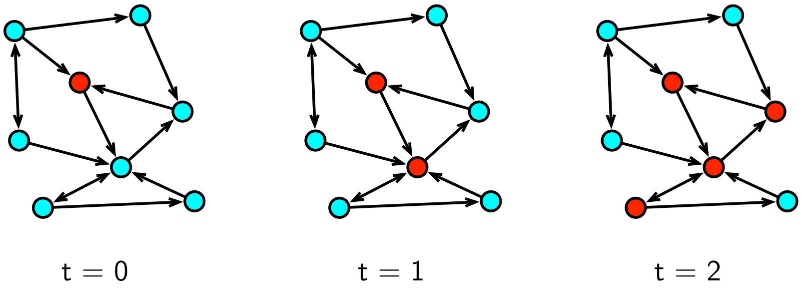
- **New goal:** Seed a network to "infect" as many nodes as possible
- Useful for information dissemination, marketing, etc.



$t = 0$ $t = 1$ $t = 2$

Influence maximization (with Justin Khim and Varun Jog)

- **New goal:** Seed a network to "infect" as many nodes as possible
- Useful for information dissemination, marketing, etc.

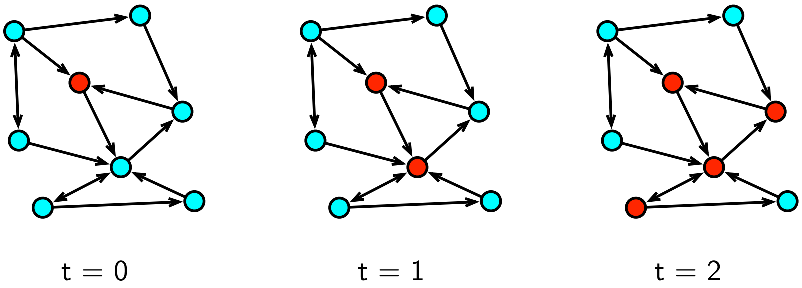


Questions

1. If k nodes may be infected initially, which nodes should be selected to maximize infection spread?

Influence maximization (with Justin Khim and Varun Jog)

- **New goal:** Seed a network to "infect" as many nodes as possible
- Useful for information dissemination, marketing, etc.



Questions

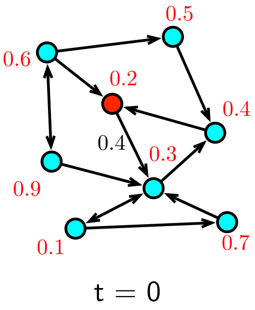
1. If k nodes may be infected initially, which nodes should be selected to maximize infection spread?
2. How to determine maximal set efficiently?

Model: Linear threshold model (broadly, triggering models)

- Edges have weights (b_{ij}), satisfying $\sum_j b_{ji} \leq 1$
- Nodes choose thresholds $\theta_i \in [0, 1]$ i.i.d., uniformly at random

Model: Linear threshold model (broadly, triggering models)

- Edges have weights (b_{ij}), satisfying $\sum_j b_{ji} \leq 1$
- Nodes choose thresholds $\theta_i \in [0, 1]$ i.i.d., uniformly at random

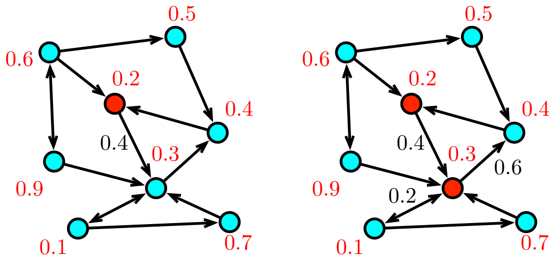


- On each round, uninfected nodes compute total weight of infected neighbors and become infected if

$$\sum_{j \text{ is infected}} b_{ji} > \theta_i$$

Model: Linear threshold model (broadly, triggering models)

- Edges have weights (b_{ij}), satisfying $\sum_j b_{ji} \leq 1$
- Nodes choose thresholds $\theta_i \in [0,1]$ i.i.d., uniformly at random



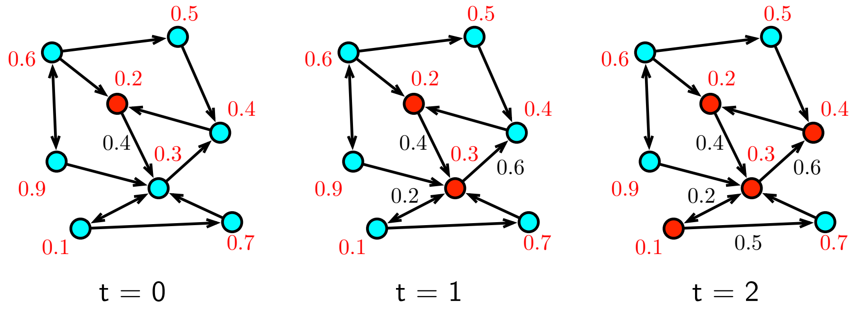
t = 0 t = 1

- On each round, uninfected nodes compute total weight of infected neighbors and become infected if

$$\sum_{j \text{ is infected}} b_{ji} > \theta_i$$

Model: Linear threshold model (broadly, triggering models)

- Edges have weights (b_{ij}), satisfying $\sum_j b_{ji} \leq 1$
- Nodes choose thresholds $\theta_i \in [0, 1]$ i.i.d., uniformly at random



- On each round, uninfected nodes compute total weight of infected neighbors and become infected if

$$\sum_{j \text{ is infected}} b_{ji} > \theta_i$$

Previous work

- Monotonicity, **submodularity** of influence function in triggering models (Kempe et al. '03)

Previous work

- Monotonicity, **submodularity** of influence function in triggering models (Kempe et al. '03)
- $\implies$ Greedy algorithm yields $\left(1 - \frac{1}{e}\right)$-approximation to

$$\max_{A \subseteq V : |A| \leq k} \mathcal{I}(A)$$

Previous work

- Monotonicity, **submodularity** of influence function in triggering models (Kempe et al. '03)
- $\implies$ Greedy algorithm yields $\left(1 - \frac{1}{e}\right)$-approximation to

$$\max_{A \subseteq V : |A| \leq k} \mathcal{I}(A)$$

- However, method involves approximating $\mathcal{I}$ at each iteration of greedy algorithm via simulations

Key contributions

1. Computable upper and lower bounds for influence function in general triggering models
2. Characterization of gap between bounds

Key contributions

1. Computable upper and lower bounds for influence function in general triggering models
2. Characterization of gap between bounds
3. Proof of monotonicity, submodularity for family of lower bounds
   $\implies \left(1 - \frac{1}{e}\right)$-approximation for sequential greedy algorithm

Key contributions

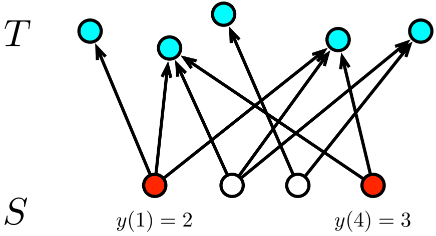
1. Computable upper and lower bounds for influence function in general triggering models
2. Characterization of gap between bounds
3. Proof of monotonicity, submodularity for family of lower bounds $\implies (1 - \frac{1}{e})$-approximation for sequential greedy algorithm

- Leads to significant speed-ups:

	LB_1	LB_2	UB	Simulation
Erdös-Renyi	**1.00**	2.36	27.43	710.58
Preferential attachment	**1.00**	2.56	28.49	759.83
$2D$-grid	**1.00**	2.43	47.08	1301.73

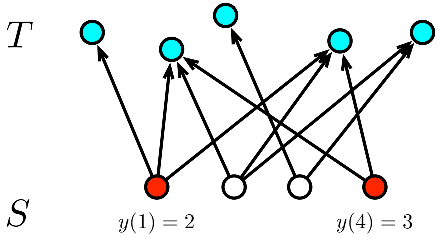
Budget allocation (with Ashley Hou)

- **Problem:** Given fixed budget to distribute amongst influencers, how to optimally allocate resources?



Budget allocation (with Ashley Hou)

- **Problem:** Given fixed budget to distribute amongst influencers, how to optimally allocate resources?

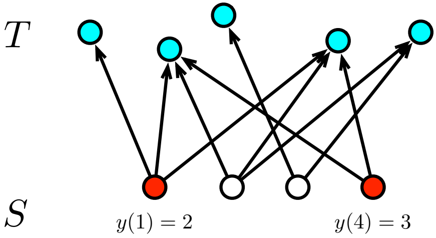


- **Mathematical formulation:** If resources $\{y(s)\}_{s \in S}$ are allocated among source nodes S, probability of influencing customer t is

$$I_t(y) = 1 - \prod_{(s,t) \in E} (1 - p_{st})^{y(s)}$$

Budget allocation (with Ashley Hou)

- **Problem:** Given fixed budget to distribute amongst influencers, how to optimally allocate resources?



- **Mathematical formulation:** If resources $\{y(s)\}_{s \in S}$ are allocated among source nodes S, probability of influencing customer t is

$$I_t(y) = 1 - \prod_{(s,t) \in E} (1 - p_{st})^{y(s)}$$

so we solve $\max \sum_{t \in T} I_t(y)$ s.t. $\sum_{s \in S} y(s) \leq B$

Robust variant

- In practice, might not know edge parameters $p = \{p_{st}\}$, or even edge structure

Robust variant

- In practice, might not know edge parameters $p = \{p_{st}\}$, or even edge structure
- Robust optimization framework:

$$\max_{\sum_{s \in S} y(s) \leq B} \left\{ \min_{p \in \Sigma} \sum_{t \in T} I_t^p(y) \right\}$$

Robust variant

- In practice, might not know edge parameters $p = \{p_{st}\}$, or even edge structure
- Robust optimization framework:

$$\max_{\sum_{s \in S} y(s) \leq B} \left\{ \min_{p \in \Sigma} \sum_{t \in T} I_t^p(y) \right\}$$

- **Goal:** Develop efficient algorithms for robust budget allocation with provable approximation guarantees

Robust variant

- In practice, might not know edge parameters $p = \{p_{st}\}$, or even edge structure
- Robust optimization framework:

$$\max_{\sum_{s \in S} y(s) \leq B} \left\{ \min_{p \in \Sigma} \sum_{t \in T} I_t^p(y) \right\}$$

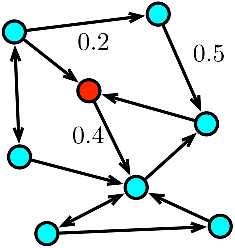
- **Goal:** Develop efficient algorithms for robust budget allocation with provable approximation guarantees
- Ingredients: Maximization of min of submodular functions, extensions to integer lattices and budget constraints

Network immunization (with Wen Yan)

- **Goal:** Given a budget of interventions at nodes/edges of a graph, how to optimally distribute resources to retard an epidemic?

Network immunization (with Wen Yan)

- **Goal:** Given a budget of interventions at nodes/edges of a graph, how to optimally distribute resources to retard an epidemic?
- Interested in *fractional immunization*, which only decreases infectiveness of nodes/edges



Network immunization

- Formulation as influence maximization problem:

$$\min_{\sum \theta_{ij} \leq B} \left\{ \max_{A \subseteq V : |A| \leq k} \mathcal{I}\left(A; \{b_{ij}\} - \{\theta_{ij}\}\right) \right\}$$

Network immunization

- Formulation as influence maximization problem:

$$\min_{\sum \theta_{ij} \leq B} \left\{ \max_{A \subseteq V : |A| \leq k} \mathcal{I}\left(A; \{b_{ij}\} - \{\theta_{ij}\}\right) \right\}$$

- **Challenges:**
  1. Bilevel optimization problem involving discrete and continuous variables
  2. No computable closed-form expression for $\mathcal{I}$ or $\nabla \mathcal{I}$

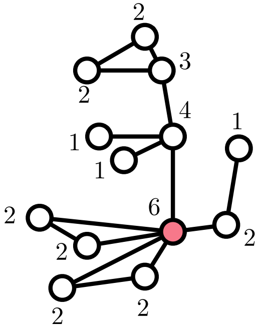
Local algorithms

Muni Pydi
(UW-Madison)

Varun Jog
(UW-Madison)

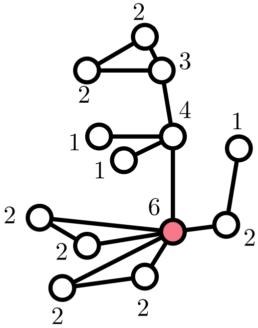
Maximizing graph functions

- Given function f defined on nodes of a graph
- **Examples:** Degree, age of node, power/population level, etc.



Maximizing graph functions

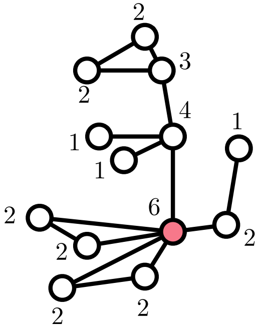
- Given function f defined on nodes of a graph
- **Examples:** Degree, age of node, power/population level, etc.



- **Goal:** Maximize f by "walking" along edges and querying values

Maximizing graph functions

- Given function f defined on nodes of a graph
- **Examples:** Degree, age of node, power/population level, etc.



- **Goal:** Maximize f by "walking" along edges and querying values
- Could use "vanilla random walk" with transition probabilities $P_{ij} = \frac{w_{ij}}{d_i}$, but can we leverage smoothness/structure of graph function?

Metropolis-Hastings algorithm

- MH algorithm specified by target density p_f and proposal distribution Q (stochastic matrix)

Metropolis-Hastings algorithm

- MH algorithm specified by target density p_f and proposal distribution Q (stochastic matrix)
- Transition matrix:

$$P_{ij} = \begin{cases} Q_{ij} \min\left\{1, \frac{p_f(j) Q_{ji}}{p_f(i) Q_{ij}}\right\}, & j \neq i, \\ 1 - \sum_{j \neq i} P_{ij}, & j = i \end{cases}$$

Metropolis-Hastings algorithm

- MH algorithm specified by target density p_f and proposal distribution Q (stochastic matrix)
- Transition matrix:

$$P_{ij} = \begin{cases} Q_{ij} \min\left\{1, \frac{p_f(j) Q_{ji}}{p_f(i) Q_{ij}}\right\}, & j \neq i, \\ 1 - \sum_{j \neq i} P_{ij}, & j = i \end{cases}$$

- Known convergence of MH algorithm to p_f

Metropolis-Hastings algorithm

- MH algorithm specified by target density p_f and proposal distribution Q (stochastic matrix)
- Transition matrix:

$$P_{ij} = \begin{cases} Q_{ij} \min\left\{1, \frac{p_f(j) Q_{ji}}{p_f(i) Q_{ij}}\right\}, & j \neq i, \\ 1 - \sum_{j \neq i} P_{ij}, & j = i \end{cases}$$

- Known convergence of MH algorithm to p_f
- **Idea:** Build a density p_f maximized wherever f is maximized, hope that MH algorithm finds maximizers quickly

Local algorithm

1. Initialize at random vertex i_0

Local algorithm

1. Initialize at random vertex i_0
2. Take T steps of MH algorithm according to transition matrix P

Local algorithm

1. Initialize at random vertex i_0
2. Take T steps of MH algorithm according to transition matrix P
3. Output maximum among $\{f(i_0), \dots, f(i_T)\}$

Local algorithm

1. Initialize at random vertex i_0
2. Take T steps of MH algorithm according to transition matrix P
3. Output maximum among $\{f(i_0), \ldots, f(i_T)\}$

- **Exponential walk:** $p_f(i) \propto \exp\Big(\gamma f(i)\Big)$ and $Q = D^{-1}W$
- **Laplacian walk:** $p_f(i) \propto f^2(i)$ and Q defined with respect to eigenvectors of graph Laplacian $L = D - W$

Local algorithm

1. Initialize at random vertex i_0
2. Take T steps of MH algorithm according to transition matrix P
3. Output maximum among $\{f(i_0), \ldots, f(i_T)\}$

- **Exponential walk:** $p_f(i) \propto \exp\Big(\gamma f(i)\Big)$ and $Q = D^{-1}W$
- **Laplacian walk:** $p_f(i) \propto f^2(i)$ and Q defined with respect to eigenvectors of graph Laplacian $L = D - W$

- **Theoretical results:** Rates of convergence in TV distance, hitting time bounds for both algorithms in terms of graph/function characteristics

Summary

- Many interesting data analysis problems involving network-structured data

Summary

- Many interesting data analysis problems involving network-structured data
- Problems span statistics, optimization, algorithmic design

Summary

- Many interesting data analysis problems involving network-structured data
- Problems span statistics, optimization, algorithmic design
- Need for new methods, theory, and validation on real-world datasets

Summary

- Many interesting data analysis problems involving network-structured data
- Problems span statistics, optimization, algorithmic design
- Need for new methods, theory, and validation on real-world datasets
- Modern-day algorithms should be scalable to large data sets

Summary

- Many interesting data analysis problems involving network-structured data
- Problems span statistics, optimization, algorithmic design
- Need for new methods, theory, and validation on real-world datasets
- Modern-day algorithms should be scalable to large data sets

Thank you!