How Robust are Thresholds for Community Detection?

Alex Wein (MIT)

Joint with Ankur Moitra (MIT) and Amelia Perry (MIT)

Two Worldviews

Convex Optimization	Statistical Physics

Two Worldviews

Convex Optimization

- Algorithm: semidefinite programming (SDP)
    - Most powerful known algorithm for various worst-case and average-case problems

Statistical Physics

Two Worldviews

Convex Optimization

- Algorithm: semidefinite programming (SDP)
  - Most powerful known algorithm for various worst-case and average-case problems
- Reasoning about hardness of problems
  - Integrality gaps
  - Extension complexity
  - Sum-of-squares lower bounds

Statistical Physics

Two Worldviews

Convex Optimization

- Algorithm: semidefinite programming (SDP)
  - Most powerful known algorithm for various worst-case and average-case problems

- Reasoning about hardness of problems
  - Integrality gaps
  - Extension complexity
  - Sum-of-squares lower bounds

Statistical Physics

- Algorithm: belief propagation (BP)
  - Believed/known to be statistically optimal for many average-case (random) problems

Two Worldviews

Convex Optimization

- Algorithm: semidefinite programming (SDP)
  - Most powerful known algorithm for various worst-case and average-case problems
- Reasoning about hardness of problems
  - Integrality gaps
  - Extension complexity
  - Sum-of-squares lower bounds

Statistical Physics

- Algorithm: belief propagation (BP)
  - Believed/known to be statistically optimal for many average-case (random) problems
- Reasoning about hardness of problems
  - Non-rigorous cavity/replica methods
  - Predict regimes in which problems are easy/hard/impossible (“phase transitions”)

Two Worldviews

Convex Optimization

- Algorithm: semidefinite programming (SDP)
  - Most powerful known algorithm for various worst-case and average-case problems
- Reasoning about hardness of problems
  - Integrality gaps
  - Extension complexity
  - Sum-of-squares lower bounds

Statistical Physics

- Algorithm: belief propagation (BP)
  - Believed/known to be statistically optimal for many average-case (random) problems
- Reasoning about hardness of problems
  - Non-rigorous cavity/replica methods
  - Predict regimes in which problems are easy/hard/impossible ("phase transitions")

Do these frameworks always agree?

Two Worldviews

Convex Optimization

- Algorithm: semidefinite programming (SDP)
  - Most powerful known algorithm for various worst-case and average-case problems
- Reasoning about hardness of problems
  - Integrality gaps
  - Extension complexity
  - Sum-of-squares lower bounds

Statistical Physics

- Algorithm: belief propagation (BP)
  - Believed/known to be statistically optimal for many average-case (random) problems
- Reasoning about hardness of problems
  - Non-rigorous cavity/replica methods
  - Predict regimes in which problems are easy/hard/impossible (“phase transitions”)

Do these frameworks always agree?

And if they don’t agree, which one is correct?

Stochastic Block Model (SBM)

Stochastic Block Model (SBM)

Model for community detection in graphs

Stochastic Block Model (SBM)

Model for community detection in graphs

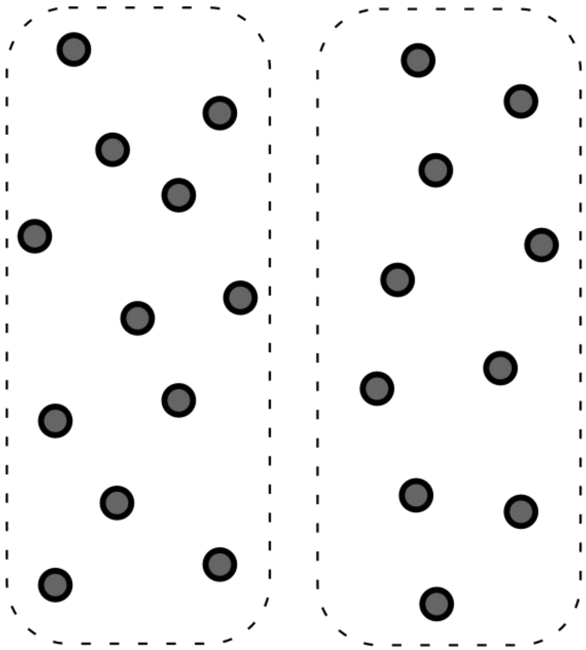
Introduced by Holland, Laskey and Leinhardt (1983)

Stochastic Block Model (SBM)

Model for community detection in graphs

Introduced by Holland, Laskey and Leinhardt (1983)

- Vertices partitioned into 2 hidden communities

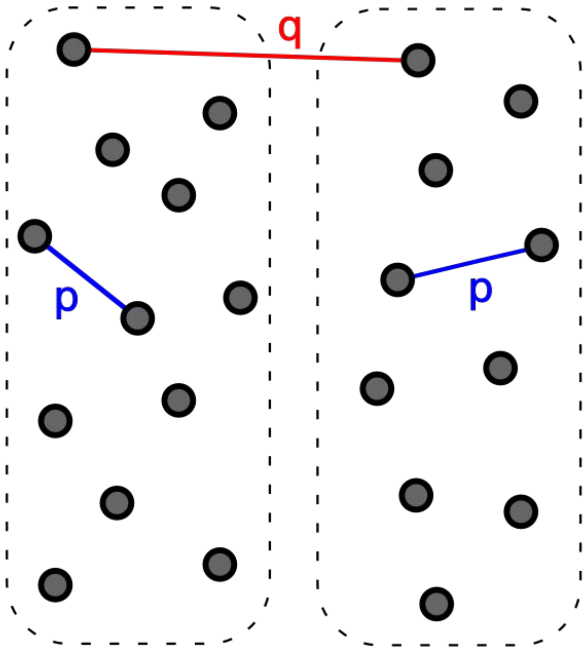


Stochastic Block Model (SBM)

Model for community detection in graphs

Introduced by Holland, Laskey and Leinhardt (1983)

- Vertices partitioned into 2 hidden communities
- Connection probabilities $p > q$

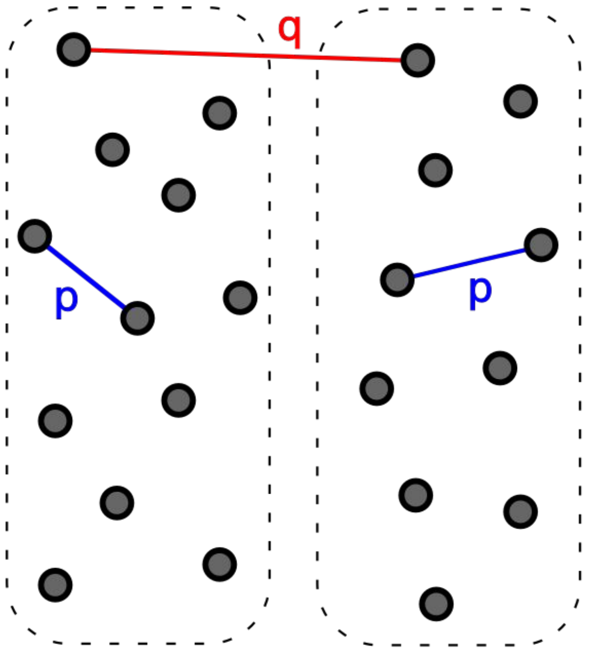


Stochastic Block Model (SBM)

Model for community detection in graphs

Introduced by Holland, Laskey and Leinhardt (1983)

- Vertices partitioned into 2 hidden communities
- Connection probabilities $p > q$
- Edges independent



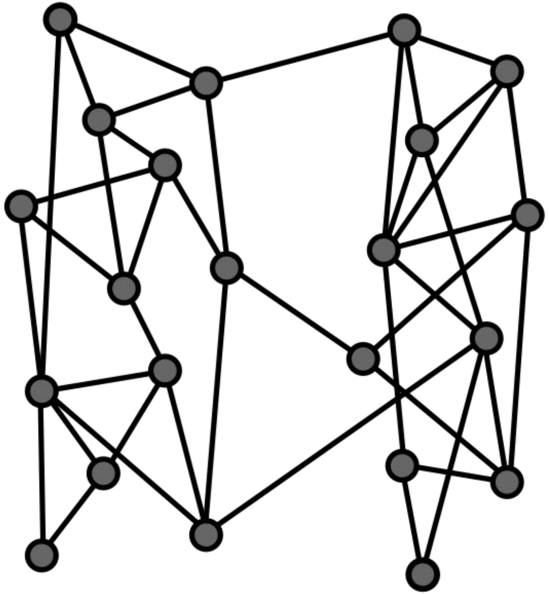
Stochastic Block Model (SBM)

Model for community detection in graphs

Introduced by Holland, Laskey and Leinhardt (1983)

- Vertices partitioned into 2 hidden communities
- Connection probabilities $p > q$
- Edges independent

Goal: recover communities (exactly or approximately)

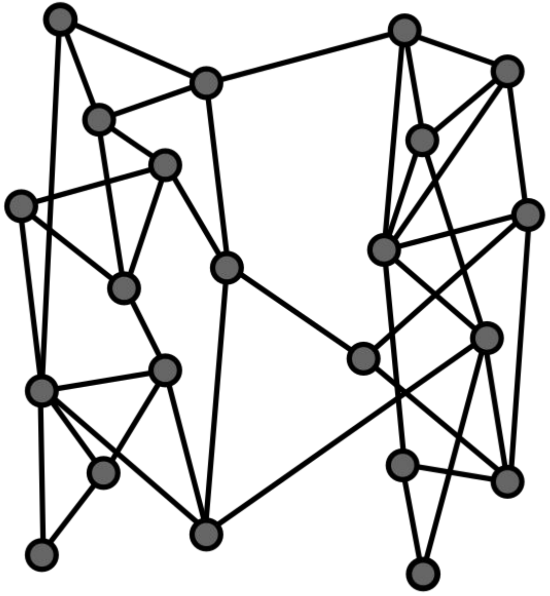


Stochastic Block Model (SBM)

Model for community detection in graphs

Introduced by Holland, Laskey and Leinhardt (1983)

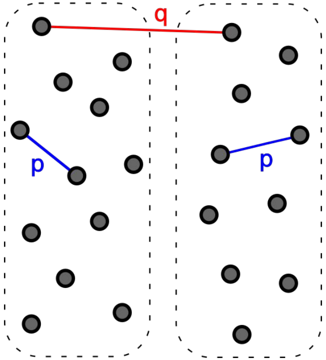
- Vertices partitioned into 2 hidden communities
- Connection probabilities p > q
- Edges independent



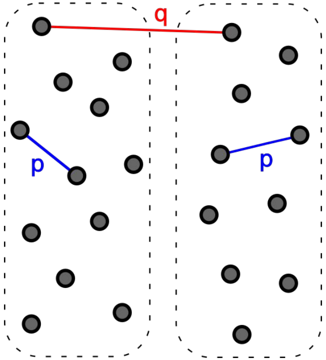
Goal: recover communities (exactly or approximately)

Studied in statistics, information theory, computer science, statistical physics, ...

Sharp Threshold Behavior



n -- num vertices
p -- within-community edge prob
q -- between-community edge prob

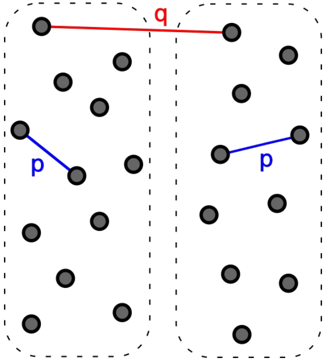


Sharp Threshold Behavior

n -- num vertices
p -- within-community edge prob
q -- between-community edge prob

Dense regime: $p = a \log(n)/n$, $q = b \log(n)/n$, $a > b > 0$

Average degree: $O(\log n)$



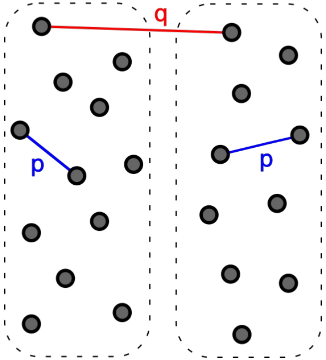
Sharp Threshold Behavior

n -- num vertices
p -- within-community edge prob
q -- between-community edge prob

Dense regime: p = a log(n)/n, q = b log(n)/n, a > b > 0
Average degree: O(log n)

Theorem [Abbe-Bandeira-Hall ’14, Mossel-Neeman-Sly ‘14]:
Possible to achieve exact recovery* iff

$$\sqrt{a} - \sqrt{b} \geq \sqrt{2}$$



Sharp Threshold Behavior

n -- num vertices
p -- within-community edge prob
q -- between-community edge prob

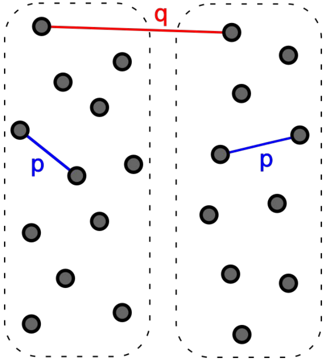
Dense regime: p = a log(n)/n, q = b log(n)/n, a > b > 0
Average degree: O(log n)

Theorem [Abbe-Bandeira-Hall ’14, Mossel-Neeman-Sly ‘14]:
Possible to achieve exact recovery* iff

$$\sqrt{a} - \sqrt{b} \geq \sqrt{2}$$

* recover communities exactly, with probability $\to 1$ as n $\to \infty$

Sharp Threshold Behavior



n -- num vertices
p -- within-community edge prob
q -- between-community edge prob

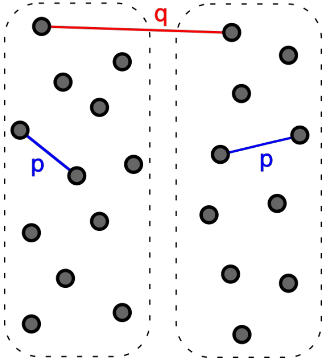
Dense regime: p = a log(n)/n, q = b log(n)/n, a > b > 0
Average degree: O(log n)

Theorem [Abbe-Bandeira-Hall '14, Mossel-Neeman-Sly '14]:
Possible to achieve exact recovery* iff

$$\sqrt{a} - \sqrt{b} \geq \sqrt{2}$$

* recover communities exactly, with probability $\to 1$ as n $\longrightarrow \infty$

Sharp Threshold Behavior



n -- num vertices
p -- within-community edge prob
q -- between-community edge prob

Dense regime: p = a log(n)/n, q = b log(n)/n, a > b > 0
Average degree: O(log n)

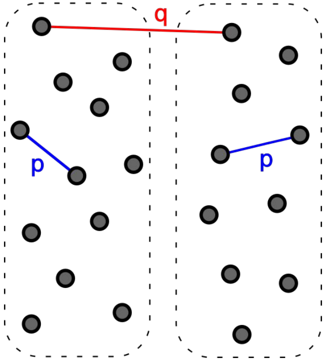
Theorem [Abbe-Bandeira-Hall ’14, Mossel-Neeman-Sly ‘14]:
Possible to achieve exact recovery* iff

$$\sqrt{a} - \sqrt{b} \geq \sqrt{2}$$

Sparse regime: p = a/n, q = b/n, a > b
Average degree: O(1)

* recover communities exactly, with probability $\to 1$ as n $\to \infty$

Sharp Threshold Behavior



n -- num vertices
p -- within-community edge prob
q -- between-community edge prob

Dense regime: p = a log(n)/n, q = b log(n)/n, a > b > 0
Average degree: O(log n)

Theorem [Abbe-Bandeira-Hall '14, Mossel-Neeman-Sly '14]:
Possible to achieve exact recovery* iff

$$\sqrt{a} - \sqrt{b} \geq \sqrt{2}$$

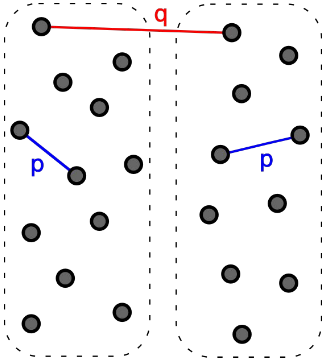
Sparse regime: p = a/n, q = b/n, a > b
Average degree: O(1)

Theorem [Mossel-Neeman-Sly '13, '14; Massoulie '14]:
Possible to achieve partial recovery* iff

$$(a - b)^2 > 2(a + b)$$

* recover communities exactly, with probability $\rightarrow 1$ as n $\longrightarrow \infty$

Sharp Threshold Behavior



n -- num vertices
p -- within-community edge prob
q -- between-community edge prob

Dense regime: p = a log(n)/n, q = b log(n)/n, a > b > 0
Average degree: O(log n)

Theorem [Abbe-Bandeira-Hall '14, Mossel-Neeman-Sly '14]:
Possible to achieve exact recovery* iff

$$\sqrt{a} - \sqrt{b} \geq \sqrt{2}$$

* recover communities exactly, with probability $\to 1$ as n $\to \infty$

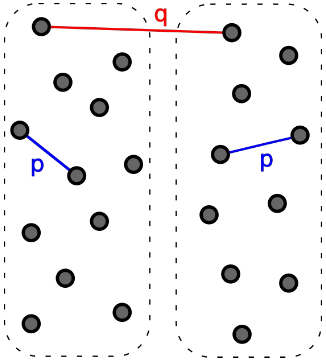
Sparse regime: p = a/n, q = b/n, a > b
Average degree: O(1)

Theorem [Mossel-Neeman-Sly '13, '14; Massoulie '14]:
Possible to achieve partial recovery* iff

$$(a-b)^2 > 2(a+b)$$

* find partition with $\frac{1}{2}+\varepsilon$ correlation with truth, with probability $\to 1$ as n $\to \infty$

Sharp Threshold Behavior



n -- num vertices
p -- within-community edge prob
q -- between-community edge prob

Dense regime: p = a log(n)/n, q = b log(n)/n, a > b > 0
Average degree: O(log n)

Theorem [Abbe-Bandeira-Hall ’14, Mossel-Neeman-Sly ‘14]:
Possible to achieve exact recovery* iff

$$\sqrt{a} - \sqrt{b} \geq \sqrt{2}$$

Sparse regime: p = a/n, q = b/n, a > b
Average degree: O(1)

Theorem [Mossel-Neeman-Sly ’13, ‘14; Massoulie ’14]:
Possible to achieve partial recovery* iff

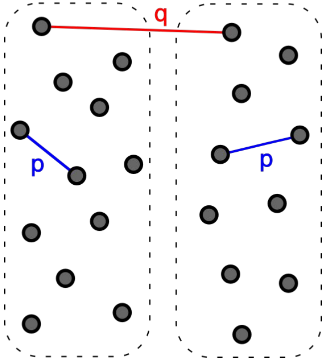
$$(a - b)^2 > 2(a + b)$$

First conjectured by statistical physics [DKMZ’11]

* recover communities exactly, with probability $\to 1$ as n $\to \infty$

* find partition with $\frac{1}{2} + \varepsilon$ correlation with truth, with probability $\to 1$ as n $\to \infty$

Sharp Threshold Behavior



n -- num vertices
p -- within-community edge prob
q -- between-community edge prob

Dense regime: $p = a \log(n)/n$, $q = b \log(n)/n$, $a > b > 0$
Average degree: $O(\log n)$

Theorem [Abbe-Bandeira-Hall '14, Mossel-Neeman-Sly '14]:
Possible to achieve exact recovery* iff

$$\sqrt{a} - \sqrt{b} \geq \sqrt{2}$$

* recover communities exactly, with probability $\to 1$ as $n \to \infty$

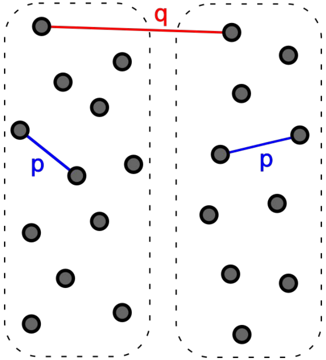
Sparse regime: $p = a/n$, $q = b/n$, $a > b$
Average degree: $O(1)$

Theorem [Mossel-Neeman-Sly '13, '14; Massoulie '14]:
Possible to achieve partial recovery* iff

$$(a - b)^2 > 2(a + b)$$

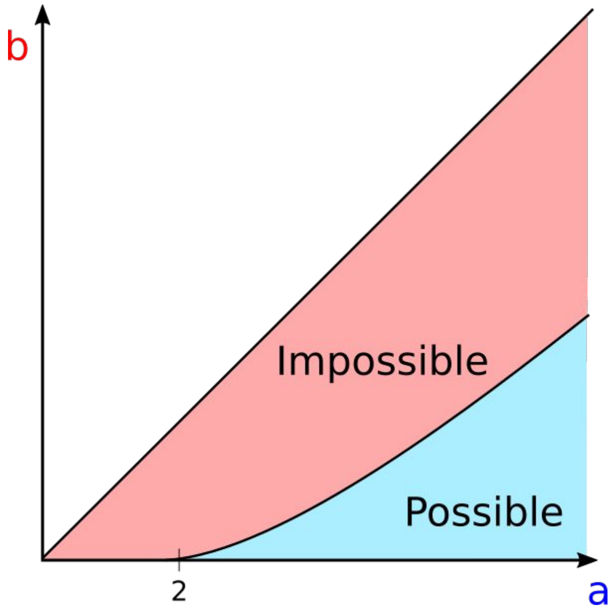
First conjectured by statistical physics [DKMZ'11]

* find partition with $\frac{1}{2} + \varepsilon$ correlation with truth, with probability $\to 1$ as $n \to \infty$



Sharp Threshold Behavior

n -- num vertices
p -- within-community edge prob
q -- between-community edge prob



Sparse regime: p = a/n, q = b/n, a > b
Average degree: O(1)

Theorem [Mossel-Neeman-Sly '13, '14; Massoulie '14]:
Possible to achieve partial recovery* iff

$$(a-b)^2 > 2(a+b)$$

First conjectured by statistical physics [DKMZ'11]

* find partition with $\frac{1}{2}+\varepsilon$ correlation with truth, with probability $\to 1$ as n $\to \infty$

Optimal Algorithms

In both settings, there are efficient algorithms known to work up to the threshold:

Optimal Algorithms

In both settings, there are efficient algorithms known to work up to the threshold:

Dense regime (exact recovery):

Sparse regime (partial recovery):

Optimal Algorithms

In both settings, there are efficient algorithms known to work up to the threshold:

Dense regime (exact recovery):

- Spectral/combinatorial clustering + local refinement [MNS'14, AS'15]

Sparse regime (partial recovery):

Optimal Algorithms

In both settings, there are efficient algorithms known to work up to the threshold:

Dense regime (exact recovery):

- Spectral/combinatorial clustering + local refinement [MNS'14, AS'15]
- Semidefinite programming (SDP) [ABH'14, HWX'15, Ban'15]

Sparse regime (partial recovery):

Optimal Algorithms

In both settings, there are efficient algorithms known to work up to the threshold:

Dense regime (exact recovery):

- Spectral/combinatorial clustering + local refinement [MNS'14, AS'15]
- Semidefinite programming (SDP) [ABH'14, HWX'15, Ban'15]

Sparse regime (partial recovery):

- (Linearized) belief propagation + variants [Mas'14, MNS'13, BLM'15, MNS'14]

Optimal Algorithms

In both settings, there are efficient algorithms known to work up to the threshold:

Dense regime (exact recovery):

- Spectral/combinatorial clustering + local refinement [MNS'14, AS'15]
- Semidefinite programming (SDP) [ABH'14, HWX'15, Ban'15]

Sparse regime (partial recovery):

- (Linearized) belief propagation + variants [Mas'14, MNS'13, BLM'15, MNS'14]
- SDPs can get close to the threshold, but haven't been able to reach it [GV'15, MS'15]

Optimal Algorithms

In both settings, there are efficient algorithms known to work up to the threshold:

Dense regime (exact recovery):

- Spectral/combinatorial clustering + local refinement [MNS'14, AS'15]
- Semidefinite programming (SDP) [ABH'14, HWX'15, Ban'15]

Sparse regime (partial recovery):

- (Linearized) belief propagation + variants [Mas'14, MNS'13, BLM'15, MNS'14]
- SDPs can get close to the threshold, but haven't been able to reach it [GV'15, MS'15]

Can SDPs reach the threshold in the sparse regime, or are they suboptimal?

Optimal Algorithms

In both settings, there are efficient algorithms known to work up to the threshold:

Dense regime (exact recovery):

- Spectral/combinatorial clustering + local refinement [MNS'14, AS'15]
- Semidefinite programming (SDP) [ABH'14, HWX'15, Ban'15]

Sparse regime (partial recovery):

- (Linearized) belief propagation + variants [Mas'14, MNS'13, BLM'15, MNS'14]
- SDPs can get close to the threshold, but haven't been able to reach it [GV'15, MS'15]

Can SDPs reach the threshold in the sparse regime, or are they suboptimal?

Answer: We will give evidence that SDPs cannot reach the threshold! — but only because they are actually solving a harder problem.

Semirandom Models

Semirandom Models

Between average-case and worst-case

Semirandom Models

Between average-case and worst-case

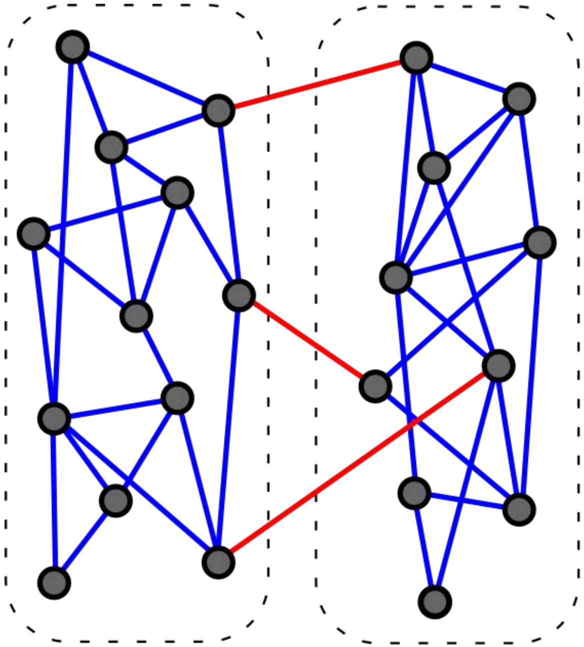
For stochastic block model [Feige-Kilian ’00]:

Semirandom Models

Between average-case and worst-case

For stochastic block model [Feige-Kilian ’00]:

1. Draw a random graph from the usual SBM



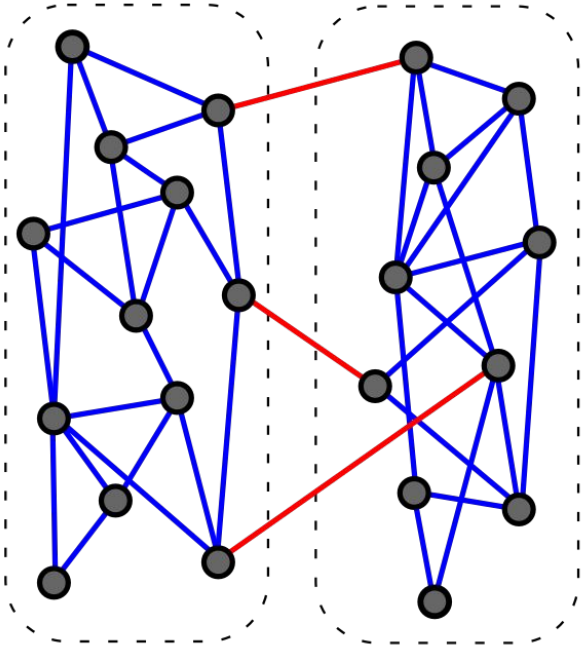
Semirandom Models

Between average-case and worst-case

"random model"

For stochastic block model [Feige-Kilian '00]:

1. Draw a random graph from the usual SBM



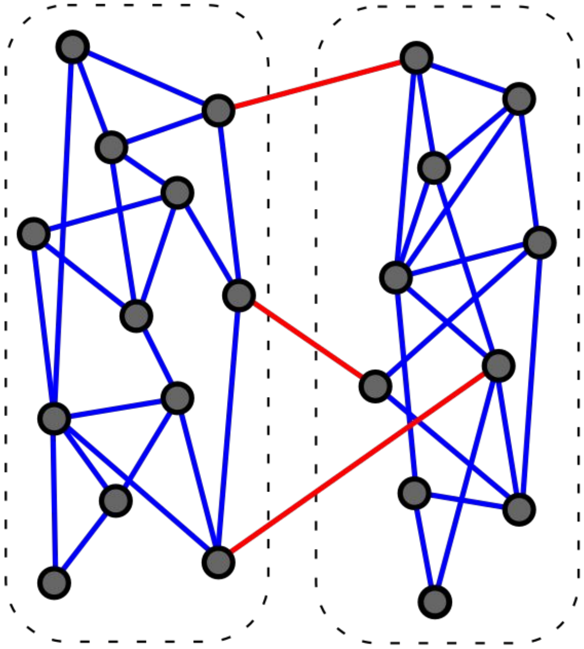
Semirandom Models

Between average-case and worst-case

"random model"

For stochastic block model [Feige-Kilian '00]:

1. Draw a random graph from the usual SBM
2. An adversary can perform any number of ***monotone*** ('helpful') changes:
   a. Add edges within communities
   b. Remove edges between communities



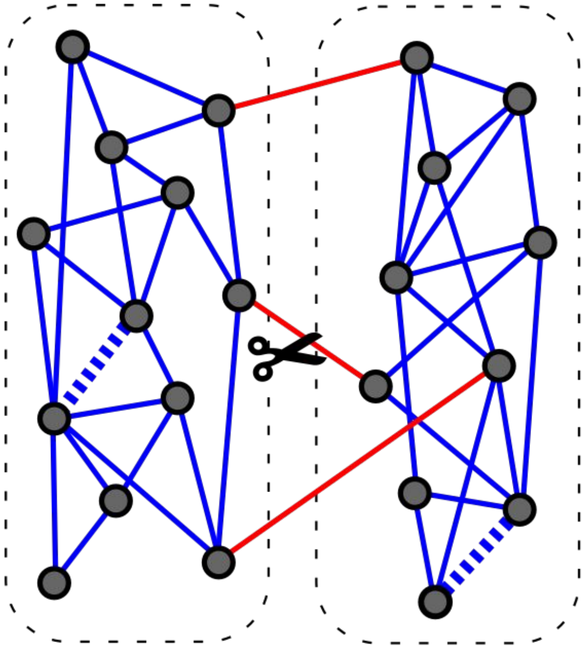
Semirandom Models

Between average-case and worst-case

“random model”

For stochastic block model [Feige-Kilian ’00]:

1. Draw a random graph from the usual SBM
2. An adversary can perform any number of ***monotone*** (‘helpful’) changes:
    a. Add edges within communities
    b. Remove edges between communities



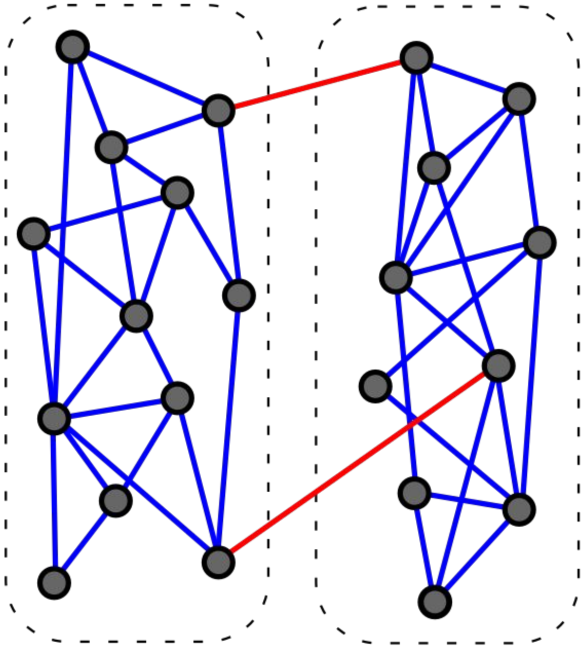
Semirandom Models

Between average-case and worst-case

“random model”

For stochastic block model [Feige-Kilian ’00]:

1. Draw a random graph from the usual SBM
2. An adversary can perform any number of ***monotone*** (‘helpful’) changes:
   a. Add edges within communities
   b. Remove edges between communities



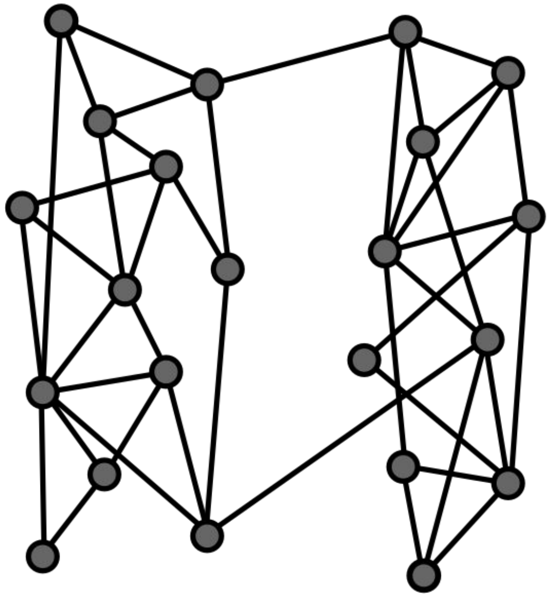
Semirandom Models

Between average-case and worst-case

“random model”

For stochastic block model [Feige-Kilian ’00]:

1. Draw a random graph from the usual SBM
2. An adversary can perform any number of ***monotone*** (‘helpful’) changes:
    a. Add edges within communities
    b. Remove edges between communities



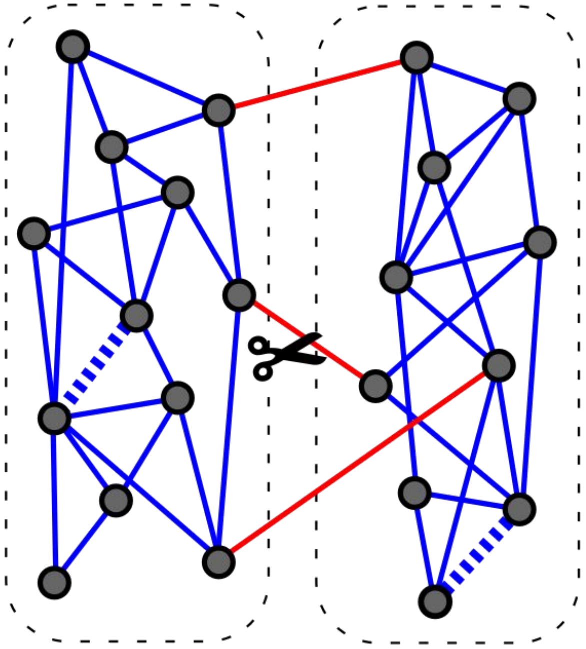
Semirandom Models

Between average-case and worst-case

"random model"

For stochastic block model [Feige-Kilian '00]:

1. Draw a random graph from the usual SBM
2. An adversary can perform any number of ***monotone*** ('helpful') changes:
   a. Add edges within communities
   b. Remove edges between communities



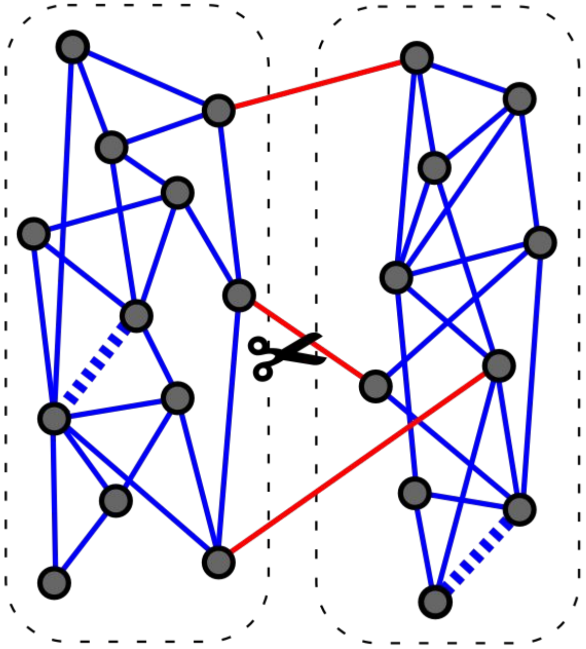
Semirandom Models

Between average-case and worst-case

“random model”

For stochastic block model [Feige-Kilian ’00]:

1. Draw a random graph from the usual SBM
2. An <u>adversary</u> can perform any number of ***monotone*** (‘helpful’) changes:
    a. Add edges within communities
    b. Remove edges between communities



Prevents algorithms from over-tuning to specific model statistics (degree distribution, spectrum, etc.)

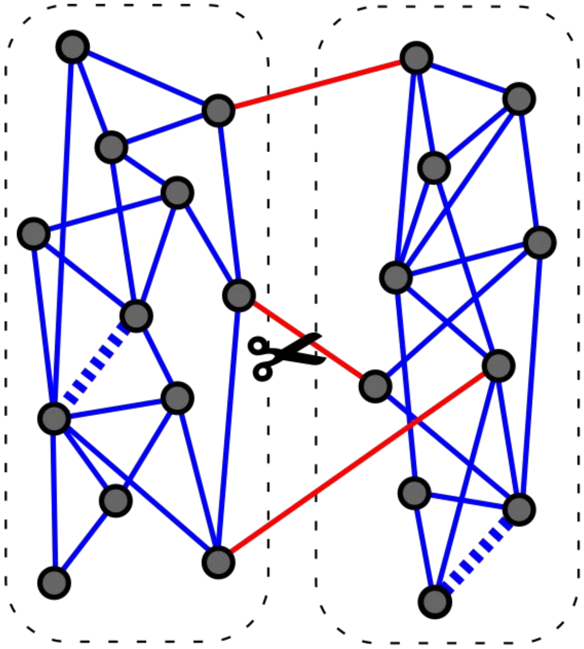
Semirandom Models

Between average-case and worst-case

"random model"

For stochastic block model [Feige-Kilian '00]:

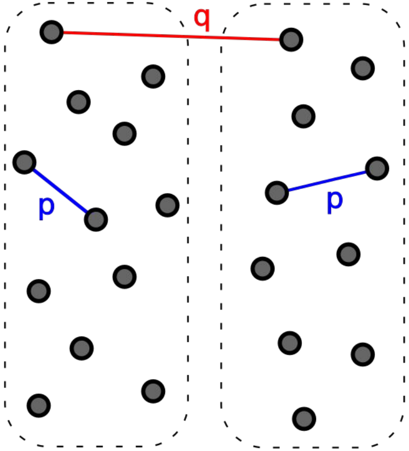
1. Draw a random graph from the usual SBM
2. An <u>adversary</u> can perform any number of ***monotone*** ('helpful') changes:
   a. Add edges within communities
   b. Remove edges between communities



Prevents algorithms from over-tuning to specific model statistics (degree distribution, spectrum, etc.)

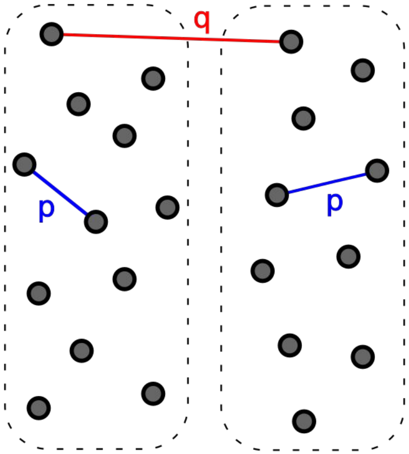
Captures some notion of 'robustness'

A Non-robust Algorithm



Example: $p = ½$, $q = ¼$, $n \to \infty$, exact recovery

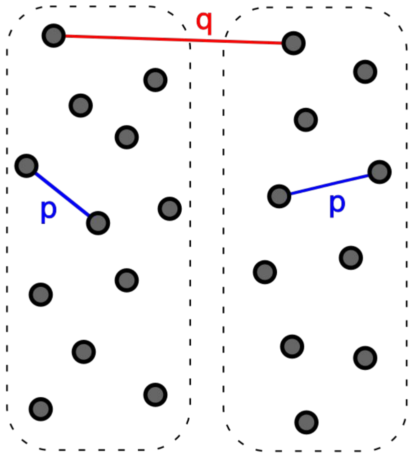
A Non-robust Algorithm



Example: p = ½, q = ¼, $n \to \infty$, exact recovery

Easy algorithm for the **random** model: count common neighbors

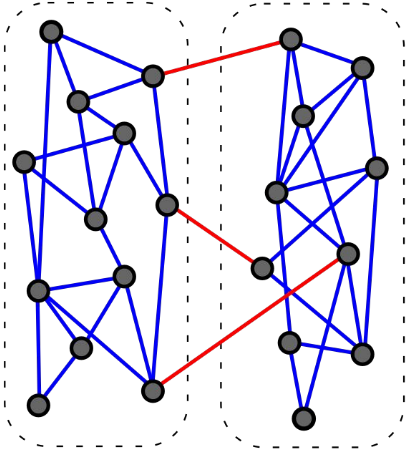
A Non-robust Algorithm



Example: p = ½, q = ¼, n $\rightarrow \infty$, exact recovery

Easy algorithm for the **random** model: count common neighbors

- 2 same-side vertices have $\approx \frac{5}{32}n$ common neighbors
- 2 opposite-side vertices have $\approx \frac{4}{32}n$ common neighbors



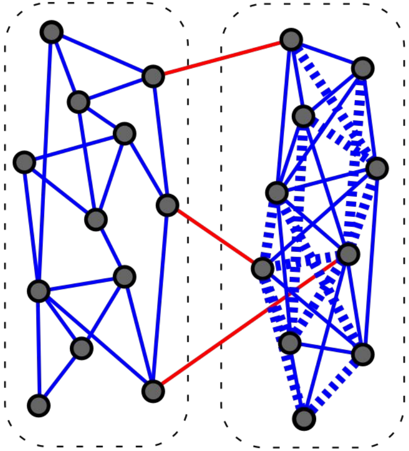
A Non-robust Algorithm

Example: $p = ½$, $q = ¼$, $n \rightarrow \infty$, exact recovery

Easy algorithm for the **random** model: count common neighbors

- 2 same-side vertices have $\approx \frac{5}{32}n$ common neighbors
- 2 opposite-side vertices have $\approx \frac{4}{32}n$ common neighbors

Semirandom model: adversary can break this — add a clique on one community



A Non-robust Algorithm

Example: p = ½, q = ¼, n $\to \infty$, exact recovery

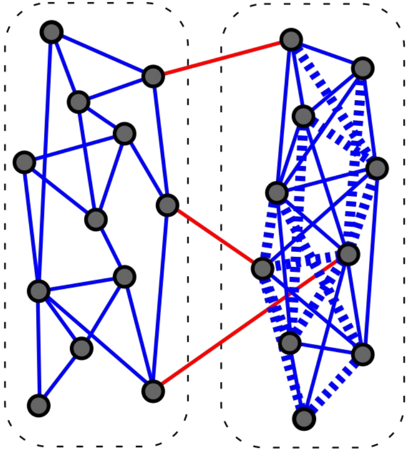
Easy algorithm for the **random** model: count common neighbors

- 2 same-side vertices have $\approx \frac{5}{32}n$ common neighbors
- 2 opposite-side vertices have $\approx \frac{4}{32}n$ common neighbors

Semirandom model: adversary can break this — add a clique on one community

- 2 left-side vertices still have $\approx \frac{5}{32}n$ common neighbors
- 2 opposite-side vertices now have $\approx \frac{6}{32}n$ common neighbors

A Non-robust Algorithm



Example: p = ½, q = ¼, $n \rightarrow \infty$, exact recovery

Easy algorithm for the **random** model: count common neighbors

- 2 same-side vertices have $\approx \frac{5}{32}n$ common neighbors
- 2 opposite-side vertices have $\approx \frac{4}{32}n$ common neighbors

Semirandom model: adversary can break this — add a clique on one community

- 2 left-side vertices still have $\approx \frac{5}{32}n$ common neighbors
- 2 opposite-side vertices now have $\approx \frac{6}{32}n$ common neighbors

The vast majority of algorithms fail against the semirandom model!

Robust Algorithms

Monotone-robust algorithm: succeeds against the semirandom model

Robust Algorithms

Monotone-robust algorithm: succeeds against the semirandom model

- In this talk, "robust" means monotone-robust

Robust Algorithms

Monotone-robust algorithm: succeeds against the semirandom model

- In this talk, “robust” means monotone-robust

Only one method is known to be robust: convex programming!

Robust Algorithms

Monotone-robust algorithm: succeeds against the semirandom model

- In this talk, "robust" means monotone-robust

Only one method is known to be robust: convex programming!

For exact recovery: SDP is robust up to the threshold [Feige–Kilian '00, Hajek–Wu–Xu '15]

Robust Algorithms

Monotone-robust algorithm: succeeds against the semirandom model

- In this talk, "robust" means monotone-robust

Only one method is known to be robust: convex programming!

For exact recovery: SDP is robust up to the threshold [Feige–Kilian '00, Hajek–Wu–Xu '15]

For partial recovery: harder…

Robust Algorithms

Monotone-robust algorithm: succeeds against the semirandom model

- In this talk, "robust" means monotone-robust

Only one method is known to be robust: convex programming!

For exact recovery: SDP is robust up to the threshold [Feige–Kilian '00, Hajek–Wu–Xu '15]

For partial recovery: harder…

- In random model, SDP works when $(a-b)^2 > C(a+b)$ [Guédon–Vershynin '15]

Robust Algorithms

Monotone-robust algorithm: succeeds against the semirandom model

- In this talk, “robust” means monotone-robust

Only one method is known to be robust: convex programming!

For exact recovery: SDP is robust up to the threshold [Feige–Kilian ’00, Hajek–Wu–Xu ’15]

For partial recovery: harder… recall: threshold is $(a-b)^2 > 2(a+b)$

- In random model, SDP works when $(a-b)^2 > C(a+b)$ [Guédon–Vershynin ‘15]

Robust Algorithms

Monotone-robust algorithm: succeeds against the semirandom model

- In this talk, “robust” means monotone-robust

Only one method is known to be robust: convex programming!

For exact recovery: SDP is robust up to the threshold [Feige–Kilian ’00, Hajek–Wu–Xu ’15]

For partial recovery: harder… recall: threshold is $(a-b)^2 > 2(a+b)$

- In random model, SDP works when $(a-b)^2 > C(a+b)$ [Guédon–Vershynin ’15]
- SDP is robust under same condition
  [Moitra–Perry–W ’15, Makarychev–Makarychev–Vijayaraghavan ’15]

Robust Algorithms

Monotone-robust algorithm: succeeds against the semirandom model

- In this talk, "robust" means monotone-robust

Only one method is known to be robust: convex programming!

For exact recovery: SDP is robust up to the threshold [Feige–Kilian '00, Hajek–Wu–Xu '15]

For partial recovery: harder… recall: threshold is $(a-b)^2 > 2(a+b)$

- In random model, SDP works when $(a-b)^2 > C(a+b)$ [Guédon–Vershynin '15]
- SDP is robust under same condition
  [Moitra–Perry–W '15, Makarychev–Makarychev–Vijayaraghavan '15]
- Open: Can [Montanari–Sen '15] analysis be made robust?

Main Result

Main Result

Theorem: Partial recovery is strictly harder in the semirandom model than in the random model — ‘helpful’ changes can hurt!

Main Result

Theorem: Partial recovery is strictly harder in the semirandom model than in the random model — ‘helpful’ changes can hurt!

Random: impossible iff $(a-b)^2 \leq 2(a+b)$

Main Result

Theorem: Partial recovery is strictly harder in the semirandom model than in the random model — ‘helpful’ changes can hurt!

Random: impossible iff $(a-b)^2 \leq 2(a+b)$

Semirandom: impossible if $(a-b)^2 \leq C_a(a+b)$

where $C_a > 2$ for all $a > 2$

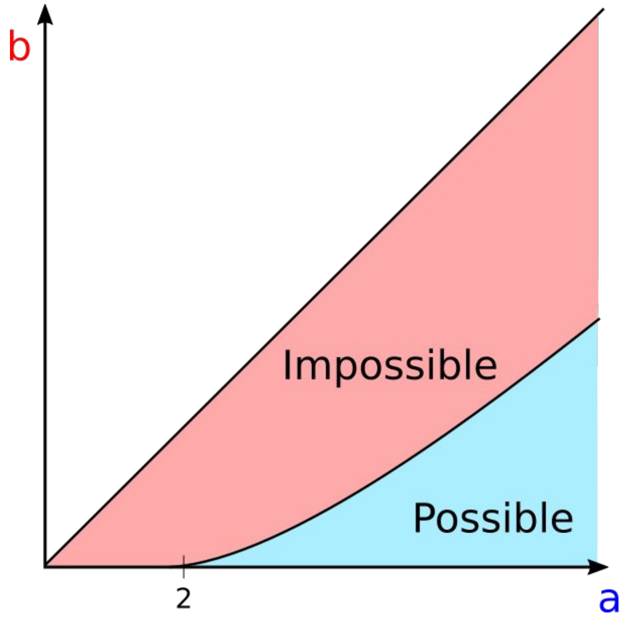
Main Result

Theorem: Partial recovery is strictly harder in the semirandom model than in the random model — ‘helpful’ changes can hurt!

Random: impossible iff $(a-b)^2 \leq 2(a+b)$

Semirandom: impossible if $(a-b)^2 \leq C_a(a+b)$

where $C_a > 2$ for all $a > 2$



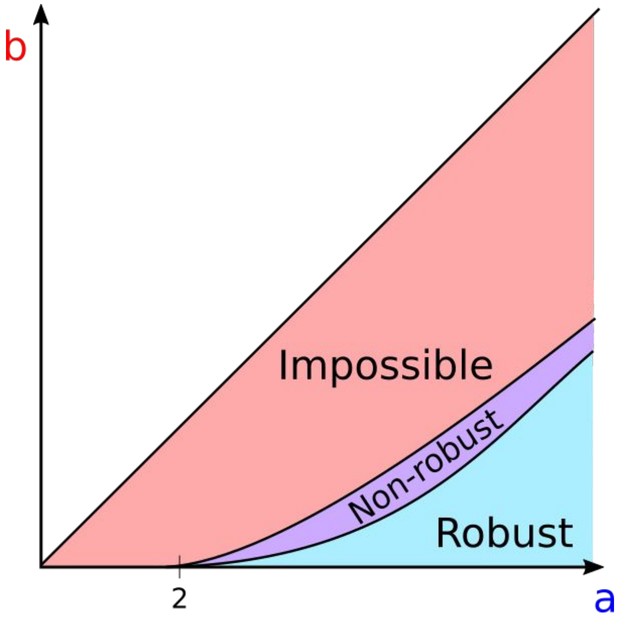
Main Result

Theorem: Partial recovery is strictly harder in the semirandom model than in the random model — 'helpful' changes can hurt!

Random: impossible iff $(a-b)^2 \leq 2(a+b)$

Semirandom: impossible if $(a-b)^2 \leq C_a(a+b)$

where $C_a > 2$ for all $a > 2$



Main Result

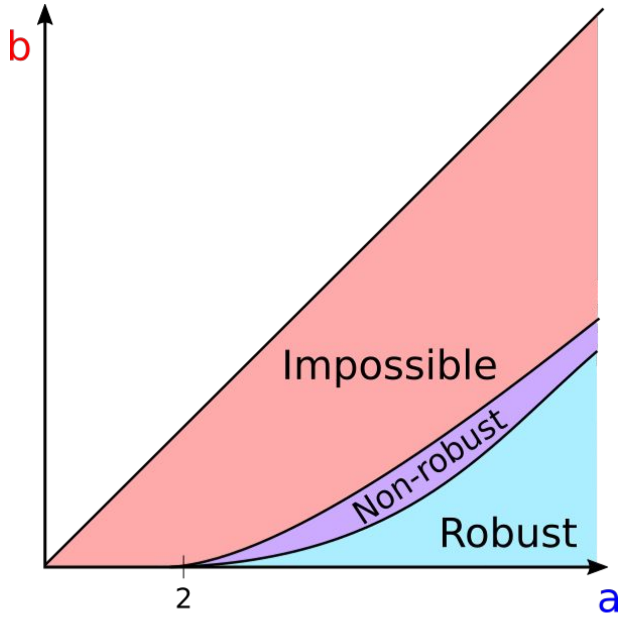
Theorem: Partial recovery is strictly harder in the semirandom model than in the random model — ‘helpful’ changes can hurt!

Random: impossible iff $(a-b)^2 \leq 2(a+b)$

Semirandom: impossible if $(a-b)^2 \leq C_a(a+b)$

where $C_a > 2$ for all $a > 2$

- No algorithm can robustly reach the threshold!



Main Result

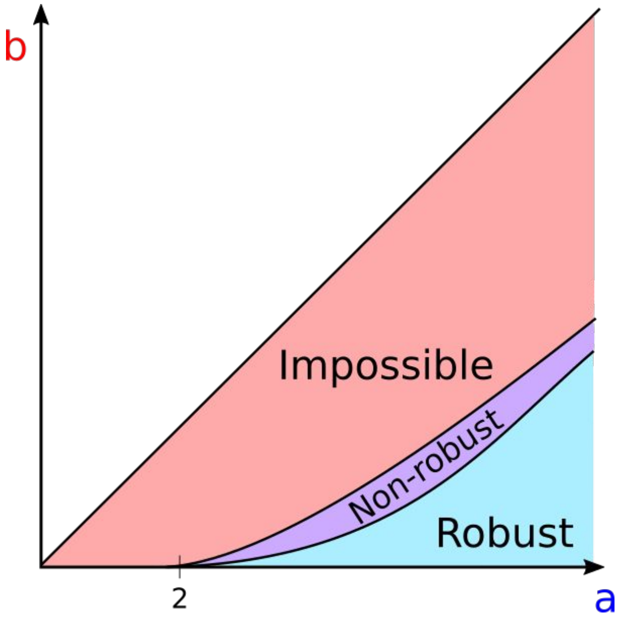
Theorem: Partial recovery is strictly harder in the semirandom model than in the random model — ‘helpful’ changes can hurt!

Random: impossible iff $(a-b)^2 \le 2(a+b)$

Semirandom: impossible if $(a-b)^2 \le C_a(a+b)$

where $C_a > 2$ for all $a > 2$

- No algorithm can robustly reach the threshold!
- First random-to-semirandom gap



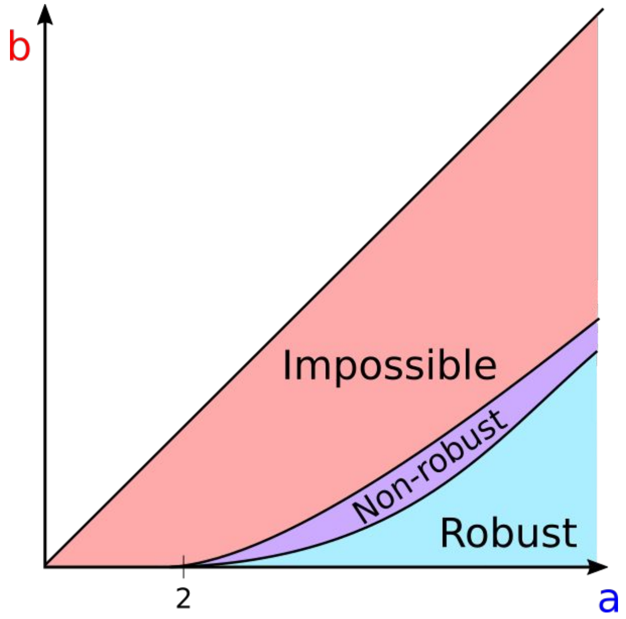
Main Result

Theorem: Partial recovery is strictly harder in the semirandom model than in the random model — ‘helpful’ changes can hurt!

Random: impossible iff $(a-b)^2 \leq 2(a+b)$

Semirandom: impossible if $(a-b)^2 \leq C_a(a+b)$

where $C_a > 2$ for all $a > 2$



- No algorithm can robustly reach the threshold!
- First random-to-semirandom gap
- Gap only exists for partial recovery

Can SDPs reach the threshold?

Can SDPs reach the threshold?

Our result: No algorithm for partial recovery can **robustly** reach the threshold

Can SDPs reach the threshold?

Our result: No algorithm for partial recovery can **robustly** reach the threshold

Doesn’t technically imply that SDPs cannot reach the threshold

- No proof that if SDP succeeds in random model, then it is robust
  (i.e. succeeds in the semirandom model for the same range of parameters a,b)

Can SDPs reach the threshold?

Our result: No algorithm for partial recovery can **robustly** reach the threshold

Doesn’t technically imply that SDPs cannot reach the threshold

- No proof that if SDP succeeds in random model, then it is robust
  (i.e. succeeds in the semirandom model for the same range of parameters a,b)

But it does give evidence that SDPs cannot reach the threshold

Can SDPs reach the threshold?

Our result: No algorithm for partial recovery can **robustly** reach the threshold

Doesn't technically imply that SDPs cannot reach the threshold

- No proof that if SDP succeeds in random model, then it is robust
  (i.e. succeeds in the semirandom model for the same range of parameters a,b)

But it does give evidence that SDPs cannot reach the threshold

- Formally: No [GV'15]-type SDP analysis succeeds up to threshold

Can SDPs reach the threshold?

Our result: No algorithm for partial recovery can **robustly** reach the threshold

Doesn't technically imply that SDPs cannot reach the threshold

- No proof that if SDP succeeds in random model, then it is robust
  (i.e. succeeds in the semirandom model for the same range of parameters a,b)

But it does give evidence that SDPs cannot reach the threshold

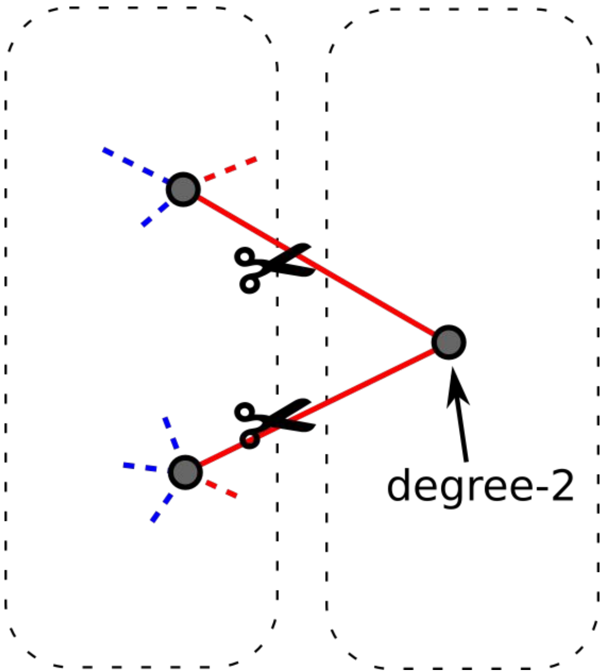
- Formally: No [GV'15]-type SDP analysis succeeds up to threshold

Additional evidence: statistical physics predicts (non-rigorous) that SDP misses the threshold [JMR'15]

Proof Idea: How can ‘helpful’ changes hurt?

Proof Idea: How can 'helpful' changes hurt?

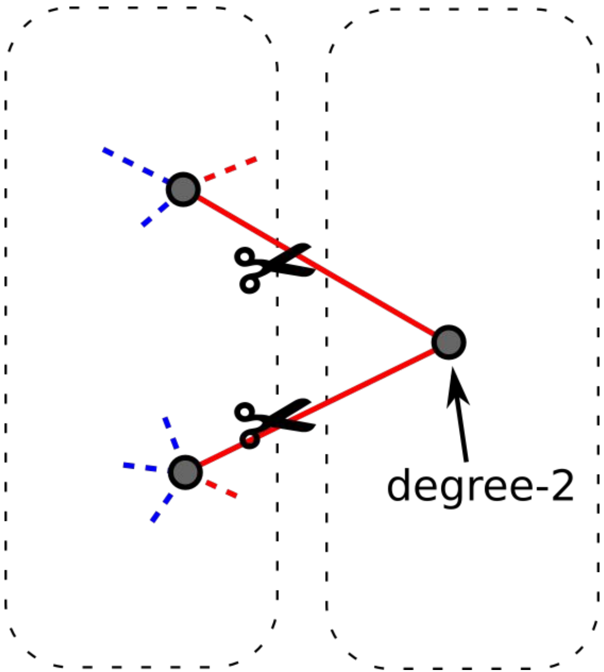
Our adversary: look for degree-2 nodes with 2 opposite-side neighbors; cut both edges



Proof Idea: How can ‘helpful’ changes hurt?

Our adversary: look for degree-2 nodes with 2 opposite-side neighbors; cut both edges

Sparse graph: this occurs often

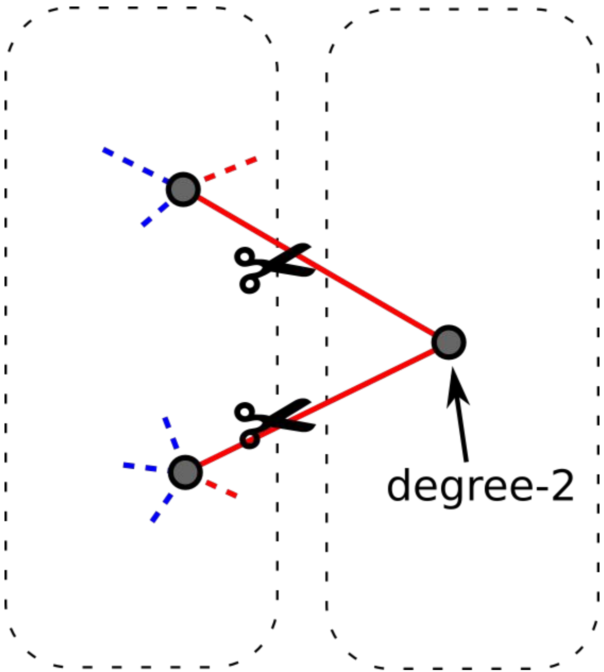


Proof Idea: How can ‘helpful’ changes hurt?

Our adversary: look for degree-2 nodes with 2 opposite-side neighbors; cut both edges

Sparse graph: this occurs often

“Long edge” within a community



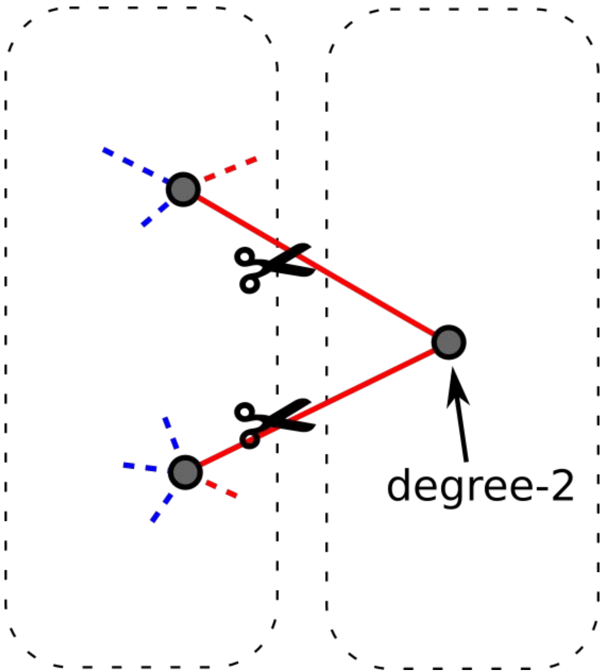
Proof Idea: How can 'helpful' changes hurt?

Our adversary: look for degree-2 nodes with 2 opposite-side neighbors; cut both edges

Sparse graph: this occurs often

"Long edge" within a community

We prove that this makes partial recovery strictly harder (information-theoretically)



Proof Idea: How can ‘helpful’ changes hurt?

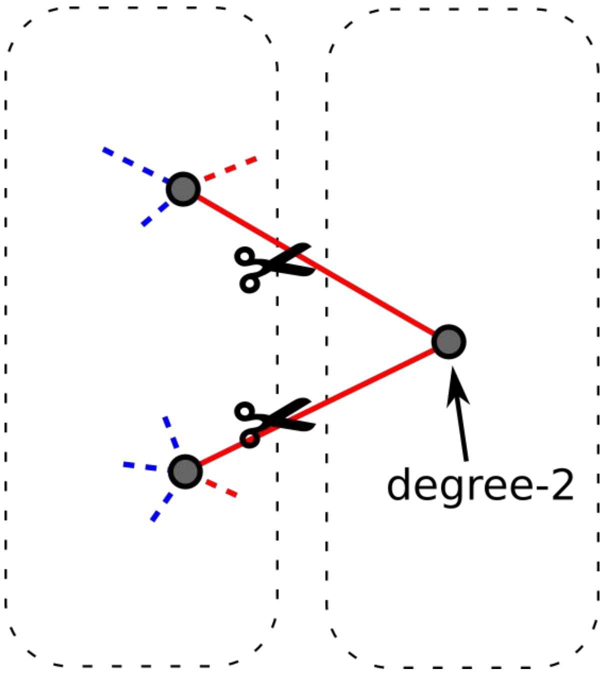
Our adversary: look for degree-2 nodes with 2 opposite-side neighbors; cut both edges

Sparse graph: this occurs often

“Long edge” within a community

We prove that this makes partial recovery strictly harder (information-theoretically)

Interpretation: algorithms reaching the threshold (e.g. linearized belief propagation) rely on the distribution of these structures in the noise



Proof Idea

Goal: show that with our adversary, partial recovery is impossible in some region strictly above the threshold

Proof Idea

Goal: show that with our adversary, partial recovery is impossible in some region strictly above the threshold

We adapt the original proof of [Mossel–Neeman–Sly ’13] that shows impossibility below the threshold (in the random model)

Proof Idea

Goal: show that with our adversary, partial recovery is impossible in some region strictly above the threshold

We adapt the original proof of [Mossel–Neeman–Sly ’13] that shows impossibility below the threshold (in the random model)

Sparse graphs are locally-tree-like

Proof Idea

Goal: show that with our adversary, partial recovery is impossible in some region strictly above the threshold

We adapt the original proof of [Mossel–Neeman–Sly ’13] that shows impossibility below the threshold (in the random model)

Sparse graphs are locally-tree-like

- A vertex’s O(log n)-radius neighborhood is a tree with high probability

Proof Idea

Goal: show that with our adversary, partial recovery is impossible in some region strictly above the threshold

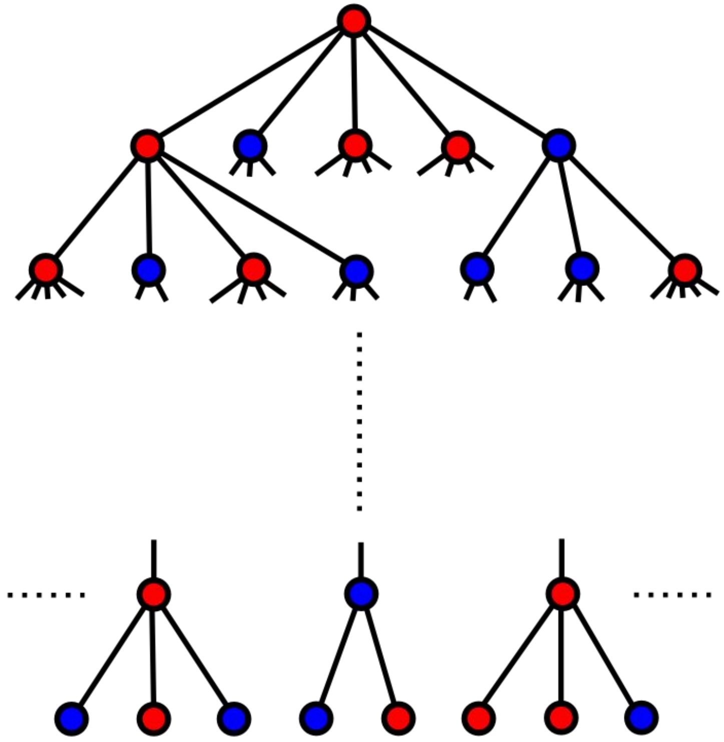
We adapt the original proof of [Mossel–Neeman–Sly '13] that shows impossibility below the threshold (in the random model)

Sparse graphs are locally-tree-like

- A vertex's O(log n)-radius neighborhood is a tree with high probability

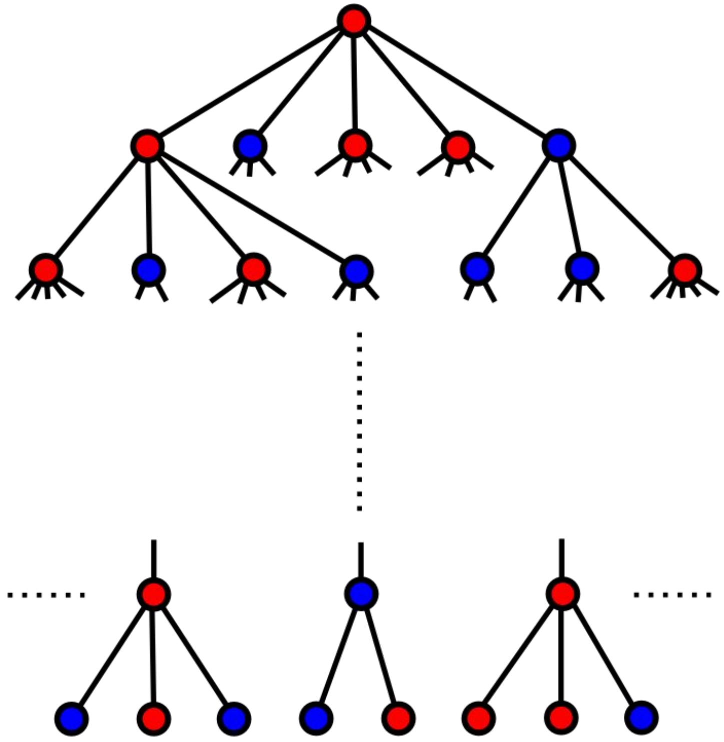
Use connection to ***broadcast tree model***

Broadcast Tree Model



Broadcast Tree Model

2 colors: red, blue (corresponding to 2 communities)

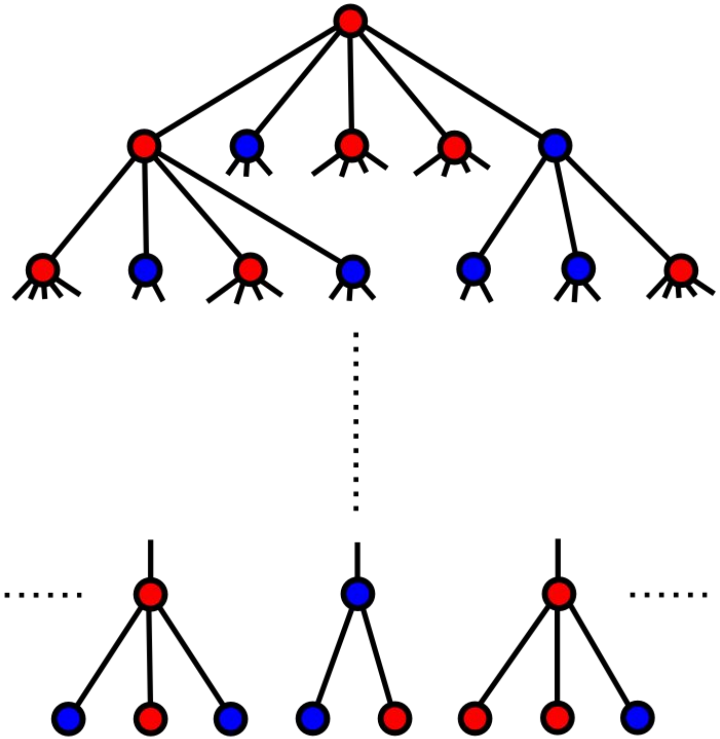


Broadcast Tree Model

2 colors: red, blue (corresponding to 2 communities)

Recursively, each node gives birth to:

- Pois(a/2) nodes of same color, and
- Pois(b/2) nodes of opposite color



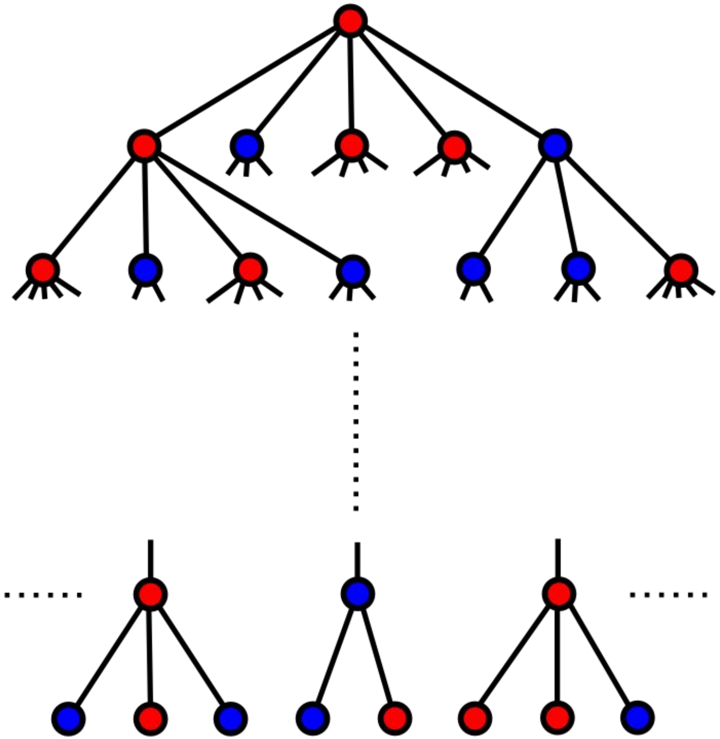
Broadcast Tree Model

2 colors: red, blue (corresponding to 2 communities)

Recursively, each node gives birth to:

- Pois(a/2) nodes of same color, and
- Pois(b/2) nodes of opposite color

(Resembles neighborhood of graph!)



Broadcast Tree Model

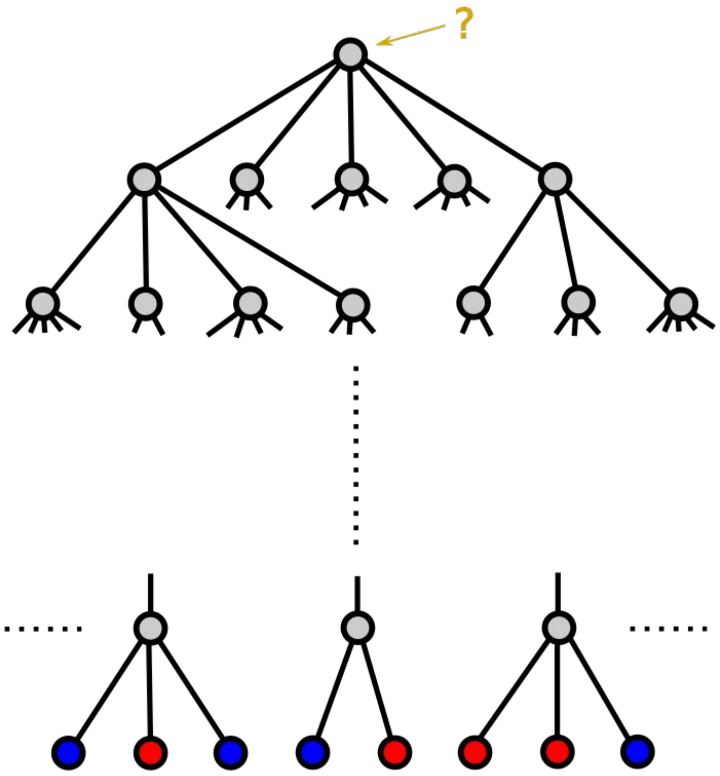
2 colors: red, blue (corresponding to 2 communities)

Recursively, each node gives birth to:

- Pois(a/2) nodes of same color, and
- Pois(b/2) nodes of opposite color

(Resembles neighborhood of graph!)

Q: When can you recover the root color from the leaf colors? (as tree depth $\to \infty$)



Broadcast Tree Model

2 colors: red, blue (corresponding to 2 communities)

Recursively, each node gives birth to:

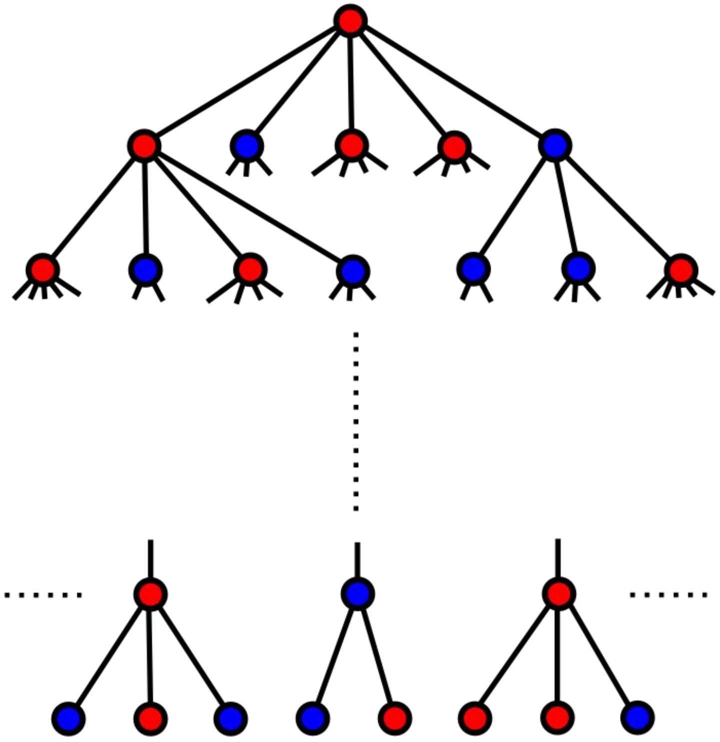
- Pois(a/2) nodes of same color, and
- Pois(b/2) nodes of opposite color

(Resembles neighborhood of graph!)

Q: When can you recover the root color from the leaf colors? (as tree depth $\to \infty$)

Answer: when $(a-b)^2 > 2(a+b)$ Look familiar?

[Kesten-Stigum '66, Evans-Kenyon-Peres-Schulman '00]



Summary

Summary

New phenomenon: random-to-semirandom gap (only in partial recovery)

Summary

New phenomenon: random-to-semirandom gap (only in partial recovery)

- Does this phenomenon occur elsewhere?

Summary

New phenomenon: random-to-semirandom gap (only in partial recovery)

- Does this phenomenon occur elsewhere?

Statistical physics (i.e. belief propagation) exactly achieves the recovery threshold

Summary

New phenomenon: random-to-semirandom gap (only in partial recovery)

- Does this phenomenon occur elsewhere?

Statistical physics (i.e. belief propagation) exactly achieves the recovery threshold

- But at what cost? Lacks robustness.

Summary

New phenomenon: random-to-semirandom gap (only in partial recovery)

- Does this phenomenon occur elsewhere?

Statistical physics (i.e. belief propagation) exactly achieves the recovery threshold

- But at what cost? Lacks robustness.

Convex optimization (i.e. SDP) falls slightly short of the threshold but holds onto robustness.

Summary

New phenomenon: random-to-semirandom gap (only in partial recovery)

- Does this phenomenon occur elsewhere?

Statistical physics (i.e. belief propagation) exactly achieves the recovery threshold

- But at what cost? Lacks robustness.

Convex optimization (i.e. SDP) falls slightly short of the threshold but holds onto robustness.

- Missing the threshold is necessary — robust problem is strictly harder.

Summary

New phenomenon: random-to-semirandom gap (only in partial recovery)

- Does this phenomenon occur elsewhere?

Statistical physics (i.e. belief propagation) exactly achieves the recovery threshold

- But at what cost? Lacks robustness.

Convex optimization (i.e. SDP) falls slightly short of the threshold but holds onto robustness.

- Missing the threshold is necessary — robust problem is strictly harder.

What price do we pay (in terms of robustness) in order to reach information-theoretic thresholds?

Summary

New phenomenon: random-to-semirandom gap (only in partial recovery)

- Does this phenomenon occur elsewhere?

Statistical physics (i.e. belief propagation) exactly achieves the recovery threshold

- But at what cost? Lacks robustness.

Convex optimization (i.e. SDP) falls slightly short of the threshold but holds onto robustness.

- Missing the threshold is necessary — robust problem is strictly harder.

What price do we pay (in terms of robustness) in order to reach information-theoretic thresholds?

Thanks! Questions?