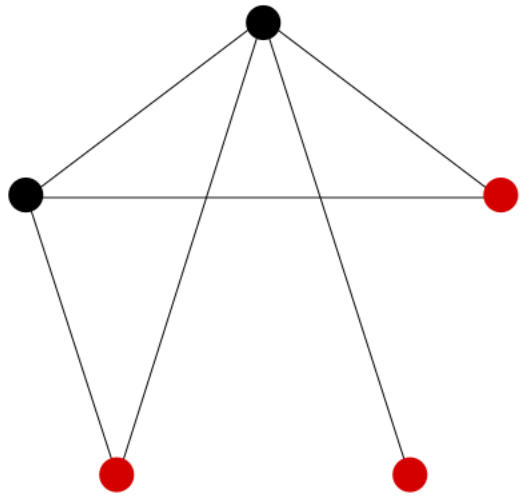
A New Algorithm for the Robust Semi-Random Independent Set Problem

Theo McKenzie
UC Berkeley

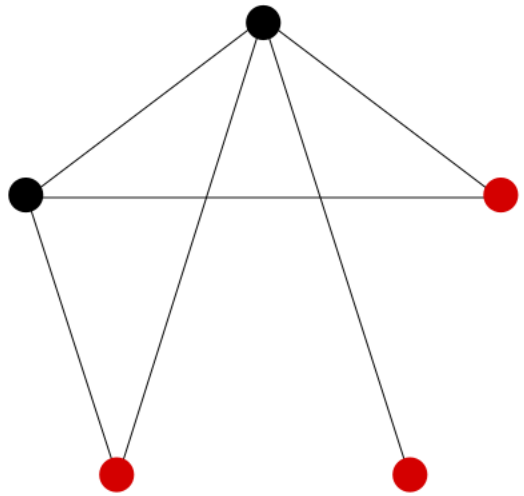
Joint with Hermish Mehta (UC Berkeley) and
Luca Trevisan (Bocconi University)

Maximum Independent Set Problem



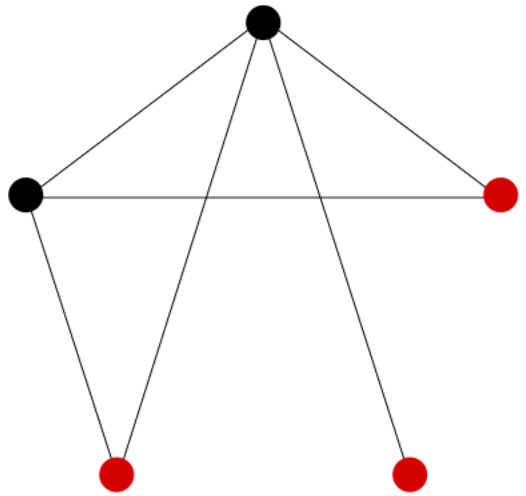
- Given a graph $G = (V, E)$ what is the maximum size independent set? The decision problem of whether there is an independent set of size at least k is on Richard Karp's original list of 21 NP complete problems.

Maximum Independent Set Problem



- Given a graph $G = (V, E)$ what is the maximum size independent set? The decision problem of whether there is an independent set of size at least k is on Richard Karp's original list of 21 NP complete problems.
- **[Håstad '96]** If there is an $n^{1-\epsilon}$ approximation algorithm for constant $\epsilon > 0$ in polynomial time, then P=NP.

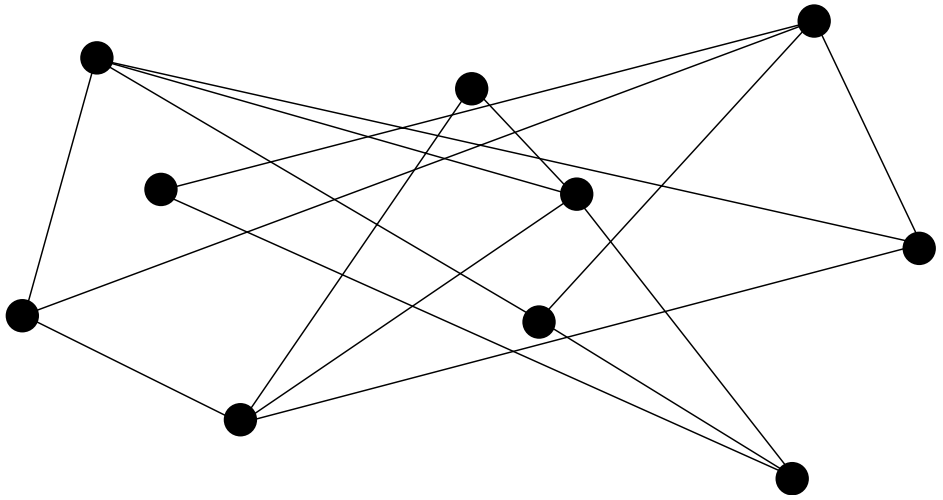
Maximum Independent Set Problem



- Given a graph $G = (V, E)$ what is the maximum size independent set? The decision problem of whether there is an independent set of size at least k is on Richard Karp's original list of 21 NP complete problems.
- **[Håstad '96]** If there is an $n^{1-\epsilon}$ approximation algorithm for constant $\epsilon > 0$ in polynomial time, then P=NP.
- This means that selecting a single vertex is not far from the best solution in polynomial time!

Average Case Analysis

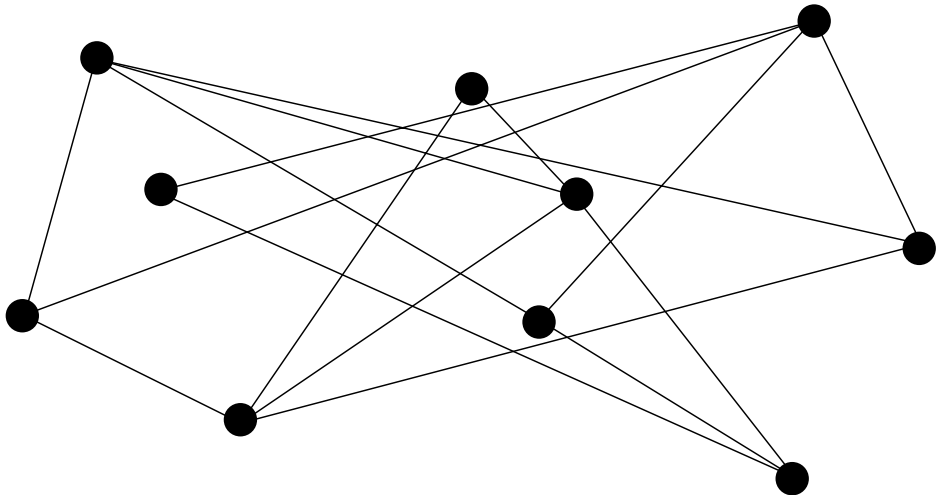
- However, this may be focusing on the idiosyncrasies of unlikely graphical models. Can we instead solve an “average” case?



Average Case Analysis

- However, this may be focusing on the idiosyncrasies of unlikely graphical models. Can we instead solve an “average” case?

- For example if each edge exists with probability *p* (an Erdös-Rényi $G(n,p)$ random graph) can I recover the maximal independent set?



Average Case Analysis Continued

- By doing a simple random greedy algorithm, with high probability we obtain a $\log_{\frac{1}{1-p}} n$ sized independent set.

Average Case Analysis Continued

- By doing a simple random greedy algorithm, with high probability we obtain a $\log_{\frac{1}{1-p}} n$ sized independent set.
- However, with high probability, the maximum independent set is of size $2\log_{\frac{1}{1-p}} n$.

Average Case Analysis Continued

- By doing a simple random greedy algorithm, with high probability we obtain a $\log_{\frac{1}{1-p}} n$ sized independent set.
- However, with high probability, the maximum independent set is of size $2\log_{\frac{1}{1-p}} n$.
- As this is the best known polynomial time algorithm, we can get a constant factor approximation, but not exact recovery.

Planted Independent Set

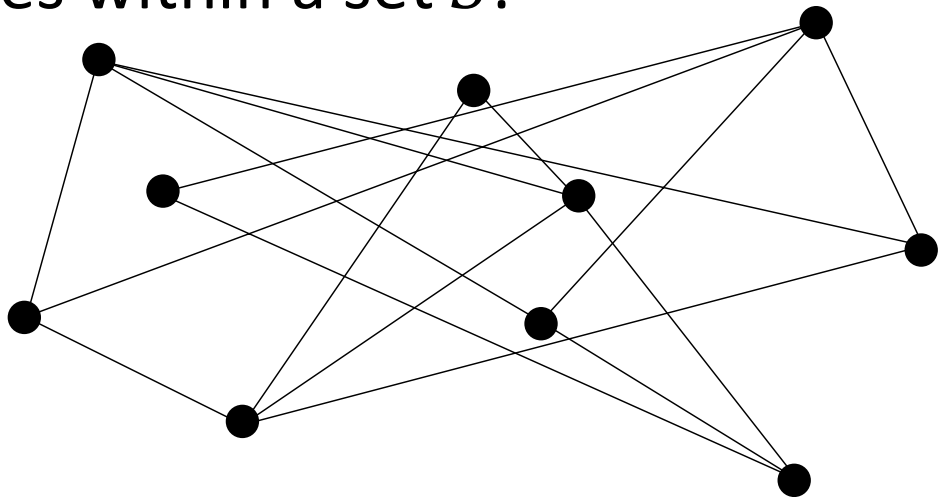
- We have more success when we plant an independent set within our random model.

Planted Independent Set

- We have more success when we plant an independent set within our random model.
- Specifically, we create a graph according to the $G(n, p)$ distribution, then delete all edges within a set S.

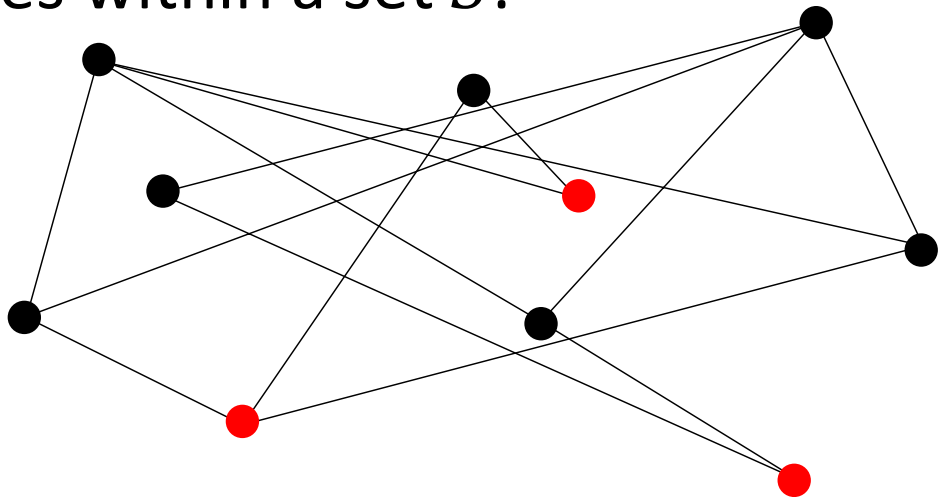
Planted Independent Set

- We have more success when we plant an independent set within our random model.
- Specifically, we create a graph according to the $G(n, p)$ distribution, then delete all edges within a set S.



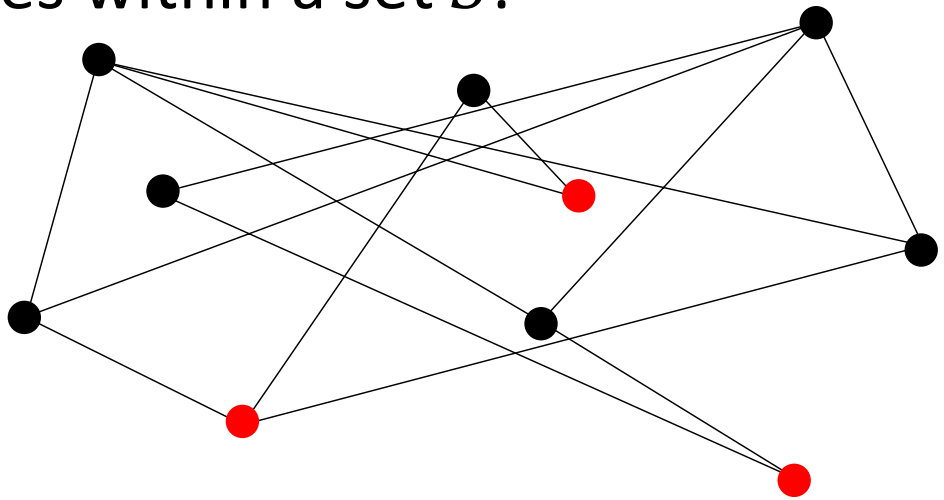
Planted Independent Set

- We have more success when we plant an independent set within our random model.
- Specifically, we create a graph according to the $G(n, p)$ distribution, then delete all edges within a set S.



Planted Independent Set

- We have more success when we plant an independent set within our random model.
- Specifically, we create a graph according to the $G(n, p)$ distribution, then delete all edges within a set S.



- **[Alon-Krivelevich-Sudakov '98]** If the planted set is of size $\Omega(\sqrt{n})$ and $p = 1/2$, then we can recover the set in polynomial time w.h.p.

Have We Assumed Too Much?

- Feige and Kilian believed that full recovery was possible without requiring the randomness of all non independent set edges. The only randomness we need is of edges adjacent to the planted independent set.

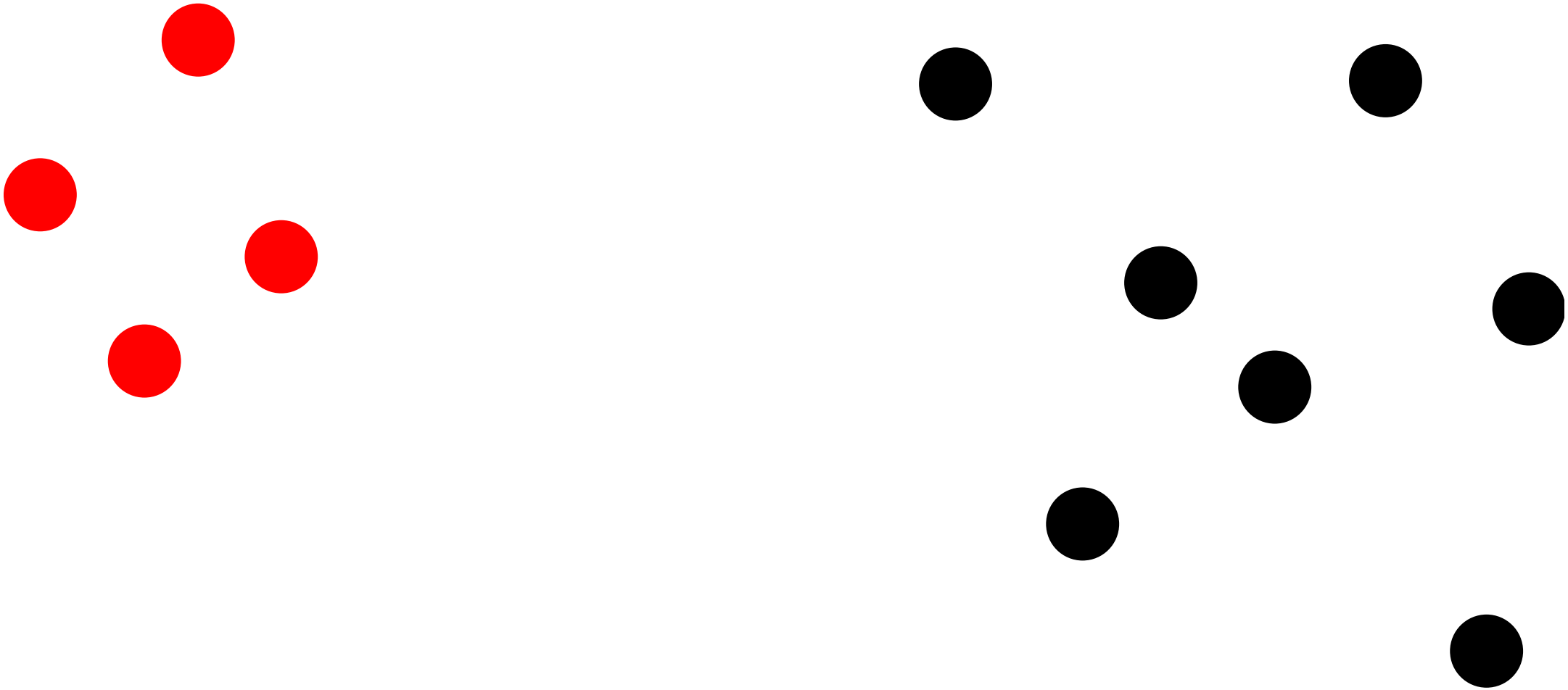
- Therefore they created a more general model.

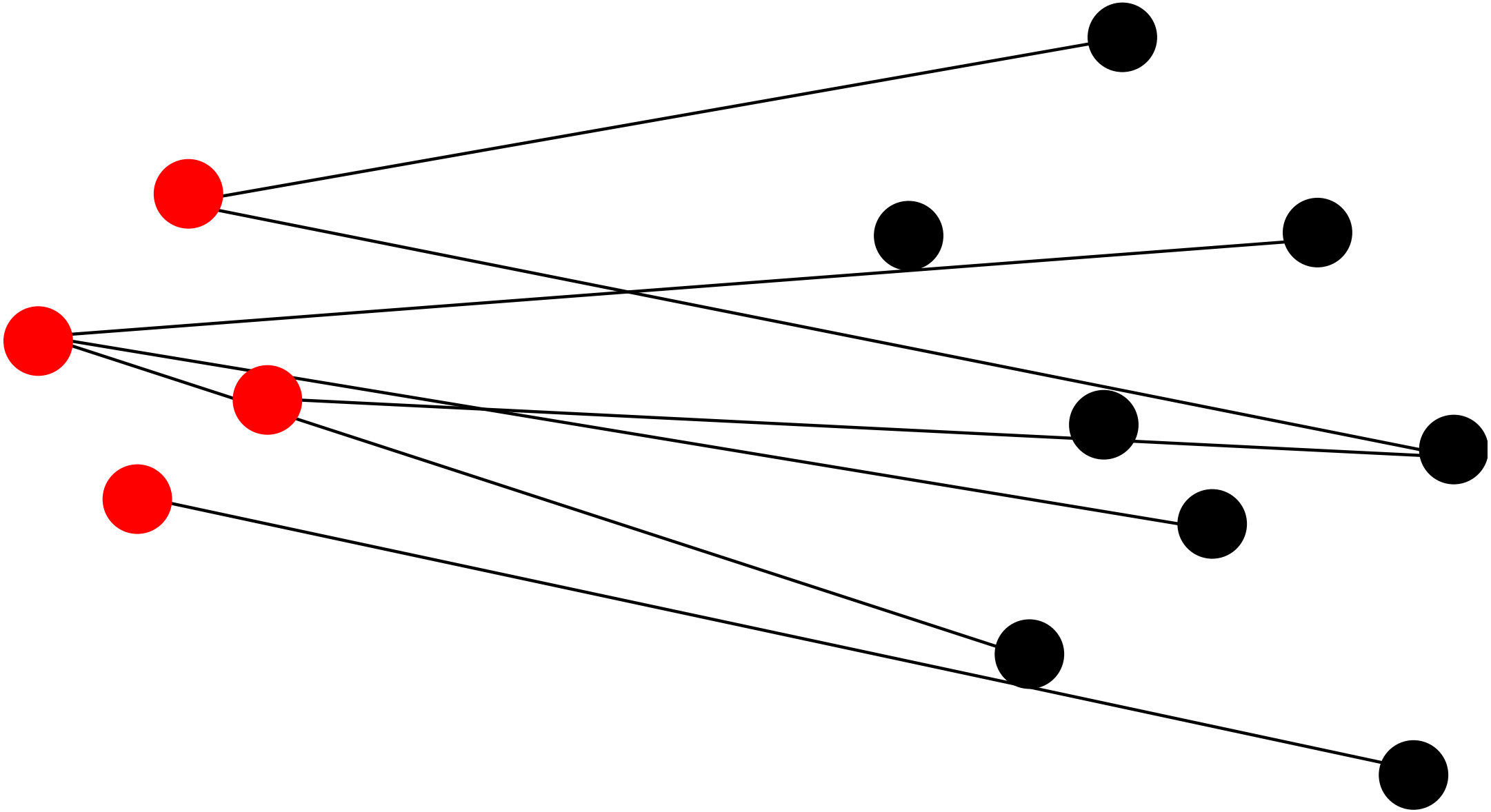
Feige and Kilian Model

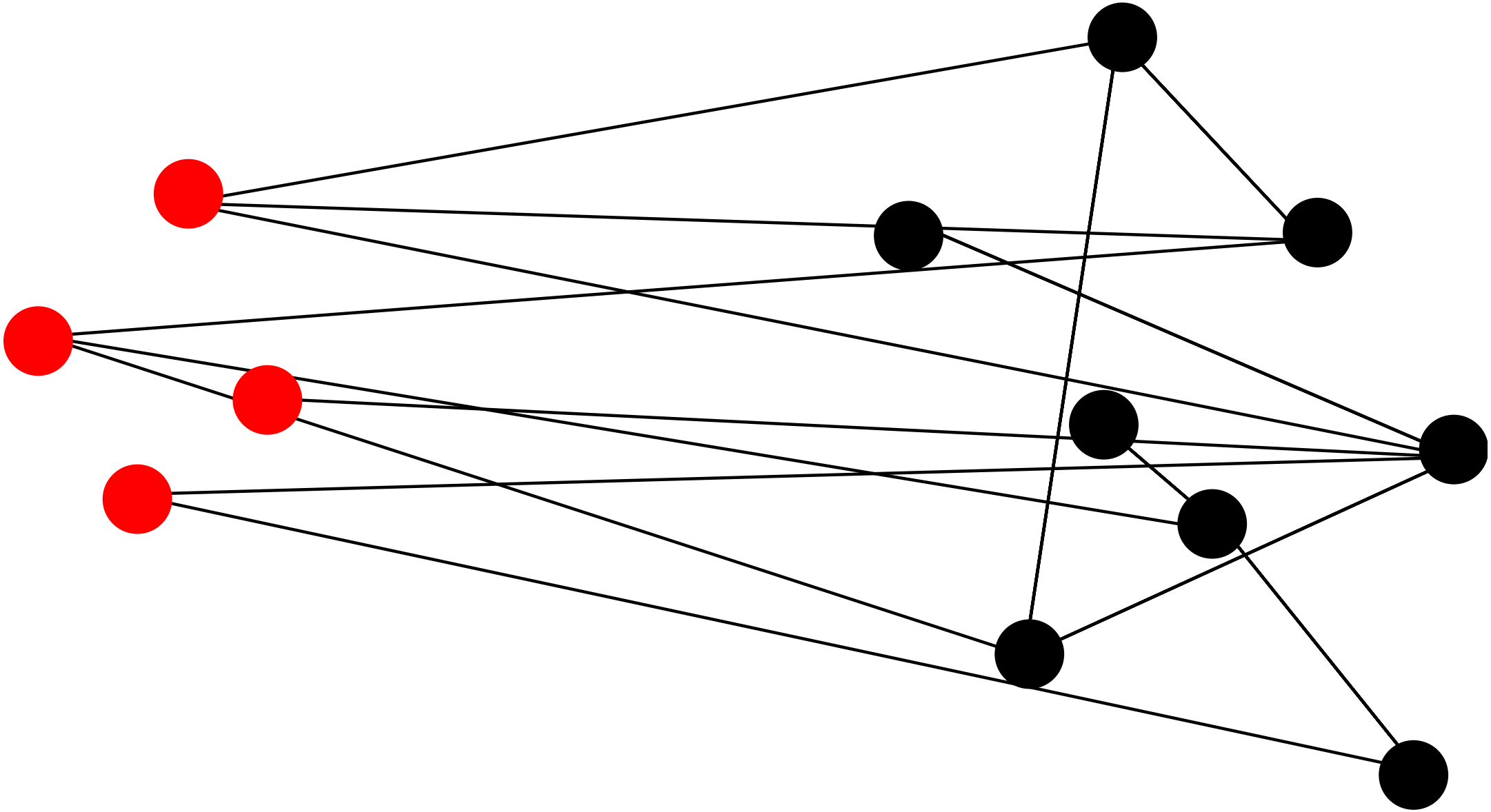
- Given a set $S \subset V$, to all vertex pairs of the form $(u, v) \in S \times \overline{S}$, we add an edge with probability p.

Feige and Kilian Model

- Given a set $S \subset V$, to all vertex pairs of the form $(u, v) \in S \times \overline{S}$, we add an edge with probability p.
- An adversary then adds edges to the model, as long as S remains an independent set.







Feige and Kilian Result

- **[Feige and Kilian '01]**
  - If $|S| = \alpha n$ and $p > ((1+\epsilon)\ln n)/(\alpha n)$, for constants $\epsilon, \alpha > 0$, w.h.p. it is possible to find an independent set of size **at least** αn utilizing a randomized polynomial time algorithm.

Feige and Kilian Result

- **[Feige and Kilian '01]**
  - If $|S| = \alpha n$ and $p > ((1+\epsilon)\ln n)/(\alpha n)$, for constants $\epsilon, \alpha > 0$, w.h.p. it is possible to find an independent set of size **at least** αn utilizing a randomized polynomial time algorithm.
  - Moreover if $p < ((1-\epsilon)\ln n)/(\alpha n)$, the problem is not solvable in polynomial time unless NP⊂BPP.

Feige and Kilian Result

- **[Feige and Kilian '01]**
  - If $|S| = \alpha n$ and $p > ((1+\epsilon)\ln n)/(\alpha n)$, for constants $\epsilon, \alpha > 0$, w.h.p. it is possible to find an independent set of size **at least** αn utilizing a randomized polynomial time algorithm.
  - Moreover if $p < ((1-\epsilon)\ln n)/(\alpha n)$, the problem is not solvable in polynomial time unless NP⊂BPP.
- Despite improvements in weaker models, until now there has been no improvement for the full Feige and Kilian model.

Our Results

- **[Our First Result]** Call $|S| = k$**.** $\exists c_1 > 0$ such that if $k > c_1 \frac{n^{2/3}}{p^{1/3}}$, w.h.p. it is possible to find an independent set of size at least $.99k$ utilizing a deterministic polynomial time algorithm.

Our Results

- **[Our First Result]** Call $|S| = k$**.** $\exists c_1 > 0$ such that if $k > c_1 \frac{n^{2/3}}{p^{1/3}}$, w.h.p. it is possible to find an independent set of size at least $.99k$ utilizing a deterministic polynomial time algorithm.
- **[Second Result]** $\exists c_2 > 0$ such that if $k > c_2 \frac{n^{2/3}}{p}$, w.h.p. this independent set can be increased to an independent set of size at least k (also deterministic polynomial time).

Comparison of Results

- For full recovery with $k = \alpha n$, the Feige Kilian algorithm works for $p = \Omega(\ln n/n)$ while ours only works for $p = \Omega(1/n^{1/3})$.

Comparison of Results

- For full recovery with $k = \alpha n$, the Feige Kilian algorithm works for $p = \Omega(\ln n/n)$ while ours only works for $p = \Omega(1/n^{1/3})$.

- For $k = \alpha n$, our approximation algorithm works for $p = \Omega(1/n)$.

Comparison of Results

- For full recovery with $k = \alpha n$, the Feige Kilian algorithm works for $p = \Omega(\ln n/n)$ while ours only works for $p = \Omega(1/n^{1/3})$.

- For $k = \alpha n$, our approximation algorithm works for $p = \Omega(1/n)$.

- For $k = o(n)$ all of our results are new. For example, for constant p, we only require $k = \Omega(n^{2/3})$ for approximation and recovery.

“Crude” SDPs

- Semirandom Unique Games **[Kolla-Makarychev-Makarychev ‘11]**
- Semirandom Small Set Expansion **[Makarychev-Makarychev-Vijayaraghavan ‘12]**

“Crude” SDPs

[Feige-Krauthgamer ‘00] Lovász Theta Function can be used to solve the “traditional” planted clique problem.

$$\begin{aligned} \text{maximize} \quad & \sum_{u,v} \langle x_u, x_v \rangle \\ \text{subject to} \quad & \\ & \sum_u ||x_u||^2 = 1 \\ & \langle x_u, x_v \rangle = 0, \forall (u,v) \in E \end{aligned}$$

“Crude” SDPs

[Feige-Krauthgamer ‘00] Lovász Theta Function can be used to solve the “traditional” planted clique problem.

$$\begin{aligned} \text{maximize} \quad & \sum_{u,v} \langle x_u, x_v \rangle \\ \text{subject to} \quad & \\ & \sum_u ||x_u||^2 = 1 \\ & \langle x_u, x_v \rangle = 0, \forall (u,v) \in E \end{aligned}$$

- The “hidden” solution is to place all vertices in S at the same vector x where $\|x\| = 1/\sqrt{k}$.

“Crude” SDPs

[Feige-Krauthgamer ‘00] Lovász Theta Function can be used to solve the “traditional” planted clique problem.

$$\begin{aligned} \text{maximize} \quad & \sum_{u,v} \langle x_u, x_v \rangle \\ \text{subject to} \quad & \\ & \sum_u ||x_u||^2 = 1 \\ & \langle x_u, x_v \rangle = 0, \forall (u,v) \in E \end{aligned}$$

- The “hidden” solution is to place all vertices in S at the same vector x where $\|x\| = 1/\sqrt{k}$.
- However in the Feige and Kilian model, the adversary can place in $V - S$ a graph that cannot be well approximated, so the optimal solution “ignores” the planted set and sends it to 0.

“Crude” SDPs

[Feige-Krauthgamer ‘00] Lovász Theta Function can be used to solve the “traditional” planted clique problem.

$$\begin{aligned} \text{maximize} \quad & \sum_{u,v} \langle x_u, x_v \rangle \\ \text{subject to} \quad & \\ & \sum_u ||x_u||^2 = 1 \\ & \langle x_u, x_v \rangle = 0, \forall (u,v) \in E \end{aligned}$$

- The “hidden” solution is to place all vertices in S at the same vector x where $\|x\| = 1/\sqrt{k}$.
- However in the Feige and Kilian model, the adversary can place in $V - S$ a graph that cannot be well approximated, so the optimal solution “ignores” the planted set and sends it to 0.

“Crude” SDP

$$\begin{aligned} \text{maximize} \quad & \sum_{u,v} \langle x_u, x_v \rangle \\ \text{subject to} \quad & \\ & ||x_u||^2 = 1, \forall u \in V \\ & \langle x_u, x_v \rangle = 0, \forall (u,v) \in E \end{aligned}$$

“Crude” SDPs

[Feige-Krauthgamer ‘00] Lovász Theta Function can be used to solve the “traditional” planted clique problem.

$$\begin{aligned} \text{maximize} \quad & \sum_{u,v} \langle x_u, x_v \rangle \\ \text{subject to} \quad & \\ & \sum_u ||x_u||^2 = 1 \\ & \langle x_u, x_v \rangle = 0, \forall (u,v) \in E \end{aligned}$$

- The “hidden” solution is to place all vertices in S at the same vector x where $\|x\| = 1/\sqrt{k}$.
- However in the Feige and Kilian model, the adversary can place in $V - S$ a graph that cannot be well approximated, so the optimal solution “ignores” the planted set and sends it to 0.

“Crude” SDP

$$\begin{aligned} \text{maximize} \quad & \sum_{u,v} \langle x_u, x_v \rangle \\ \text{subject to} \quad & \\ & ||x_u||^2 = 1, \forall u \in V \\ & \langle x_u, x_v \rangle = 0, \forall (u,v) \in E \end{aligned}$$

- We sacrifice our program being a direct relaxation but maintain information about all vertices.

“Crude” SDPs

[Feige-Krauthgamer ‘00] Lovász Theta Function can be used to solve the “traditional” planted clique problem.

$$\begin{aligned} \text{maximize} \quad & \sum_{u,v} \langle x_u, x_v \rangle \\ \text{subject to} \quad & \\ & \sum_u ||x_u||^2 = 1 \\ & \langle x_u, x_v \rangle = 0, \forall (u,v) \in E \end{aligned}$$

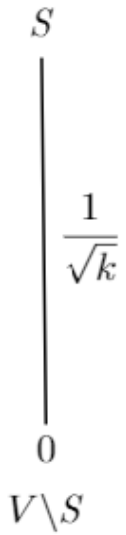
- The “hidden” solution is to place all vertices in S at the same vector x where $\|x\| = 1/\sqrt{k}$.
- However in the Feige and Kilian model, the adversary can place in $V - S$ a graph that cannot be well approximated, so the optimal solution “ignores” the planted set and sends it to 0.

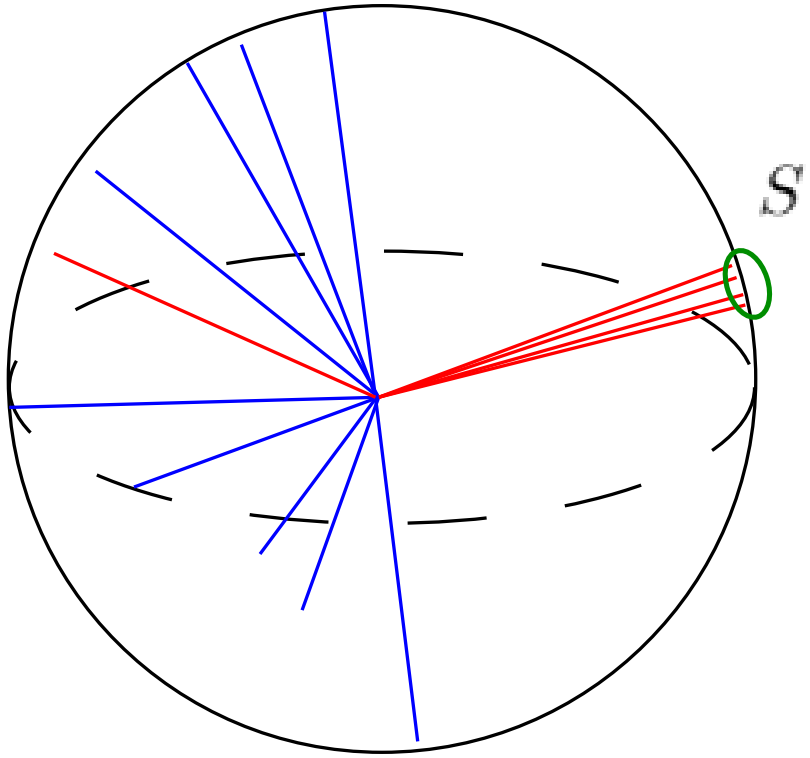
“Crude” SDP

$$\begin{aligned} \text{maximize} \quad & \sum_{u,v} \langle x_u, x_v \rangle \\ \text{subject to} \quad & \\ & ||x_u||^2 = 1, \forall u \in V \\ & \langle x_u, x_v \rangle = 0, \forall (u,v) \in E \end{aligned}$$

- We sacrifice our program being a direct relaxation but maintain information about all vertices.
- Our goal is to show that the planted set will form a cluster separated from all other vertices.

Comparison of Optimal Solutions





Analysis of Clustering

- We compare our optimal solution to an altered solution, where we take all vectors corresponding to vertices of our planted set and set them to a vector e orthogonal to all other vectors.

Analysis of Clustering

- We compare our optimal solution to an altered solution, where we take all vectors corresponding to vertices of our planted set and set them to a vector e orthogonal to all other vectors.
- Comparing where these two cost functions differ, for the solution to be optimal,

$$\left(\sum_{u,v\in S\times S}\langle x_u^*, x_v^*\rangle\right) + 2\left(\sum_{u,v\in S\times \bar{S}}\langle x_u^*, x_v^*\rangle\right) \geq k^2$$

where x_u^* is the vector corresponding to u in the optimal solution.

Analysis of Clustering

- We compare our optimal solution to an altered solution, where we take all vectors corresponding to vertices of our planted set and set them to a vector e orthogonal to all other vectors.
- Comparing where these two cost functions differ, for the solution to be optimal,

$$\left(\sum_{u,v\in S\times S}\langle x_u^*, x_v^*\rangle\right) + 2\left(\sum_{u,v\in S\times \bar{S}}\langle x_u^*, x_v^*\rangle\right) \geq k^2$$

  where x_u^* is the vector corresponding to u in the optimal solution.
- To show that the vectors corresponding to S cluster, we bound the second sum.

Analysis of Clustering

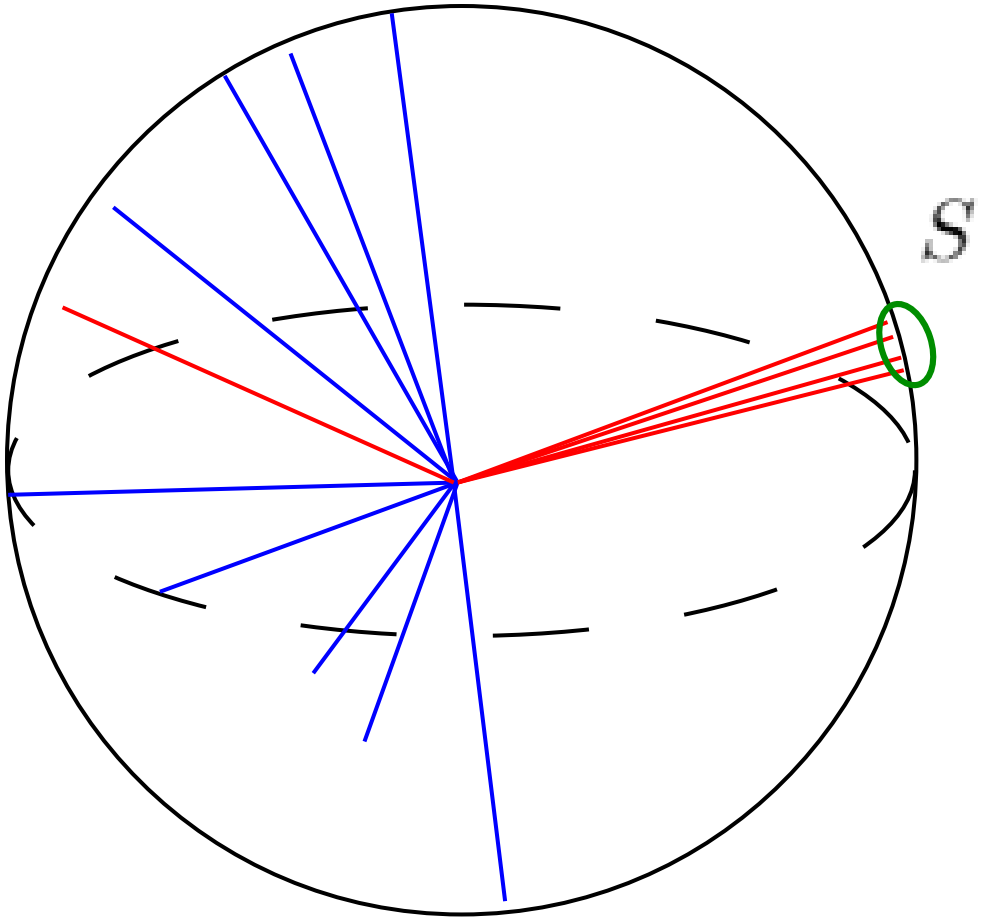
This is the key advantage to crude SDPs. Regardless of the configuration, we can deduce information about S.

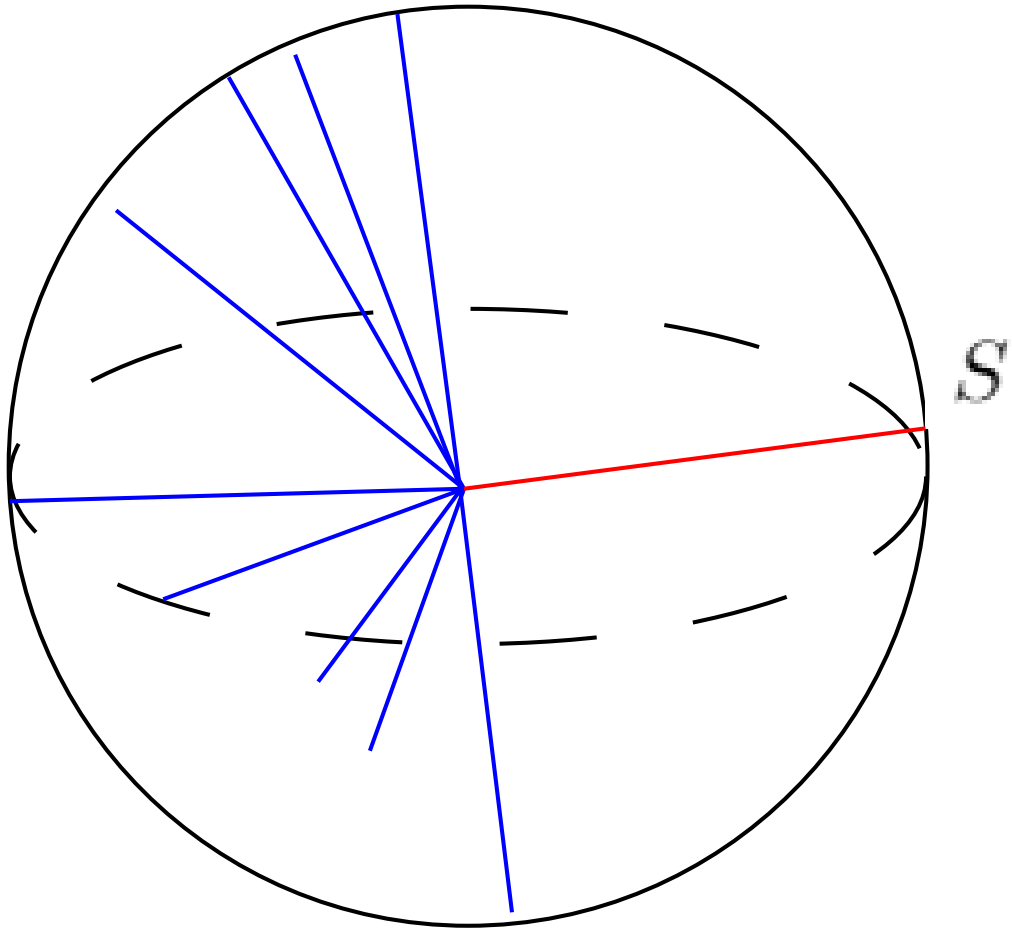
- We compare our optimal solution to an [illegible] solution, where we take all vectors corresponding to vertice[illegible]ur planted set and set them to a vector e orthogonal to all oth[illegible]ectors.
- Comparing where these two cost funct[illegible]ns differ, for the solution to be optimal,

$$\left(\sum_{u,v\in S\times S}\langle x_u^*, x_v^*\rangle\right) + 2\left(\sum_{u,v\in S\times \bar{S}}\langle x_u^*, x_v^*\rangle\right) \geq k^2$$

  where x_u^* is the vector corresponding to u in the optimal solution.
- To show that the vectors corresponding to S cluster, we bound the second sum.

Optimal vs. Adjusted Solutions





Bounding Size of Sum

- We wish to show that $2\left(\sum_{u,v\in S\times\bar{S}}\langle x_u^*, x_v^*\rangle\right)$ is small.

Bounding Size of Sum

- We wish to show that $2\left(\sum_{u,v\in S\times\bar{S}}\langle x_u^*, x_v^*\rangle\right)$ is small.
- To do this, we show that the inner product of the randomly added edges approximates the inner product of the entire barrier. If F is the set of randomly added edges, we have

$$\max\left|\left(\sum_{u,v\in S\times\bar{S}}\langle x_u, x_v\rangle\right)\right| = \max\left|\left(\sum_{u,v\in S\times\bar{S}}\langle x_u, x_v\rangle\right) - \frac{1}{p}\left(\sum_{u,v\in F}\langle x_u, x_v\rangle\right)\right|$$

Grothendieck Inequality

- As these are unit vectors, we can use Grothendieck's inequality to discretize our vectors.

$$\max \left| \left(\sum_{u,v \in S \times \bar{S}} \langle x_u, x_v \rangle \right) - \frac{1}{p} \left(\sum_{u,v \in F} \langle x_u, x_v \rangle \right) \right|$$

$$\leq c \max_{x_1,\ldots,x_n,y_1,\ldots,y_n \in \{\pm 1\}^{2n}} \left| \left(\sum_{u,v \in S \times \bar{S}} x_u y_v \right) - \frac{1}{p} \left(\sum_{u,v \in F} x_u y_v \right) \right|$$

where c is a constant less than 2.

- By union bounding over the 2^{2n} possible choices of the x_u and y_u for each vertex and a standard Chernoff bound, we show that with high probability,

$$\max_{x_1,\ldots,x_n,y_1,\ldots,y_n\in\{\pm 1\}^{2n}} \left| \left(\sum_{u,v\in S\times\bar{S}} x_u y_v \right) - \frac{1}{p}\left(\sum_{u,v\in F} x_u y_v \right) \right| = O\left(\frac{n\sqrt{k}}{\sqrt{p}} \right)$$

- By union bounding over the 2^{2n} possible choices of the x_u and y_u for each vertex and a standard Chernoff bound, we show that with high probability,

$$\max_{x_1,\ldots,x_n,y_1,\ldots,y_n\in\{\pm 1\}^{2n}} \left| \left(\sum_{u,v\in S\times\bar{S}} x_u y_v \right) - \frac{1}{p} \left(\sum_{u,v\in F} x_u y_v \right) \right| = O\left(\frac{n\sqrt{k}}{\sqrt{p}} \right)$$

- As the second sum is 0, we have that with high probability, over any configuration of vectors,

$$\left| \sum_{u,v\in S\times\bar{S}} \langle x_u, x_v \rangle \right| = O\left(\frac{n\sqrt{k}}{\sqrt{p}} \right)$$

$$\left(\sum_{u,v\in S\times S}\langle x_u^*, x_v^*\rangle\right) + 2\left(\sum_{u,v\in S\times \bar{S}}\langle x_u^*, x_v^*\rangle\right) \geq k^2$$

$$\left(\sum_{u,v\in S\times S}\langle x_u^*, x_v^*\rangle\right) + 2\left(\sum_{u,v\in S\times \bar{S}}\langle x_u^*, x_v^*\rangle\right) \geq k^2$$

- Now knowing that the second sum is $O\left(\frac{n\sqrt{k}}{\sqrt{p}}\right)$, the average inner product between vertices of S is $1-O\left(\frac{n}{k^{3/2}\sqrt{p}}\right)$.

$$\left(\sum_{u,v\in S\times S}\langle x_u^*, x_v^*\rangle\right) + 2\left(\sum_{u,v\in S\times \bar{S}}\langle x_u^*, x_v^*\rangle\right) \geq k^2$$

- Now knowing that the second sum is $O\left(\frac{n\sqrt{k}}{\sqrt{p}}\right)$, the average inner product between vertices of S is $1-O\left(\frac{n}{k^{3/2}\sqrt{p}}\right)$.
- Using a Markov inequality, if $k \geq c_1 \frac{n^{2/3}}{p^{1/3}}$, $\exists u \in S$ such that $.99k$ vertices of S have an inner product $> 1/\sqrt{2}$ with u.

$$\left(\sum_{u,v\in S\times S}\langle x_u^*, x_v^*\rangle\right) + 2\left(\sum_{u,v\in S\times \bar{S}}\langle x_u^*, x_v^*\rangle\right) \geq k^2$$

- Now knowing that the second sum is $O\left(\frac{n\sqrt{k}}{\sqrt{p}}\right)$, the average inner product between vertices of S is $1-O\left(\frac{n}{k^{3/2}\sqrt{p}}\right)$.
- Using a Markov inequality, if $k \geq c_1 \frac{n^{2/3}}{p^{1/3}}$, $\exists u \in S$ such that $.99k$ vertices of S have an inner product $> 1/\sqrt{2}$ with u.
- None of the vectors in this cluster are orthogonal, so these vertices form an independent set.

$$\left(\sum_{u,v\in S\times S}\langle x_u^*, x_v^*\rangle\right) + 2\left(\sum_{u,v\in S\times \bar{S}}\langle x_u^*, x_v^*\rangle\right) \geq k^2$$

- Now knowing that the second sum is $O\left(\frac{n\sqrt{k}}{\sqrt{p}}\right)$, the average inner product between vertices of S is $1-O\left(\frac{n}{k^{3/2}\sqrt{p}}\right)$.
- Using a Markov inequality, if $k \geq c_1 \frac{n^{2/3}}{p^{1/3}}$, $\exists u \in S$ such that $.99k$ vertices of S have an inner product $> 1/\sqrt{2}$ with u.
- None of the vectors in this cluster are orthogonal, so these vertices form an independent set.
- If we run our SDP then take the list of clusters around each vertex, each of these is independent and the one corresponding to u is of size $.99k$.

Exact Recovery

- With the slightly stronger condition that $k = \Omega\left(\frac{n^{2/3}}{p}\right)$ we add a greedy step: for each cluster on our list, add all vertices with no edges to the cluster.

Exact Recovery

- With the slightly stronger condition that $k = \Omega\left(\frac{n^{2/3}}{p}\right)$ we add a greedy step: for each cluster on our list, add all vertices with no edges to the cluster.
- With high probability, no point in $V \backslash S$ has less than $.01k$ neighbors in S, so u's cluster will become S with this greedy step.

Exact Recovery

- With the slightly stronger condition that $k = \Omega\left(\frac{n^{2/3}}{p}\right)$ we add a greedy step: for each cluster on our list, add all vertices with no edges to the cluster.
- With high probability, no point in $V \backslash S$ has less than $.01k$ neighbors in S, so u's cluster will become S with this greedy step.
- Remember Feige and Kilian's result that when $p < ((1-\epsilon)\ln n)/k$, the problem is not solvable in polynomial time unless NP⊂BPP. Therefore we do not hope for exact recovery under the full generality of the original constraints.

Recovery of Original Set

- If we are given a random vertex of S, we can analyze our clusters to recover S exactly.

Further Questions

- Is there a matching hardness result for the size of k?
  - **[Steinhardt '17]** For $p = \frac{1}{2}$ we must have $k = \Omega(\sqrt{n})$ to solve in poly time.
- Can we use some of the extra partitioning steps of the Feige and Kilian algorithm to further improve the analysis?

Thank you!

- Thank you to the NSF, Ford Foundation and UC Berkeley for funding this research and my travel.