How to Reliably Find a Hidden Clique

Brendan Ames

Department of Mathematics
The University of Alabama

UA Applied Math Seminar
April 14, 2017

Cliques of a graph

Given graph $G = (V, E)$, a **clique** of G is a pairwise adjacent subset of V.

The vertex set $C \subseteq V$ is a clique of G if $uv \in E$ for all $u, v \in C$.

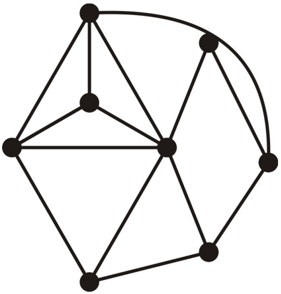
The subgraph $G(C)$ induced by C is **complete**.

Cliques of a graph

Given graph $G = (V, E)$, a **clique** of G is a pairwise adjacent subset of V.

The vertex set $C \subseteq V$ is a clique of G if $uv \in E$ for all $u, v \in C$.

The subgraph $G(C)$ induced by C is **complete**.

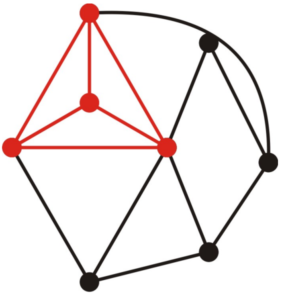


Cliques of a graph

Given graph $G = (V, E)$, a **clique** of G is a pairwise adjacent subset of V.

The vertex set $C \subseteq V$ is a clique of G if $uv \in E$ for all $u, v \in C$.

The subgraph $G(C)$ induced by C is **complete**.



The Clique problem

Optimization version: Find the clique of G of maximum size. Size of the largest clique is the **clique number** $\omega(G)$.

Decision version: Given graph G, integer k: does G contain a clique of cardinality at least k.

Complexity: NP-complete, cannot approximate within a ratio of $N^{1-\epsilon}$ for any $\epsilon > 0$.

Many applications: communication, biological, and social networks. Find large group of related objects.

The planted case

Hardness results are worst case.

There should be instances we should be able to solve efficiently.

In particular, if G has a clique of size k, we should be able to find it if k is large.

The planted case

Hardness results are worst case.

There should be instances we should be able to solve efficiently.

In particular, if G has a clique of size k, we should be able to find it if k is large.

Alon et al. 1998, Feige and Krauthgamer 2000: if $k \geq \Omega(\sqrt{N})$ and all other edges are added independently at random then we can find the maximum clique in polynomial time.

A more general model?

These recovery guarantees rely heavily on the fact that G is an undirected graph:

- e.g., symmetry of $\boldsymbol{A_G}$, the fact that a stable set of $\bar{G}$ is a clique of G, etc.

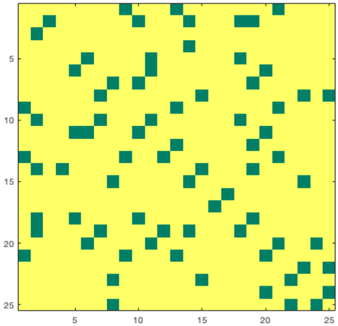
Would like an approach that translates to finding other "clique-like" objects with minimal effort.

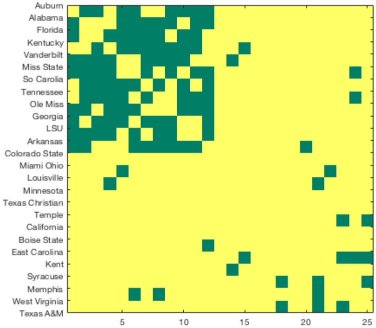
e.g., the maximum biclique of a bipartite graph, fully dense block in a matrix.

Example: Community Detection in Social Networks

NCAA forms a social network. Schools are “friends” if football teams play each other at least once (here in Fall 2000).

A random selection of teams should be unstructured (left), but the network does contain community structure via athletic conferences (right).

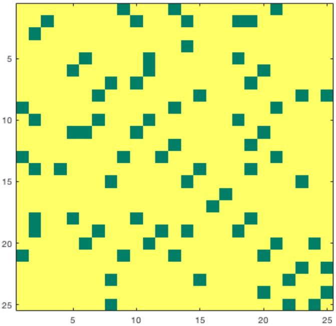


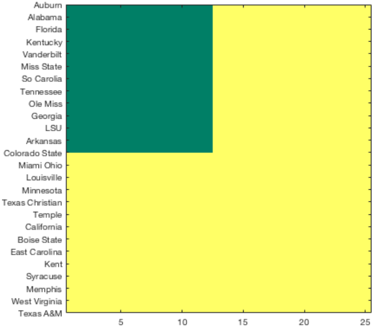


Example: Community Detection in Social Networks

NCAA forms a social network. Schools are "friends" if football teams play each other at least once (here in Fall 2000).

A random selection of teams should be unstructured (left), but the network does contain community structure via athletic conferences (right).

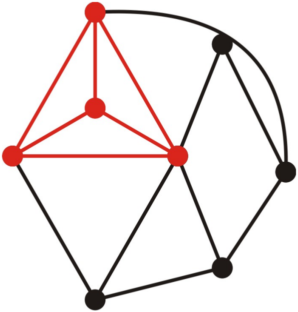


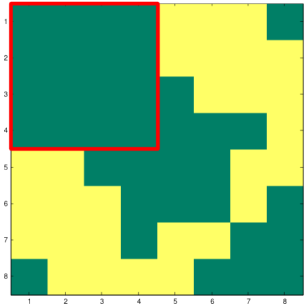


Cliques and low-rank matrices

Every clique C (with characteristic vector $\boldsymbol{v}$) of the graph $G = (V, E)$ defines a rank-one matrix by $\boldsymbol{X} = \boldsymbol{v}\boldsymbol{v}^T$.

Moreover, nonzero entries of X form a $|C| \times |C|$ rank-one block in $\boldsymbol{A_G} + I$.





Clique as rank minimization

G has a clique of cardinality at least k if and only if there exists rank-one symmetric binary matrix $\boldsymbol{X}$ such that

$$\sum \sum X_{ij} \geq k^2$$

$$X_{ij} = 0 \quad \forall\, ij \notin E,\; i \neq j.$$

Otherwise $\omega(G) < k$.

Therefore **Clique** is equivalent to the rank minimization problem:

$$\min_{\substack{\boldsymbol{X} \in \{0,1\}^{V \times V} \\ \boldsymbol{X} \in \Sigma^V}} \left\{ \operatorname{rank}(\boldsymbol{X}) : \boldsymbol{e}^T \boldsymbol{X} \boldsymbol{e} \geq k^2, X_{ij} = 0 \text{ if } (i,j) \in \tilde{E} \right\}$$

where $\tilde{E} = V \times V - \left\{ E \cup \{(u,u) : u \in V\} \right\}$.

Rank minimization

Affine rank minimization problem: find matrix with minimum rank satisfying linear constraints:

$$\min\{\operatorname{rank}(\boldsymbol{X}) : \mathcal{A}(\boldsymbol{X}) = \boldsymbol{b}\}.$$

Well-known to be NP-hard.

Relax $\operatorname{rank}(\boldsymbol{X})$ with nuclear norm $\|\boldsymbol{X}\|_* = \sigma_1(\boldsymbol{X}) + \cdots + \sigma_N(\boldsymbol{X})$:

$$\operatorname{rank}(\boldsymbol{X}) = \|\boldsymbol{\sigma}(\boldsymbol{X})\|_0, \quad \|\boldsymbol{X}\|_* = \|\boldsymbol{\sigma}(\boldsymbol{X})\|_1.$$

If $\mathcal{A}$ satisfies certain "niceness" conditions then the minimum nuclear norm solution is the minimum rank solution.

Nuclear norm relaxation of Clique

We solve the convex relaxation

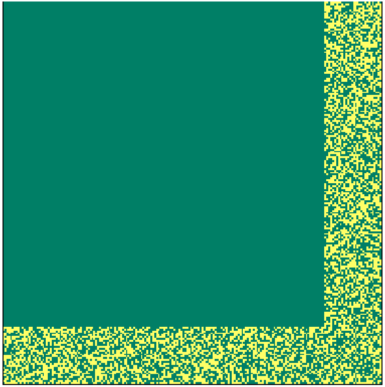
$$\min \left\{ \|\boldsymbol{X}\|_* : \boldsymbol{e}^T X \boldsymbol{e} \geq k^2, X_{ij} = 0 \text{ if } (i,j) \in \tilde{E} \right\} \quad (\textbf{NNR})$$

Does the linear operator defining the constraints satisfy RIP/incoherence/null space conditions?

Failure of RIP

Consider the constraints $X_{ij} = 0$ if $(i, j) \in \tilde{E}$.

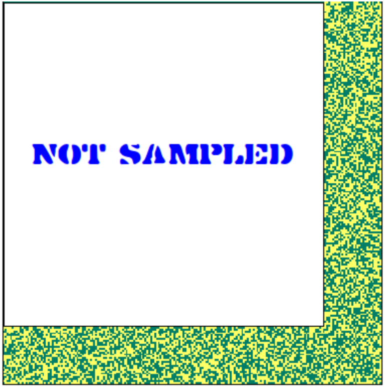
Only sample information from entries corresponding to nonadjacent nodes: not evenly distributed among $V \times V$!



Failure of RIP

Consider the constraints $X_{ij} = 0$ if $(i, j) \in \tilde{E}$.

Only sample information from entries corresponding to nonadjacent nodes: not evenly distributed among $V \times V$!



Why the relaxation works

Low-rank matrix completion fails when the matrix to be recovered lies in the null space of the sampling operator.

In this case: can't distinguish from the all zero matrix $\mathbf{0}$.

For our problem: solutions are bounded away from 0 by the constraint $\boldsymbol{e}^T X \boldsymbol{e} \geq k^2$.

We want the sum constraint to be satisfied using only the clique entries, and all other entries can be set to 0.

The planted case

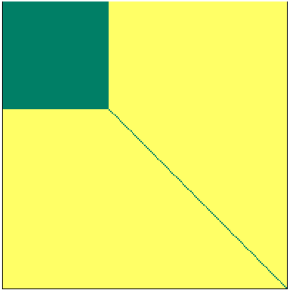
Construction:

- Add all potential edges between nodes in vertex set V^* of size k.
- Then some of the remaining potential edges are added as noise at random.
- By construction, V^* is a clique of G (called a planted or hidden clique).

The planted case

Construction:

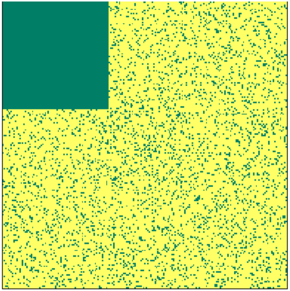
- Add all potential edges between nodes in vertex set V^* of size k.
- Then some of the remaining potential edges are added as noise at random.
- By construction, V^* is a clique of G (called a planted or hidden clique).



The planted case

Construction:

- Add all potential edges between nodes in vertex set V^* of size k.
- Then some of the remaining potential edges are added as noise at random.
- By construction, V^* is a clique of G (called a planted or hidden clique).



Recovery guarantee (Random case)

Theorem

- *Suppose that noise edges are added independently with fixed probability p.*
- *There exists scalar $c > 0$ such that if*

$$k \geq c\sqrt{N}$$

*then V^* is the unique maximum clique of G and $\boldsymbol{X}^* = \boldsymbol{v}\boldsymbol{v}^T$ is the unique optimal solution of (**NNR**) with probability tending exponentially to 1 as $N \to \infty$.*

Proof Idea

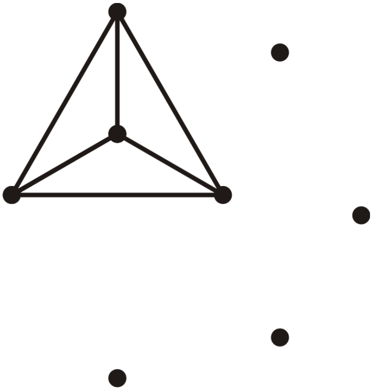
Apply KKT conditions and SDP duality to derive conditions ensuring optimality and uniqueness of $\boldsymbol{X}^*$.

Propose a choice of Lagrange multipliers corresponding to $\boldsymbol{X}^*$.

Use bounds on concentration of random variables to establish that these multipliers satisfy the optimality and uniqueness conditions (with high probability).

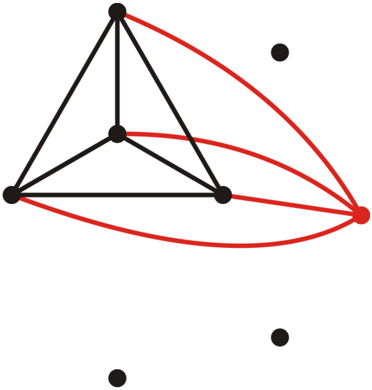
Adversarial bounds

- Suppose that an adversary can add k edges from $v \in V - V^*$ to V^*.



Adversarial bounds

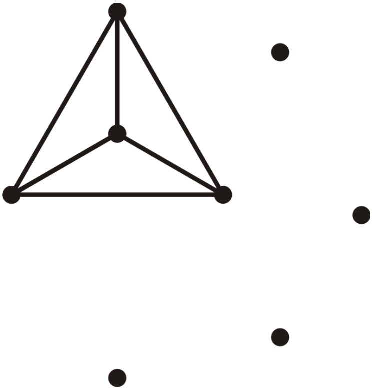
- Suppose that an adversary can add k edges from $v \in V - V^*$ to V^*.
- Then can expand the planted clique V^* to $V^* \cup \{v\}$.



Adversarial bounds

- Suppose that an adversary can add k edges from $v \in V - V^*$ to V^*.
- Then can expand the planted clique V^* to $V^* \cup \{v\}$.

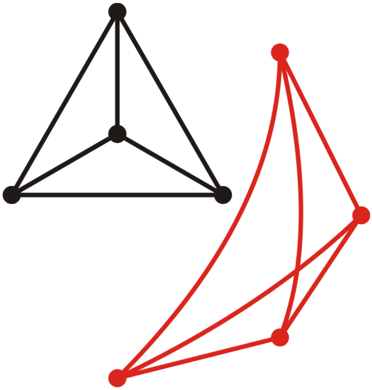
- Suppose the adversary can add $k(k-1)/2$ edges.



Adversarial bounds

- Suppose that an adversary can add k edges from $v \in V - V^*$ to V^*.
- Then can expand the planted clique V^* to $V^* \cup \{v\}$.

- Suppose the adversary can add $k(k-1)/2$ edges.
- Can make a new clique of size k.

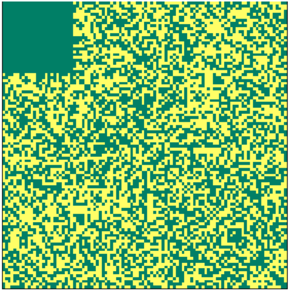


What happens if edges are deleted?

These guaranteees do not tolerate edge deletion noise.

Suppose the graph is corrupted so that edge uv is deleted for some $u, v \in V^*$.

Then V^* is not a clique and $\boldsymbol{X}^*$ is not feasible for (**NNR**).

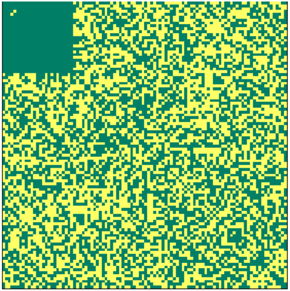


What happens if edges are deleted?

These guaranteees do not tolerate edge deletion noise.

Suppose the graph is corrupted so that edge uv is deleted for some $u, v \in V^*$.

Then V^* is not a clique and $\boldsymbol{X}^*$ is not feasible for (**NNR**).



The densest k-subgraph problem

Want to find a **dense** subgraph of size k, not necessarily a clique.

Densest k-subgraph problem (DKS): Given a graph G, find subgraph $H \subseteq G$ on k nodes with maximum density:

$$d(H) = \frac{|E(H)|}{|V(H)|} = \frac{|E(H)|}{k}.$$

NP-hard: proof is by reduction to **Clique**; hard to approximate.

Maximizing $d(H)$ is equivalent to maximizing $|E(H)|$ over all k-node subgraphs.

Duality of density and number of missing edges

Let $V^* \subseteq V$ be a k-subset with characteristic vector $\boldsymbol{v}$.

Introduce a new variable $\boldsymbol{Y}$: acts as a correction for entries of $\boldsymbol{X} = \boldsymbol{v}\boldsymbol{v}^T$ that should be 0:

$$Y_{ij} = \begin{cases} -X_{ij}, & \text{if } ij \in \tilde{E} \\ 0, & \text{otherwise.} \end{cases}$$

If V^* is almost a clique then $G(V^*)$ should be very dense and $\boldsymbol{Y}$ should be very sparse.

Cardinality of $\boldsymbol{Y}$ acts as a dual of density of $G(V^*)$:

$$|E(G(V^*))| = \binom{k}{2} - \frac{\|\boldsymbol{Y}\|_0}{2}$$

Formulation as sparse plus low-rank decomposition

Can formulate (**DKS**) as

$$\begin{aligned} \min \quad & \operatorname{rank}(\boldsymbol{X}) + \gamma \|\boldsymbol{Y}\|_0 \\ st \quad & \boldsymbol{e}^T X \boldsymbol{e} = k^2 \\ & X_{ij} + Y_{ij} = 0 \text{ if } ij \in \tilde{E} \\ & \boldsymbol{X} \in \{0, 1\}^{V \times V} \\ & \boldsymbol{X} \in \Sigma^V \end{aligned}$$

where γ is a regularization parameter.

Formulation as sparse plus low-rank decomposition

Can formulate (**DKS**) as

$$\begin{aligned} \min \quad & \|\boldsymbol{X}\|_* + \gamma \|\boldsymbol{Y}\|_1 \\ st \quad & \boldsymbol{e}^T X \boldsymbol{e} = k^2 \\ & X_{ij} + Y_{ij} = 0 \text{ if } ij \in \tilde{E} \\ & \boldsymbol{X} \in [0,1]^{V \times V} \end{aligned}$$

where γ is a regularization parameter.

Relax $\|\boldsymbol{Y}\|_0$ using the ℓ_1-norm $\|\boldsymbol{Y}\|_1$, $\mathrm{rank}(\boldsymbol{X})$ with the nuclear norm $\|\boldsymbol{X}\|_*$

Formulation as sparse plus low-rank decomposition

Can formulate (**DKS**) as

$$\begin{aligned} \min \quad & \|\boldsymbol{X}\|_* - \gamma \boldsymbol{e}^T \boldsymbol{Y} \boldsymbol{e} \\ st \quad & \boldsymbol{e}^T X \boldsymbol{e} = k^2 \\ & X_{ij} + Y_{ij} = 0 \text{ if } ij \in \tilde{E} \\ & \boldsymbol{X} \in [0,1]^{V \times V}, \boldsymbol{Y} \geq 0 \end{aligned}$$

where γ is a regularization parameter.

Relax $\|\boldsymbol{Y}\|_0$ using the ℓ_1-norm $\|\boldsymbol{Y}\|_1$, $\operatorname{rank}(\boldsymbol{X})$ with the nuclear norm $\|\boldsymbol{X}\|_*$

Planted case

Start with N nodes V.

Add all edges between nodes in $V^* \subseteq V$.

Add noise:

- Add some of the remaining potential edges.
- Delete some edges in $V^* \times V^*$.

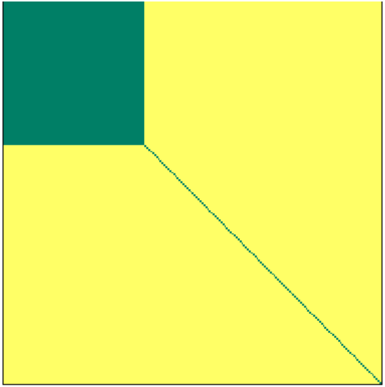
Planted case

Start with N nodes V.

Add all edges between nodes in $V^* \subseteq V$.

Add noise:

- Add some of the remaining potential edges.
- Delete some edges in $V^* \times V^*$.



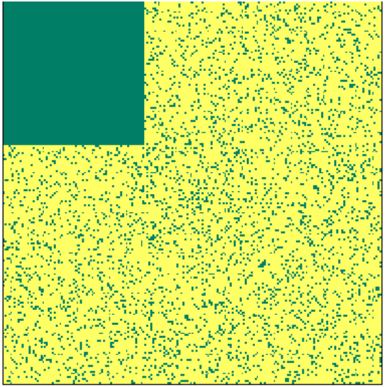
Planted case

Start with N nodes V.

Add all edges between nodes in $V^* \subseteq V$.

Add noise:

- Add some of the remaining potential edges.
- Delete some edges in $V^* \times V^*$.



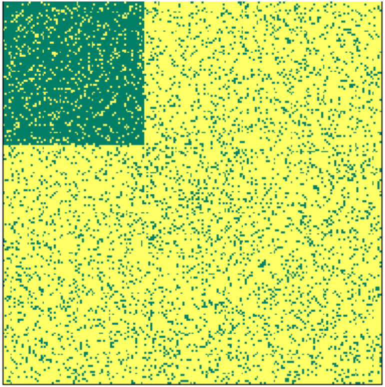
Planted case

Start with N nodes V.

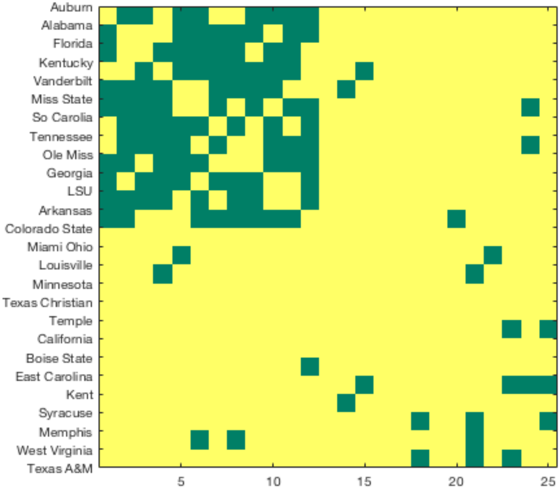
Add all edges between nodes in $V^* \subseteq V$.

Add noise:

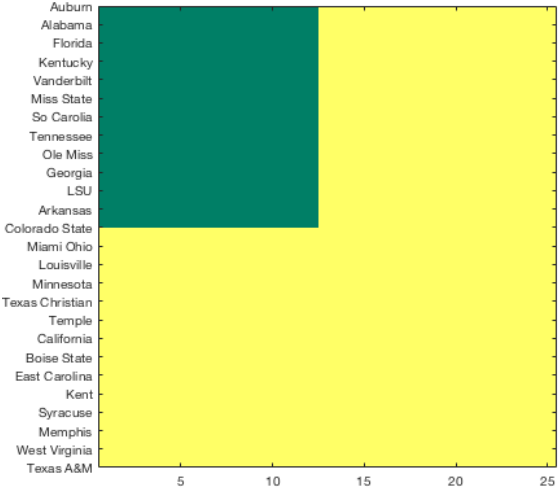
- Add some of the remaining potential edges.
- Delete some edges in $V^* \times V^*$.



Back to the SEC Example



Back to the SEC Example



Random case guarantee

Theorem (Ames 2017)

- *Suppose that G is sampled from the planted dense k-subgraph model with constants $c_1, c_2 > 0$ such that*

$$p(1-p) \geq c_1 \frac{\log N}{N} \qquad q - p \geq c_2 \frac{\log k}{k}.$$

- *Then there exist constant $c_3, c_4 > 0$ depending on p, q such that if*

$$k \geq \frac{c_3}{q-p} \sqrt{\left(\frac{p}{1-p} N \log N\right)}$$

*then $G(V^*)$ is the unique maximum density k-subgraph of G and $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ is the unique optimal solution of (**DKSR**) for regularization parameter $\gamma = c_4/(q-p)k$ with high probability.*

Example: Dense Case

Suppose that p, q are fixed or shrink very slowly, i.e., $p, 1 - q > 1/\log k$.

Then we can recover the planted subgraph with high probability provided that

$$k \geq C\sqrt{N \log N}.$$

Ignoring log-term, we have the same results as before.

- can modify analysis to eliminate the $\log N$.

Sparse Graphs

In most practical examples, the following are not necessarily true:

1. $k = \Omega(\sqrt{N})$.
2. The noise probabilities p, q are not fixed.

Example: Community Detection. In most real-world social networks, community size does not grow as the number of users increases. (Seems to be capped at a very small fraction of the total population).

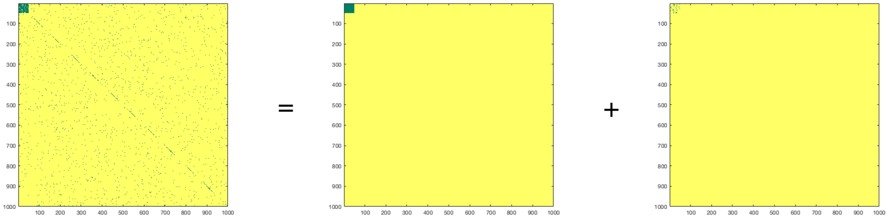
Need to modify model to use **sparse** noise: p and/or q tend to zero as $N \to \infty$.

Example: Sparse Case

Suppose that noise is **sparse**.

Suppose q is fixed and $p = (\log N)^3/N$.

Then we have exact recovery when $k \geq C(\log N)^3$



Conclusion

Proposed new heuristics for the **Clique** and **Densest k-subgraph** problems.

Established theoretical guarantees for exact recovery.

Open problems:

- How to efficiently solve the relaxations?
- Are the random bounds tight? Can we relax $\Omega(N^{1/2})$ to $\Omega(N^{1/2-\epsilon})$?

Thank you!

References:

- B. Ames and S. Vavasis. Nuclear norm minimization for the planted clique and biclique problems. *Mathematical Programming*, 129(1):1-21, 2011.
- B. Ames. Guaranteed recovery of planted cliques and dense subgraphs by convex relaxation. *Journal of Optimization Theory and Applications*, 167(2), 653-675, 2015.
- B. Ames. Finding dense subgraphs in sparse graphs. In preparation.

(see **bpames.people.ua.edu/publications** for more)