How to find a hidden clique

Brendan Ames

Institute for Mathematics and its Applications
University of Minnesota

Clemson University
Thursday, April 11, 2013

Outline/Contributions

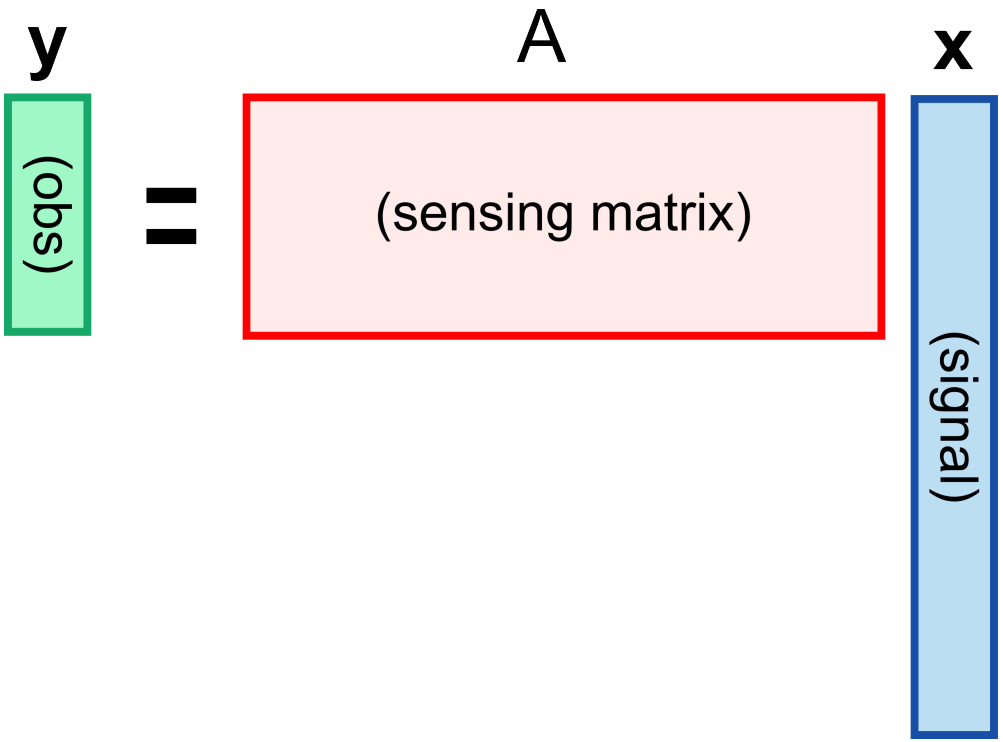
- Brief review of sparse optimization/recovery.
- Present a new convex relaxation for the maximum clique problem based on sparse optimization techniques.
- Explain how this relaxation approach can be extended to graph clustering.
- Give a probabilistic model for "clusterable" data and graphs, and theoretical guarantees of recovery of the clusters using our relaxation.
- Experimental results and open problems.
- Joint work with Stephen Vavasis and Ting-Kei Pong, University of Waterloo.

Classical Signal Compression

- Measure signal $\mathbf{x} \in \mathbf{R}^n$, which is sparse in some basis.
- Compress signal as $\mathbf{y} = A\mathbf{x} \in \mathbf{R}^m$, where $m << n$.
- Measuring the full signal $\mathbf{x}$ may be costly if n is very large.

Compressed Sensing

- Compressed Sensing: measure a sparse signal $\mathbf{x} \in \mathbf{R}^n$ directly using m linear observations:



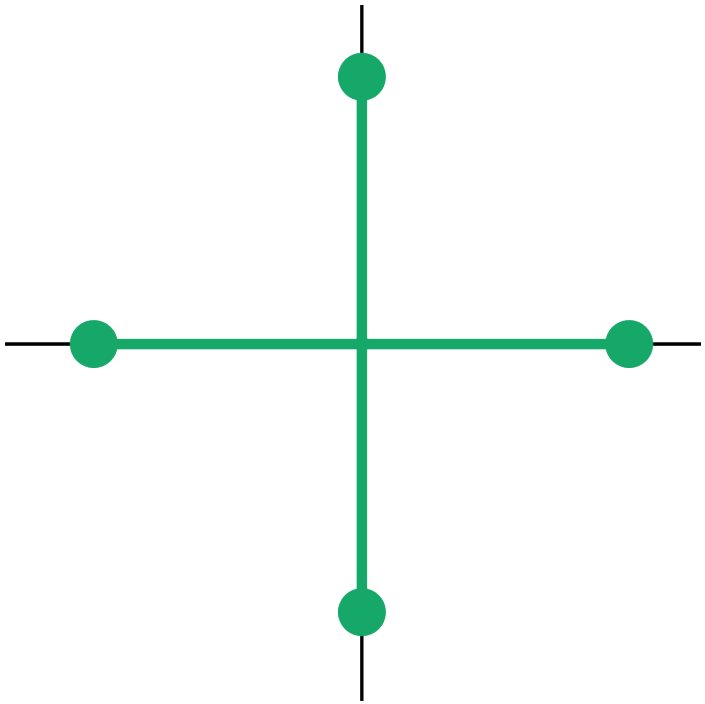
- Never record a full copy of $\mathbf{x}$!
- To recover $\mathbf{x}$ from observed $\mathbf{y}$ solve $\min\{\|\mathbf{x}\|_0 : A\mathbf{x} = \mathbf{y}\}$ where $\|\mathbf{x}\|_0 := \#\{i : x_i \neq 0\}$.
- Problem is NP-hard.

ℓ_1-norm relaxation

- Relax by replacing $\|\mathbf{x}\|_0$ with the ℓ_1 norm:

$$\|\mathbf{x}\|_0 \Rightarrow \|\mathbf{x}\|_1 = \sum |x_i|$$

- $\|\cdot\|_1$ is the convex envelope of $\|\cdot\|_0$ on $B_\infty = \{\mathbf{x} : |x_i| \leq 1\}$

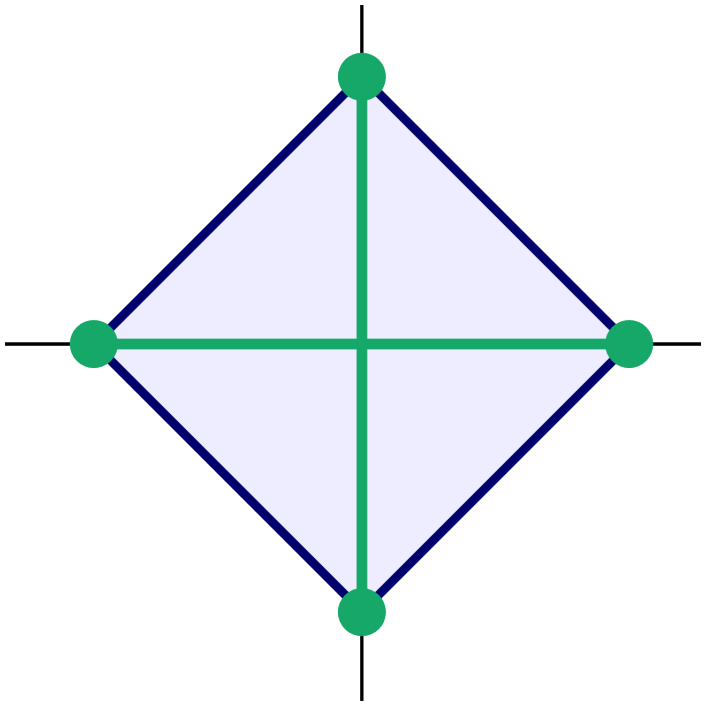


ℓ_1-norm relaxation

- Relax by replacing $\|\mathbf{x}\|_0$ with the ℓ_1 norm:

$$\|\mathbf{x}\|_0 \Rightarrow \|\mathbf{x}\|_1 = \sum |x_i|$$

- $\|\cdot\|_1$ is the convex envelope of $\|\cdot\|_0$ on $B_\infty = \{\mathbf{x} : |x_i| \leq 1\}$

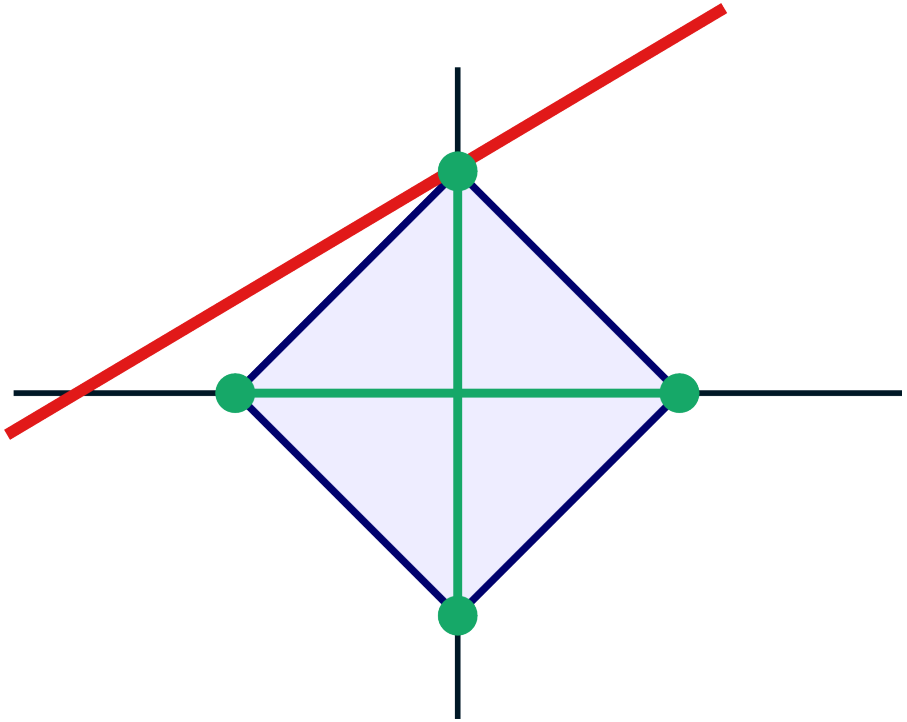


Sparse recovery

- Donoho 2006, Candès, Romberg, Tao 2006: If A and $\mathbf{y}$ are given by sufficiently many ($m = \Omega(s \log(n/s))$ random observations of sparse vector $\mathbf{x}_0$ then

$$\mathbf{x}_0 = \arg\min\{\|\mathbf{x}\|_0 : A\mathbf{x} = \mathbf{y}\} = \arg\min\{\|\mathbf{x}\|_1 : A\mathbf{x} = \mathbf{y}\}$$

- Idea: recovery is guaranteed if A samples roughly same amount of information from each element of $\mathbf{x}_0$.



Affine Rank minimization

- Affine rank minimization problem: find minimum rank solution of a given set of linear equations:

$$\min\{\mathrm{rank}(X) : \mathcal{A}(X) = \mathbf{b}\}.$$

- Well-known to be NP-hard.
- Fazel 2002: Relax using the nuclear norm $\|X\|_*$

$$\mathrm{rank}(X) = \|\sigma(X)\|_0 \;\Rightarrow\; \|X\|_* = \sum_i \sigma_i(X)$$

Matrix completion/Netflix Problem

- Netflix problem: matrix M contains incomplete user movie ratings.
  - M is low rank: few factors influence user prefs

Movies

Users								
Alice	3			4		2		2
Bob			3			3		
Charlie	1				4		3	

- Try to recover the full-matrix M of user ratings from submitted ratings:

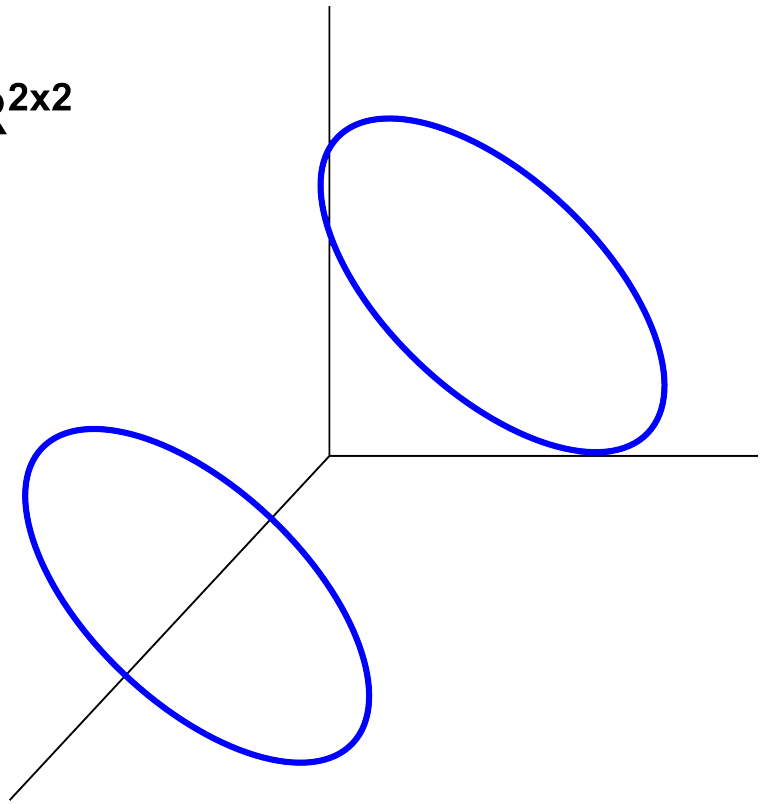
$$\min\{\operatorname{rank} X : X_{ij} = M_{ij} \text{ if user } i \text{ has rated movie } j\}$$

Geometry of nuclear norm relaxation

- Relax by replacing $\mathrm{rank}(X)$ with $\|X\|_* = \sum \sigma_i(X)$
- $\|\cdot\|_*$ is the convex envelope of rank on $B = \{X : \|X\| \leq 1\}$.

$X = \begin{bmatrix} x & y \\ y & z \end{bmatrix}$ in $\mathbf{R}^{2x2}$

rank(X) = 1
|x + z| =1

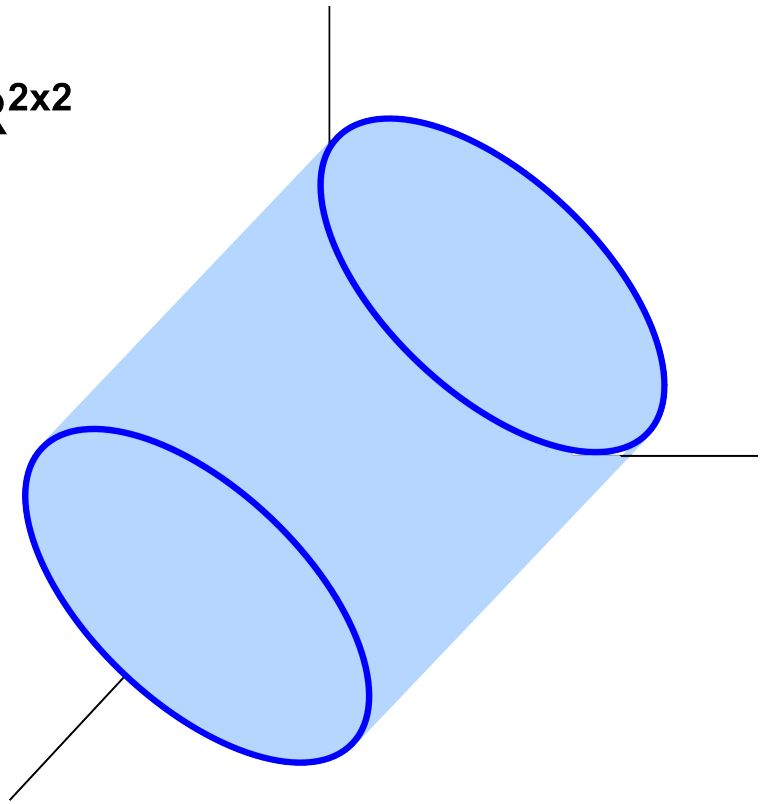


Geometry of nuclear norm relaxation

- Relax by replacing $\mathrm{rank}(X)$ with $\|X\|_* = \sum \sigma_i(X)$
- $\|\cdot\|_*$ is the convex envelope of rank on $B = \{X : \|X\| \leq 1\}$.

$X = \begin{bmatrix} x & y \\ y & z \end{bmatrix}$ in $\mathbf{R}^{2x2}$

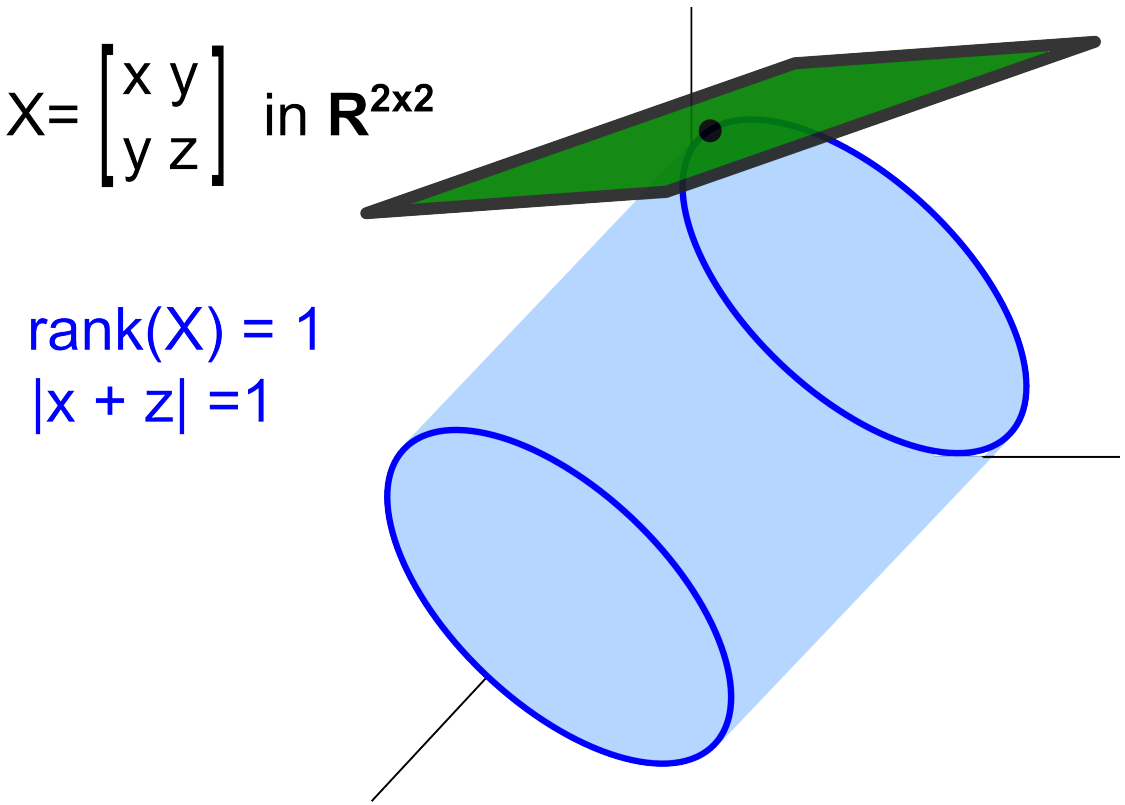
rank(X) = 1
|x + z| =1



Guaranteed low-rank recovery

- Recht, Fazel, Parrilo 2007, Candés and Recht 2008:
  If $\mathcal{A}$ and **b** are given by sufficiently many ($m = \Omega(rn \log n)$ random observations of a low-rank matrix then

$$\arg\min\{\mathrm{rank}(X) : \mathcal{A}(X) = \mathbf{b}\} = \arg\min\{\|X\|_* : \mathcal{A}(X) = \mathbf{b}\}$$



Clustering

- Clustering: Want to partition a given data set so that items in each cluster are similar to each other and items not in the same cluster are dissimilar.
- Fundamental problem in statistics and machine learning:
  - applications include pattern recognition, computational biology, image processing/computer vison, network analysis.
- Intractable in general: usually modeled as some NP-hard problem.
- Consider two model problems: **clique** and **graph partitioning**.

Graph clustering

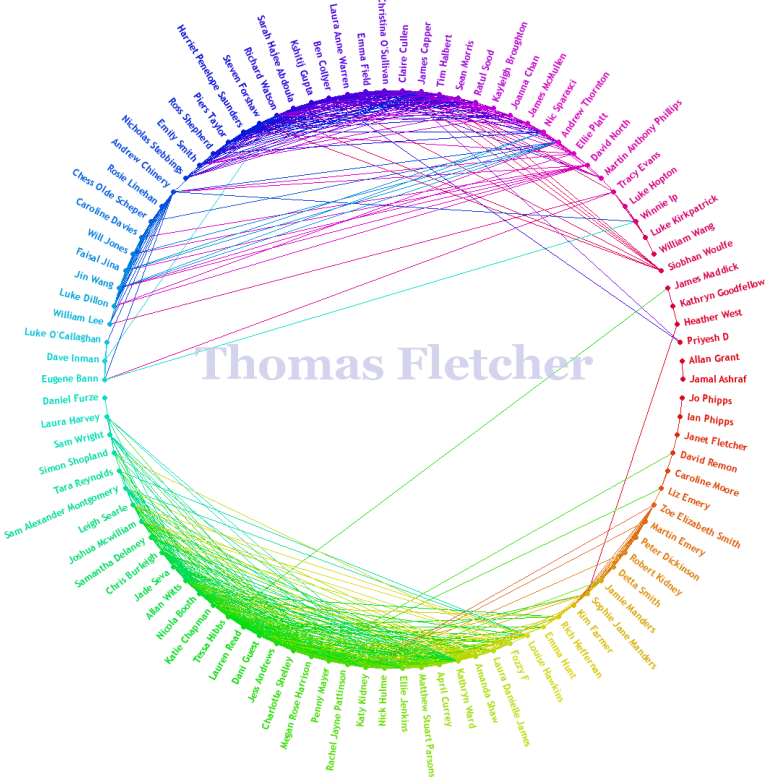
- The Similarity Graph: we may represent a data set as a graph where items correspond to nodes and edges indicate similarity.

- To cluster the data set into k clusters, we want to divide the graph into k subgraphs representing these clusters.

- E.g. partition the similarity graph into k dense subgraphs:
  - Dense = large average degree

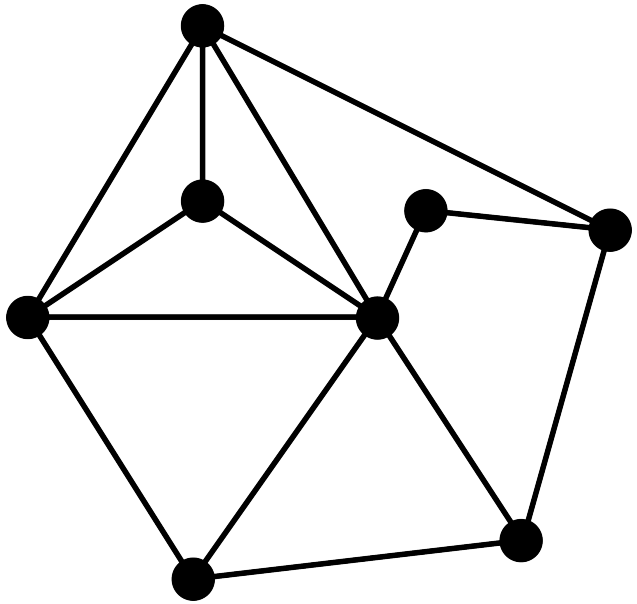
Example: Communities in Social Networks

- Nodes = users
- Edges = “friendship”.
- Densely connected groups = communities



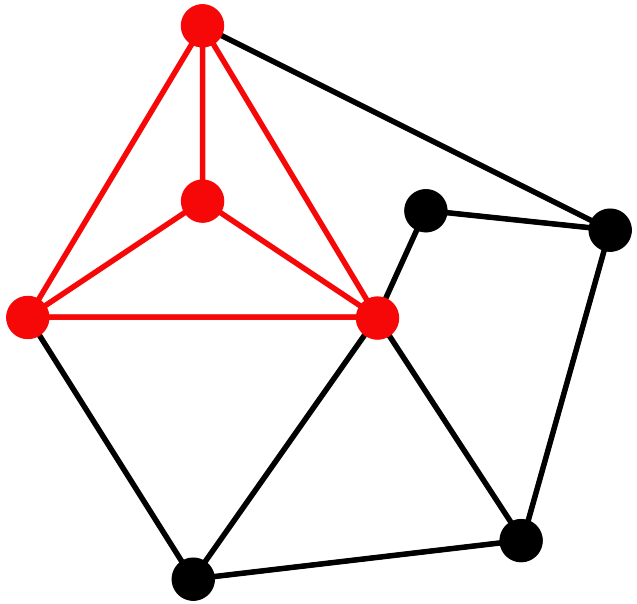
Cliques of a graph

- Given graph $G = (V, E)$, a clique of G is a pairwise adjacent subset of V.
- $C \subseteq V$ 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.
- $C \subseteq V$ 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 Maximum Clique problem

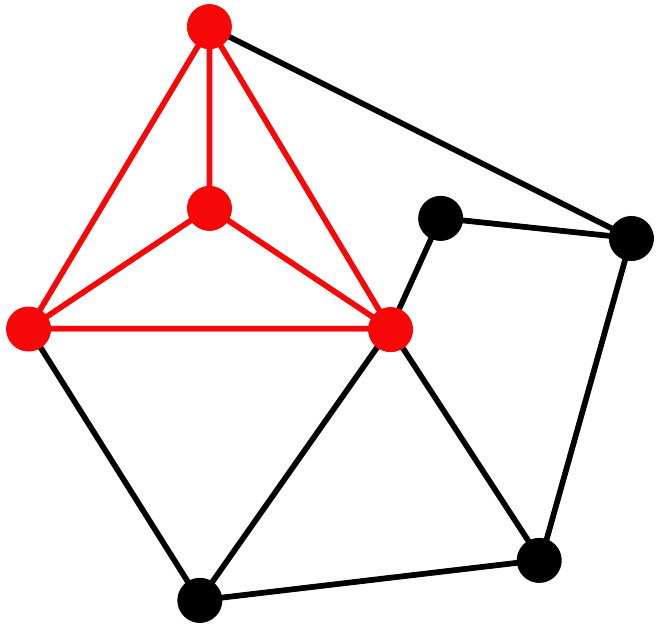
- Optimization version: Find the size of the largest clique of G; size of the largest clique is the clique number $\omega(G)$.
- Decision version: Given graph G, integer r: does G contain a clique of cardinality at least r?
- Complexity: NP-complete, cannot approximate $\omega(G)$ within a ratio of $N^{1-\epsilon}$ for any $\epsilon > 0$, where $N = |V|$.
- Many applications: communication, power, and social networks, mathematical biology, cryptography.

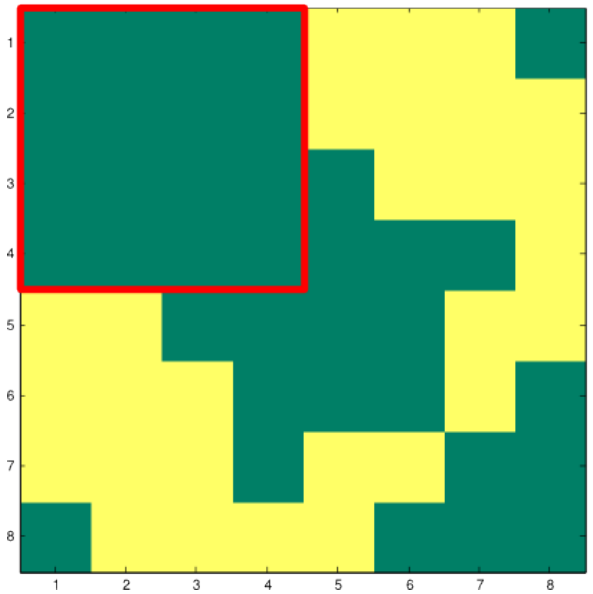
Matrix representation of cliques

- Characteristic vector of C: vector $\mathbf{v} \in \{0,1\}^V$ with

$$v_i = \begin{cases} 1, & \text{if } i \in C \\ 0, & \text{otherwise} \end{cases}$$

- If C is a clique with characteristic vector $\mathbf{v}$ let $X = \mathbf{v}\mathbf{v}^T$.
- Nonzero entries of X form a $|C| \times |C|$ all-ones block in $A_G + I$.





Max Clique as rank minimization

- G has a r-clique if and only if there exists rank-one symmetric binary matrix X such that

$$\sum\sum X_{ij} \geq r^2$$
$$X_{ij} = 0 \quad \forall\, ij \notin E,\ i \neq j.$$

- Otherwise $\omega(G) < r$.

- Max Clique is equivalent to the rank minimization problem:

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

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

Nuclear norm relaxation of Clique

- We have the rank minimization problem:

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

- Relax by replacing rank with the nuclear norm and ignoring the binary and symmetry constraints:

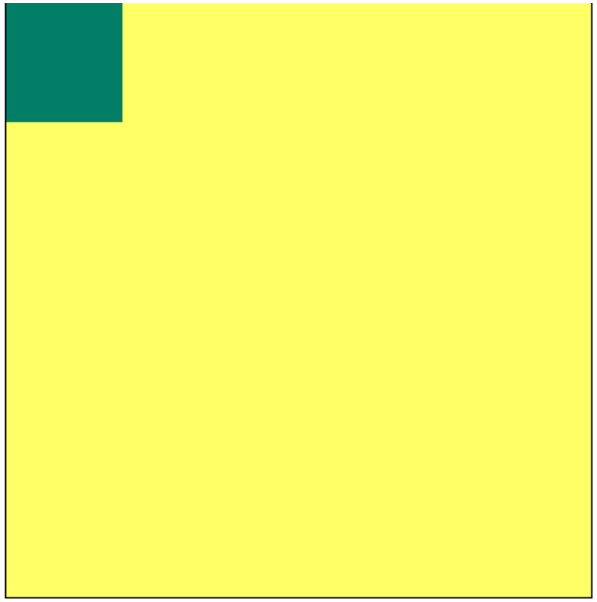
$$\min \left\{ \|X\|_* : \mathbf{e}^T X \mathbf{e} \geq r^2, X_{ij} = 0 \text{ if } (i,j) \in \tilde{E} \right\} \quad (\mathbf{NNR})$$

- When does the solution of the relaxation coincide with that of the rank minimization problem?

The planted case

Construction:

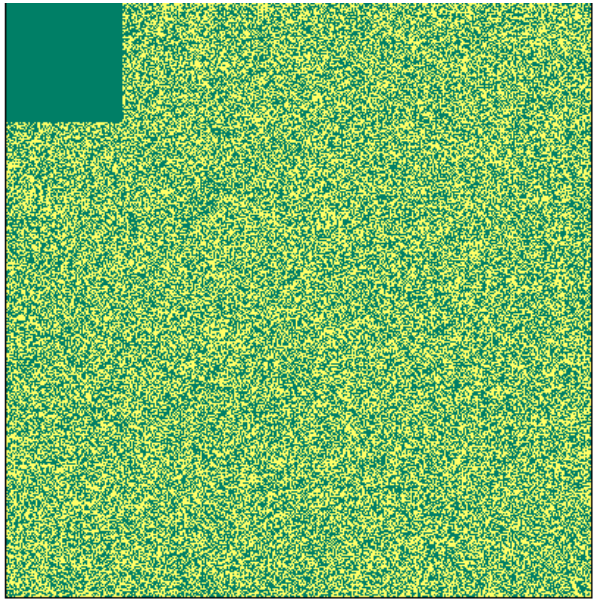
- Add all potential edges between nodes in vertex set V^* of size r.



The planted case

Construction:

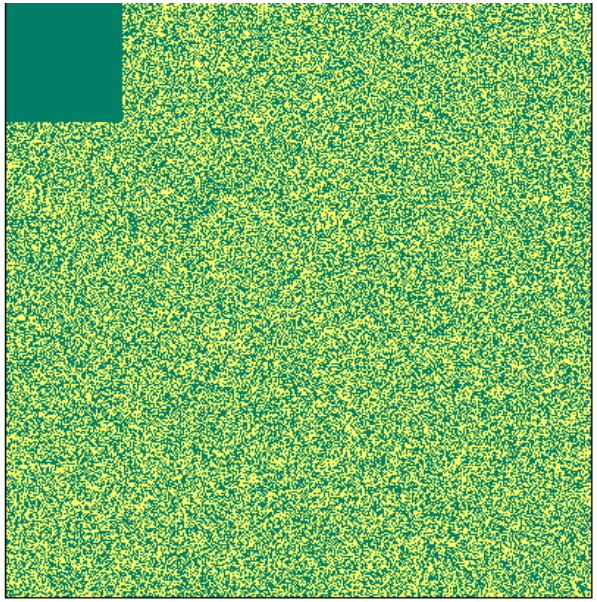
- Add all potential edges between nodes in vertex set V^* of size r.
- Each remaining potential edge is added to E independently with probability p.



The planted case

Construction:

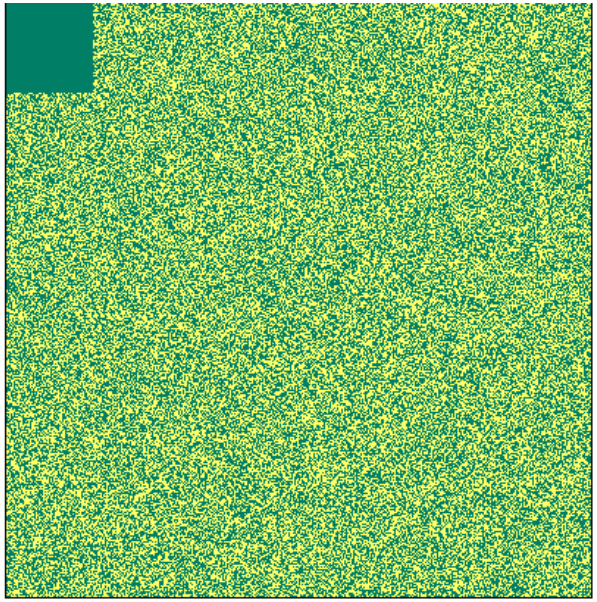
- Add all potential edges between nodes in vertex set V^* of size r.
- Each remaining potential edge is added to E independently with probability p.
- By construction, V^* is a clique of G (called a planted or hidden clique).



The planted case

Construction:

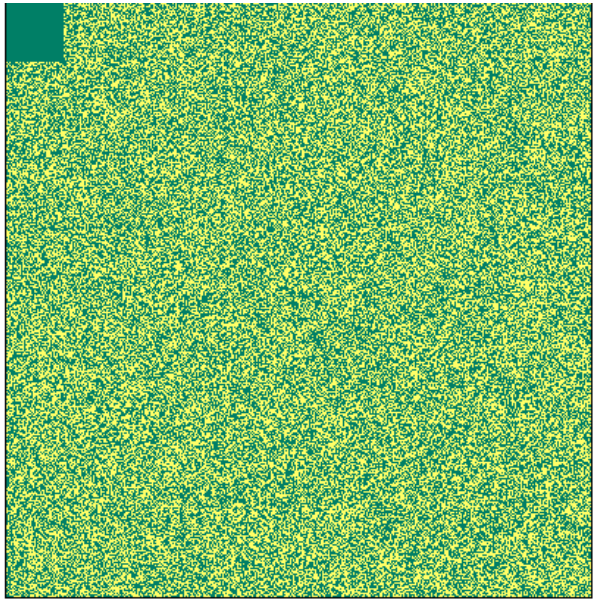
- Add all potential edges between nodes in vertex set V^* of size r.
- Each remaining potential edge is added to E independently with probability p.
- By construction, V^* is a clique of G (called a planted or hidden clique).



The planted case

Construction:

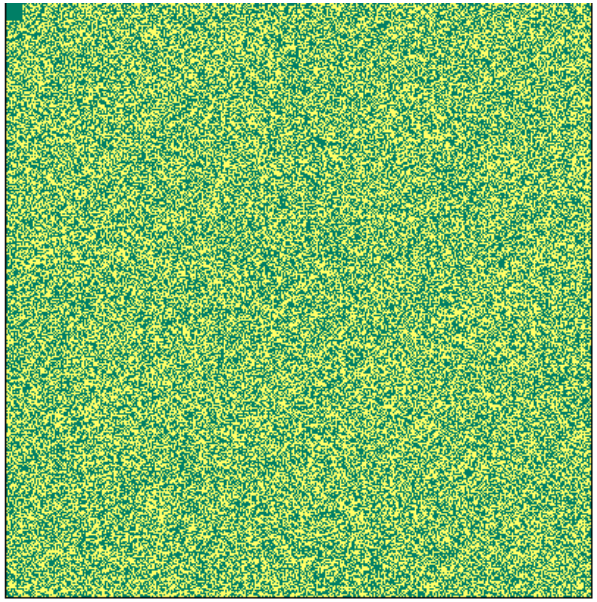
- Add all potential edges between nodes in vertex set V^* of size r.
- Each remaining potential edge is added to E independently with probability p.
- By construction, V^* is a clique of G (called a planted or hidden clique).



The planted case

Construction:

- Add all potential edges between nodes in vertex set V^* of size r.
- Each remaining potential edge is added to E independently with probability p.
- By construction, V^* is a clique of G (called a planted or hidden clique).



Recovery guarantee in the planted case

Theorem (Ames-Vavasis 2011)

There exists scalar $c > 0$ (depending only on p) such that if

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

then

- V^* *is the unique maximum clique of* G*, and*
- $X^* = \mathbf{v}\mathbf{v}^T$ *is the unique optimal solution of* (**NNR**)

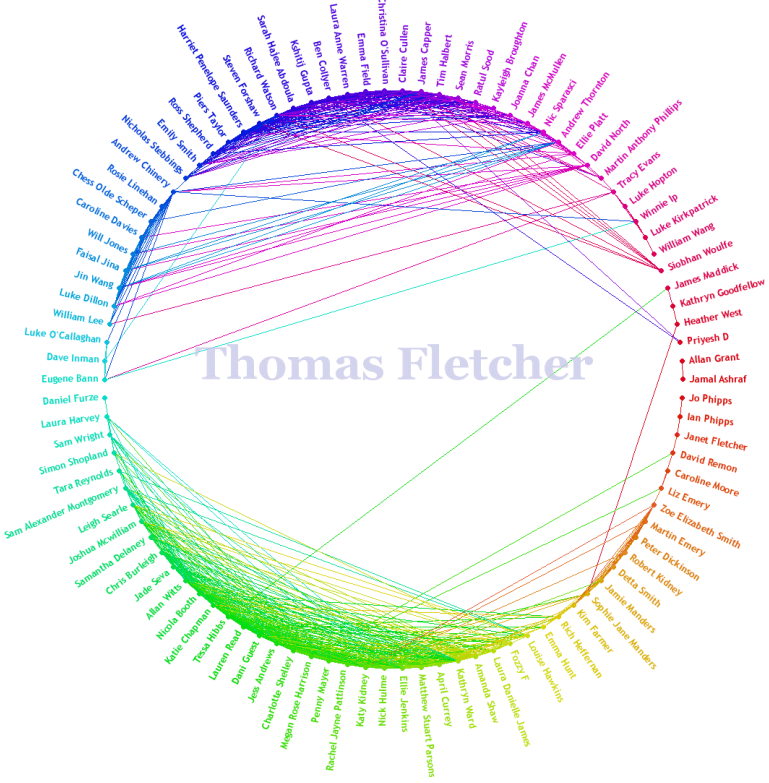
with probability tending exponentially to 1 *as* $N \to \infty$*.*

The Weighted Similarity Graph

- Suppose we are given a set of data and affinity function f indicating similarity between any two items in the data set.
- We can model the data using the weighted similarity graph $G_S = (V, E, W)$ as follows:
  - Each item in the data set is represented by a node in V.
  - We add an edge between each pair of two nodes i, j with edge weight $W_{ij} = f(i, j)$.
  - W_{ij} is large if i and j are highly similar.

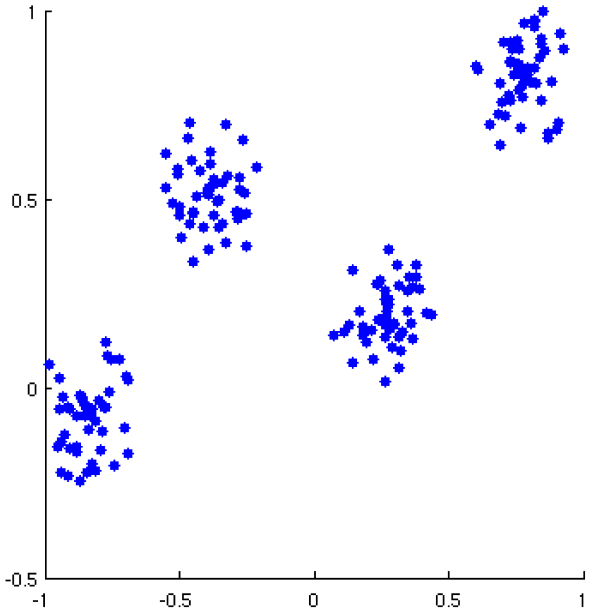
Example: Communities in Social Networks

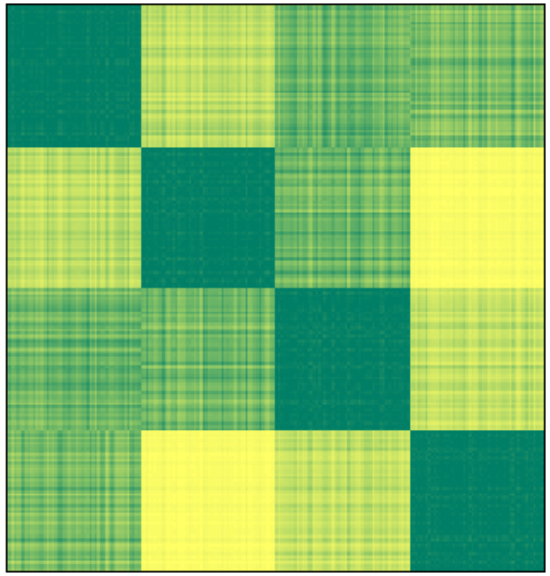
- Nodes = users
- Edges = “friendship”.
- Densely connected groups = communities



Example: Clustered Euclidean data

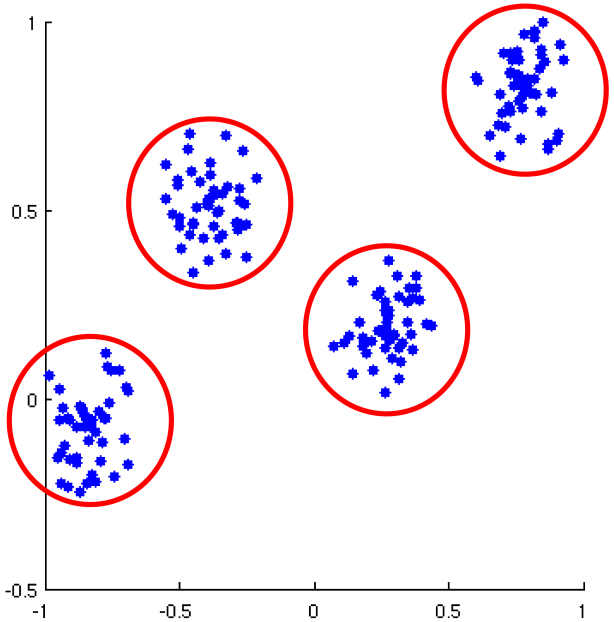
- Suppose each data point in the ith cluster C_i is placed uniformly at random in a ball centered at $c_i \in \mathbf{R}^d$.
- If the centers are sufficiently far apart, then the distance within clusters will be small compared to the distance between clusters.
- Choose $W_{ij} = \exp(-\|\mathbf{x}^i - \mathbf{x}^j\|^2)$.

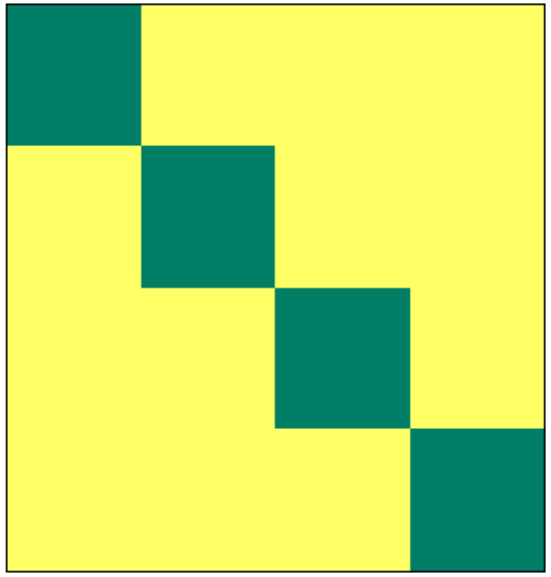




Example: Clustered Euclidean data

- Suppose each data point in the ith cluster C_i is placed uniformly at random in a ball centered at $c_i \in \mathbf{R}^d$.
- If the centers are sufficiently far apart, then the distance within clusters will be small compared to the distance between clusters.
- Choose $W_{ij} = \exp(-\|\mathbf{x}^i - \mathbf{x}^j\|^2)$.





The Densest k-Disjoint Clique Problem

- To cluster the data we want to partition the graph into cliques whose adjacency matrices have heavy support.
- A k-disjoint-clique subgraph of a graph G is a subgraph of G whose set of nodes consists of k disjoint cliques.
- Densest k-disjoint-clique problem (KDC): find a k-disjoint-clique subgraph such that the sum of the densities of the k complete subgraphs induced by the cliques is maximized.
  - Density of a complete subgraph induced by C:

$$d(C) = \frac{1}{|C|} \sum_{i \in C} \sum_{j \in C} W_{ij} = \frac{\mathbf{v}^T W \mathbf{v}}{\mathbf{v}^T \mathbf{v}}$$

where $\mathbf{v}$ is the characteristic vector of C.

Lifting procedure for **KDC**

- Let $\{C_1, \ldots, C_k\}$ define a k-disjoint-clique subgraph with corresponding characteristic vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$
- We lift the k characteristic vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ to the rank k matrix variable X:

$$X = \sum \frac{\mathbf{v}_i \mathbf{v}_i^T}{\|\mathbf{v}_i\|^2}.$$

- X has trace exactly equal to k and satisfies

$$\mathrm{Tr}(WX) = \sum \frac{\mathbf{v}_i^T W \mathbf{v}_i}{\|\mathbf{v}_i\|^2}$$

Relaxation to SDP

- Ignoring rank constraint and relaxing combinatorial constraints on X gives the semidefinite program:

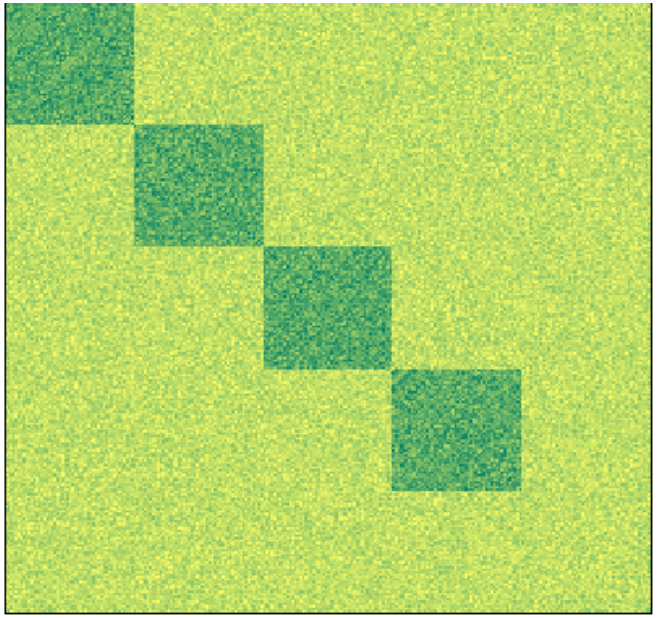
$$\begin{aligned} \max \quad & \mathrm{Tr}(WX) \\ \text{st} \quad & X\mathbf{e} \leq \mathbf{e} \\ & \mathrm{Tr}(X) = k \\ & X_{ij} \geq 0 \quad \forall\, i, j \in V \\ & X \succeq 0. \end{aligned}$$

- $\mathrm{Tr}(X) = \|X\|_* = k$ acts a surrogate for $\mathrm{rank}(X) = k$

The Planted cluster model

Randomly generate weights $W \in [0,1]^{N \times N}$ according to the following model:

- Start with clusters $C_1, \ldots, C_k$ of sizes $r_1, \ldots, r_k$.
- Sample entries of $W(C_i, C_i)$ i.i.d. from probability distribution Ω_1 with mean α.
- Sample remaining entries of W i.i.d. from distribution Ω_2 with mean $\beta << \alpha$.



Guaranteed recovery

- Let $X^* = \sum_{i=1}^{k} \frac{\mathbf{v}_i \mathbf{v}_i^T}{r_i}$ be the feasible solution corresponding the planted clusters where $\mathbf{v}_i$ is the characteristic vector of C_i for all i.

Theorem (Ames 2012)

There exist scalars $c_1, c_2, c_3 > 0$ such that if

$$c_1\sqrt{N} + c_2\sqrt{kr_{k+1}} + c_3 r_{k+1} \leq (\alpha + \alpha\delta_{0,r_{k+1}} - 2\beta)\hat{r},$$

where $\hat{r} = \min\{r_1, r_2, \ldots, r_k\}$, then

- *X^* is the unique optimal solution of SDP relaxation, and*
- *$\{C_1, \ldots, C_k\}$ is the unique densest k-disjoint-clique subgraph*

with probability tending exponentially to 1 as $\hat{r} \to \infty$.

Rehnquist Supreme Court

- Data set is the set of U.S. Supreme Court Justices (serving from 1994-95 to 2003-04).
- First consider by Hubert and Steinley 2005.
- Assign edge-weights corresponding to fraction of decisions on which Justices agreed:

	St	Br	Gi	So	Oc	Ke	Re	Sc	Th
1 St	1	0.62	0.66	0.63	0.33	0.36	0.25	0.14	0.15
2 Br	0.62	1	0.72	0.71	0.55	0.47	0.43	0.25	0.24
3 Gi	0.66	0.72	1	0.78	0.47	0.49	0.43	0.28	0.26
4 So	0.63	0.71	0.78	1	0.55	0.5	0.44	0.31	0.29
5 Oc	0.33	0.55	0.47	0.55	1	0.67	0.71	0.54	0.54
6 Ke	0.36	0.47	0.49	0.5	0.67	1	0.77	0.58	0.59
7 Re	0.25	0.43	0.43	0.44	0.71	0.77	1	0.66	0.68
8 Sc	0.14	0.25	0.28	0.31	0.54	0.58	0.66	1	0.79
9 Th	0.15	0.24	0.26	0.29	0.54	0.59	0.68	0.79	1

Rehnquist Supreme Court: results, pt 1

- Solve KDC with $k = 2$ to get the following partition of the Supreme court:

1: "Liberal"	2: "Conservative"
Stevens (St) Breyer (Br) Ginsberg (Gi) Souter (So)	O'Connor (Oc) Kennedy (Ke) Rehnquist (Re) Scalia (Sc) Thomas (Th)

Rehnquist Supreme Court: results, pt 2

- Algorithm is sensitive to choice of k.
- Solve with $k = 3$:

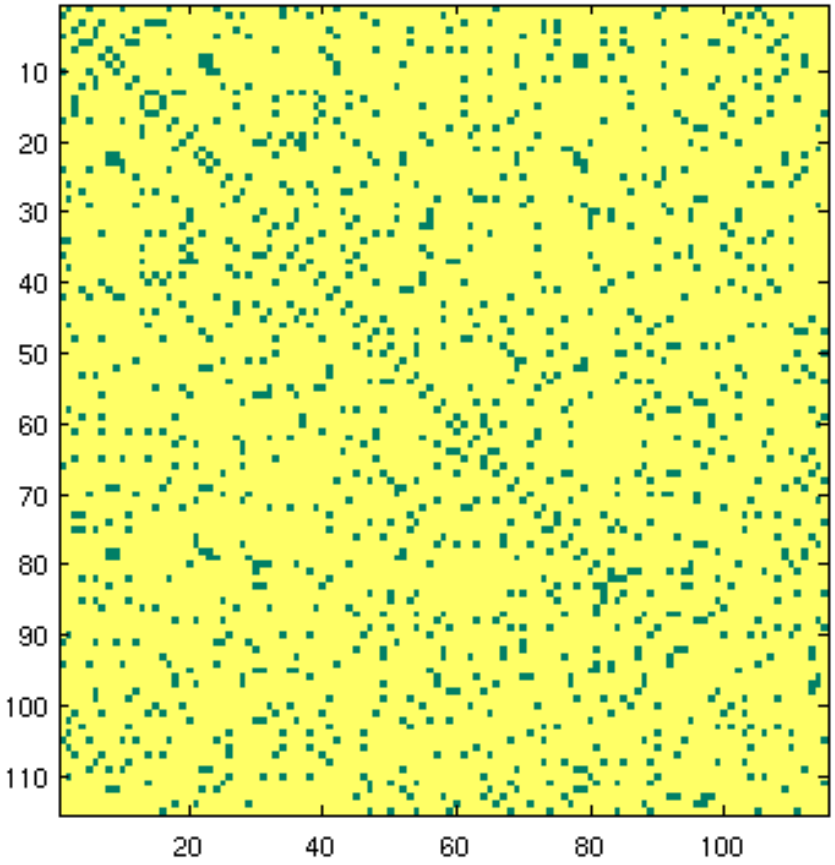
1: “Most Conservative”	2: “Moderate Conservative”	3: ”Liberal”
Thomas (Th) Scalia (Sc)	O’Connor (Oc) Kennedy (Ke) Rehnquist (Re)	Stevens (St) Breyer (Br) Ginsberg (Gi) Souter (So)

The NCAA Football Network

- First considered in Girvan and Newman 2001.
- Nodes are the 115 Division I football teams.
- Teams are deemed adjacent if they played at least one game against each other in the 2000 season.
- Games are typically scheduled based on membership within athletic conferences and geography.
- Should display community structure based on membership in conferences.

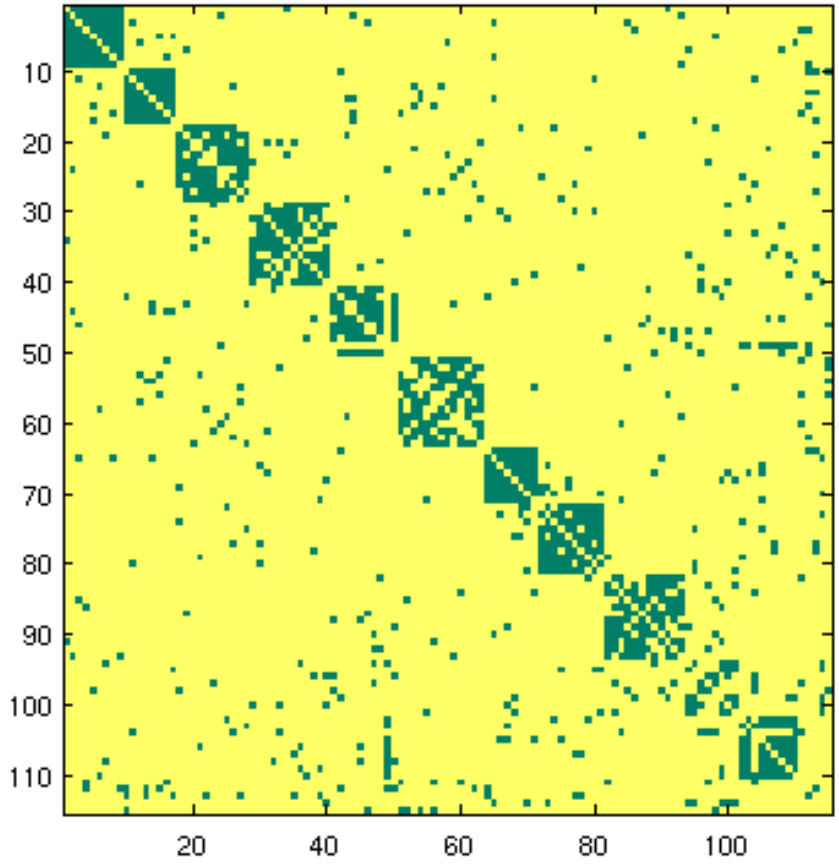
The NCAA Football Network, pt 2

- The node-node adjacency matrix for the football network.



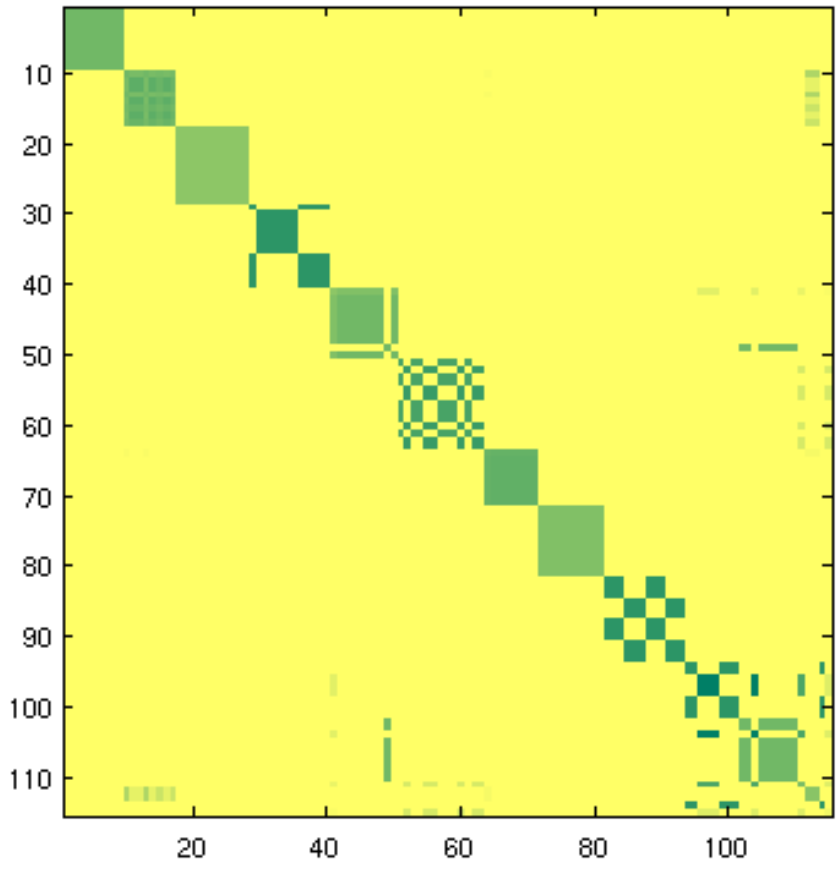
The NCAA Football Network, pt 3

- The adjacency matrix with rows/cols reordered according to conferences.



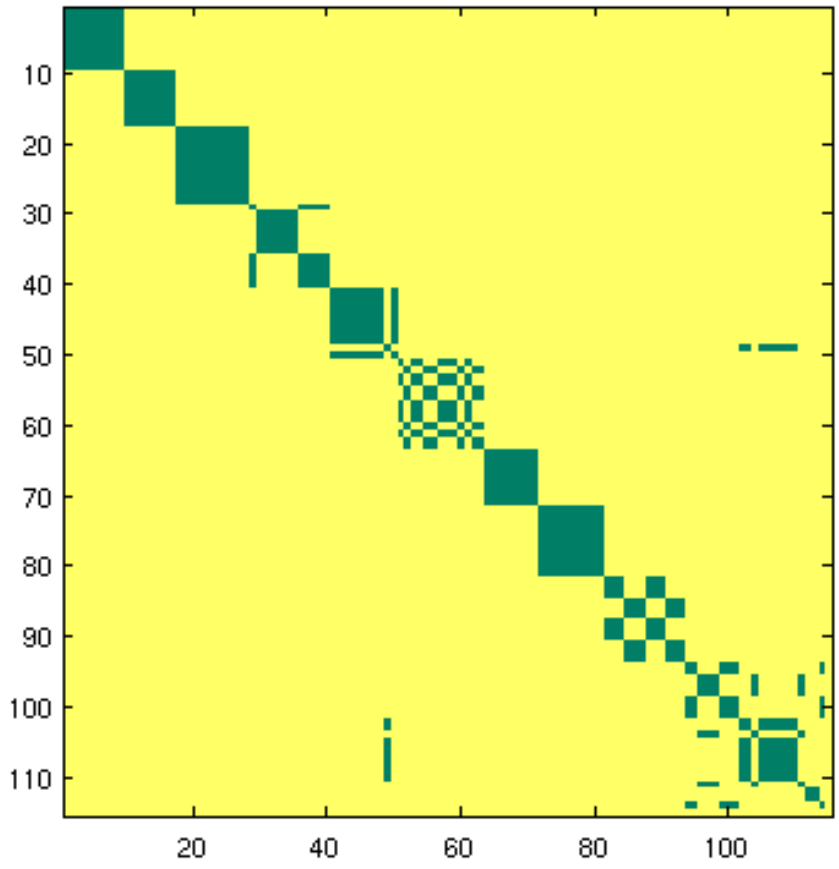
The NCAA Football Network, pt 4

- Solve instance of KDC with $W = A$, $k = 15$.
  - Use ADMM to solve the SDP
- Obtain optimal solution X^*:



The NCAA Football Network, pt 4

- Solve instance of KDC with $W = A$, $k = 15$.
  - Use ADMM to solve the SDP
- Obtain optimal solution X^*:



The NCAA Football Network, pt 5

Atlantic Coast
Clemson
Duke
Florida State
Georgia Tech
Maryland
NC State
North Carolina
Virginia
Wake Forest

Big East
Boston College
Miami
Pittsburgh
Rutgers
Syracuse
Temple
Virginia Tech
West Virginia

Big Ten
Illinois
Indiana
Iowa
Michigan
Michigan State
Minnesota
Northwestern
Ohio State
Penn State
Purdue
Wisconsin

Big Twelve 1
Colorado
Kansas
Kansas State
Iowa State
Missouri
Nebraska

Big Twelve 2
Oklahoma State
Oklahoma
Texas Tech
Baylor
Texas A&M
Texas

Conference USA
Alabama-Birm
Army
Cincinnati
East Carolina
Houston
Louisville
Memphis
So. Miss.
Tulane

Outlier
Connecticut

MAC West
Ball State
C. Michigan
E. Michigan
N. Illinois
Toledo
W. Michigan

MAC East
Akron
BGSU
Buffalo
Kent
Marshall
Miami Ohio
Ohio

Mountain West
Air Force
BYU
Colorado St
New Mexico
SDSU
UNLV
Utah
Wyoming

Pacific Ten
Arizona
Arizona State
California
Oregon
Oregon State
Stanford
UCLA
USC
Washington
WSU

SEC West
Alabama
Arkansas
Auburn
LSU
Miss St
Mississippi

SEC East
Florida
Georgia
Kentucky
South Carolina
Tennessee
Vanderbilt

Sun Belt 1
Arkansas State
*Boise State**
Idaho
NMSU
North Texas
*Utah State**

Sun Belt 2
*Central Florida**
L-Lafayette
L-Monroe
*Louisiana Tech**
MTSU

Western Athletic
Fresno St
Hawaii
Nevada
Rice
SJSU
SMU
*Texas Christian**
Texas El Paso
Tulsa

Independents
Navy
Notre Dame

Conclusions

- Have presented new convex relaxation based heuristics for the maximum clique and graph clustering problems.
- These relaxations are based on the relationship between cliques and low-rank submatrices of the adjacency graph, and the nuclear norm relaxation for rank minimization.
- If data is sufficiently clusterable (i.e. highly similar within clusters, dissimilar across clusters) then these heuristics successfully recover the correct partition of the data.

Open problems

- Overlapping communities?
- Sparse graphs? Can we improve recovery guarantees?
  - Sparse in the sense that $\alpha > \beta$ are both tending to 0 as $N \to \infty$
- How to choose k in the clustering SDP?
- Algorithmic issues: how to efficiently solve large-scale SDP?

Thank you!

- Papers/talk slides: ima.umn.edu/∼bpames
- Matlab code available upon request.