Provably Correct Clustering via Convex Relaxation

Brendan Ames

Department of Mathematics
The University of Alabama

UA Applied Statistics Seminar

Friday February 19, 2021

Agenda

Consider semidefinite relaxations for the maximum clique, densest submatrix, and graph clustering problems.

Give a probabilistic model for ‘‘clusterable’’ data and graphs, and theoretical recovery guarantees.

Open problems.

Joint work with Stephen Vavasis, University of Waterloo, Polina Bombina, UA, Phineas Agar, UA and Aleksis Pirinen, Lund University.

Clustering

Clustering: partition data 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:

- pattern recognition, computational biology, image processing/computer vison, network analysis.

No consensus on what constitutes a **good** clustering; depends heavily on application.

Intractable: usually modeled as some NP-hard problem (e.g., clique, normalized cut, k-means).

A sanity check

Clustering seems to be a very difficult/ill-posed problem.

Many heuristics seem to work well in practice.

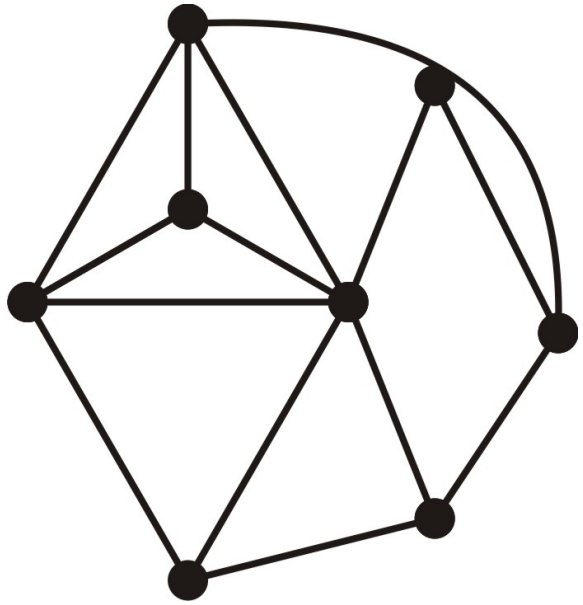
Question: can we show that we can cluster "clusterable" data?
How do we model **clusterable** data?

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**.

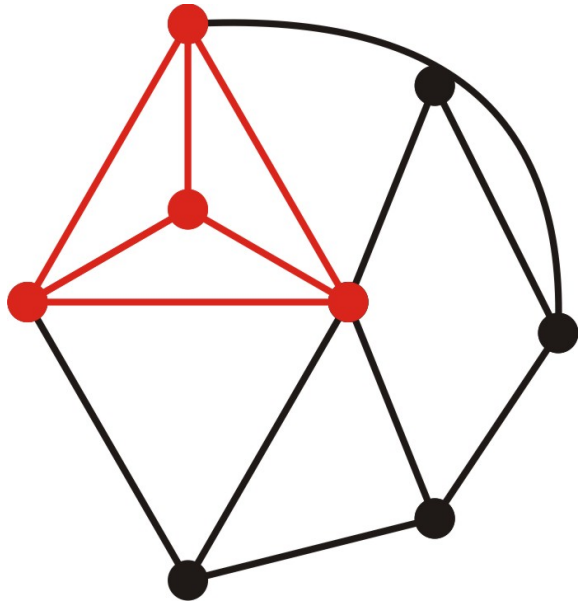


Cliques of a graph

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.

Alon et al. 1998, Feige and Krauthgamer 2000, Ames and Vavasis 2011: 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.

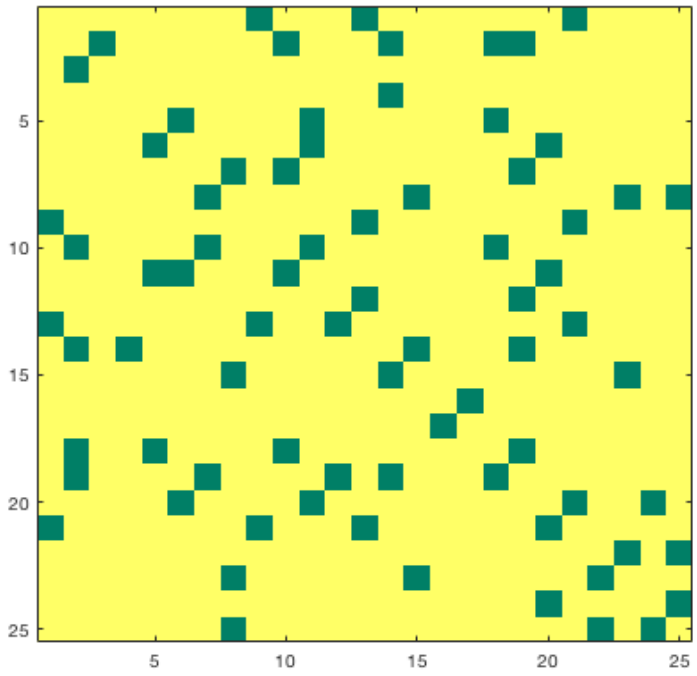
Would like an approach that translates to finding other "clique-like" objects with minimal effort.

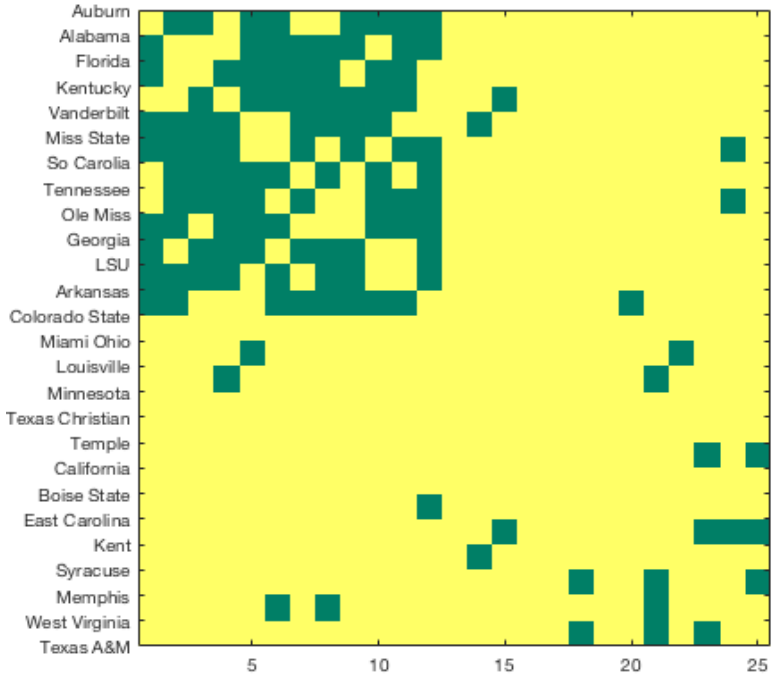
e.g., the maximum biclique of a bipartite graph, fully dense block in a matrix.

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).

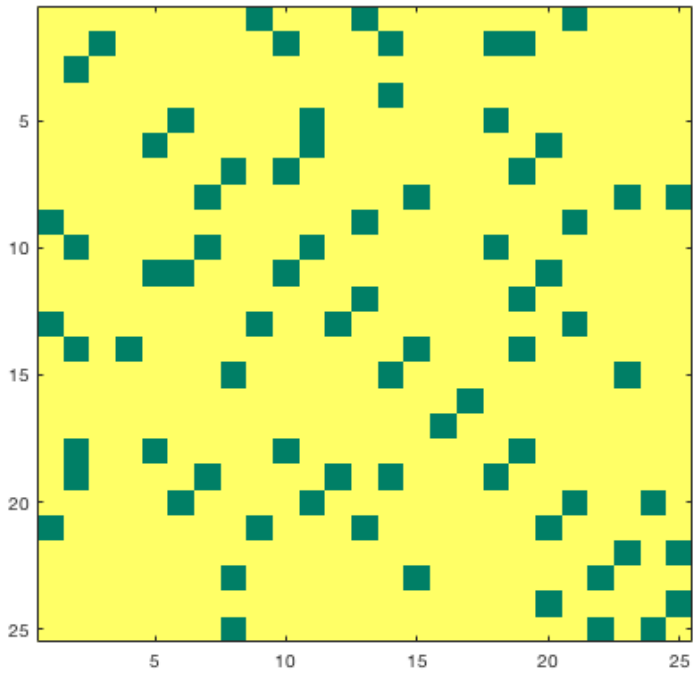


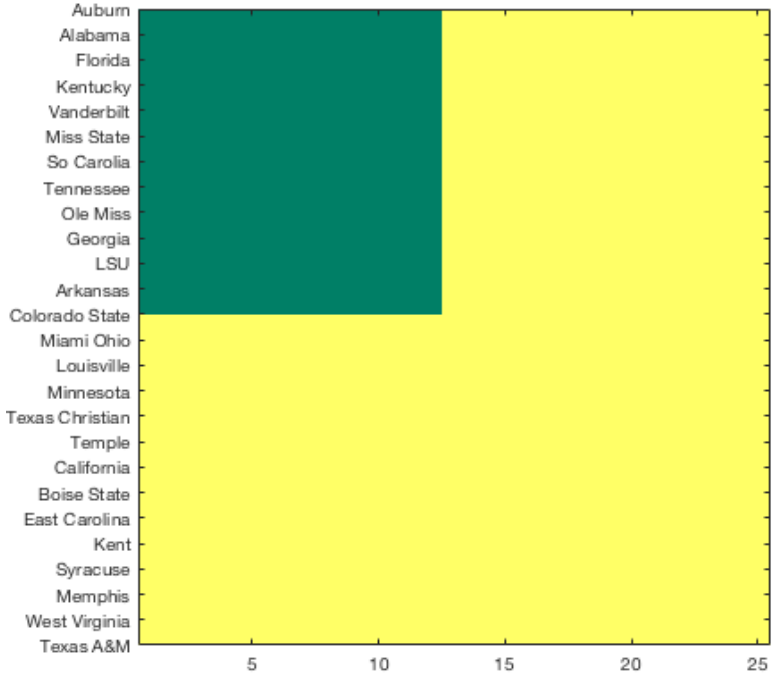


Example: Community Detection in Social Networks

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).

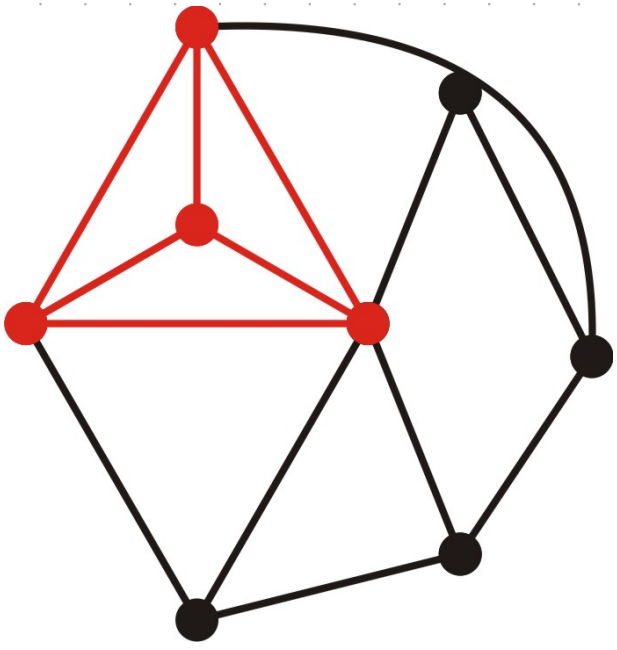


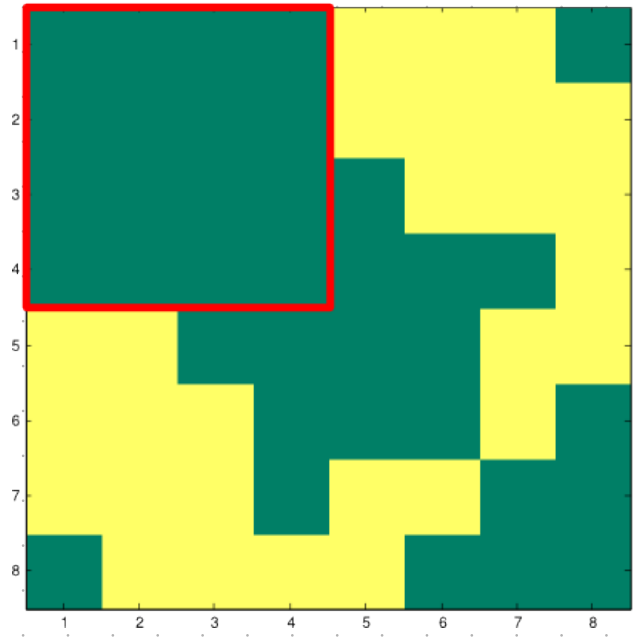


Cliques and low-rank matrices

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 $\boldsymbol{A_G} + \boldsymbol{I}$.





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:

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

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:

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

Well-known to be NP-hard.

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

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

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

The densest (m,n)-submatrix problem

We want to find a **dense $k \times k$ submatrix** in $\boldsymbol{A}_G + \boldsymbol{I}$, not necessarily a clique.

Densest $m \times n$-submatrix problem (DSM): Given a matrix $\boldsymbol{A} \in \mathbf{R}^{M \times N}$, find submatrix with m rows and n columns with maximum number of nonzero entries.

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

Duality of density and number of missing edges / zero entries

Let U and V be a subsets of $\{1, 2, \ldots, M\}$ and $\{1, 2, \ldots, N\}$ with characteristic vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ respectively.

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

$$y_{ij} = \begin{cases} -x_{ij}, & \text{if } a_{ij} = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Cardinality of $\boldsymbol{Y}$ acts as a dual of density of $\boldsymbol{A}(U, V)$:

$$\text{card}\left(\boldsymbol{A}(U, V)\right) = mn - \sum_{i=1}^{M} \sum_{j=1}^{N} y_{ij}$$

Formulation as sparse plus low-rank decomposition

Can formulate (**DSM**) as

$$\begin{aligned} \min \quad & \operatorname{rank} \boldsymbol{X} + \gamma \operatorname{card} \boldsymbol{Y} \\ \text{s.t.} \quad & \boldsymbol{e}^T \boldsymbol{X} \boldsymbol{e} = mn \\ & x_{ij} + y_{ij} = 0 \text{ if } a_{ij} = 0 \\ & x_{ij} \in \{0, 1\} \end{aligned}$$

where γ is a regularization parameter.

Formulation as sparse plus low-rank decomposition

Can formulate (**DSM**) as

$$\begin{aligned} \min \quad & \|X\|_* + \gamma \|Y\|_1 \\ \text{s.t.} \quad & \boldsymbol{e}^T \boldsymbol{X} \boldsymbol{e} = mn \\ & x_{ij} + y_{ij} = 0 \text{ if } a_{ij} = 0 \\ & 0 \leq x_{ij} \leq 1 \end{aligned}$$

where γ is a regularization parameter.

Relax card $\boldsymbol{Y}$ using the ℓ_1-norm $\|\boldsymbol{Y}\|_1$, and rank $\boldsymbol{X}$ with the nuclear norm $\|\boldsymbol{X}\|_*$.

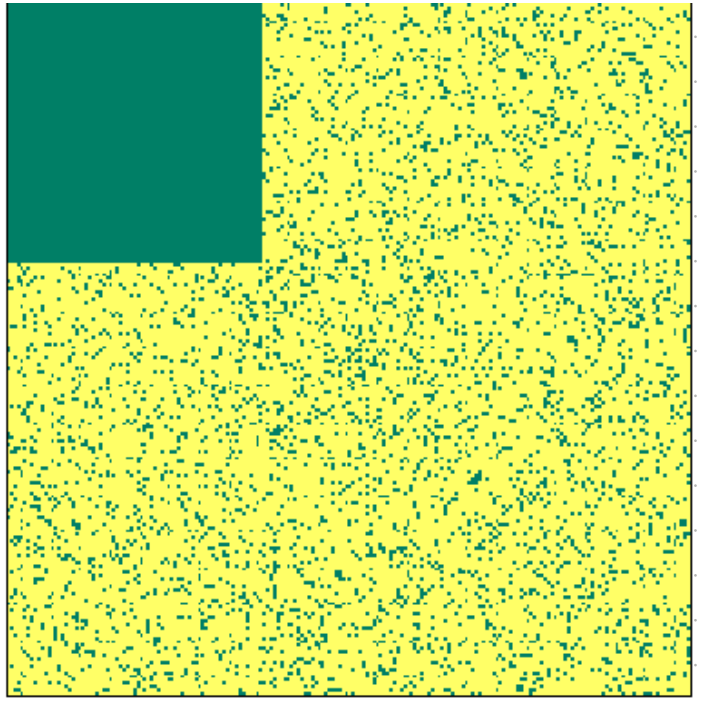
Planted case

Start with $M \times N$ all-zeros matrix $\boldsymbol{A}$.

Set all entries in $m \times n$ block equal to 1.

Add noise:

- Add some of the remaining potential entries with probability p.
- Delete some entries in $m \times n$ block with probability $1 - q$, $q > p$.



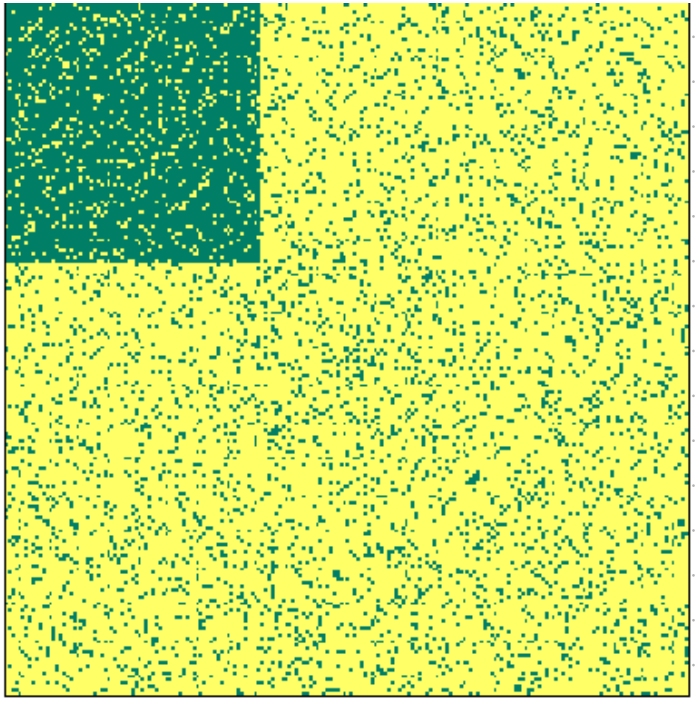
Planted case

Start with $M \times N$ all-zeros matrix $\boldsymbol{A}$.

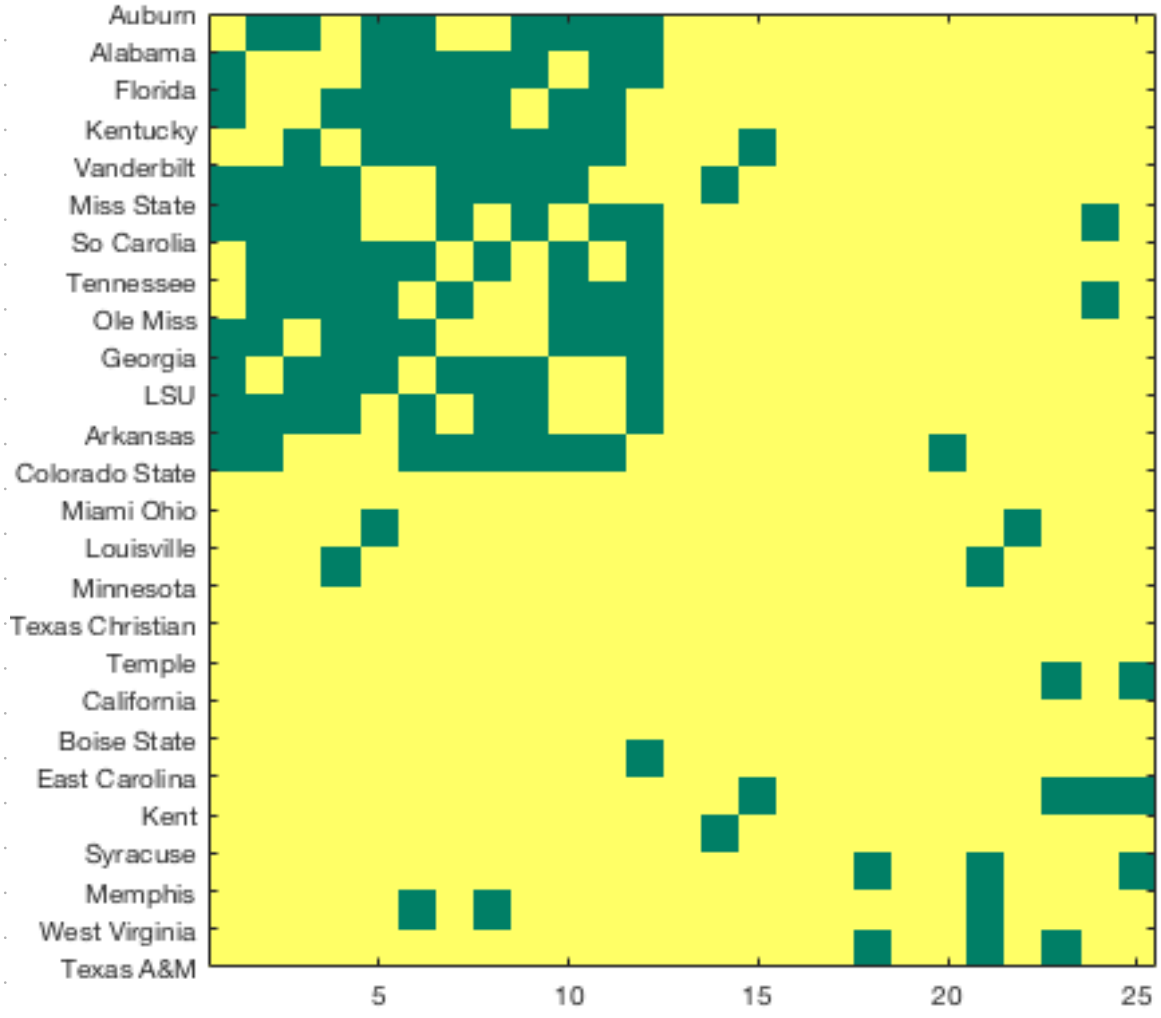
Set all entries in $m \times n$ block equal to 1.

Add noise:

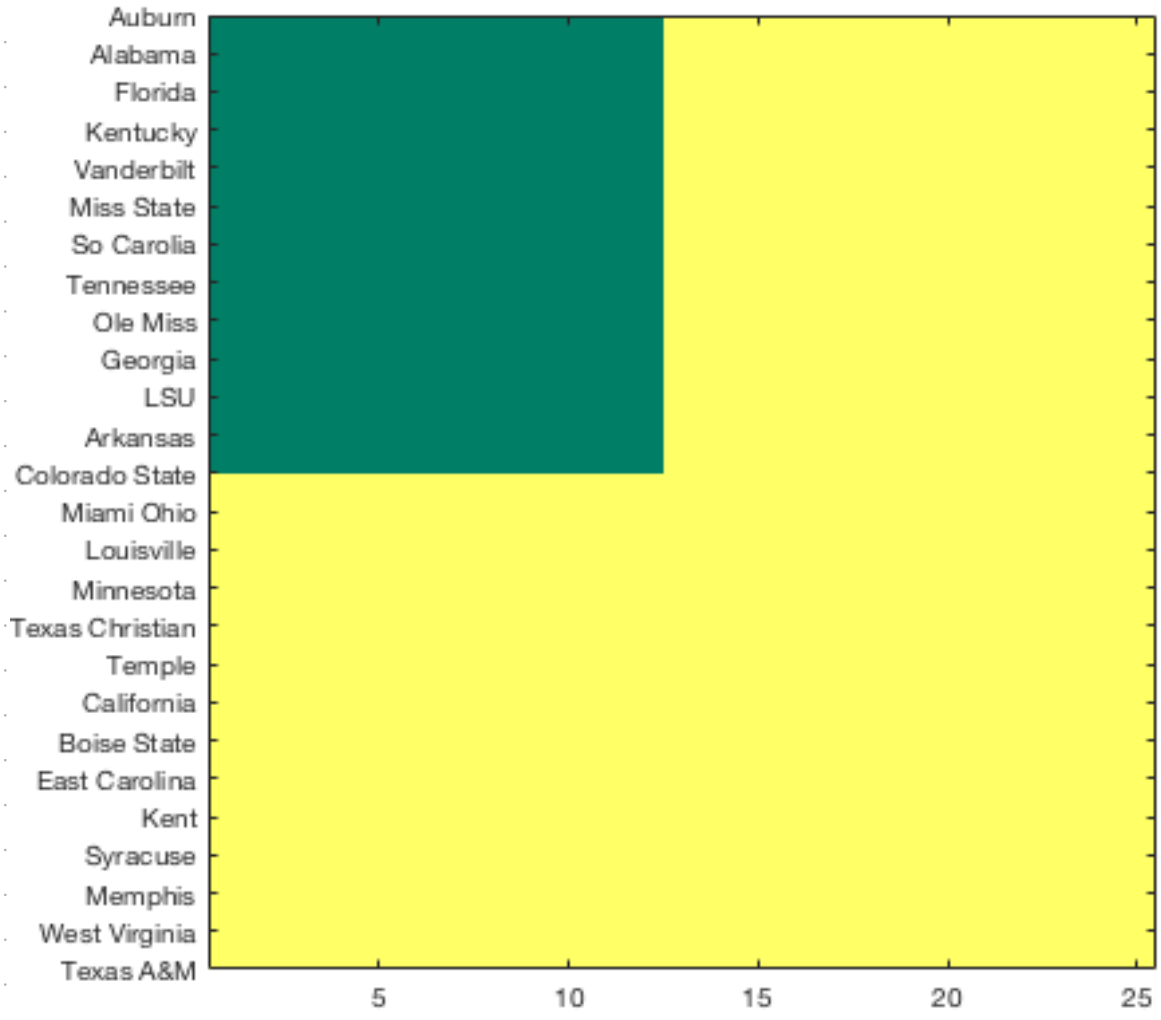
- Add some of the remaining potential entries with probability p.
- Delete some entries in $m \times n$ block with probability $1 - q$, $q > p$.



Back to the SEC Example



Back to the SEC Example



Recovery Guarantee

Theorem (Bombina-Ames 2020)

Suppose that $\boldsymbol{A}$ *is sampled from the planted dense* $m \times n$*-submatrix model with edge probabilities* q *and* p*.*

Let $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ *denote the matrix representation of the planted submatrix and assume* $m \leq n$*,* $M \leq N$*.*

Then there exists constants $c_1, c_2, c_3 > 0$ *such that if*

$$q - p \geq c_1 \max\left\{ \sqrt{\max\{\sigma_q^2, \sigma_p^2\}} \frac{\log N}{m}, \frac{\log N}{m} \sqrt{\sigma_p^2 N}, \frac{(\log N)^{3/2}}{m} \right\}$$

then $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ *is the* **unique optimal solution** *of* **(DSM)** *for regularization parameter*

$$\gamma = \frac{t}{(q-p)m}, \qquad c_2 \leq t \leq c_3$$

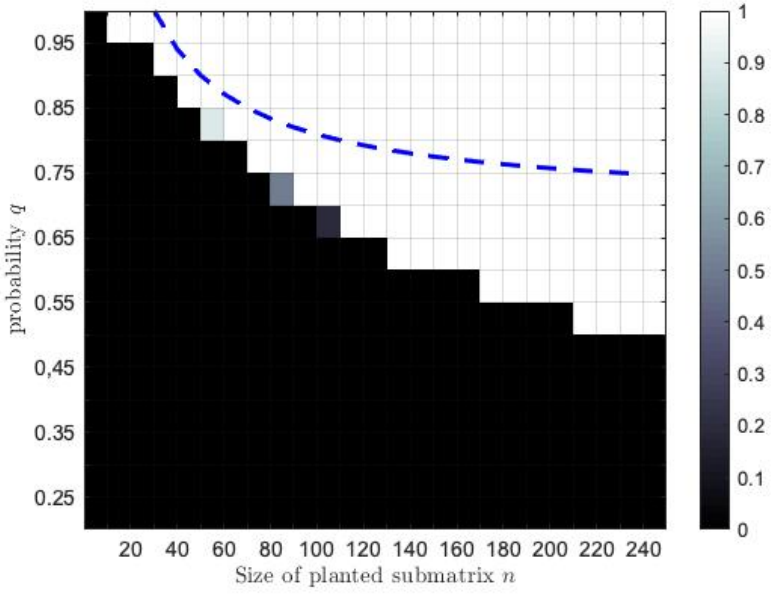
with high probability.

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 submatrix with high probability provided that

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



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

Sparse Graphs

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

1. $m = \Omega(\sqrt{N})$.
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).

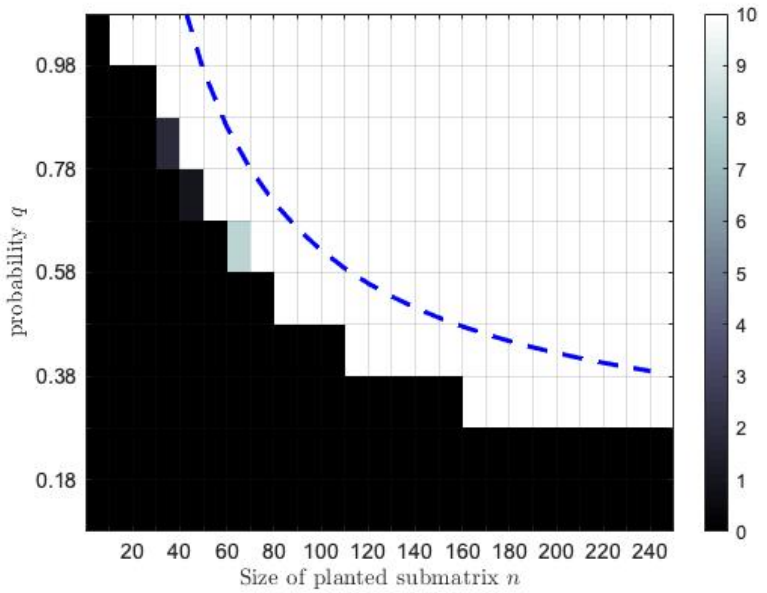
Need to modify model to use **sparse** noise: p and/or q tend to zero as $N \to \infty$.

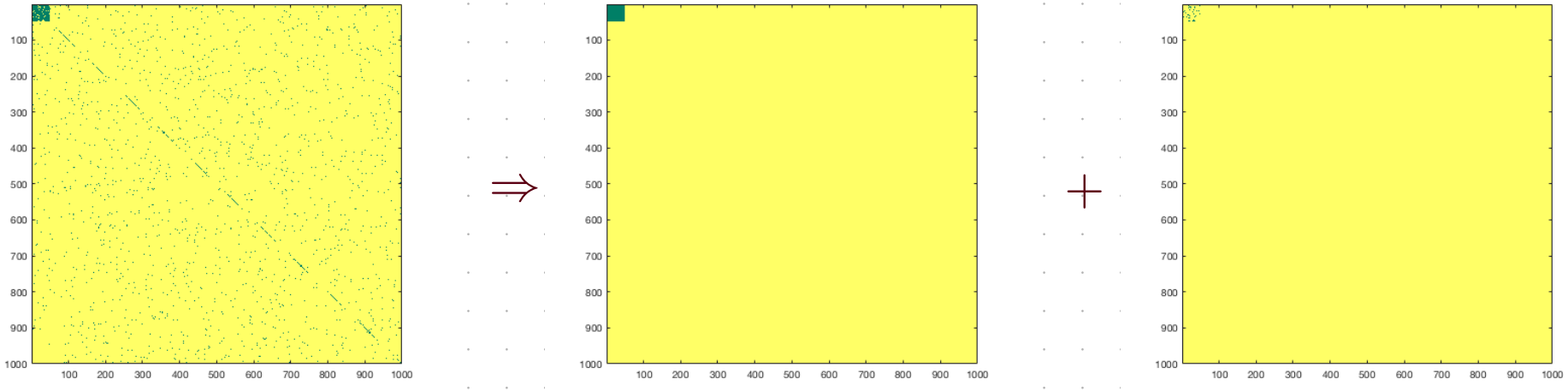
Example: Sparse Case

Suppose that noise is **sparse**.

Suppose q is fixed and $p \leq \log N / N$.

Then we have exact recovery w.h.p. if $m \geq C(\log N)^{3/2}$





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 norms of random matrices to establish that these multipliers satisfy the optimality and uniqueness conditions (with high probability).

The Weighted Similarity Graph

Given data and affinity function f indicating similarity between any two items.

Can model the data as weighted similarity graph $G_S = (V, E, \boldsymbol{W})$ as follows:

- Each item 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: Rehnquist Supreme Court

Data drawn from U.S. Supreme Court decisions (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:

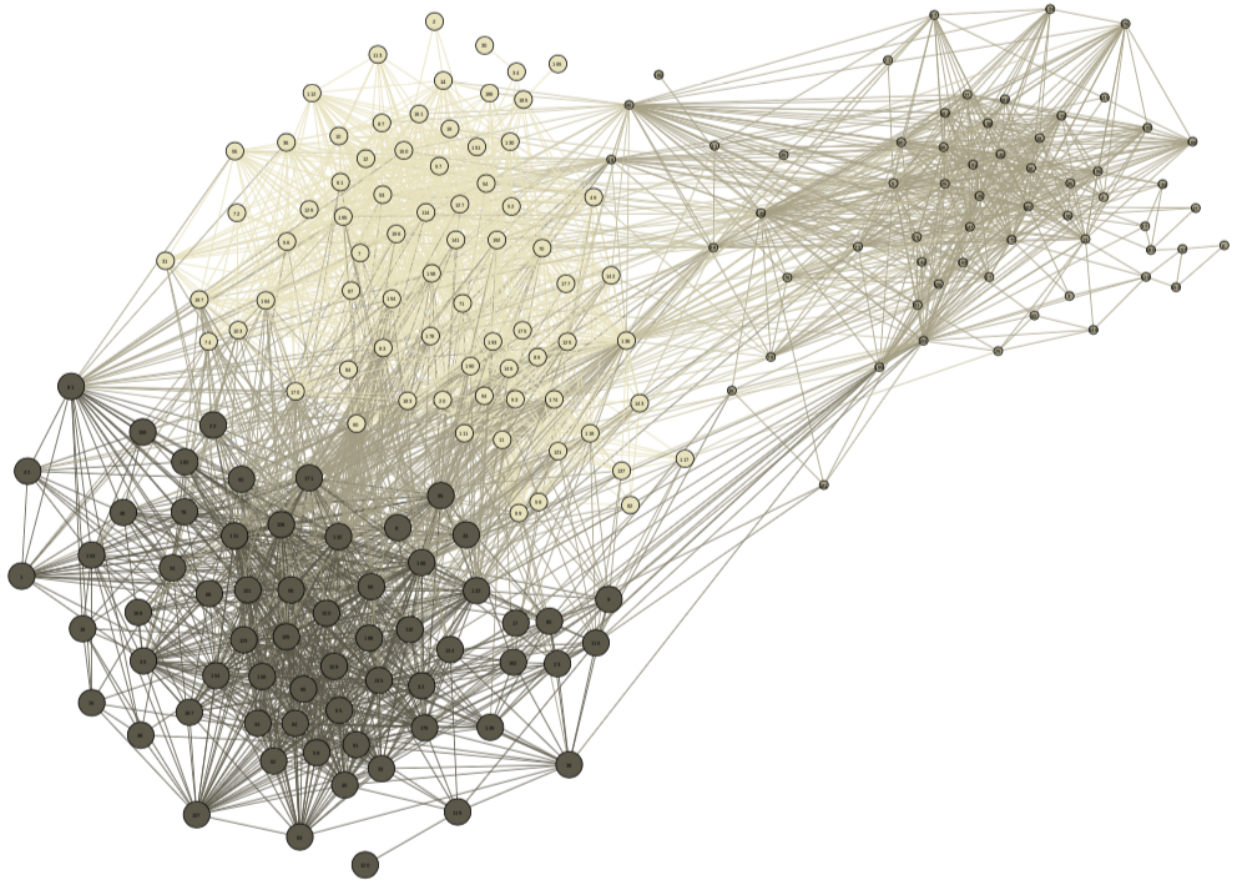
	St	Br	Gi	So	Oc	Ke	Re	Sc	Th
St	**1**	**0.62**	**0.66**	**0.63**	0.33	0.36	0.25	0.14	0.15
Br	**0.62**	**1**	**0.72**	**0.71**	0.55	0.47	0.43	0.25	0.24
Gi	**0.66**	**0.72**	**1**	**0.78**	0.47	0.49	0.43	0.28	0.26
So	**0.63**	**0.71**	**0.78**	**1**	0.55	0.5	0.44	0.31	0.29
Oc	0.33	0.55	0.47	0.55	**1**	**0.67**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

Example: Communities in Social Networks

Nodes = users.

Edges = "friendship".

Densely connected groups = communities.



The Densest k-Disjoint Clique Problem

To cluster the data we want to partition the graph into cliques with heavy support.

A **k-disjoint-clique subgraph** of a graph G is a subgraph of G induced by 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 complete subgraph induced by C:

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

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

Lifting procedure for KDC

Let $\{C_1, \ldots, C_k\}$ define a k-disjoint-clique subgraph with characteristic vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k\}$

Lift the k characteristic vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k\}$ to the rank-k matrix variable $\boldsymbol{X}$:

$$\boldsymbol{X} = \sum_{i=1}^{k} \frac{\boldsymbol{v}_i \boldsymbol{v}_i^T}{\|\boldsymbol{v}_i\|^2} = \sum_{i=1}^{k} \frac{\boldsymbol{v}_i \boldsymbol{v}_i^T}{|C_i|}$$

Want to find $\boldsymbol{X}$ that maximizes

$$\mathrm{tr}(\boldsymbol{W}\boldsymbol{X}) = \sum_{i=1}^{k} \frac{\boldsymbol{v}_i^T \boldsymbol{W} \boldsymbol{v}_i}{\|\boldsymbol{v}_i\|^2} = \sum_{i=1}^{k} d(C_i)$$

SDP Relaxation

Ignoring rank constraint and relaxing combinatorial constraints on $\boldsymbol{X}$ gives the semidefinite program:

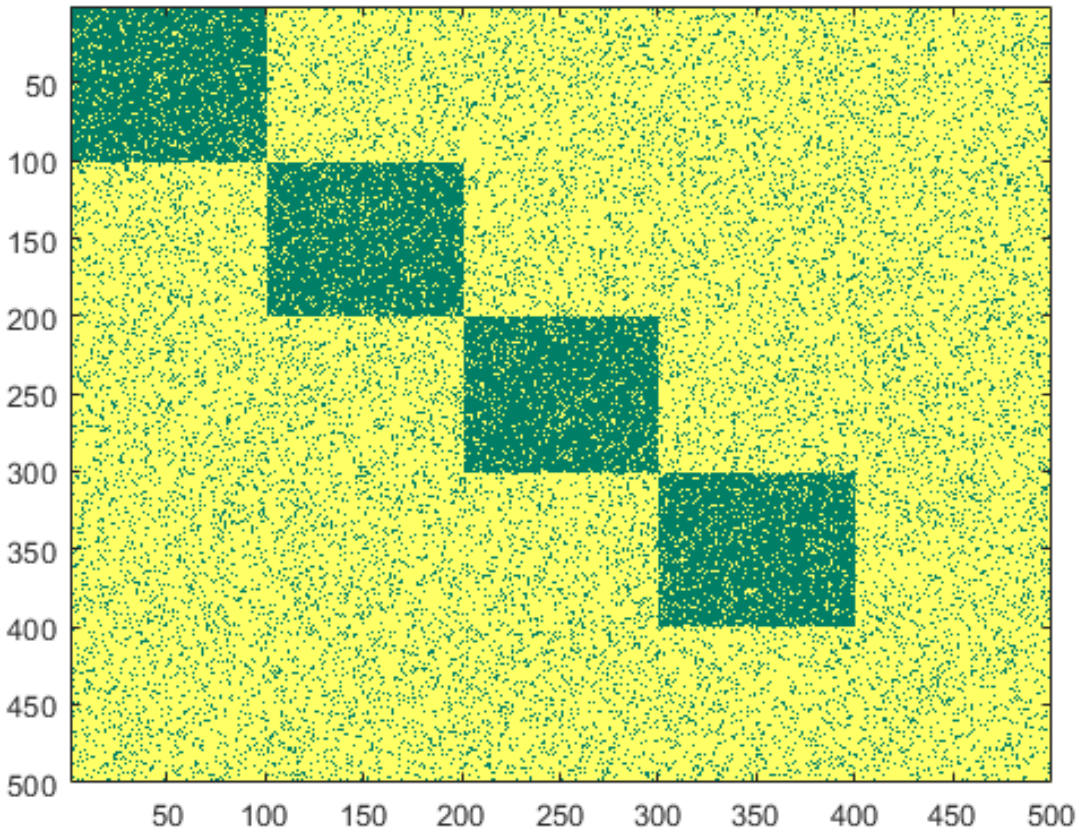
$$\begin{aligned} \max \quad & \operatorname{tr}(\boldsymbol{W}\boldsymbol{X}) \\ \text{s.t.} \quad & \boldsymbol{X}\boldsymbol{e} \leq \boldsymbol{e} \\ & \operatorname{tr}(\boldsymbol{X}) = k \\ & \boldsymbol{X} \geq \boldsymbol{0}, \;\; \boldsymbol{X} \succeq 0. \end{aligned}$$

Question: When does the optimal solution of this relaxation recover underlying cluster structure in similarity graph?

The Stochastic Block Model

Stochastic Block Model (SBM): generate random graph containing k clusters of size r:

- edges within clusters are added independently with probability p
- edges between-clusters are added with probability $q < p$.



Recovery Guarantees under the SBM

Chen/Xu (2014) characterize when graphs sampled from the SBM are:

- trivial to cluster,
- easy to cluster (have polynomial-time algorithm),
- hard to cluster (via NP-hard maximum likelihood estimation)
- impossible to cluster (data has no meaningful cluster structure).

A n-node graph sampled from SBM is **easy** to cluster if

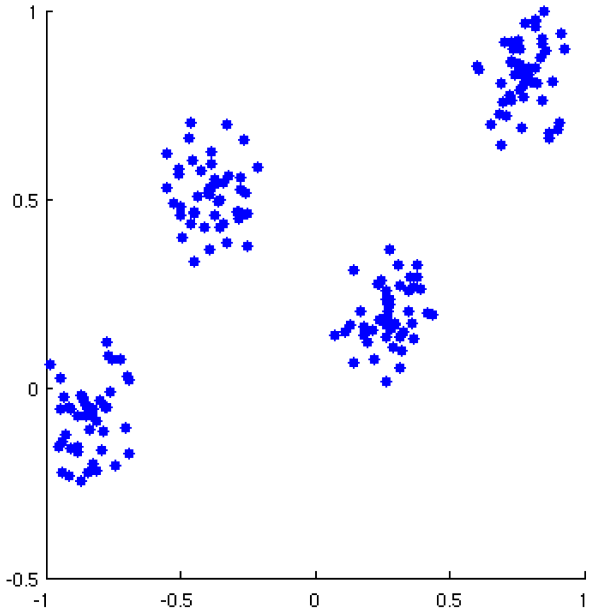
$$\frac{(p-q)^2}{q(1-q)} = \Omega\left(\frac{n}{r^2}\right).$$

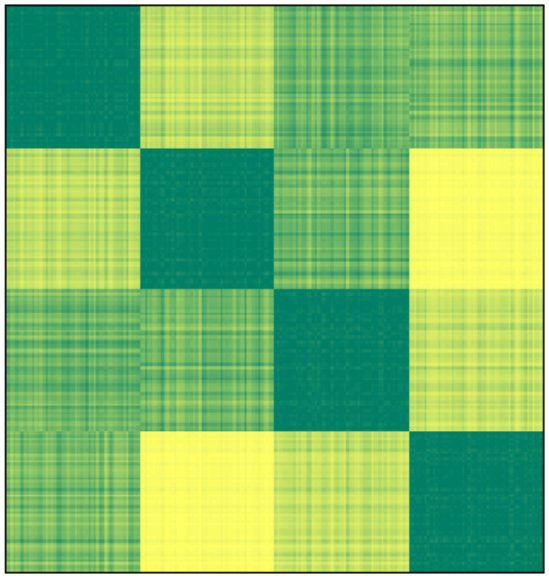
Example: Clustered Euclidean data

Suppose each data point in the ith cluster C_i is placed uniformly at random in a ball centered at $c_i \in \mathbf{R}^d$.

Distance within clusters will be small compared to the distance between clusters if centers are well-separated.

Choose $w_{ij} = \exp(-\|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2)$.





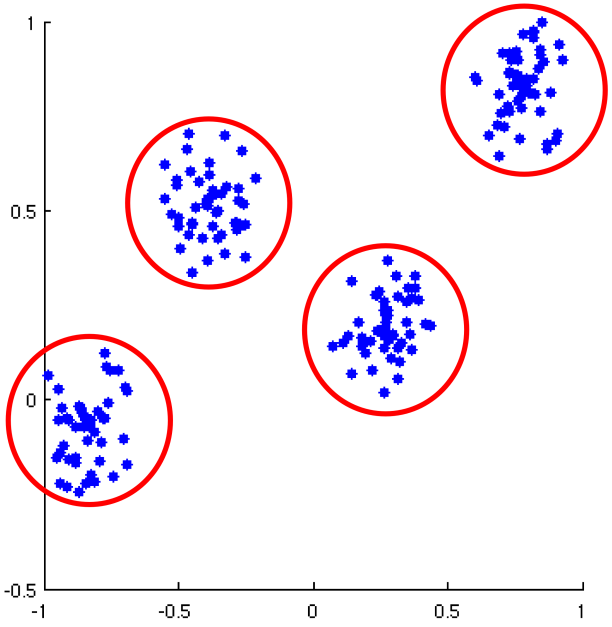
DOES NOT FIT STOCHASTIC BLOCK MODEL!!

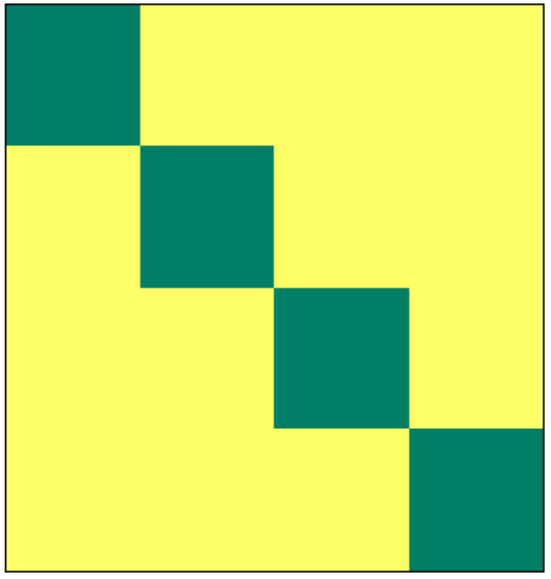
Example: Clustered Euclidean data

Suppose each data point in the ith cluster C_i is placed uniformly at random in a ball centered at $c_i \in \mathbf{R}^d$.

Distance within clusters will be small compared to the distance between clusters if centers are well-separated.

Choose $w_{ij} = \exp(-\|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2)$.





DOES NOT FIT STOCHASTIC BLOCK MODEL!!

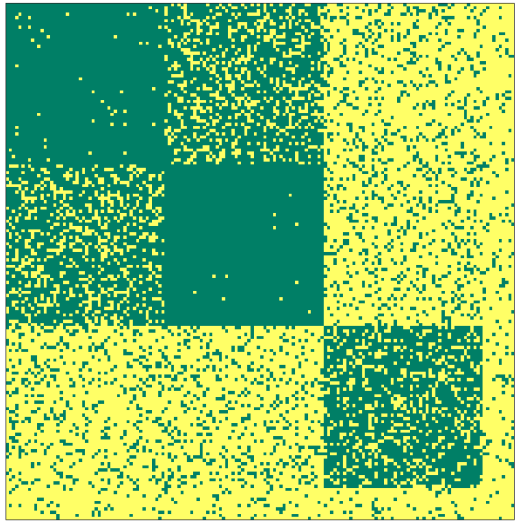
The Heterogeneous Planted Cluster Model

Assume that each node belongs to one of k clusters $C_1, C_2, \ldots, C_k$.

For each $u \in C_i$ and $v \in C_j$ we sample edge weight $w_{uv} = w_{vu}$ from distribution Ω_{ij} with

$$\boldsymbol{E}[w_{uv}] = \mu_{ij} \qquad \mathrm{Var}[w_{uv}] = \sigma_{ij}^2 \qquad 0 \leq w_{uv} \leq 1.$$

Weights within the same block are i.i.d., but weight might not be identically distributed across blocks.



Rethinking the Gap Condition

In the stochastic block model, we have perfect recovery if the gap constant $\gamma = q - p$ is sufficiently large.

In the heterogeneous case, we have perfect recovery if the **weak assortativity constant**

$$\gamma = \min_{\substack{q,s=1,2,\ldots,k \\ q \neq s}} \{\mu_{qq} - \mu_{qs}\}$$

is sufficiently large.

The Recovery Guarantee

Theorem (Pirinen-Ames 2020)

Let $\hat{\sigma} := \max_q \sigma_{qq}$ *and* $\tilde{\sigma} := \max_{q,s} \sigma_{q,s}$.

Let $\hat{r}$ *denote size of the* **smallest** *planted cluster and* r_{k+1} *denote the number of outlier nodes.*

Then there exists constant $c > 0$ *such that if*

$$\gamma \hat{r} \geq c \max \Big\{ \sqrt{\tilde{\sigma}^2 n},\ \sqrt{\tilde{\sigma}^2 \hat{r} \log n},\ \sqrt{\hat{\sigma}^2 k r_{k+1}},\ \sqrt{k r_{k+1} \log n / \hat{r}},\ \mu_{k+1,k+1} r_{k+1},\ \log n \Big\}.$$

then we have **perfect recovery with high probability**.

Signal-to-noise ratio

Suppose that the edge weight is homogeneous: $\alpha = \mu_{qq}$, $\beta = \mu_{qs}$ for all $q \neq s$.
We can recover the planted clusters w.h.p. if

$$\frac{(\alpha - \beta)^2}{\tilde{\sigma}^2} = \Omega\left(\frac{n}{\hat{r}^2}\right).$$

The left-hand side acts as a signal-to-noise ratio: ratio of difference between expected edge weights to noise variance.

This agrees with/generalizes the easy regime for cluster recovery proposed by Chen and Xu (2014).

The relaxation is mostly parameter free: SDP needs number of clusters k but doesn't need estimate of cluster sizes r_i, gap statistic $\alpha - \beta$, etc., seen in similar theoretical guarantees.

Current projects: Generalization of SBMs

More realistic planted models are needed:

- Overlapping clusters/communities;
- Finding largest of several planted clusters, possibly overlapping (without finding all clusters);
- Random graphs with **dependent** edges;
- Time-varying graphs; etc.,

Current projects: Improved Numerical Methods

Current state of the art for solving clustering SDP requires $O(n^3)$ floating point operations for singular value decomposition each iteration; algorithm converges linearly.

Cannot solve large-scale problem instances.

Investigating intermediate relaxation via non-convex quadratic programming (QP).

Solve QP using linearized ADMM, with much lower iteration complexity.

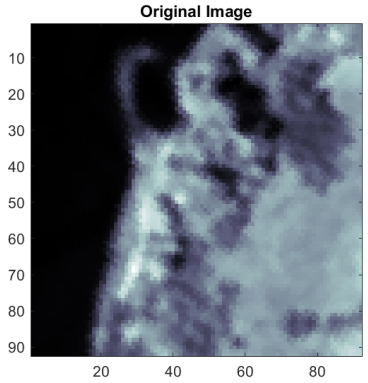
- When do we have perfect recovery?
- When does our algorithm converge?

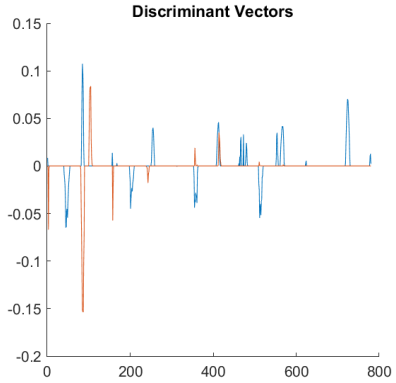
Current projects: Applied Data Analysis

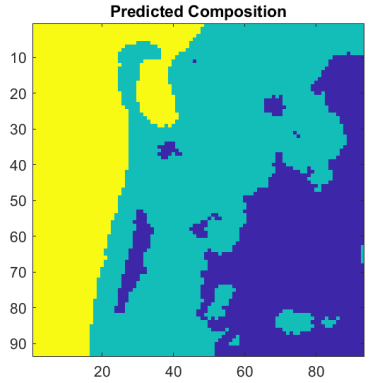
Ford et al. 2021: applied novel classification algorithm to identify comprehension of language via EEG.

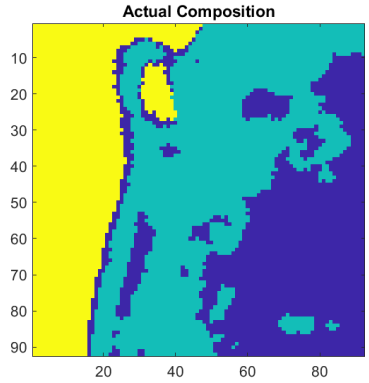
Hyperspectral segmentation. (Joint work with Mikal Webb, Carter Yancey, Julia Cartwright, Ayan Ghosh, Adam Hauser)

- Remote sensing (land-cover classification), biomedical samples (malignant vs. benign cells), geological samples (compositional/chronometric analysis)









Thank you!

P. Bombina and B. Ames. *Convex optimization for the densest subgraph and densest submatrix problems.* SN Operations Research Forum. Year: 2020, Vol: 1, No: 3.
https://link.springer.com/article/10.1007/s43069-020-00020-5

A. Pirinen and B. Ames. *Exact clustering of weighted graphs via semidefinite programming.* Journal of Machine Learning Research. Year: 2019, Vol: 20, Issue: 30, pp. 1-34.
http://jmlr.org/beta/papers/v20/16-128.html

Software available from bpames.people.ua.edu/software

B. Ames supported by NSF Grant #2012554; UA Cyberseed Grant SP14572; University of Alabama RGC grants RG14678 and RG14838.