Non-convex relaxations for the densest submatrix problem

Brendan Ames

Department of Mathematics
The University of Alabama

UA Applied Math Seminar

Friday August 26, 2022

Agenda

Consider convex and non-convex relaxations for the **maximum clique** and **densest submatrix** problems.

Give a probabilistic model for **''clusterable''** data and graphs, and theoretical recovery guarantees.

Propose efficient first-order methods for solving these relaxations.

Joint work with **Polina Bombina, UA**.

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.

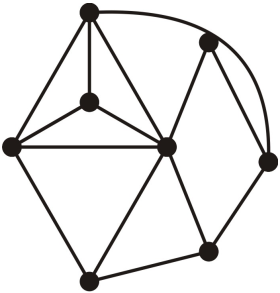
Question: can we show that we can cluster "clusterable" data?
How do we model **clusterable** data?

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**.

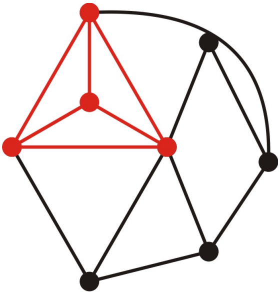


Cliques of a graph

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.

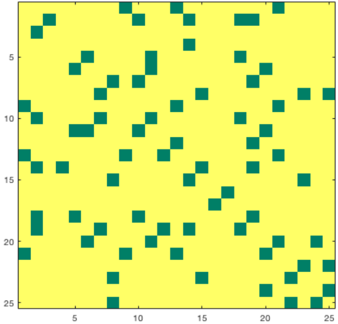
Would like an approach that translates to finding other "clique-like" objects with minimal effort.

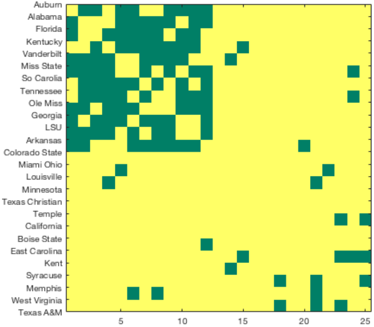
e.g., the maximum biclique of a bipartite graph, fully dense block in a matrix.

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).

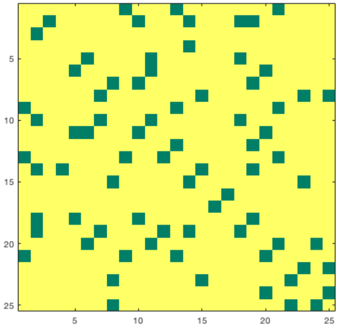


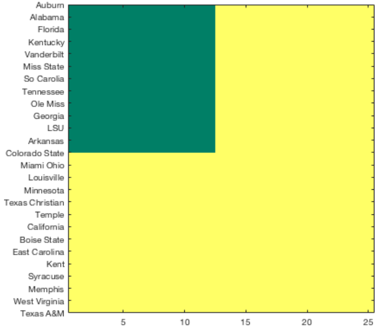


Example: Community Detection in Social Networks

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).

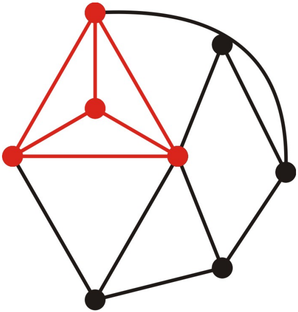


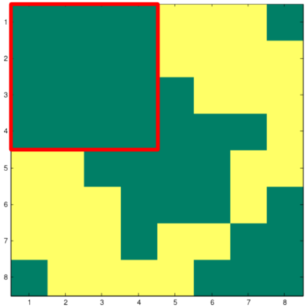


Cliques and low-rank matrices

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

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 $\|\boldsymbol{X}\|_* = \sigma_1(\boldsymbol{X}) + \cdots + \sigma_N(\boldsymbol{X})$:

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

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 $\boldsymbol{A}(U, V)$:

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

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

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

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 $\|\boldsymbol{X}\|_*$.

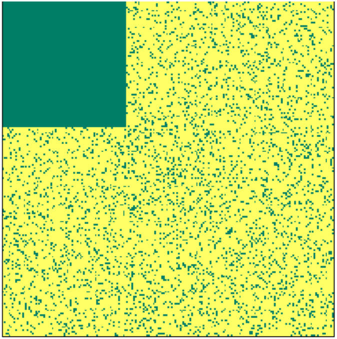
Planted case

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 $1 - q$, $q > p$.



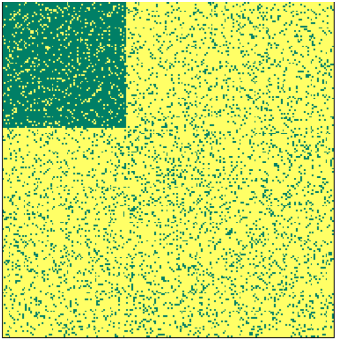
Planted case

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

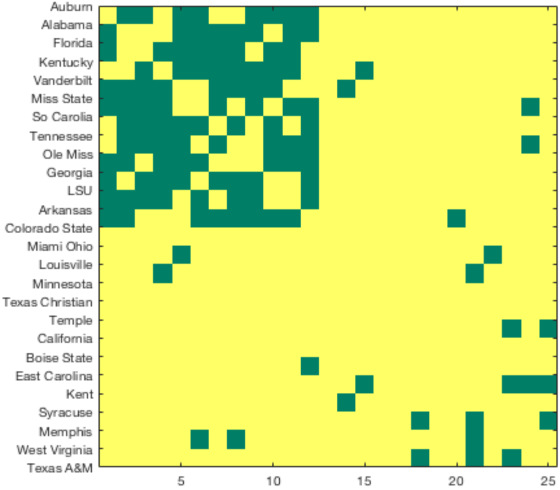
Set all entries in $m \times n$ block equal to 1.

Add noise:

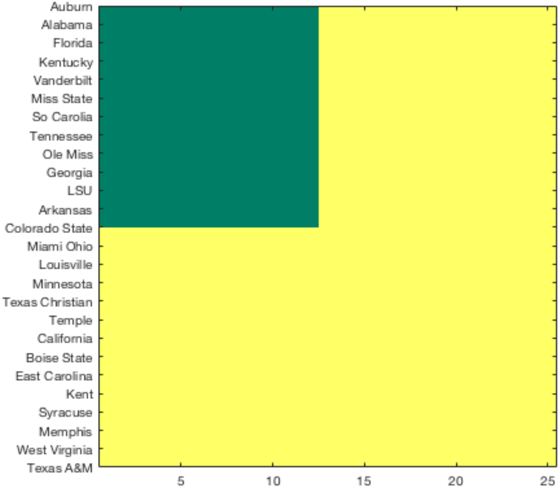
- Add some of the remaining potential entries with probability p.
- Delete some entries in $m \times n$ block with probability $1 - q$, $q > p$.



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*

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

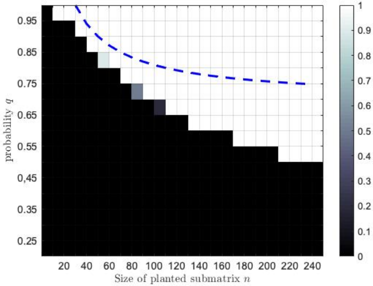
with high probability.

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.)

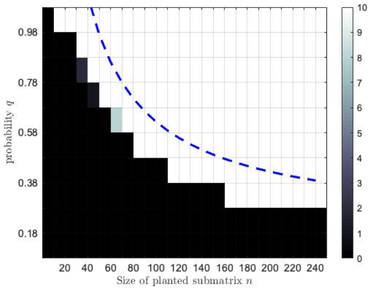
Need to modify model to use **sparse** noise: p and/or q tend to zero as $N \to \infty$.

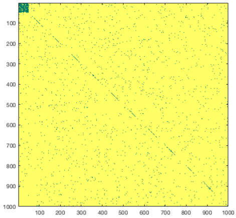
Example: Sparse Case

Suppose that noise is **sparse**.

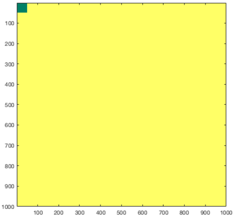
Suppose q is fixed and
$p \leq \log N/N$.

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

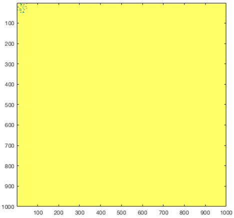




$\Rightarrow$



$+$



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).

ADMM Approach

Introduce artificial variables $\boldsymbol{Q}$, $\boldsymbol{W}$, $\boldsymbol{Z}$ to obtain the equivalent convex optimization problem

$$\min \quad \|\boldsymbol{X}\|_* + \gamma\|\boldsymbol{Y}\|_1 + \mathbf{1}_{\Omega_Q}(\boldsymbol{Q}) + \mathbf{1}_{\Omega_W}(\boldsymbol{W}) + \mathbf{1}_{\Omega_Z}(\boldsymbol{Z})$$

$$\boldsymbol{X} = \boldsymbol{Y} = \boldsymbol{Q},\ \boldsymbol{X} - \boldsymbol{W} = \mathbf{0},\ \boldsymbol{X} - \boldsymbol{Z} = \mathbf{0},$$

where $\Omega_Q, \Omega_W, \Omega_Z$ denote the constraint sets

$$\begin{aligned}
\Omega_Q &:= \{\boldsymbol{Q} : P_{\tilde{N}}(\boldsymbol{Q}) = \mathbf{0}\}, \\
\Omega_W &:= \{\boldsymbol{W} : \boldsymbol{e}^T \boldsymbol{W} \boldsymbol{e} = mn\}, \\
\Omega_Z &= \{\boldsymbol{Z} : Z_{ij} \leq 1\ \forall (i,j) \in M \times N\},
\end{aligned}$$

and $\mathbf{1}_S : \mathbf{R}^{M \times M} \to \{0, +\infty\}$ is the indicator function of the set $S \subseteq \mathbf{R}^{M \times N}$ $\Big(\mathbf{1}_S(\boldsymbol{X}) = 0$ if $\boldsymbol{X} \in S$, and $+\infty$ otherwise$\Big)$.

ADMM Idea

We solve using the **Alternating Direction Method of Multipliers (ADMM)**.

We update each primal variable by minimizing the augmented Lagrangian in Gauss-Seidel fashion with respect to each primal variable. Then the dual variables are updated using approximate gradient ascent.

ADMM Update Steps

The augmented Lagrangian is given by

$$
\begin{aligned}
L_\tau = & \|\boldsymbol{X}\|_* + \gamma \|\boldsymbol{Y}\|_1 + \mathbf{1}_{\Omega_Q}(\boldsymbol{Q}) + \mathbf{1}_{\Omega_W}(\boldsymbol{W}) + \mathbf{1}_{\Omega_Z}(\boldsymbol{Z}) \\
& + \operatorname{tr}(\boldsymbol{\Lambda}_{\boldsymbol{Q}}(\boldsymbol{X} - \boldsymbol{Y} - \boldsymbol{Q})) + \operatorname{tr}(\boldsymbol{\Lambda}_{\boldsymbol{W}}(\boldsymbol{X} - \boldsymbol{W})) + \operatorname{tr}(\boldsymbol{\Lambda}_{\boldsymbol{Z}}(\boldsymbol{X} - \boldsymbol{Z})) \\
& + \frac{\tau}{2} \left(\|\boldsymbol{X} - \boldsymbol{Y} - \boldsymbol{Q}\|_F^2 + \|\boldsymbol{X} - \boldsymbol{W}\|_F^2 + \|\boldsymbol{X} - \boldsymbol{Z}\|_F^2 \right),
\end{aligned}
$$

where τ is a regularization parameter chosen so that L_τ is strongly convex in each primal variable.

Update $\boldsymbol{Q}, \boldsymbol{W}$ and $\boldsymbol{Z}$ by projection onto each of the sets Ω_Q, Ω_W and Ω_Z.

Update $\boldsymbol{X}$ and $\boldsymbol{Y}$ using proximal operators of $\|\cdot\|_*$ and $\|\cdot\|_1$ respectively.

The Algorithm

```
while convergence==0  % Repeat until converged.
    % Update Q. Project onto support of A.
    Q = (X - Y + mu*LambdaQ).*A;

    % Update X by singular value shrinkage.
    X = mat_shrink(1/3*(Y + Q + Z + W
        - mu*(LambdaQ + LambdaW + LambdaZ)), 1/(3*tau));

    % Update Y as projection of residual onto nonnegative cone.
    Y = max(X-Q-gamma*ones(M,N)*mu + LambdaQ*mu, zeros(M,N));

    % Scale/shift W so that entries sum to m*n.
    newW = X + mu*LambdaW;
    alfa = (m*n-sum(newW(:)))/(M*N);
    W = newW + alfa*ones(M,N);

    % Update Z.
    Z = X+ mu*LambdaZ;  Z = min(max(Z,0),1);

    % Update dual variables by approximate gradient ascent.
    LambdaQ = LambdaQ + tau*(X-Y-Q);
    LambdaW = LambdaW + tau*(X-W);
    LambdaZ = LambdaZ + tau*(X-Z);
end
```

A Problem

ADMM algorithm requires $O(N^3)$ floating point operations for **singular value decomposition** each iteration; algorithm converges linearly.

Cannot solve large-scale problem instances.

Limited to graphs/matrices with $N = O(1000)$.

Quadratic Programming Relaxation

If $\operatorname{rank} \boldsymbol{X} = 1$ then $\boldsymbol{X} = \boldsymbol{u}\boldsymbol{v}^T \in \mathbf{R}^{M \times N}$ for some $\boldsymbol{u} \in \mathbf{R}^M$, $\boldsymbol{v} \in \mathbf{R}^N$.

(DSM) can be relaxed as

$$
\begin{array}{ll}
\min & \frac{\lambda}{2}\left(\|\boldsymbol{u}\|_2^2 + \|\boldsymbol{v}\|_2^2\right) + \boldsymbol{u}^T \bar{\boldsymbol{A}} \boldsymbol{v} \\
\text{s.t.} & \sum u_i = m, \qquad \sum v_i = n \\
& 0 \le u_i \le 1, \qquad 0 \le v_i \le 1
\end{array}
$$

This is a **non-convex** quadratic program in $\boldsymbol{u}$ and $\boldsymbol{v}$.

A Translation of Recovery Guarantees

Theorem

Suppose that the **nuclear norm relaxation is exact***.*

That is $\boldsymbol{X}^* = \boldsymbol{u}^*(\boldsymbol{v}^*)^T$*, is the optimal solution for* **(DSM)** *and the nuclear norm relaxation with regularization parameter* γ*.*

Then $(\boldsymbol{u}^*, \boldsymbol{v}^*)$ *is the optimal solution of the non-convex QP relaxation with*

$$\lambda \leq \frac{1}{2\gamma} \min\left\{\sqrt{\frac{m}{n}}, \sqrt{\frac{n}{m}}\right\}.$$

Proof Idea: Use optimality of $\boldsymbol{X}^*$ to establish that

$$\frac{\lambda}{2}\left(\|\boldsymbol{u}\|_2^2 + \|\boldsymbol{v}\|_2^2\right) + \boldsymbol{u}^T\bar{\boldsymbol{A}}\boldsymbol{v} \geq \frac{\lambda}{2}\left(\|\boldsymbol{u}^*\|_2^2 + \|\boldsymbol{v}^*\|_2^2\right) + (\boldsymbol{u}^*)^T\bar{\boldsymbol{A}}\boldsymbol{v}^*$$

for every feasible $\boldsymbol{u}$ and $\boldsymbol{v}$ for this choice of γ and λ.

LADMM setup

We can write the QP relaxation as

$$\begin{aligned} \min \quad & \frac{\lambda}{2}(\|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2) + \boldsymbol{u}^T \bar{\boldsymbol{A}} \boldsymbol{v} + \mathbf{1}_{\Omega_1}(\boldsymbol{x}) + \mathbf{1}_{\Omega_2}(\boldsymbol{w}) \\ \text{s. t.} \quad & \boldsymbol{u} = \boldsymbol{x}, \boldsymbol{v} = \boldsymbol{w}, \end{aligned}$$

where

$$\begin{aligned} \Omega_1 &= \{\boldsymbol{x} : \mathbf{0} \le \boldsymbol{x} \le \boldsymbol{e}, \boldsymbol{x}^T \boldsymbol{e} = m\}, \\ \Omega_2 &= \{\boldsymbol{w} : \mathbf{0} \le \boldsymbol{w} \le \boldsymbol{e}, \boldsymbol{w}^T \boldsymbol{e} = n\}. \end{aligned}$$

The augmented Lagrangian is given by:

$$\begin{aligned} L_\tau = & \frac{\lambda}{2}(\|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2) + \boldsymbol{u}^T \bar{\boldsymbol{A}} \boldsymbol{v} + \mathbf{1}_{\Omega_1}(\boldsymbol{x}) + \mathbf{1}_{\Omega_2}(\boldsymbol{w}) \\ & + \boldsymbol{\Lambda_1}^T(\boldsymbol{u} - \boldsymbol{x}) + \boldsymbol{\Lambda_2}^T(\boldsymbol{v} - \boldsymbol{w}) + \frac{\tau}{2}(\|\boldsymbol{u} - \boldsymbol{x}\|^2 + \|\boldsymbol{v} - \boldsymbol{w}\|^2) \end{aligned}$$

Outline of the Algorithm

Minimization of the augmented Lagrangian with respect to each of the artificial primal variables $\boldsymbol{x}$ and $\boldsymbol{w}$ is equivalent to projection onto the capped simplex.

To update $\boldsymbol{u}$, we replace $\boldsymbol{u}^T \bar{\boldsymbol{A}} \boldsymbol{v}^i + \frac{\lambda}{2}\|\boldsymbol{u}\|^2$ by

$$\langle \boldsymbol{u} - \boldsymbol{u}^i, \bar{\boldsymbol{A}}\boldsymbol{v}^i + \lambda \boldsymbol{u}^i \rangle + \frac{\ell_u}{2}\|\boldsymbol{u} - \boldsymbol{u}^i\|^2,$$

where ℓ_u is a regularization term.

Similarly for $\boldsymbol{v}$: we replace $\boldsymbol{u}^T \bar{\boldsymbol{A}} \boldsymbol{v} + \frac{\lambda}{2}\|\boldsymbol{v}\|^2$ by

$$\langle \boldsymbol{v} - \boldsymbol{v}^i, \bar{\boldsymbol{A}}^T \boldsymbol{u}^{i+1} + \lambda \boldsymbol{v}^i \rangle + \frac{\ell_v}{2}\|\boldsymbol{v} - \boldsymbol{v}^i\|^2,$$

where ℓ_v is a regularization term.

The LADMM Algorithm

```
while convergence==0
    %update x
    y0 = u + 1/tau*Lambda_x;
    x = projection(y0,m,tau);

    % Update u
    u = 1/(L_v+tau)*(tau*x-Lambda_x-A_bar*v+L_v*u_old-lambda*u_old);

    %update w
    y1 = v + 1/tau*Lambda_w;
    w = projection(y1,n,tau);

    % Update v
    v = 1/(L_v+tau)*(tau*w-Lambda_w-A_bar'*u+L_v*v_old-lambda*v_old);

    % Update dual variables
    Lambda_x_old = Lambda_x;
    Lambda_x = Lambda_x_old+tau*(u-k);

    Lambda_w_old = Lambda_w;
    Lambda_w = Lambda_w_old + tau*(v-w);
end
```

Remarks

The sequences of iterates $\{\boldsymbol{u}^k\}, \{\boldsymbol{v}^k\}, \{\boldsymbol{x}^k\}, \{\boldsymbol{w}^k\}$ are convergent if we choose regularization parameter τ and linearization parameters ℓ_u, ℓ_v in a certain range.

The QP relaxation is **degenerate** (i.e., doesn't satisfy usual constraint qualifications) at binary feasible solutions.

Can show that there is a non-zero duality gap between the QP relaxation and its dual for modestly large planted solutions, even when we have perfect recovery.

In practice, method converges quickly with initial solution $\boldsymbol{u}^0 = \boldsymbol{e}/m \in \mathbf{R}^M$ and $\boldsymbol{v}^0 = \boldsymbol{e}/n \in \mathbf{R}^N$.

Improvement: Adaptive LADMM

Performance depends on augmented Lagrangian parameter τ.

Number of iterations and run-time increase significantly if τ is too small or too large.

Need to automate choice of τ:

1. **Residual balancing:** increment/decrement τ^i to tune between primal and dual residuals.
2. **Line-search** to choose τ^i ensuring sufficient decrease in residual each iteration.

Empirical Trials

We randomly generate 500×500 matrices with randomly generated planted densest $m \times n$ submatrices according to the planted submatrix model with

$$n \in \{10, 20, 30, \ldots, 250\} \qquad m = 2n$$
$$p = 0.25 \qquad q \in \{0.3, 0.4, 0.5, \ldots, 1\}.$$

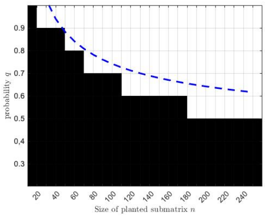
We use **ADMM**, **LADMM**, and adaptive ADMM with **line search (AdaLADMM-LS)** and **residual balancing (AdaLADMM-RB)** with $\gamma = 6/(q-p)n$ and $\lambda = (q-p)n/10$.

Augmented Lagrangian parameters and adaptation parameters are chosen to ensure convergence.

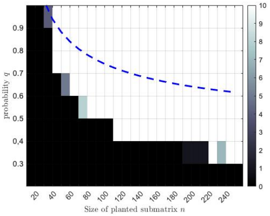
Stop each algorithm with stopping tolerance $\epsilon = 10^{-4}$ and maximum number of iterations 2000.

Recovery Rates for Randomly Generated Matrices

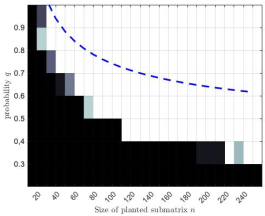
Declare DSM recovered if relative error between planted solution and calculated solution is within 10^{-2}. Repeat 10 times.



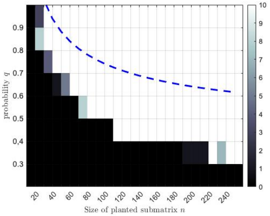
ADMM



LADMM

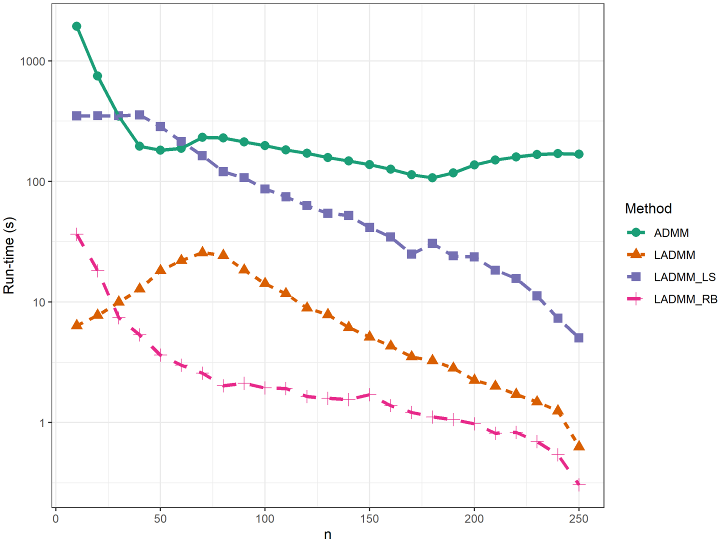


AdaLADMM:RB



AdaLADMM:LS

Run Time



1000
100
10
1
Run-time (s)
0
50
100
150
200
250
n
Method
ADMM
LADMM
LADMM_LS
LADMM_RB

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

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

B. Ames supported by **NSF Grants #2012554** and **#2108645**; **UA Cyberseed Grant SP14572**; University of Alabama RGC grants **RG14678** and **RG14838**.

P. Bombina supported by **Alabama EPSCoR Graduate Research Scholars Program**.