# Convex Relaxation for the Clique and Clustering Problems

Brendan Ames[1] Stephen Vavasis[1]

[1]Department of Combinatorics and Optimization
University of Waterloo

MS83: An Algebraic View of Sparse Optimization

2011 SIAM Conference on Optimization, Darmstadt, May 16-19

# Outline

# The clustering problem

- 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.
- Intractible in general.
- If data is actually clustered, can we identify the clusters efficiently?

# Clustering as graph partitioning

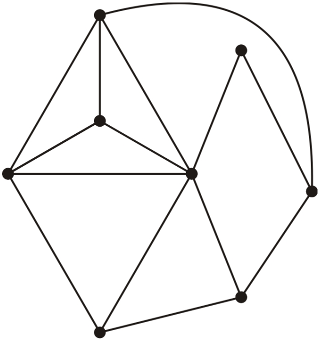
- Can model data as graph $G = (V, E)$.
- $V$ is the set of items in the data.
- Similarity is indicated by $E$:
  - Either by adjacency ($ij \in E$ iff $i, j$ are similar), or
  - By assigning edge-weights to each edge of complete graph.
- A set of nodes $C \subseteq V$ is a clique of $G$ if $ij \in E$ for all $i, j \in C$.
- Optimal partitioning of the graph into cliques gives an optimal clustering of the data.

# Cliques and rank-one matrices

- Every clique $C$ (with char vector $\mathbf{v}$) defines rank-one matrix $\mathbf{v}\mathbf{v}^T$ (equal to all-ones block in $A(G) + I$):

# Cliques and rank-one matrices

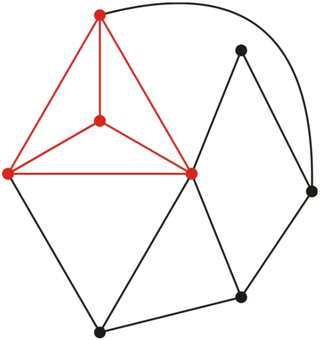
- Every clique $C$ (with char vector $\mathbf{v}$) defines rank-one matrix $\mathbf{v}\mathbf{v}^T$ (equal to all-ones block in $A(G) + I$):



$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

# Cliques and rank-one matrices

- Every clique $C$ (with char vector $\mathbf{v}$) defines rank-one matrix $\mathbf{v}\mathbf{v}^T$ (equal to all-ones block in $A(G) + I$):



$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

# The k-disjoint-clique problem

- Given graph $G = (V, E)$ and integer $k \in [1, |V|]$, a k-disjoint-clique-subgraph is a subgraph of $G$ composed of $k$ disjoint cliques.
- We consider the problem of identifying the largest $k$-disjoint-clique subgraph in a given graph $G$.
- binary case: maximize number of nodes.
- weighted case: maximize normalized edge-weights.

# The maximum node KDC problem

The maximum node k-disjoint-clique problem **(KDC)**:

Fnd a $k$-disjoint-clique subgraph containing the maximum number of nodes of input graph $G$.

# The maximum node KDC problem

**The maximum node k-disjoint-clique problem (KDC):**

Fnd a $k$-disjoint-clique subgraph containing the maximum number of nodes of input graph $G$.

**Special case:** $k = 1$ The maximum clique problem

**[A-V 09]** if input graph contains a sufficiently large clique ($\Omega(\sqrt{N})$), can solve the maximum clique problem exactly by formulating as rank minimization and relaxing to nuclear norm minimization (even though $\mathcal{A}$ does not satisfy RIP or incoherence).

# What if we want to find many cliques?

- In general, can't formulate $k$-disjoint-clique as rank minimization.
- $k$ may be very large, don't want to recover a low-rank matrix.
- Can't generate series of cuts using the maximum clique because not in the planted case.
- Solution: Relax rank constraints with combination of nuclear norm constraint and semidefinite constraint.

# Relaxation as SDP

- We relax **KDC** as the SDP

$$(*) \qquad \begin{aligned} \max \quad & \sum \sum X_{ij} \\ \text{s.t.} \quad & X\mathbf{e} \le \mathbf{e}, \\ & X_{ij} = 0, \qquad \forall (i,j) \notin E, i \neq j \\ & tr(X) = k, \\ & X \succeq 0. \end{aligned}$$

- $X\mathbf{e} \le \mathbf{e} \Rightarrow \|X\|_* \le rank(X)$ for all feasible $X \ge 0$.
- $X \succeq 0 \Rightarrow \|X\|_* = tr(X)$ for all feasible $X$.

# The planted case

- Consider graphs constructed as follows:
- Start with disjoint cliques $C_1, \ldots, C_k$ of size $r_1, \ldots, r_k$.
- Noise: add set $C_{k+1}$ containing $r_{k+1}$ additional nodes and additional edges either deterministically by an adversary or at random independently with fixed prob $p$.
- $C_1, \ldots, C_k$ induce a feasible solution of $(*)$:

$$X^* = \sum_{i=1}^{k} \frac{1}{r_i} \mathbf{v}_i \mathbf{v}_i^T$$

where $\mathbf{v}_i \in \mathbf{R}^V$ is the characteristic vector of $C_i$.

# Results

- if not too much extra noise is added then, we can recover $X^*$ (and hence $\{C_1, \ldots, C_k\}$) by solving $(*)$.
- For both formulations, can add at most $r_{k+1} = O(\hat{r}^2)$ extra nodes, where $\hat{r} = \min_{i=1,\ldots,k} r_i$.

- Adv case: Can add up to $O(\hat{r}^2)$ additional edges, provided there is at most $O(\min\{r_q, r_{cl(v)}\})$ edges from $v$ to $C_q$.
- Random case: "too many" noise edges quantified by number of cliques $k$ and the discrepancy between their sizes: need

$$\left(\sum_{s=1}^{k} {r_s}^2\right)^{1/2} \left(\sum_{q=1}^{k} \frac{1}{r_q}\right)^{1/2} \leq O(\hat{r}).$$

# Results: Random noise

- Max clique $k = 1$: $X^*$ optimal if $r_1 = \Omega(\sqrt{N})$.
- Constant clique size: $r_1 = \cdots = r_k = N^\alpha$. Can find at most $k = O(N^{\alpha/2})$ disjoint planted cliques for $\alpha \in [1/2, 2/3]$.
- Different sized cliques: e.g. $1$ large clique of size $O(N^{2/3})$ and O($N^{1/6}$) smaller cliques of size $O(N^{1/2})$.

# Proof idea

- Want to show that $X^*$ satisfies the KKT conditions for $(*)$ provided not too much noise is present.
- Difficulty arises from constructing a multiplier $S \succeq 0$ for the constraint $X \succeq 0$.
- KKT conditions give explicit formulas for the entries of $S$ corresponding to edges in $G$ but remaining entries are unknown.
- Once $S$ is chosen can establish that $S \succeq 0$ using norm bounds for the off-diagonal blocks and $S(C_{k+1}, C_{k+1})$.

# The maximum mean weight KDC problem

## Mean weight k-disjoint-clique problem **(WKDC)**:

Given a complete graph $K_N$ and edge-weights $W$, find a $k$-disjoint-clique subgraph of $K_N$ that maximizes the sum of average weights covered by each clique.

We relax **WKDC** as the SDP

$$(w*) \quad \begin{array}{rl} \max & tr(WX) \\ \text{s.t.} & X\mathbf{e} \le \mathbf{e}, \\ & X \ge 0, \\ & tr(X) = k, \\ & X \succeq 0 \end{array}$$

# Weighted planted case

- Unlike **(KDC)**, planted case does not correspond to any particular structure in input graph.
- Instead, is induced by edge-weight matrix $W$: entries of $W$ corresponding to the planted $k$-disjoint-clique subgraph are larger than the rest.
- Consider random symmetric $k+1 \times k+1$ block matrix $W \in \Sigma^N$ with i.i.d. entries in $[0,1]$ in each block such that

$$E[W_{C_q,C_s}] = E[W_{C_s,C_q}] = \begin{cases} \alpha \mathbf{e}\mathbf{e}^T, & \text{if } q = s,\ 1 \leq q,s \leq k \\ \beta \mathbf{e}\mathbf{e}^T, & \text{otherwise} \end{cases}$$

for $\alpha > 2\beta$ and partitioning $\{C_1, \dots, C_{k+1}\}$ of $V$.

# Results: weighted case

- If not too much noise, the relaxation $(w*)$ of **WKDC** is exact with extremely high probability.
- Can tolerate up to $O(\hat{r})$ additional nodes.
- As before, too much edge noise is quantified by number of cliques $k$ and the discrepancy between their sizes:

$$\left( \sum_{s=1}^{k+1} r_s \right)^{1/2} \leq O(\hat{r}).$$

# Rehnquist Supreme Court

- Data set is the set of U.S. Supreme Court Justices (serving from 1994-95 to 2003-04).
- 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

- We solved **WKDC** with $k = 2$ using SeDuMi. We obtained the following partition of the supreme court:

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

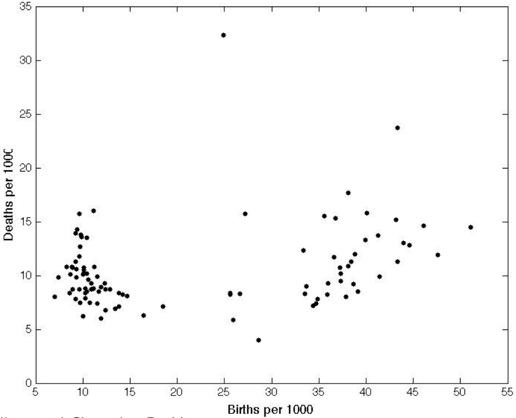
# Rehnquist Supreme Court: results

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

| 1: "Most Conservative" | 2: "Moderate Conservative" | 3: "Liberal" |
|---|---|---|
| Thomas (Th)<br>Scalia (Sc) | O'Connor (Oc)<br>Kennedy (Ke)<br>Rehnquist (Re) | Stevens (St)<br>Breyer (Br)<br>Ginsberg (Gi)<br>Souter (So) |

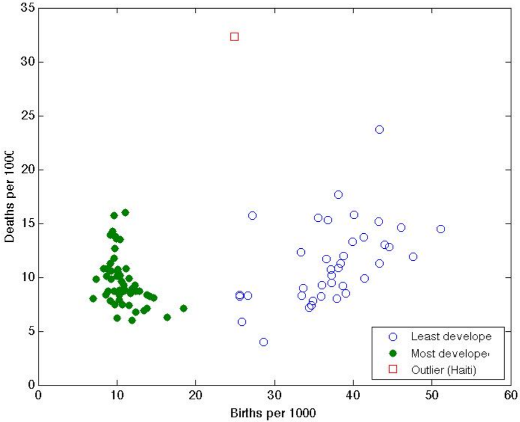
# Birth/Death rates

- Source: *2010 Crude Birth & Death, Net Migration, & Growth Rates, Births, Deaths, & Migrants*, U.S. Census Bureau, International Data Base.



# Birth/Death rates: results

- Use $W = \exp(-\|\mathbf{v}_i - \mathbf{v}_j\|^2/\sigma^2)$.
- Solve using SDPNAL with $k = 2$:



# Conclusions

- The clique and clustering problems be relaxed as SDPs.
- We obtain the exact solution in polynomial time if the data contains the desired hidden structure.
- Future work:
  - extension to other problems: eg. other graph-partitioning objectives (e.g. normalized cut).
  - unify spectral clustering results: if in planted case is the partitioning given by eigenvectors of $A(G)$ or $W$ exact?

# Thank you!


- B. Ames and S. Vavasis (2010). *Convex optimization for the planted k-disjoint-clique problem* arXiv:1008.2814
- B. Ames and S. Vavasis (2009). *Nuclear norm minimization for the planted clique and biclique problems* arXiv:0901.3348