Convex Relaxation for the Clique and Clustering Problems

Brendan Ames\textsuperscript{1} \hspace{1cm} Stephen Vavasis\textsuperscript{1}

\textsuperscript{1}Department of Combinatorics and Optimization
University of Waterloo

MB2: Semidefinite and Convex Relaxations for Discrete Optimization Problems

2011 Optimization Days, May 2-4
• **Data mining/Information retrieval**: want to find a pattern hidden in an apparently unstructured set of data

• These problems are intractible in general.

• What if the data consists of the desired structure hidden by random noise?

• **e.g. Clustering**: if the data is actually clustered, can we identify the clusters efficiently?
Rank minimization

- **Affine rank minimization problem (RMP):** find matrix with minimum rank satisfying linear constraints:

\[
\min \{ \text{rank}(X) : \mathcal{A}(X) = b \}
\]

(here \(\mathcal{A} : \mathbb{R}^{m \times n} \to \mathbb{R}^p\) linear, \(b \in \mathbb{R}^p\)).

- NP-hard but can be solved in the case that \(\mathcal{A}\) is well-structured by relaxing to convex programming.

- Replace \(\text{rank}(X)\) with nuclear norm

\[
\|X\|_* = \sigma_1(X) + \cdots + \sigma_m(X). \quad (\text{NNM})
\]

Cliques and rank-one matrices

- Given a graph $G = (V, E)$, a set of nodes $C \subseteq V$ is clique of $G$ if $ij \in E$ for all $i, j \in C$.
  - $C$ induces a complete subgraph of $G$.
- Relation to rank-min: every clique $C$ (with char vector $v$) defines rank-one matrix $vv^T$.
  - Block in adjacency matrix of $G$ plus extra 1s on diagonal.
- [A-V 09] if input graph contains sufficiently large clique ($\Omega(\sqrt{N})$), can solve the maximum clique problem in polynomial time by formulating as RMP and relaxing to NNM.
• **Clustering**: Given set of data, group them into clusters such that items in each cluster are similar each other and items not in the same cluster are dissimilar.

• 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 (\( ij \in E \text{ iff } i, j \) are similar) or by edge-weights.

• If we find optimal partitioning of the graph into cliques, then have an optimal clustering of the data.
The k-disjoint-clique problem

- Given graph $G = (V, E)$ with $|V| = N$ and integer $k \in [1, N]$, a **k-disjoint-clique-subgraph** is a subgraph of $G$ composed of $k$ disjoint cliques.

- **Maximum node k-disjoint-clique problem (KDC)**: find a $k$-disjoint-clique subgraph containing the maximum number of nodes of $G$.

- **Mean weight k-disjoint-clique problem (WKDC)**: Given 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.
What if we want to find *many* cliques?

- Can't formulate $k$-disjoint-clique as rank minimization.
- $k$ may be very large, don’t want to recover a low-rank matrix.
- **Solution:** Relax rank constraints with combination of nuclear norm constraint and semidefinite constraint.
Relaxation as SDP

• Max node $k$-disjoint-clique can be relaxed as

\[
\begin{align*}
\text{max} & \quad \sum \sum X_{ij} \\
\text{s.t.} & \quad Xe \leq e, \\
\text{(*)} & \quad X_{ij} = 0, \quad \forall (i,j) \notin E, i \neq j \\
& \quad tr(X) = k, \\
& \quad X \succeq 0
\end{align*}
\]

• $Xe \leq e \Rightarrow \|X\|_* \leq \text{rank}(X)$ for all feasible $X$.

• $X \succeq 0 \Rightarrow \|X\|_* = tr(X)$ 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 (*):

$$X^* = \sum_{i=1}^{k} \frac{1}{r_i} v_i v_i^T$$

where $v_i \in \mathbb{R}^V$ 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 $(\ast)$.
- 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 $O(N^{1/2})$. 
Weighted case SDP relaxation

- Relax \((WKDC)\) as

\[
\max \quad \text{tr}(WX) \\
\text{s.t.} \quad Xe \leq e, \\
\quad (w^*) \\
\quad X \geq 0, \\
\quad \text{tr}(X) = k, \\
\quad X \succeq 0
\]
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

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

for $\alpha > 2\beta$ and partitioning $\{C_1, \ldots, C_{k+1}\}$ of $V$. 
Results: weighted case

- If not too much noise, can solve \((WKDC)\) by solving \((w^*)\).
- 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: need
  
  \[
  \left( \sum_{s=1}^{k+1} r_s \right)^{1/2} \leq O(\hat{r}).
  \]
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:

<table>
<thead>
<tr>
<th></th>
<th>St</th>
<th>Br</th>
<th>Gi</th>
<th>So</th>
<th>Oc</th>
<th>Ke</th>
<th>Re</th>
<th>Sc</th>
<th>Th</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>St</td>
<td>1</td>
<td>0.62</td>
<td>0.66</td>
<td>0.63</td>
<td>0.33</td>
<td>0.36</td>
<td>0.25</td>
<td>0.14</td>
</tr>
<tr>
<td>2</td>
<td>Br</td>
<td>0.62</td>
<td>1</td>
<td>0.72</td>
<td>0.71</td>
<td>0.55</td>
<td>0.47</td>
<td>0.43</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td>Gi</td>
<td>0.66</td>
<td>0.72</td>
<td>1</td>
<td>0.78</td>
<td>0.47</td>
<td>0.49</td>
<td>0.43</td>
<td>0.28</td>
</tr>
<tr>
<td>4</td>
<td>So</td>
<td>0.63</td>
<td>0.71</td>
<td>0.78</td>
<td>1</td>
<td>0.55</td>
<td>0.5</td>
<td>0.44</td>
<td>0.31</td>
</tr>
<tr>
<td>5</td>
<td>Oc</td>
<td>0.33</td>
<td>0.55</td>
<td>0.47</td>
<td>0.55</td>
<td>1</td>
<td>0.67</td>
<td>0.71</td>
<td>0.54</td>
</tr>
<tr>
<td>6</td>
<td>Ke</td>
<td>0.36</td>
<td>0.47</td>
<td>0.49</td>
<td>0.5</td>
<td>0.67</td>
<td>1</td>
<td>0.77</td>
<td>0.58</td>
</tr>
<tr>
<td>7</td>
<td>Re</td>
<td>0.25</td>
<td>0.43</td>
<td>0.43</td>
<td>0.44</td>
<td>0.71</td>
<td>0.77</td>
<td>1</td>
<td>0.66</td>
</tr>
<tr>
<td>8</td>
<td>Sc</td>
<td>0.14</td>
<td>0.25</td>
<td>0.28</td>
<td>0.31</td>
<td>0.54</td>
<td>0.58</td>
<td>0.66</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Th</td>
<td>0.15</td>
<td>0.24</td>
<td>0.26</td>
<td>0.29</td>
<td>0.54</td>
<td>0.59</td>
<td>0.68</td>
<td>0.79</td>
</tr>
</tbody>
</table>
Rehnquist Supreme Court: results

- We solved \textbf{WKDC} with \( k = 2 \) using SeDuMi. We obtained the following partition of the supreme court:

<table>
<thead>
<tr>
<th>1: “Liberal”</th>
<th>2: “Conservative”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stevens (St)</td>
<td>O’Connor (Oc)</td>
</tr>
<tr>
<td>Breyer (Br)</td>
<td>Kennedy (Ke)</td>
</tr>
<tr>
<td>Ginsberg (Gi)</td>
<td>Rehnquist (Re)</td>
</tr>
<tr>
<td>Souter (So)</td>
<td>Scalia (Sc)</td>
</tr>
<tr>
<td></td>
<td>Thomas (Th)</td>
</tr>
</tbody>
</table>
Rehnquist Supreme Court: results

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

<table>
<thead>
<tr>
<th>1: “Most Conservative”</th>
<th>2: “Moderate Conservative”</th>
<th>3: ”Liberal”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thomas (Th)</td>
<td>O’Connor (Oc)</td>
<td>Stevens (St)</td>
</tr>
<tr>
<td>Scalia (Sc)</td>
<td>Kennedy (Ke)</td>
<td>Breyer (Br)</td>
</tr>
<tr>
<td></td>
<td>Rehnquist (Re)</td>
<td>Ginsberg (Gi)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Souter (So)</td>
</tr>
</tbody>
</table>
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 $W = \exp\left(-\|v_i - v_j\|^2/\sigma^2\right)$.
- Solve using SDPNAL with $k = 2$:

![Graph showing birth and death rates](image)
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: densest subgraph, other graph-partitioning objectives (e.g. normalized cut).
• unify spectral clustering results: if in planted case is the partitioning given by eigenvectors of $W$ exact?
Thank you!