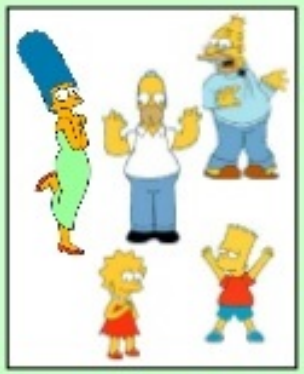
Spectral Clustering of Graph Vertex Subsets via Krylov Subspace Model Reduction

Alexander V. Mamonov[1],
Vladimir Druskin[2] and Mikhail Zaslavsky[3]

[1]University of Houston,
[2]Worcester Polytechnic Institute,
[3]Schlumberger-Doll Research Center

Support: NSF DMS-1619821, ONR N00014-17-1-2057

Clustering problem



Simpson's Family

School Employees

Females

Males

- **Unsupervised** machine learning
- **Input:** unlabeled data set $V = \{x_1, \ldots x_N\} \subset \mathbb{R}^d$, number of clusters K and some **measure of similarity** between data points
- **Output:** clusters $C_1, \ldots, C_K \subset V$ such that
  - $V = \sqcup_{k=1}^{K} C_k$ (hard assignment)
  - $x_i, x_j \in C_k$ if x_i is "similar" to x_j

Classical approach to hard assignment: K-means

- Assume similarity measure is **Eucledean distance**
- **Integer optimization problem**:

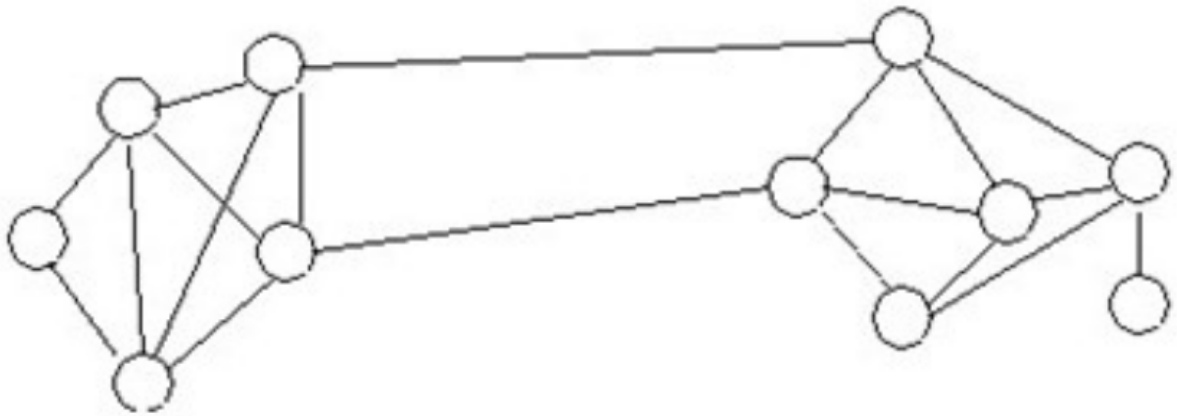
$$\sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \to min,$$

  where **centroids** are $\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$
- **Highly non-convex** objective
- **Global minimization is NP-hard**
- Simplest greedy relaxation: update centroids via fixed point iteration (**Lloyd's algorithm**), gets stuck in **local minima**
- **Robust relaxations are expensive**: Semi-Definite Programming (SDP) handles problems up to $N = 200$ in reasonable time
- How to overcome prohibitive computational cost?
- We will try to **reduce both** d **and** N

Graph-based formulation

- Consider **undirected weighted graph** $G = (V; E; \mathbf{W})$ with N vertices (data) V, edges E and **positive weights W** assigned to each edge
- Entries of **adjacency matrix** $\mathbf{W} = [w_{ij}]_{i,j=1}^{N}$ encode the **similarity** between data points x_i and x_j (assume $w_{ii} = 0$)
- **Vertex degrees** are $d_i = \sum_{j=1}^{N} w_{ij}$ and

  $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_N)$ is the **degree matrix**
- **Volume** of a data subset $A \subseteq V$ is $\mathrm{vol}(A) = \sum_{x_j \in A} d_j$



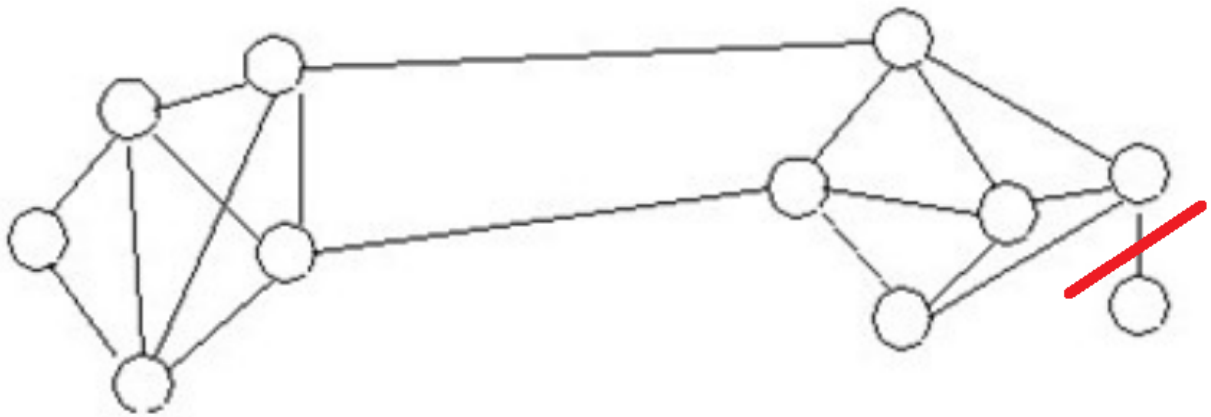
Clustering via graph cuts: 2 clusters

- **Brute-force** approach: remove edges with low similarity to split $V = C_1 \sqcup C_2$ with no edges between C_1 and C_2
- Corresponds to **"min-cut"** formulation

$$\underset{C_1, C_2}{\text{minimize}} \ \text{cut}(C_1, C_2),$$

where $\text{cut}(C_1, C_2) = \sum_{x_i \in C_1, x_j \in C_2} w_{ij}$

- Problem: creates small clusters (including single-vertex)

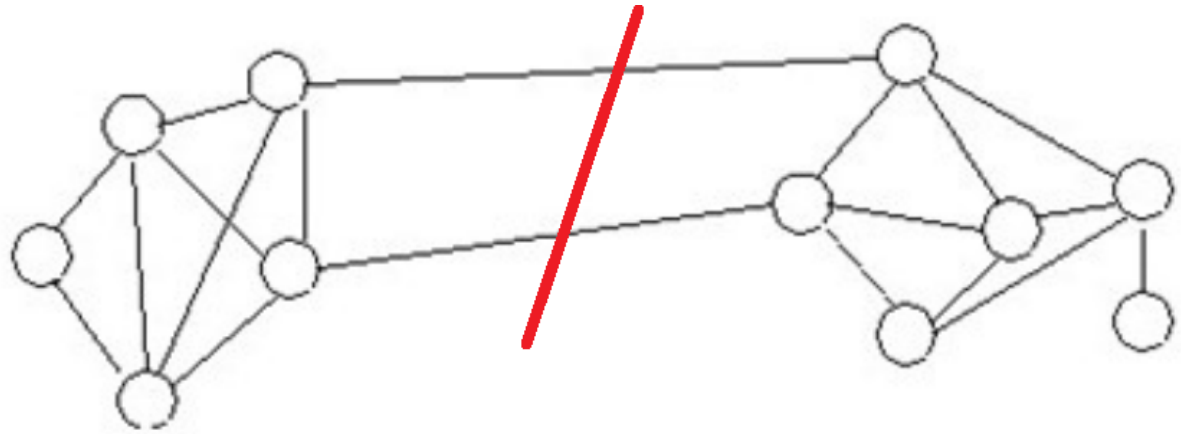


Clustering via graph cuts: 2 clusters

- Possible solution: use **normalized cut** instead

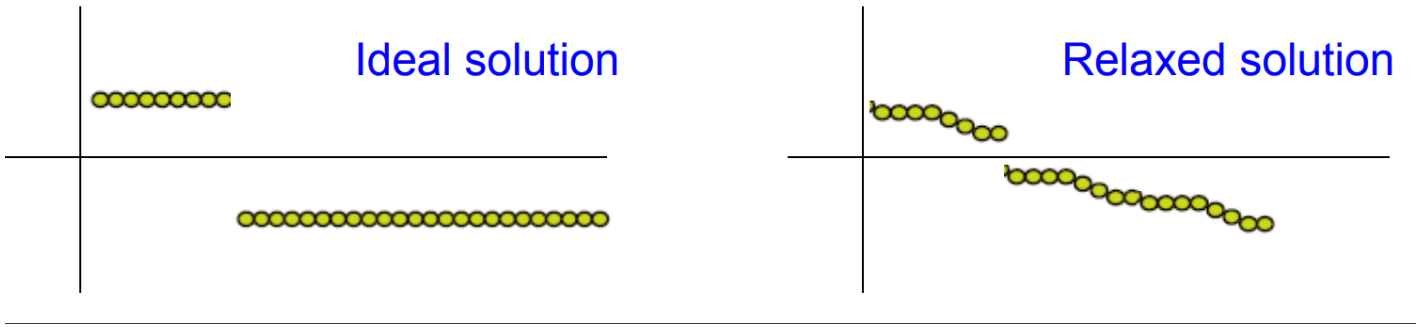
$$\mathrm{Ncut}(C_1, C_2) = \mathrm{cut}(C_1, C_2)\left(\frac{1}{\mathrm{vol}(C_1)} + \frac{1}{\mathrm{vol}(C_2)}\right)$$

- Another issue: minimizing $\mathrm{Ncut}(C_1, C_2)$ is **NP-hard**



Relaxation: Spectral Clustering (SC), 2 clusters

- **Graph-Laplacian**: $\mathbf{L} = \mathbf{D} - \mathbf{W}$, symmetric, non-negative definite
- Zero always an eigenvalue: $\mathbf{Le} = \mathbf{0}$ with $\mathbf{e} = (1, 1, \ldots, 1)^T \in \mathbb{R}^N$
- Normalized cut becomes $\text{Ncut}(C_1, C_2) = \mathbf{z}^T\mathbf{Lz}$,
  where $\mathbf{z} \in \mathbb{R}^N$ is the normalized **indicator vector**:
  $z_i = \sqrt{\frac{vol(C_2)}{vol(V)vol(C_1)}}$, if $x_i \in C_1$, $\quad z_i = -\sqrt{\frac{vol(C_1)}{vol(V)vol(C_2)}}$, if $x_i \in C_2$
- Properties: $\|\mathbf{z}\|_\mathbf{D} = 1$ and $\mathbf{z} \perp \mathbf{e}$
- **Relaxed min-Ncut:** $\underset{\mathbf{z}\perp\mathbf{e}}{\text{minimize}} \dfrac{\mathbf{z}^T\mathbf{Lz}}{\mathbf{z}^T\mathbf{Dz}}$ (Rayleigh quotient)
- **Solution**: second eigenvector of $\mathbf{L}u = \lambda\mathbf{D}u$,
  form clusters according to signs of components



Spectral Clustering (SC): general case

Algorithm:

1. Compute $K-1$ eigenvectors $\left\{\mathbf{z}^j\right\}_{j=1}^{K-1}$ of the **generalized** eigenvalue problem $\mathbf{L}u = \lambda \mathbf{D}u$ corresponding to $K-1$ **smallest** non-zero eigenvalues
2. Perform **approximate K-means** (e.g., with Lloyd's algorithm) on spectral data $\mathbf{Z} = \left[\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^{K-1}\right] \in \mathbb{R}^{N\times(K-1)}$

Spectral clustering achieves:

- **Data dimensionality reduction**: from d, unrelated to number of clusters, to $K-1$
- **Flattening of the data manifold**: spectral data $\mathbf{Z}$ provides a parametrization of the manifold closer to its "intrinsic" dimension

Problems:

- Number of data points N still large
- Main idea: **cluster small data subsets** $V_m = \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$, $m \ll N$, separately, merge afterwards

Divide-and-conquer approach via subset clustering

- Ultimate goal: **divide-and-conquer** clustering algorithm
  1. **Split** the data (and thus graph) into disjoint **target subsets** V_m with $m \ll N$ data points each
  2. **Clusterize target subsets** V_m separately
  3. **Project** $(\mathbf{L}, \mathbf{D})$ on cluster indicator vectors
  4. **Clusterize projected graph**, if still too large, repeat recursively
- Here we focus on **step 2**
- Target subset clustering must respect the **overall graph structure**, cannot just discard $V \setminus V_m$
- What to replace the rest of the graph with?
- Consider **random-walk normalized graph-Laplacian**:

$$\mathbf{L}_{RW} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$$

- Note that $\mathbf{D}^{-1}\mathbf{W} = \mathbf{I} - \mathbf{L}_{RW}$ is **Markov matrix**, a transition matrix for a **random walk** on the graph

Diffusion transfer function and model reduction

- **Long-time** random walk limit is **diffusion**
- Restrict diffusive response to target subset $V_m = \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$: consider **source/receiver** matrix $\mathbf{B} = [\mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_m}] \in \mathbb{R}^{N \times m}$
- Diffusive behavior on V_m is completely described by **discrete-time diffusion transfer function**

$$\mathbf{F}(p) = \mathbf{B}^T \mathbf{D}(\mathbf{I} - \mathbf{L}_{RW})^p \mathbf{B} \in \mathbb{R}^{m \times m}, \quad p = 1, 2, \ldots$$

- This is a transfer function of a multi-input multi-output (MIMO) **dynamical system**
- All standard **model order reduction (MOR)** techniques apply!
- Approximate the transfer function by

$$\mathbf{F}(p) \approx \widetilde{\mathbf{F}}(p) = \mathbf{E}_1^T \widetilde{\mathbf{D}}(\mathbf{I} - \widetilde{\mathbf{L}}_{RW})^p \mathbf{E}_1,$$

where $\widetilde{\mathbf{L}}_{RW} \in \mathbb{R}^{n \times n}$ is the **reduced-order graph-Laplacian (ROGL)**, $n \ll N$

Projection-based model reduction

- Look for ROGL in the form $\widetilde{\mathbf{L}}_{RW} = \widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{L}}$ where $\widetilde{\mathbf{L}}$, $\widetilde{\mathbf{D}}$ are **projections** of $\mathbf{L}$, $\mathbf{D}$ on a properly chosen **Krylov subspace**
- We are interested in $\mathbf{F}(p) \approx \widetilde{\mathbf{F}}(p)$ at **late times** $p \gg 1$ which corresponds to **lower part of the spectrum**, the one used in clustering
- Define **normalized graph-Laplacian** $\mathbf{A} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} \in \mathbb{R}^{N\times N}$
- **Ideally**, we want to project on

$$\mathcal{K}(\mathbf{A}^\dagger, \mathbf{B}) = \text{colspan}\left\{\mathbf{B}, \mathbf{A}^\dagger\mathbf{B}, \ldots, (\mathbf{A}^\dagger)^{k-1}\mathbf{B}\right\}$$

  via Lanczos process
- **Infeasible** in practice, **too expensive**: $N \gg 1$
- Replace projection on $\mathcal{K}(\mathbf{A}^\dagger, \mathbf{B})$ by **two Lanczos** processes
- Then use a **third Lanczos** process to recover a **semi-sparse** structure and obtain the ROGL

Two-stage deflated block Lanczos process for MOR

1. Construct a **deflated block-tridiagonal** proxy $\mathbf{T}_1 \in \mathbb{R}^{n_1 \times n_1}$ of $\mathbf{A}$ by projecting on

$$\mathcal{K}_{k_1}(\mathbf{A}, \mathbf{B}) = \text{colspan}\{\mathbf{B}, \mathbf{AB}, \ldots, \mathbf{A}^{k_1-1}\mathbf{B}\},$$

choosing number of steps k_1 to control approximation

$$\mathbf{F}_1(p) = \mathbf{E}_1^T(\mathbf{I} - \mathbf{T}_1)^p\mathbf{E}_1 \approx \mathbf{F}(p)$$

2. Perform second **deflated block Lanczos** process to obtain $\mathbf{T}_2 \in \mathbb{R}^{n_2 \times n_2}$, the projection of $(\mathbf{T}_1 + s_0\mathbf{I})^{-1}$ on

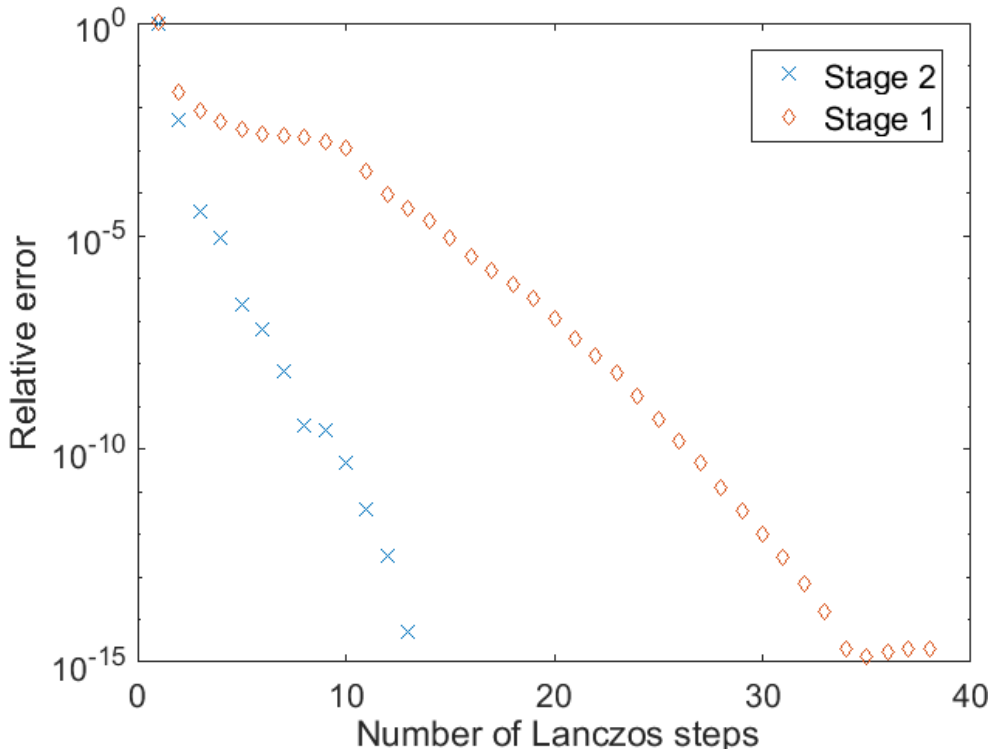
$$\mathcal{K}_{k_2}((\mathbf{T}_1+s_0\mathbf{I})^{-1}, \mathbf{E}_1) = \text{colspan}\{\mathbf{E}_1, (\mathbf{T}_1+s_0\mathbf{I})^{-1}\mathbf{E}_1, \ldots, (\mathbf{T}_1+s_0\mathbf{I})^{-k_2+1}\mathbf{E}_1\}$$

choosing k_2 to control approximation

$$\mathbf{F}_2(p) = \mathbf{E}_1^T(\mathbf{I} - (\mathbf{T}_2^{-1} - s_0\mathbf{I}))^p\mathbf{E}_1 \approx \mathbf{F}_1(p)$$

Two-stage model reduction: approximation accuracy

Convergence curves for approximations $\mathbf{F}(p) - \mathbf{F}_1(p)$ (Stage 1) and $\mathbf{F}_1(p) - \mathbf{F}_2(p)$ (Stage 2) versus the numbers of Lanczos steps k_1 and k_2



- Dataset for collaborations in arXiv:AstroPhysics $N = 18872$
- Target subset V_m with $m = 20$ vertices
- ROM transfer function accuracy 10^{-15} for $n_2 = 20 \times 13 = 260$

Reduced-order graph-Laplacian (ROGL)

- Note that $\mathbf{T}_2^{-1}$ is dense. To recover sparse ROM for the graph-Laplacian, employ the **third Lanczos process**
- Project $\mathbf{T}_2^{-1} - s_0\mathbf{I}$ on

$$\mathcal{K}_{k_2}(\mathbf{T}_2^{-1}, \mathbf{E}_1) = \text{colspan}\{\mathbf{E}_1, \mathbf{T}_2^{-1}\mathbf{E}_1, \ldots, \mathbf{T}_2^{-k_2+1}\mathbf{E}_1\}$$

  to find **deflated block-tridiagonal** $\mathbf{T}_3 \in \mathbb{R}^{n\times n}$
- Here $n = n_2$ and the **transfer function** is **preserved exactly** from the second stage
- To normalize properly, choose $\mathbf{z}_0$ in the approximate nullspace of $\mathbf{T}_3$ and let

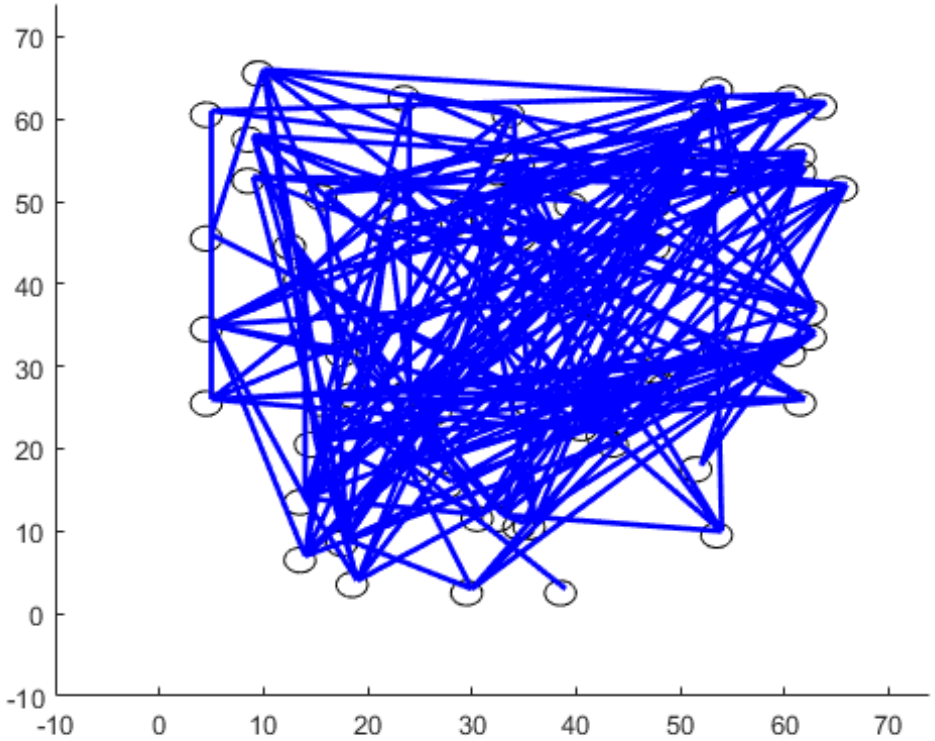
$$\widetilde{\mathbf{D}} = \text{diag}(\mathbf{z}_0)$$

- ROGL becomes

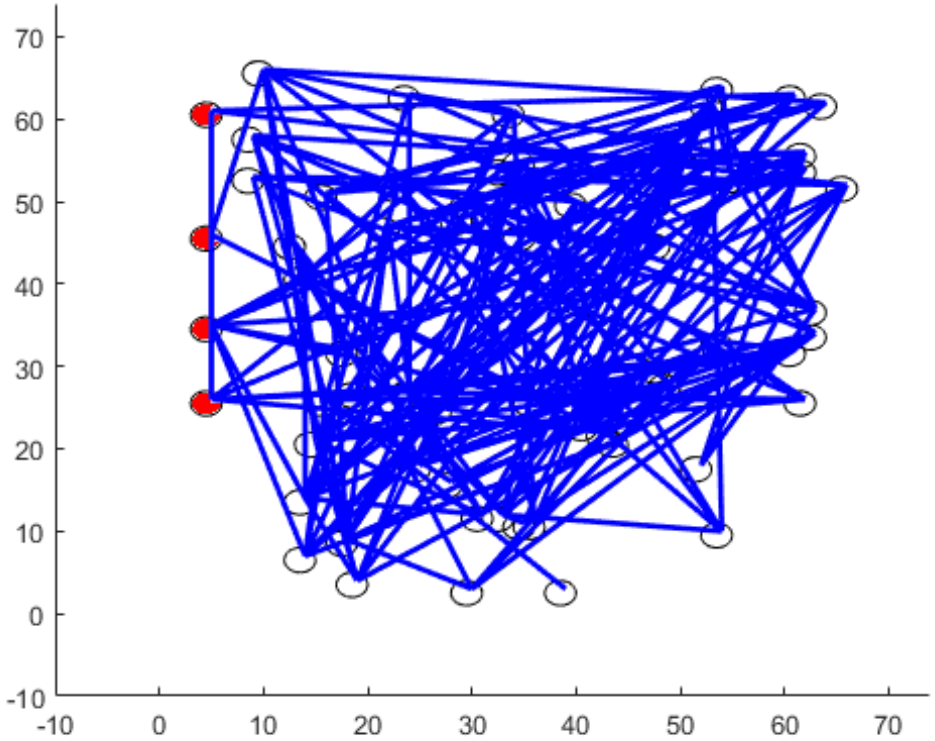
$$\widetilde{\mathbf{L}}_{RW} = \widetilde{\mathbf{D}}^{-1/2}\mathbf{T}_3\widetilde{\mathbf{D}}^{1/2}$$

- We denote graph corresponding to $\widetilde{\mathbf{L}}_{RW}$ the **reduced graph** $\widetilde{G}$

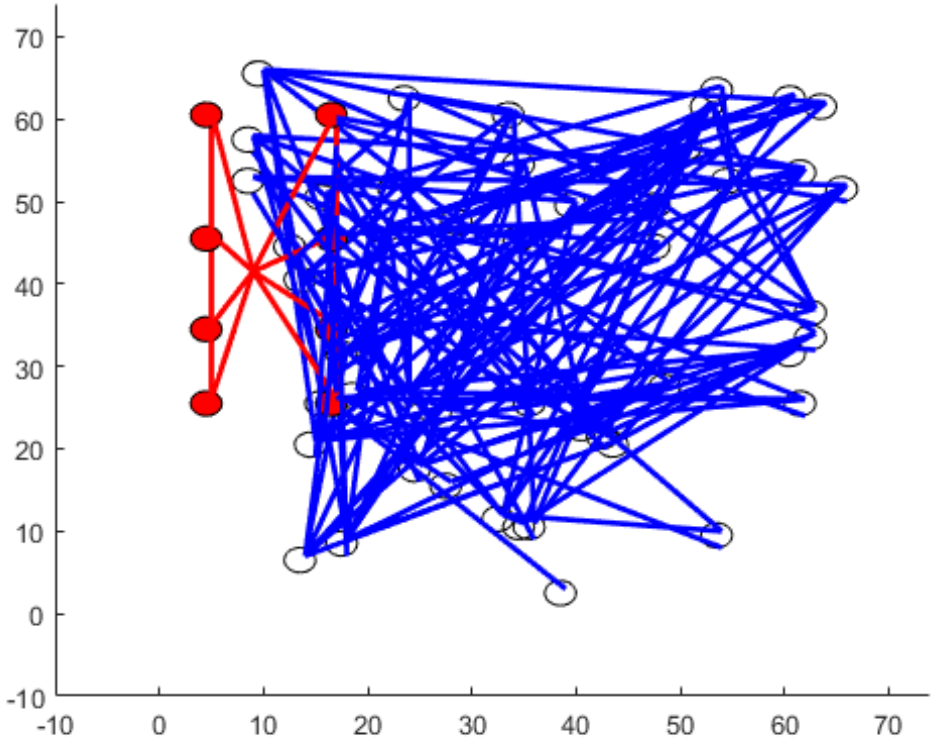
Third Lanczos process and the reduced graph $\widetilde{G}$



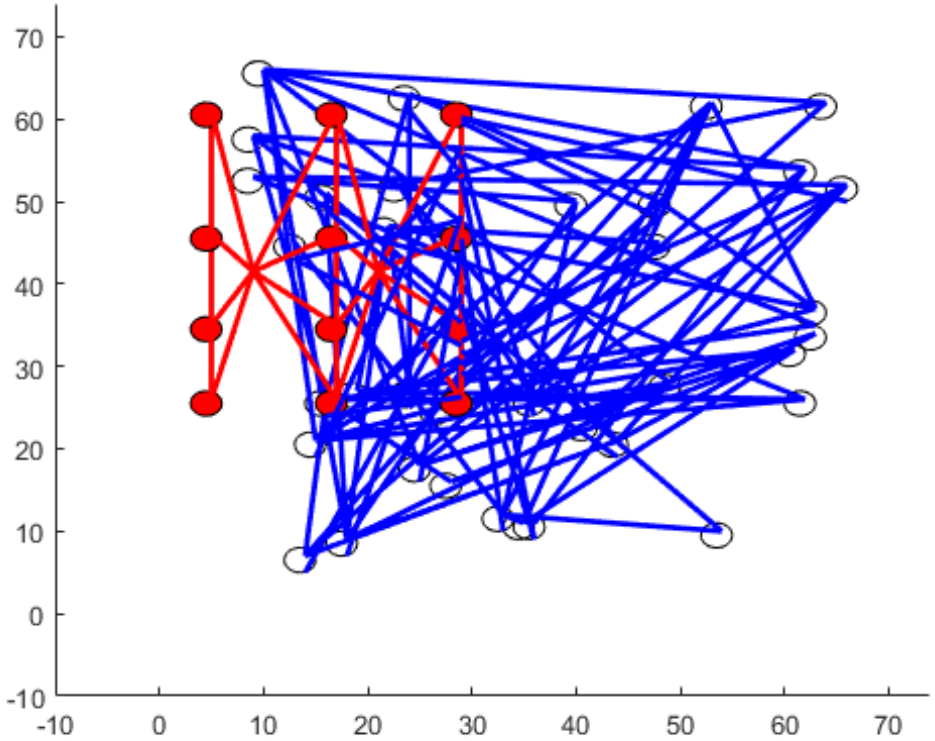
Third Lanczos process and the reduced graph $\widetilde{G}$



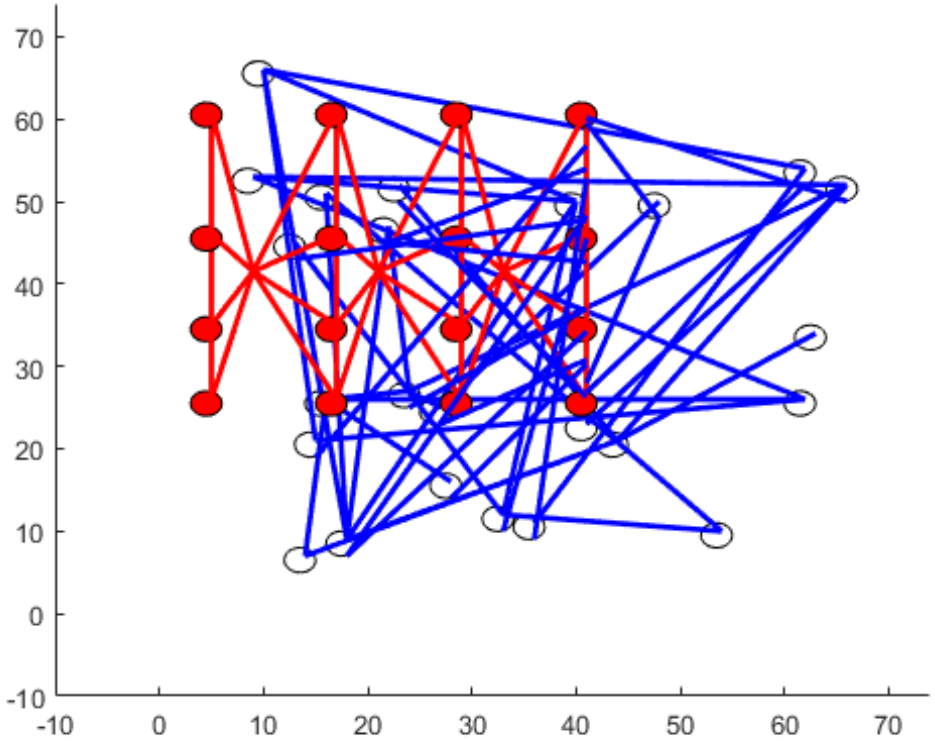
Third Lanczos process and the reduced graph $\widetilde{G}$



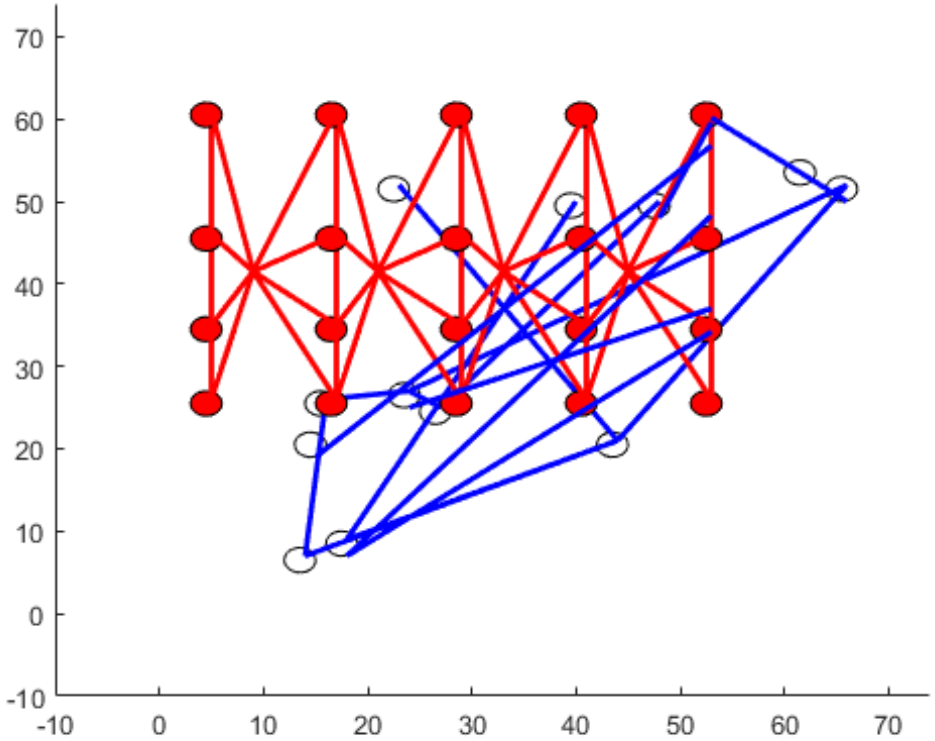
Third Lanczos process and the reduced graph $\widetilde{G}$



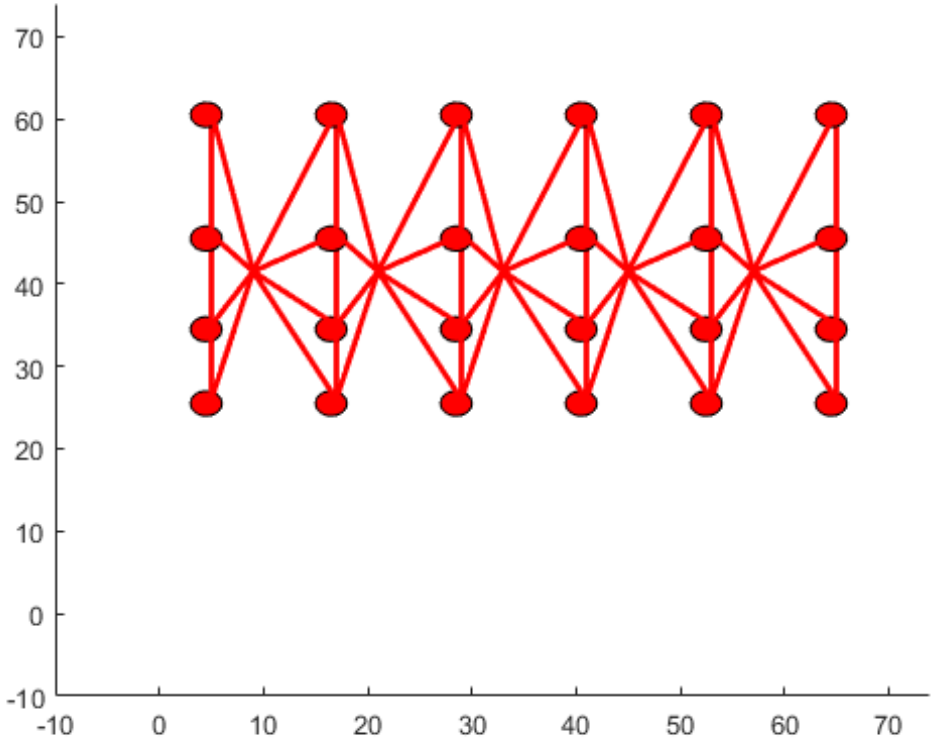
Third Lanczos process and the reduced graph $\widetilde{G}$



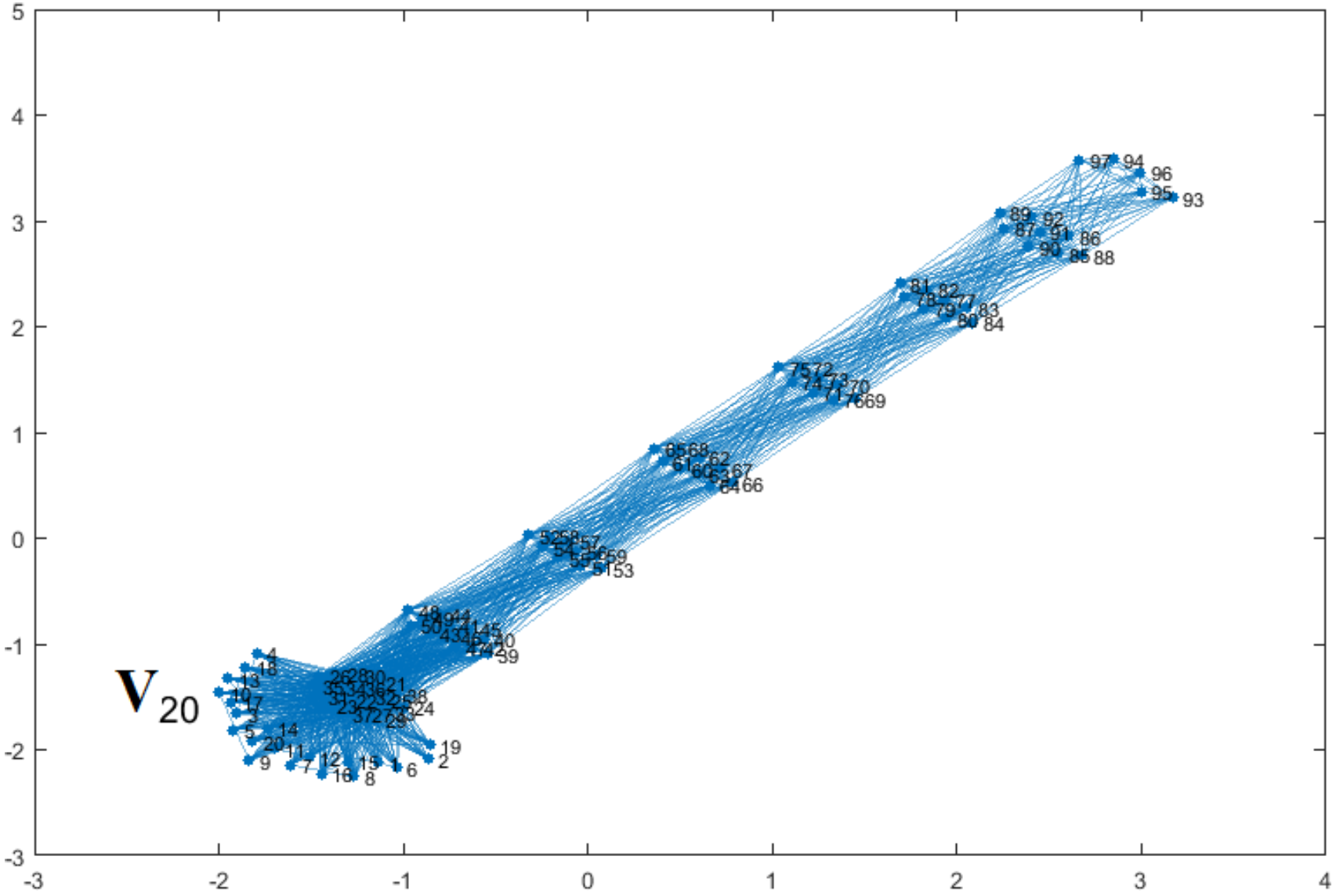
Third Lanczos process and the reduced graph $\widetilde{G}$



Third Lanczos process and the reduced graph $\widetilde{G}$



Deflated block-tridiagonal structure of $\widetilde{G}$



Target subset clustering with ROGL

Once the target subset V_m is chosen and the ROGL $\widetilde{\mathbf{L}}_{RW}$ is computed, several possibilities for clustering of V_m exist:

- Compute the **spectral data** for $\widetilde{\mathbf{L}}_{RW}$, $\widetilde{\mathbf{D}}$ and clusterize $\widetilde{G}$ using a relaxation of K-means, either
  1. **Lloyd's** algorithm (**ROGLC-L**), or
  2. Semidefinite programming (**ROGLC-S**)
- Clusterize $\widetilde{G}$ **directly** using an **SDP-relaxed min-Ncut**, bypassing spectral data computation

After the clustering

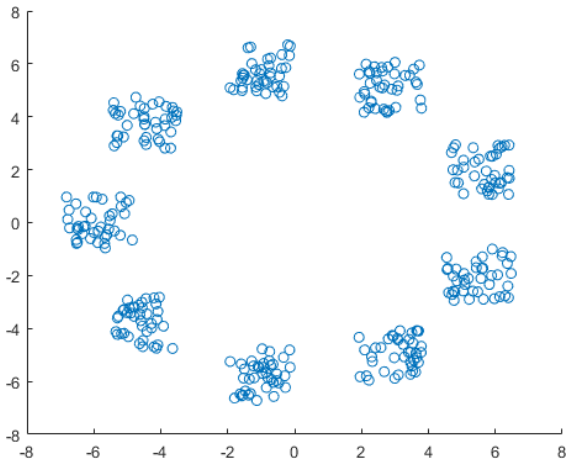
$$\widetilde{V} = \sqcup_{k=1}^{K} \widetilde{C}_k$$

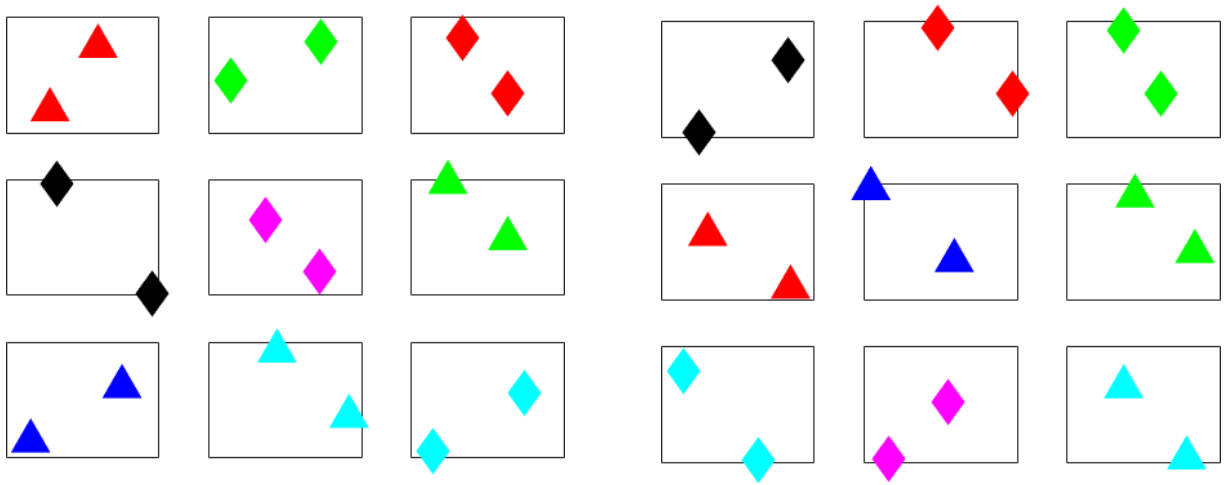
of $\widetilde{G}$ is found, the clusters of V_m are simply

$$V_m \cap \widetilde{C}_k$$

Numerical results: synthetic example

- **Synthetic dataset**: 9 "clouds" of 40 points in $\mathbb{R}^2$ each, $N = 360$ (leftmost plot)
- Clustering results for V_{18} of 2 randomly selected points from every cloud: conventional normalized SC (middle) and **ROGLC-L** (right)
- All 18 points are correctly identified as belonging to 9 separate clusters, each corresponding to a different cloud

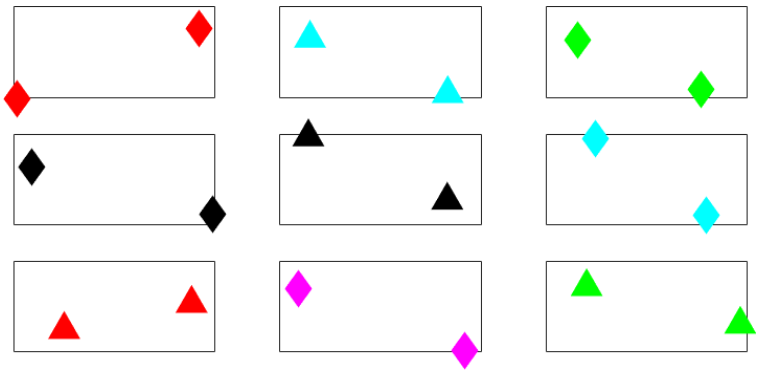




Numerical results: Astrophysics collaboration network

- Real dataset from **SNAP**: collaboration network from arXiv:Astrophysics of $N = 18872$ authors, our largest example
- No $\mathbb{R}^d$ embedding available
- **No ground-truth communities** available, used conventional SC as a reference clustering
- Two test cases:
  1. Randomly select 2 vertices from 9 ground-truth clusters $m = 18$ (left)
  2. Randomly select 2 vertices from 3 ground-truth clusters $m = 6$ (right)
- **ROGLC-L**: all vertices correctly attributed to reference clusters

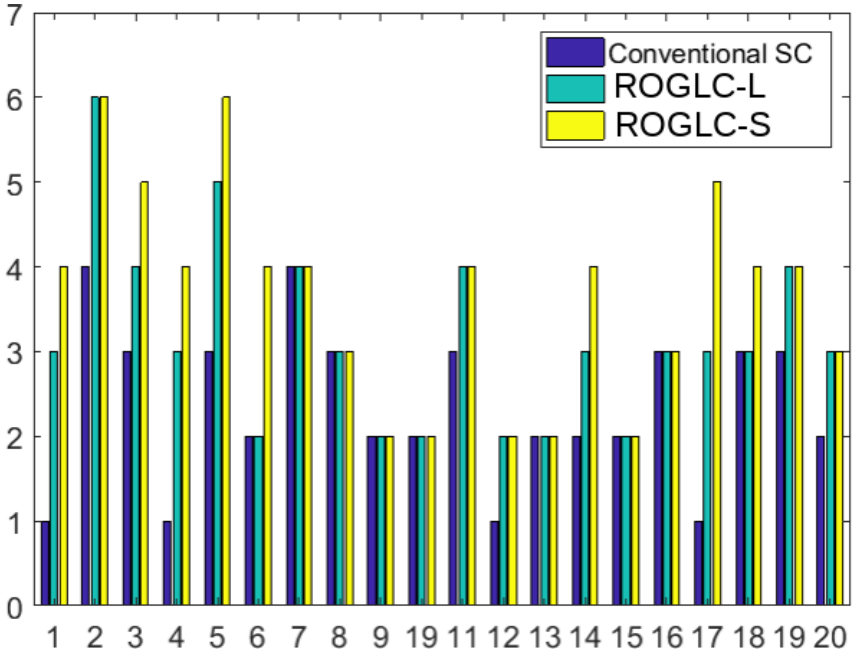
V_{18} requires $n = 146$



V_6 requires $n = 24$

Numerical results: EMail communication network

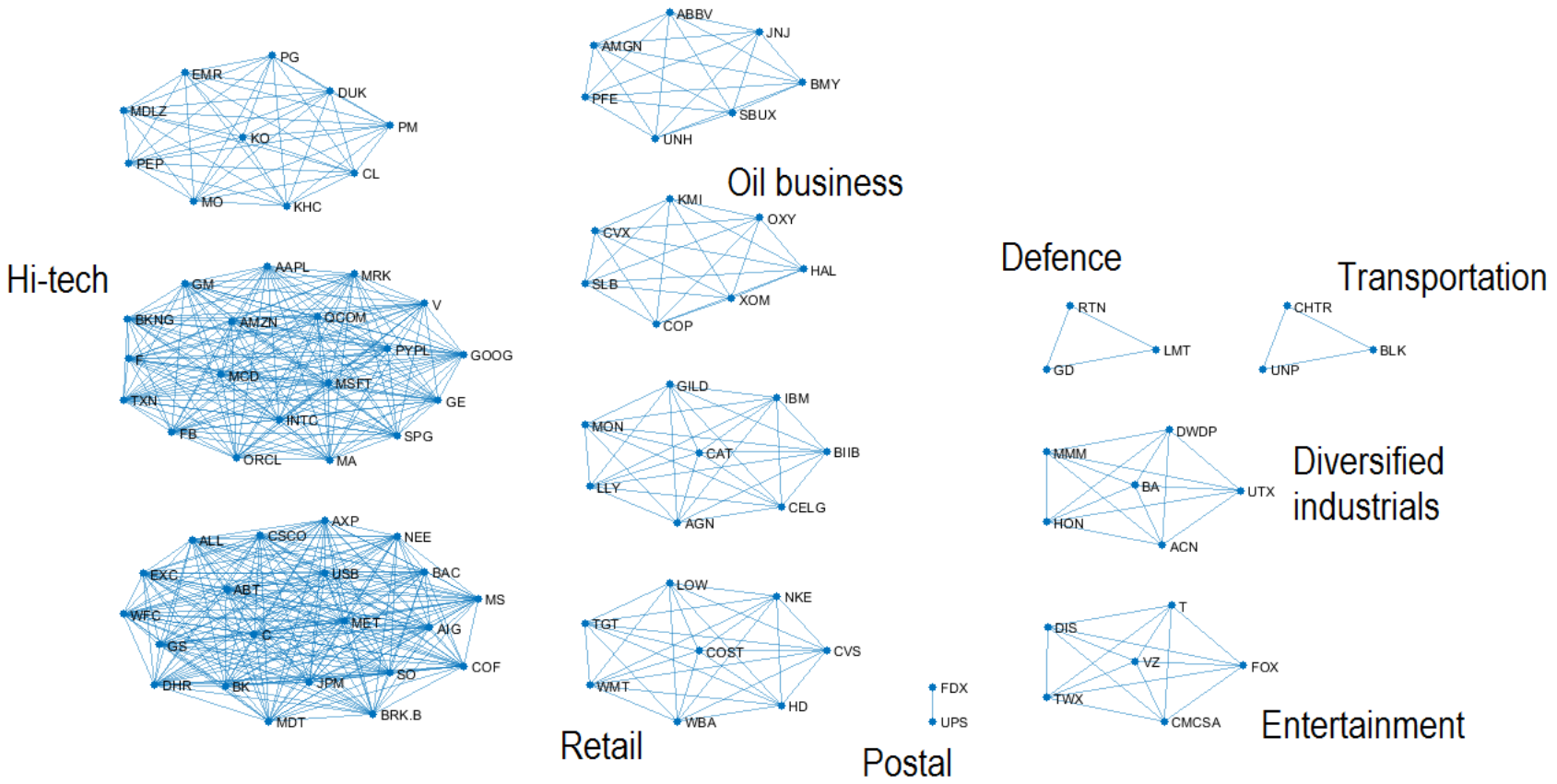
- Real dataset from **SNAP**: email communication between $N = 1,005$ correspondents, 42 **ground-truth communities** (correspondents' affiliations)
- **Comparison** between conventional SC, **ROGLC-L** and **ROGLC-S**
- Choose at random 2 vertices from 10 ground-truth communities to form V_m, repeat multiple times to check robustness w.r.t. random realization



- Numbers of correctly identified vertices from 10 ground-truth communities for 20 realizations of V_m
- **ROGL+SDP** achieves the best performance, only possible with ROGL, **too expensive otherwise**

Another possible application: stock market data

- Identify highly-correlated stocks to diversify investment portfolios
- No $\mathbb{R}^d$ embedding available (cross-correlations only)
- Significant noise in the data



Conclusions and future work

- We introduced the **reduced-order graph-Laplacian (ROGL)** for clustering graph vertex subsets
- It is a **building block** for a **divide-and-conquer** clustering algorithm currently under development
- **Advantages** compared to full graph SC:
  1. Well-suited for parallel computations
  2. Small sub-problem size enables the use of more accurate clustering algorithms (SDP-based relaxations of K-means or min-Ncut) which leads to **qualitatively better** solutions
- **Possible improvements**: finite-precision Lanczos for the first stage (currently uses reorthogonalization to achieve stability)

[1] *Clustering of graph vertex subset via Krylov subspace model reduction.* V. Druskin, A.V. Mamonov, M. Zaslavsky, 2018, submitted to JMLR, arXiv:1809.03048 [cs.LG]