Cut Improvement and Clustering using Compressive Sensing

Math. of Data Science Virtual Lecture Series
Tufts University

Ming-Jun Lai [1] Daniel McKenzie [2]

[1]University of Georgia

[2]University of California, Los Angeles

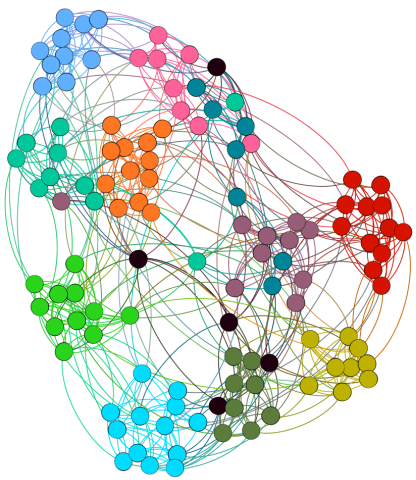
April 16, 2020

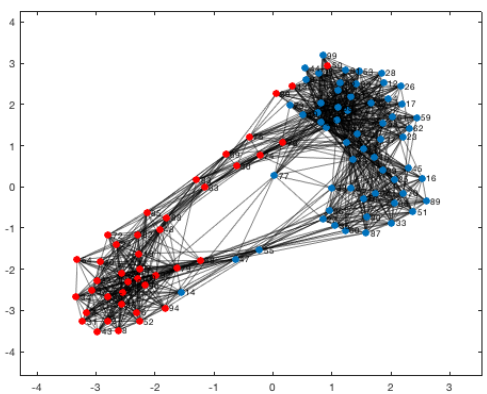
Clusters in Graphs

- All graphs $G = (V, E)$ are finite and $V := [n] = \{1, \ldots, n\}$.
- A denotes (possibly weighted) **adjacency matrix** of G.
- For any data set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^D$ can make graph:

$$A_{ij} = \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\sigma^2\right)$$

- A **cluster** $C \subset V$ has "many" internal edges and "few" external edges.





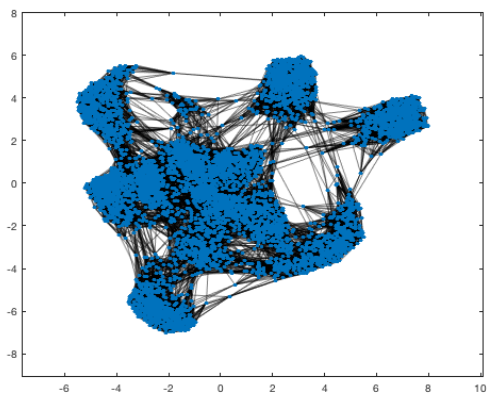


Figure: From left to right: College Football (2000 season) Girvan and Newman [2002], Senate Co-voting data for 97th congress Lewis et al. [2020], OptDigits made into a graph.

Quantifying Good Clusters

A **cluster** $C \subset V$ *has "many" internal edges and "few" external edges.*

- **Volume:** $\text{vol}(C) := \sum_{i \in C} d_i$ where $d_i = \text{degree of } i = \sum_j A_{ij}$
- **Cut:** $\text{Cut}(C) = \sum_{i \in C, j \in \bar{C}} A_{ij}$
- **Normalized Cut:** $\text{NCut}(C) = \dfrac{\text{Cut}(C)}{\text{vol}(C)\text{vol}(\bar{C})}$
- **Conductance:** $\text{Cond}(C) = \dfrac{\text{Cut}(C)}{\min(\text{vol}(C), \text{vol}(\bar{C}))}$

- Finding $C^{\#} = \min_{C \subset V} \text{Cut}(C)$ possible but non-informative.
- Finding $C^{\#} = \min_{C \subset V} \text{NCut}(C)$ or $\min_{C \subset V} \text{Cond}(C)$ informative but NP-Hard

Finding Good Clusters—Local and Global

- Global clustering (*e.g.* Spectral Clustering [1]).
  - Operates on full adj. matrix, run time $\sim O(n^2)$.
  - Typically unsupervised.
- Strongly local clustering (*e.g.* `Nibble`, `CRD`, `LocalImprove` [2]).
  - Semi-supervised: Given $\Gamma \subset V$ returns $C^{\#}$ containing Γ.
  - Only operates on neighbourhood of $C^{\#}$, run time $\sim O(\text{vol}(C^{\#}))$.
- Weakly local clustering (*e.g.* `PPR`, `HK-flow`, `CP+RWT` [3]).
  - Semi-supervised: Given $\Gamma \subset V$ returns $C^{\#}$ containing Γ.
  - Operate on whole graph, run time $\sim \tilde{O}(n)$.
- Cut improvement (*e.g.* `FlowImprove`, `LocalFlow`, `ClusterPursuit`) [4]
  - Given $\Omega \approx C$ returns $C^{\#}$ better approx to C.
  - Can be local, run time $= O(Vol(\Omega)^{\alpha})$, or global, run time $= \tilde{O}(n)$.

---

[1] Shi and Malik [2000], Ng et al. [2002]

[2] Spielman and Teng [2004, 2013], Wang et al. [2017],Veldt et al. [2016]

[3] Andersen et al. [2007],Kloster and Gleich [2014],Lai and Mckenzie [2019]

[4] Andersen and Lang [2008], Orecchia and Allen-Zhu [2014], Lai and Mckenzie [2019]

Overview of this talk

- We rephrase cut improvement as a compressive sensing problem.
- We introduce a new algorithm for cut improvement: `ClusterPursuit`.
- This algorithm enjoys theoretical guarantees on accuracy and run time.
- Numerical results are good.
- We use `ClusterPursuit` to design local & global clustering algorithms.
- Code available at: `http://danielmckenzie.github.io/`.

In cluster and between cluster graphs

- **Graph Laplacian:** $L = I - D^{-1}A$.
- Suppose G has clusters $C_1, \ldots, C_k$.
- **Key Idea:** Split $G = G^{\text{in}} \coprod G^{\text{out}}$.
- Here $E^{\text{in}} = \{\{i,j\} : \ i,j \in C_a \text{ for } a = 1,\ldots,k\}$ and $G^{\text{in}} := (V, E^{\text{in}})$.
- $E^{\text{out}} = E \setminus E^{\text{in}}$ and $G^{\text{out}} = (V, E^{\text{out}})$.
- Let A^{in} (resp. L^{in}) denote adj. matrix (resp. Laplacian) of G^{in}.
- Then $A = A^{\text{in}} + A^{\text{out}}$ and $L = L^{\text{in}} + M$.
- **Theorem**[5] $L^{\text{in}}\mathbf{1}_{C_a} = 0$ for $a = 1,\ldots,k$.
- **Observation:** $\|\mathbf{1}_{C_a}\|_0 := |\{i : \ (\mathbf{1}_{C_a})_i \neq 0\}| = |C_a| := n_a$

[5] See, for example, Von Luxburg [2007]

(Totally Perturbed) Compressive Sensing

- Compressive Sensing gives theory and algorithms for solving problem:

$$\boldsymbol{x}^{\#} = \arg\min\{\|\Phi\boldsymbol{x} - \boldsymbol{y}\|_2 : \ \|\boldsymbol{x}\|_0 \leq s\} \tag{1}$$

- Restricted Isometry Constant, $\delta_s(\Phi)$: smallest $\delta \in (0,1)$ s.t.

$$(1-\delta)\|\boldsymbol{x}\|_2^2 \leq \|\Phi\boldsymbol{x}\|_2^2 \leq (1+\delta)\|\boldsymbol{x}\|_2^2 \text{ for all } \boldsymbol{x} \in \mathbb{R}^n \text{ with } \|\boldsymbol{x}\|_0 \leq s$$

- Fast, greedy algorithms for (1): `OMP`, `CoSaMP`, `SubspacePursuit` [6].
- Robust to (additive and multiplicative) noise [7]:

$$\text{If } \boldsymbol{x}^* = \arg\min\left\{\|\hat{\Phi}\boldsymbol{x} - \hat{\boldsymbol{y}}\|_2 : \ \|\boldsymbol{x}\|_0 \leq s\right\}$$
$$\text{and } \boldsymbol{x}^{\#} = \arg\min\left\{\|\Phi\boldsymbol{x} - \boldsymbol{y}\|_2 : \ \|\boldsymbol{x}\|_0 \leq s\right\}$$
$$\text{with } \boldsymbol{y} = \hat{\boldsymbol{y}} + \boldsymbol{e} \text{ and } \Phi = \hat{\Phi} + M$$
$$\text{then } \frac{\|\boldsymbol{x}^* - \boldsymbol{x}^{\#}\|_2}{\|\boldsymbol{x}^*\|_2} \leq C(\delta_s(\Phi), \epsilon_\Phi^s, \epsilon_{\boldsymbol{y}}) \text{ where } \epsilon_\Phi^s \approx \frac{\|M\|_2}{\|\Phi\|_2} \text{ and } \epsilon_{\boldsymbol{y}} = \frac{\|\boldsymbol{e}\|_2}{\|\boldsymbol{y}\|_2}$$

[6] Tropp [2004], Needell and Tropp [2009], Dai and Milenkovic [2009]

[7] Herman and Strohmer [2010], Li [2016]

Cut improvement with compressive sensing

- **Recall:**
  - $L = L^{\text{in}} + M$. *(Think* $\hat{\Phi} = L^{in}$ *and* $\Phi = L$*)*
  - $L^{in}\mathbf{1}_{C_a} = 0$.

Cut improvement with compressive sensing

- **Recall:**
  - $L = L^{\text{in}} + M$. *(Think* $\hat{\Phi} = L^{in}$ *and* $\Phi = L$*)*
  - $L^{in}\,\mathbf{1}_{C_a} = 0$.
- Assume $\Omega \approx C_a$ given. Let $U = C_a \setminus \Omega$ and $W = \Omega \setminus C_a$. Then:

$$
\begin{aligned}
\mathbf{1}_\Omega = \mathbf{1}_{C_a} + \mathbf{1}_W - \mathbf{1}_U \implies & L^{\text{in}}\,\mathbf{1}_\Omega = L^{\text{in}}\,\mathbf{1}_{C_a} + L^{\text{in}}\,(\mathbf{1}_W - \mathbf{1}_U) \\
\implies & L^{\text{in}}\,\mathbf{1}_\Omega = 0 + L^{\text{in}}\,(\mathbf{1}_W - \mathbf{1}_U) \\
\implies & \boldsymbol{y}^{\text{in}} = L^{\text{in}}\,(\mathbf{1}_W - \mathbf{1}_U) \quad (\text{if } \boldsymbol{y}^{\text{in}} := L^{\text{in}}\,\mathbf{1}_\Omega) \\
\implies & \boldsymbol{y} \approx L\,(\mathbf{1}_W - \mathbf{1}_U) \quad (\text{if } \boldsymbol{y} := L\,\mathbf{1}_\Omega)
\end{aligned}
$$

Cut improvement with compressive sensing

- **Recall:**
  - $L = L^{\text{in}} + M$. *(Think* $\hat{\Phi} = L^{in}$ *and* $\Phi = L$*)*
  - $L^{in}\mathbf{1}_{C_a} = 0$.
- Assume $\Omega \approx C_a$ given. Let $U = C_a \setminus \Omega$ and $W = \Omega \setminus C_a$. Then:

$$\begin{aligned}
\mathbf{1}_\Omega = \mathbf{1}_{C_a} + \mathbf{1}_W - \mathbf{1}_U \implies & L^{\text{in}}\mathbf{1}_\Omega = L^{\text{in}}\mathbf{1}_{C_a} + L^{\text{in}}(\mathbf{1}_W - \mathbf{1}_U) \\
\implies & L^{\text{in}}\mathbf{1}_\Omega = 0 + L^{\text{in}}(\mathbf{1}_W - \mathbf{1}_U) \\
\implies & \boldsymbol{y}^{\text{in}} = L^{\text{in}}(\mathbf{1}_W - \mathbf{1}_U) \quad (\text{if } \boldsymbol{y}^{\text{in}} := L^{\text{in}}\mathbf{1}_\Omega) \\
\implies & \boldsymbol{y} \approx L(\mathbf{1}_W - \mathbf{1}_U) \quad (\text{if } \boldsymbol{y} := L\mathbf{1}_\Omega)
\end{aligned}$$

- Define $\boldsymbol{x}^* = \arg\min\left\{\|L^{\text{in}}\boldsymbol{x} - \boldsymbol{y}^{\text{in}}\|_2 : \ \|\boldsymbol{x}\|_0 \leq |W| + |U|\right\}$.
- Will show that $\boldsymbol{x}^* = \mathbf{1}_W - \mathbf{1}_U$.
- Define $\boldsymbol{x}^\# = \arg\min\{\|L\boldsymbol{x} - \boldsymbol{y}\|_2 : \ \|\boldsymbol{x}\|_0 \leq |W| + |U|\}$.
- Will show that $\boldsymbol{x}^\# \approx \boldsymbol{x}^*$.

Cut improvement with compressive sensing

Algorithm 1: `ClusterPursuit`

Input: Adj. matrix A, initial cut Ω, estimate $s \approx |C_a \triangle \Omega|$ and $R \in [0, 1)$.

Output: Subset $C_a^{\#}$ that approximates C_a

1 $L \leftarrow I - D^{-1}A$ and $\boldsymbol{y} \leftarrow L\,\mathbf{1}_\Omega$.

2 $\boldsymbol{x}^{\#} \leftarrow \arg\min \{\|L\boldsymbol{x} - \boldsymbol{y}\|_2 : \ \|\boldsymbol{x}\|_0\}$ using $m = O(\log(n))$ iterations of `SubspacePursuit`.

3 $U^{\#} \leftarrow \{i : \ x_i^{\#} < -R\}$ and $W^{\#} \leftarrow \{i : \ x_i^{\#} > R\}$.

4 $C_a^{\#} \leftarrow \left(\Omega \setminus W^{\#}\right) \cup U^{\#}$.

- $|C_a \triangle \Omega| = |C_a \setminus \Omega| + |\Omega \setminus C_a| = |W| + |U|$.
- Robust w.r.t parameters.
- Run time $= O(d_{\max} n \log n)$.

Cut improvement with compressive sensing.

Recall:

- $\Omega \approx C_a$.
- $\boldsymbol{y}^{\text{in}} = L^{\text{in}} \mathbf{1}_\Omega$

Theorem (Lai & M.)

$\mathbf{1}_W - \mathbf{1}_U$ is the unique solution to:

$$\arg\min \left\{ \|L^{in}\boldsymbol{x} - \boldsymbol{y}^{in}\|_2 : \ \|\boldsymbol{x}\|_0 \leq s \right\}$$

for any G with clusters $C_1, \ldots, C_k$, as long as $|C_a \triangle \Omega| \leq s < n_1/2$.

- Not a practical result! Don't know L^{in}.
- Getting from L^{in} to L requires a data model.

The Data Model

- Let $\{\mathcal{G}_n\}_{n=1}^{\infty}$ where $\mathcal{G}_n$ is prob. dist. on graphs on n vertices.
- Suppose exists $\epsilon_i = o_n(1)$ for $i = 1, 2, 3$ such that for $G \sim \mathcal{G}_n$:

(A1) $V = C_1 \cup \ldots \cup C_k$ where C_a are disjoint clusters and $k = O_n(1)$.

(A2) For all $a \in [k]$ $\lambda_2(L_{G_{C_a}}) \geq 1 - \epsilon_1$ and $\lambda_{n_a}(L_{G_{C_a}}) \leq 1 + \epsilon_1$ almost surely.

(A3) letting $r_i := d_i^{\text{out}}/d_i^{\text{in}}$, $r_i \leq \epsilon_2$ for all $i \in [n]$ almost surely.

(A4) If $d_{\text{av}}^{\text{in}} := \mathbb{E}[d_i^{\text{in}}]$ then $d_{\text{max}}^{\text{in}} \leq (1 + \epsilon_3) d_{\text{av}}^{\text{in}}$ and $d_{\text{min}}^{\text{in}} \geq (1 - \epsilon_3) d_{\text{av}}^{\text{in}}$ a.s.

From L^{in} to L

Recall:

- $M := L - L^{\text{in}}$ and $\boldsymbol{e} := \boldsymbol{y} - \boldsymbol{y}^{\text{in}}$.
- $\epsilon_{\boldsymbol{y}} = \dfrac{\|\boldsymbol{e}\|_2}{\|\boldsymbol{y}^{\text{in}}\|_2}$ and $\epsilon_L^s = \dfrac{\|M\|_2^{(s)}}{\|L^{\text{in}}\|_2^{(s)}}$
- Key parameters for perturbed compressive sensing are $\epsilon_{\boldsymbol{y}}, \epsilon_L^s$ and $\delta_s(L)$

Theorem (Lai & M.)

Suppose that $\mathcal{G}_n$ satisfies (A1)–(A4) and that $|C_1 \triangle \Omega| \leq 0.13 n_1$. Then for any $\gamma \in (0,1)$ the following hold almost surely:

1. $\epsilon_{\boldsymbol{y}} = o(1)$ *and* $\epsilon_L^{\gamma n_1} = o(1)$.
2. $\delta_{\gamma n_1}(L) \leq \gamma + o(1)$.

(Think $s = \gamma n_1$.)

Recovery Guarantee for `ClusterPursuit`

Theorem (Lai & M.)

Suppose the following:

- $\mathcal{G}_n$ *satisfies (A1)–(A4) and* $G \sim \mathcal{G}_n$.
- $|C_1 \triangle \Omega| = \epsilon n_1$ *with* $\epsilon \leq 0.13$.
- $s \leq 0.13 n_1$ *and* $R = 0.5$.

If $C_1^{\#} = \texttt{ClusterPursuit}(A, \Omega, s, R)$ *then* $\dfrac{\left|C_1 \triangle C_1^{\#}\right|}{|C_1|} = o(1)$ *a.s.*

Recovery Guarantee for `ClusterPursuit`

Theorem (Lai & M.)

Suppose the following:

- $\mathcal{G}_n$ *satisfies (A1)–(A4) and* $G \sim \mathcal{G}_n$.
- $|C_1 \triangle \Omega| = \epsilon n_1$ *with* $\epsilon \leq 0.13$.
- $s \leq 0.13 n_1$ *and* $R = 0.5$.

If $C_1^{\#} = \texttt{ClusterPursuit}(A, \Omega, s, R)$ *then* $\dfrac{\left|C_1 \triangle C_1^{\#}\right|}{|C_1|} = o(1)$ *a.s.*

Proof.

- Know $\boldsymbol{x}^* = \arg\min \left\{ \|L^{\text{in}}\boldsymbol{x} - \boldsymbol{y}^{\text{in}}\|_2 : \ \|\boldsymbol{x}\|_0 \leq s \right\} = \mathbf{1}_W - \mathbf{1}_U$.
- Data Model $\Rightarrow$ $\epsilon_{\boldsymbol{y}}, \epsilon_L^s$ and $\delta_s(L)$ are small.
- If $\boldsymbol{x}^{\#} = \arg\min \{ \|L\boldsymbol{x} - \boldsymbol{y}\|_2 : \ \|\boldsymbol{x}\|_0 \leq s \}$ then $\|\boldsymbol{x}^{\#} - \boldsymbol{x}^*\|_2$ small.
- $\{i : x_i^{\#} > 0\} \approx W$ and $\{i : \ x_i^{\#} < 0\} \approx U$

□

The stochastic block model

- Specify cluster sizes $n_1 \leq n_2 \leq \ldots \leq n_k$.
- Specify **connection probability matrix** $P \in \mathbb{R}^{k \times k}$.
- Construct partition $V = C_1 \cup \ldots \cup C_k$ with $|C_a| = n_a$.
- Generate $G \sim \mathrm{SBM}(\mathbf{n}, P)$ with $\mathbb{P}[A_{ij} = 1 | i \in C_a, j \in C_b] = P_{ab}$.

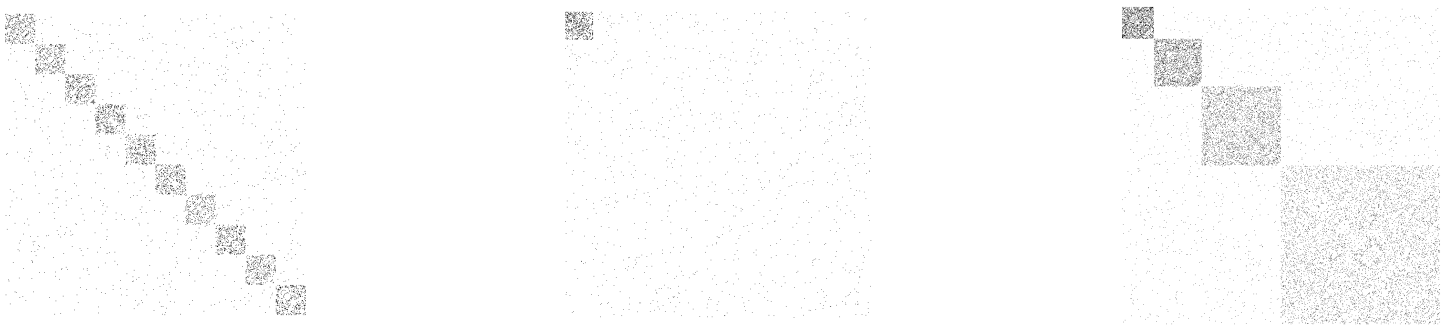


Figure: Examples of adjacency matrices for different $\mathrm{SBM}(\mathbf{n}, P)$.

Spectral Properties for L for SBM

Theorem (Lai & M.)

- *Let* $\mathcal{G}_n = SBM(\mathbf{n}, P)$ *with* $|\mathbf{n}| = \sum_{a=1}^{k} n_a = n$.
- **Assume:**
  - $n_1 \to \infty$.
  - $P_{aa} = \omega \log(n)/n_a$ *for any* ω *with* $\omega \to \infty$.
  - $P_{ab} = (\beta + o(1)) \log(n)/n$ *for all* $a \neq b$
- **Then:** $\mathcal{G}_n$ *satisfies assumptions (A1)–(A4).*

Spectral Properties for L for SBM

Theorem (Lai & M.)

- *Let* $\mathcal{G}_n = SBM(\mathbf{n}, P)$ *with* $|\mathbf{n}| = \sum_{a=1}^{k} n_a = n$.
- **Assume:**
  - $n_1 \to \infty$.
  - $P_{aa} = \omega \log(n)/n_a$ *for any* ω *with* $\omega \to \infty$.
  - $P_{ab} = (\beta + o(1)) \log(n)/n$ *for all* $a \neq b$
- **Then:** $\mathcal{G}_n$ *satisfies assumptions (A1)–(A4).*

Proof.

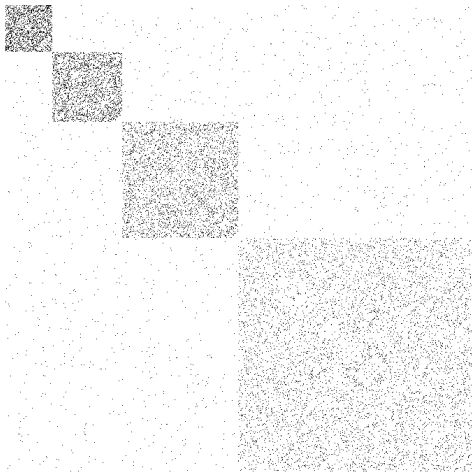
- If $G \sim \text{SBM}(\boldsymbol{n}, P)$ then each $G_{C_a} \sim \text{ER}(n_a, P_{aa})$.
- Concentration of measure for $d_i(G_{C_a})$. [a]
- Concentration of measure for $\lambda_i(G_{C_a})$. [b]

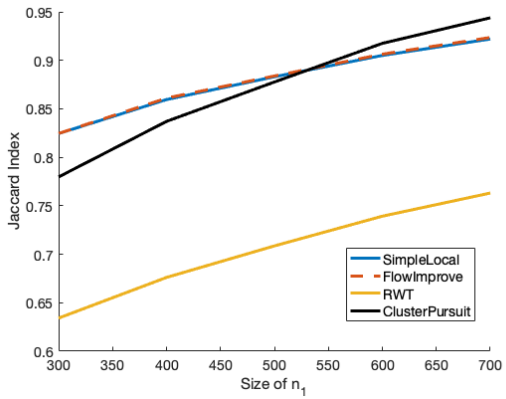
□

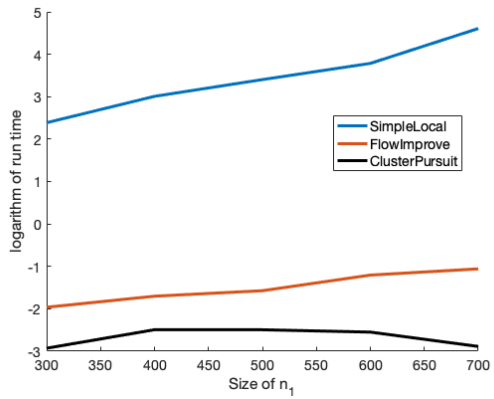
[a] Frieze and Karoński [2016]

[b] Chung and Radcliffe [2011]

Experimental Results: Stochastic block model





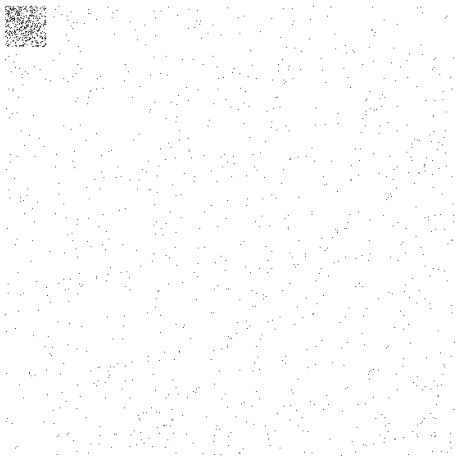


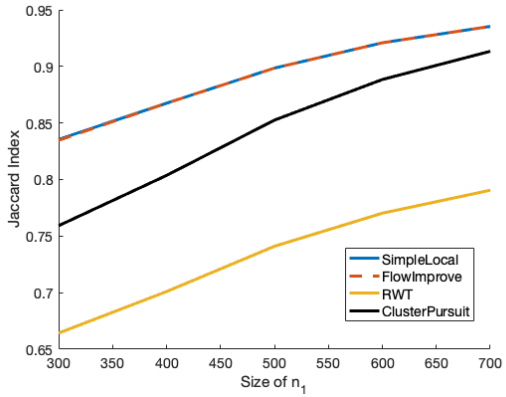
- $\mathrm{Jac}(C_1, C_1^{\#}) = |C_1 \cap C_1^{\#}|/|C_1 \cup C_1^{\#}|$. High is good.
- `FlowImprove`[8] and `SimpleLocal`[9] take essentially the same approach.
- `SimpleLocal` optimized for small clusters (*i.e.* $|C_a| = O_n(1)$).
- Yellow line is baseline (represents $\mathrm{Jac}(\Omega, C_1)$).

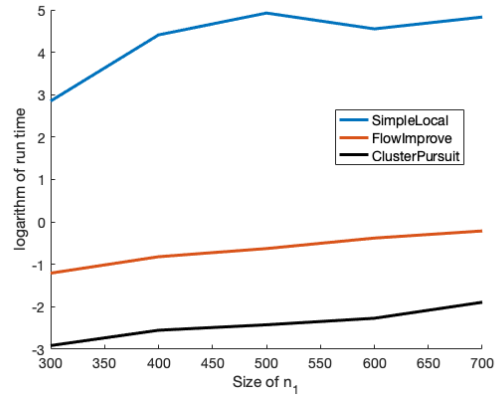
[8] Andersen and Lang [2008]

[9] Veldt et al. [2016]

Experimental Results: Stochastic block model







Extension to local clustering

- `ClusterPursuit` works well given $\Omega \approx C_a$.
- How to find Ω?

[10] Spielman and Teng [2004],Andersen et al. [2007], Kloster and Gleich [2014],Wang et al. [2017]

[11] Li et al. [2015], He et al. [2015], Veldt et al. [2019]

Extension to local clustering

- `ClusterPursuit` works well given $\Omega \approx C_a$.
- How to find Ω?
- Diffusion-based local clustering.
  - Given small set of seed vertices Γ.
  - Let $\boldsymbol{v}^{(0)} = |\Gamma|^{-1} \mathbf{1}_\Gamma$.
  - Run a diffusive process for t steps: $\boldsymbol{v}^{(t)} = P^t \boldsymbol{v}^{(0)}$.
  - $\Omega \leftarrow \{i : \ v_i^{(t)}$ "is large"$\}$.

[10] Spielman and Teng [2004],Andersen et al. [2007], Kloster and Gleich [2014],Wang et al. [2017]

[11] Li et al. [2015], He et al. [2015], Veldt et al. [2019]

Extension to local clustering

- `ClusterPursuit` works well given $\Omega \approx C_a$.
- How to find Ω?
- Diffusion-based local clustering.
  - Given small set of seed vertices Γ.
  - Let $\boldsymbol{v}^{(0)} = |\Gamma|^{-1} \mathbf{1}_\Gamma$.
  - Run a diffusive process for t steps: $\boldsymbol{v}^{(t)} = P^t \boldsymbol{v}^{(0)}$.
  - $\Omega \leftarrow \{i : \ v_i^{(t)}$ "is large"$\}$.
- Diffusive process? random walk, Pagerank, heat flow, CRD [10]

[10] Spielman and Teng [2004],Andersen et al. [2007], Kloster and Gleich [2014],Wang et al. [2017]

[11] Li et al. [2015], He et al. [2015], Veldt et al. [2019]

Extension to local clustering

- `ClusterPursuit` works well given $\Omega \approx C_a$.
- How to find Ω?
- Diffusion-based local clustering.
  - Given small set of seed vertices Γ.
  - Let $\boldsymbol{v}^{(0)} = |\Gamma|^{-1} \mathbf{1}_\Gamma$.
  - Run a diffusive process for t steps: $\boldsymbol{v}^{(t)} = P^t \boldsymbol{v}^{(0)}$.
  - $\Omega \leftarrow \{i : \ v_i^{(t)}$ "is large"$\}$.
- Diffusive process? random walk, Pagerank, heat flow, CRD [10]
- Two-step local clustering[11]: find $\Omega \approx C_a$ then refine to get $C_a^{\#}$.

[10] Spielman and Teng [2004],Andersen et al. [2007], Kloster and Gleich [2014],Wang et al. [2017]

[11] Li et al. [2015], He et al. [2015], Veldt et al. [2019]

Random Walk Thresholding

Algorithm 2: `RWThresh`

Input: Adj. matrix A, thresh. param. $\epsilon \in (0,1)$, seeds $\Gamma \subset C_a$,$\hat{n}_a \approx n_a$ and t.

Output: $\Omega \approx C_a$

1 $P \leftarrow AD^{-1}$ and $\boldsymbol{v}^{(0)} \leftarrow D\,\mathbf{1}_\Gamma$.

2 $\boldsymbol{v}^{(t)} \leftarrow P^t \boldsymbol{v}^{(0)}$.

3 $\Omega \leftarrow \{i : \ v_i^{(t)}$ amongst $(1+\epsilon)\hat{n}_1$ entries$\}$

4 $\Omega \leftarrow \Omega \cup \Gamma$.

Random Walk Thresholding

Theorem (Lai & M.)

Suppose the following:

- $\mathcal{G}_n$ *satisfies Assumptions (A1)–(A4) and* $G \sim \mathcal{G}_n$.
- $t = O(1)$, $\hat{n}_1 = n_1$ *and* $\epsilon \in (0, 1)$.
- $\Gamma \subset C_1$ *with* $|\Gamma| = g\epsilon_3^{2t-1} n_1$ *for any* $g \in (0, 1)$ *and* ϵ_3 *as in (A4)).*

If $\Omega = \texttt{RWThresh}(A, \epsilon, \Gamma, \hat{n}_1, t)$ *then* $|\Omega \triangle C_1| \leq (\epsilon + o(1))n_1$ *almost surely.*

- For SBM $\epsilon_3 = 1/\log(n)$ so $|\Gamma| = n_1/\text{polylog}(n_1)$.
- In practice, take $|\Gamma| = 0.01 n_1$ or similar.
- Other diffusive algorithms[12] take $|\Gamma| = O(1)$, but return $|\Omega| = O(1)$.
- Run time $= O(n \log(n))$.

[12] Spielman and Teng [2004], Andersen et al. [2007], Kloster and Gleich [2014], Wang et al. [2017]

Cluster pursuit for local clustering

Algorithm 3: `CP+RWT`

Input: Adj. matrix A, seed vertices $\Gamma \subset C_1$, parameters $\epsilon, R, \hat{n}_1, t$

Output: $C_1^{\#} \approx C_1$

1 $\Omega \leftarrow \texttt{RWThresh}(A, \epsilon, \Gamma, \hat{n}_1, t)$

2 $C_1^{\#} \leftarrow \texttt{ClusterPursuit}(A, s = 2\epsilon\hat{n}_1, R)$

Theorem (Lai & M.)

Suppose the following:

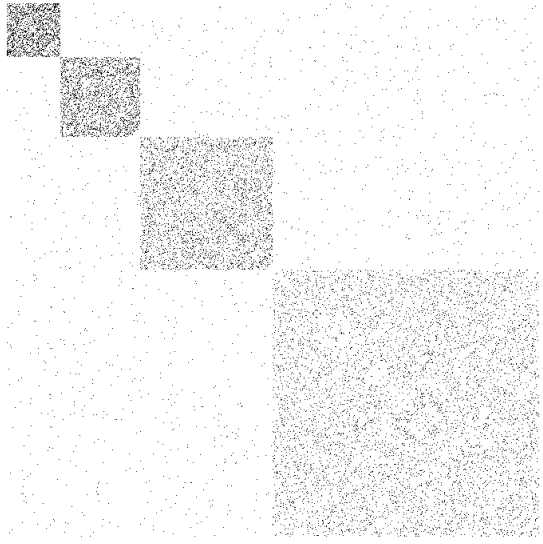
- $\mathcal{G}_n$ *satisfies Assumptions (A1)–(A4) and* $G \sim \mathcal{G}_n$.
- $t = O_n(1)$, $\hat{n}_1 = n_1$, $R = 0.5$ *and* $\epsilon \in (0, 1)$.
- $\Gamma \subset C_1$ *with* $|\Gamma| = g\epsilon_3^{2t-1} n_1$ *for any* $g \in (0, 1)$ *and* ϵ_3 *as in (A4).*

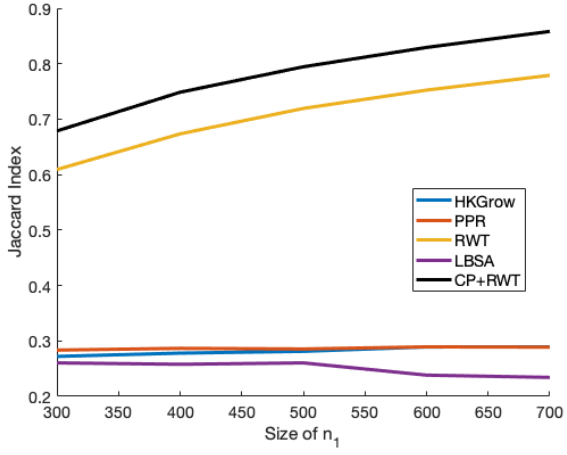
Then if $C_1^{\#} = \texttt{CP+RWT}(A, \Gamma, \epsilon, R, \hat{n}_1, t)$*:*

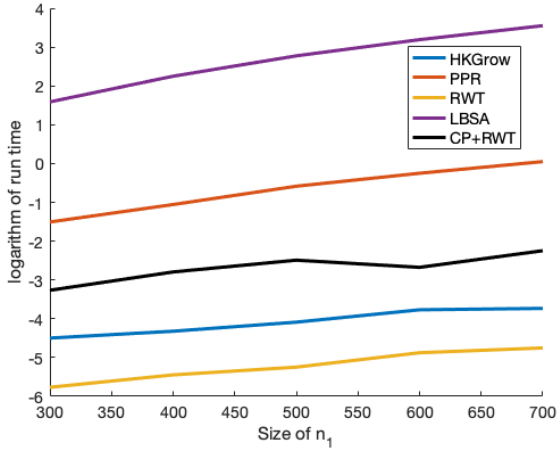
$$\frac{\left|C_1 \triangle C_1^{\#}\right|}{|C_1|} = o_n(1)$$

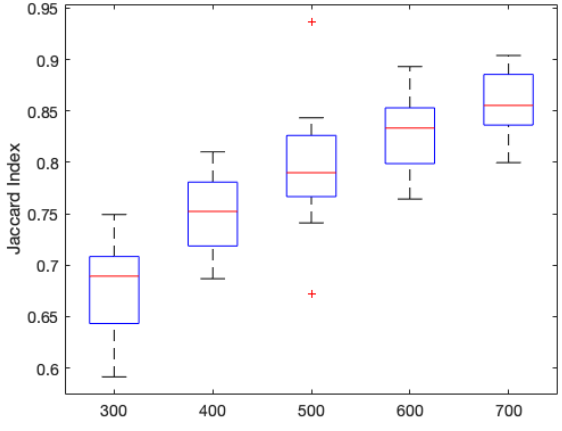
almost surely, for large enough n_1.

Experimental Results: Stochastic block model

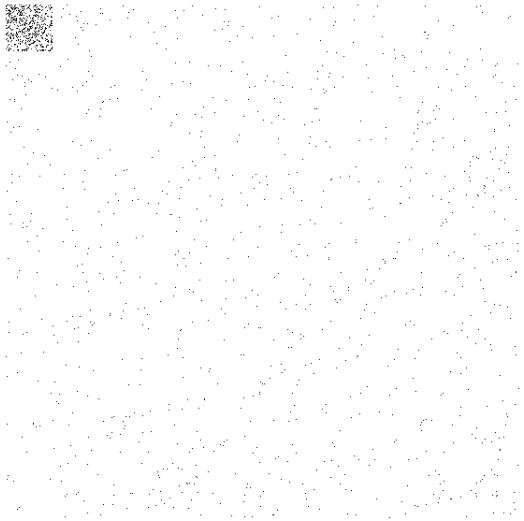


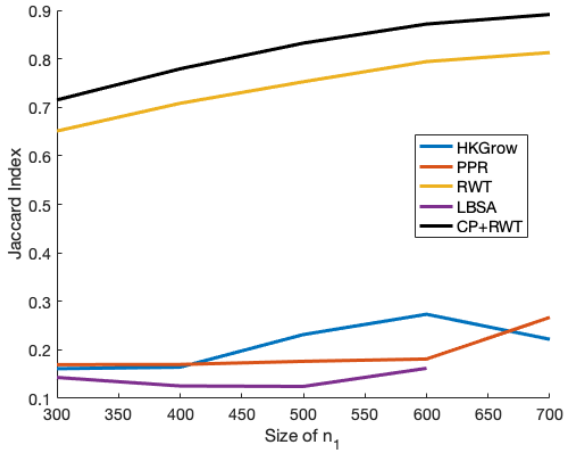


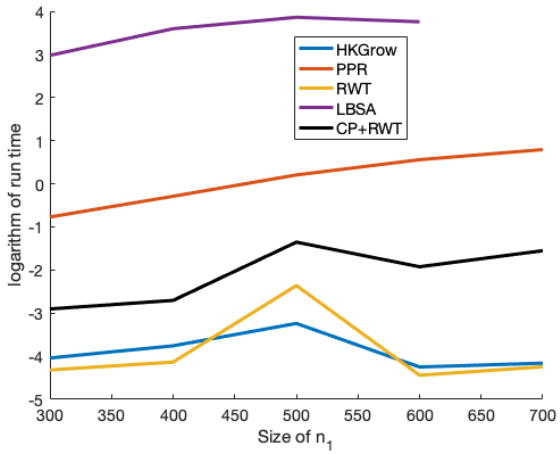


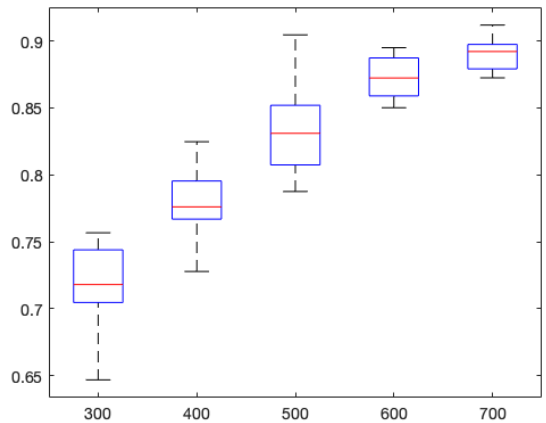


Experimental Results: Stochastic block model









Experimental Results: Social Networks

- `Facebook100`[13] dataset: Facebook networks at American universities.
- Metadata used to define ground-truth clusters.
- Considered four clusters [14]: two good, two moderately good.
- Always take $|\Gamma| = 0.02n_1$.

School	Cluster	Size of graph	Size of Cluster	Conductance
Johns Hopkins	Class of 2009	5180	910	0.21
Rice	Dorm. 203	4087	406	0.47
Simmons	Class of 2009	1518	289	0.11
Colgate	Class of 2006	3482	557	0.49

Table: Basic properties of four clusters. Lower conductance is better.

[13] Traud et al. [2012]
[14] Wang et al. [2017]

Experimental Results: Social Networks

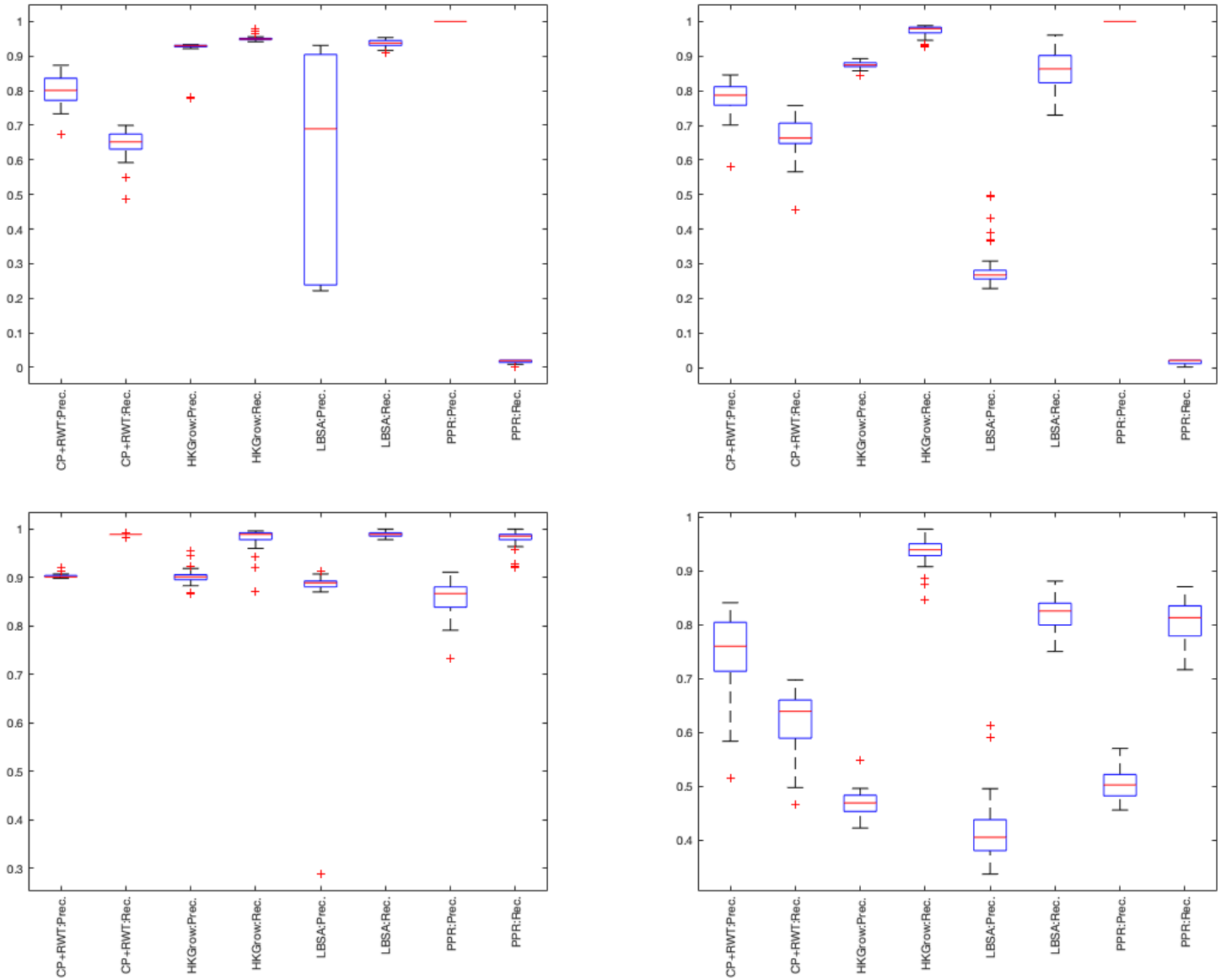


Figure: Clockwise from top left: Johns Hopkins, Rice, Colgate and Simmons.

Iterated CP+RWT for global clustering

Algorithm 4: ICP+RWT

Input: Adj. matrix A, labeled data $\Gamma_a \subset C_a$ for $a = 1, \ldots, k$. Parameters.

Output: $C_1^{\#} \approx C_1, \ldots, C_k^{\#} \approx C_k$

$G^{(1)} \leftarrow G$ and $A^{(1)} \leftarrow A$.

for $a = 1, \ldots k$ **do**

  $C_a^{\#} \leftarrow \texttt{CP+RWT}(A^{(a)}, \Gamma_a, \epsilon, R, \hat{n}_a, t)$

  $G^{(a+1)} \leftarrow G^{(a)} \setminus C_a^{\#}$ and $A^{(a+1)}$ is adj. matrix of $G^{(a+1)}$.

% Labeled Data	0.5	1	1.5	2	2.5
MNIST	96.41%	97.32%	97.44%	97.52%	97.50%
OptDigits	91.88%	95.47%	97.16%	98.06%	98.08%

Table: Classification accuracy, as a function of amount of labeled data, for ICP+RWT on two well-studied benchmark data sets. Results essentially state-of-the-art.[16]

[15] Rasmus et al. [2015], Jacobs et al. [2018], Yin and Tai [2018]

[16] Rasmus et al. [2015], Jacobs et al. [2018], Yin and Tai [2018]

Conclusion

- `ClusterPursuit` is a provably robust, provably efficient cut improvement algorithm.
- Can use `ClusterPursuit` as an algorithmic primitive to design clustering algorithms.
- Theoretical guarantees follow from novel connection between cut improvement and compressed sensing.

Conclusion

- `ClusterPursuit` is a provably robust, provably efficient cut improvement algorithm.
- Can use `ClusterPursuit` as an algorithmic primitive to design clustering algorithms.
- Theoretical guarantees follow from novel connection between cut improvement and compressed sensing.
- Thanks!
- `mckenzie@math.ucla.edu`.

References I

Reid Andersen and Kevin J Lang. An algorithm for improving graph partitions. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 651–660. Society for Industrial and Applied Mathematics, 2008.

Reid Andersen, Fan Chung, and Kevin Lang. Using pagerank to locally partition a graph. *Internet Mathematics*, 4(1):35–64, 2007.

Fan Chung and Mary Radcliffe. On the spectra of general random graphs. *The Electronic Journal of Combinatorics*, 18(1):215, 2011.

Wei Dai and Olgica Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE transactions on Information Theory*, 55(5): 2230–2249, 2009.

Alan Frieze and Michał Karoński. *Introduction to random graphs*. Cambridge University Press, 2016.

Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99 (12):7821–7826, 2002.

References II

Kun He, Yiwei Sun, David Bindel, John Hopcroft, and Yixuan Li. Detecting overlapping communities from local spectral subspaces. In *2015 IEEE International Conference on Data Mining*, pages 769–774. IEEE, 2015.

Matthew A. Herman and Thomas Strohmer. General deviants: An analysis of perturbations in compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):342–349, 2010.

Matt Jacobs, Ekaterina Merkurjev, and Selim Esedoḡlu. Auction dynamics: A volume constrained MBO scheme. *Journal of Computational Physics*, 354: 288–310, 2018.

Kyle Kloster and David F. Gleich. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1386–1395. ACM, 2014.

Ming-Jun Lai and Daniel Mckenzie. Semi-supervised cluster extraction via a compressive sensing approach. *arXiv preprint arXiv:1808.05780*, 2019.

References III

Jeffrey B. Lewis, Keith Poole, Howar Rosenthal, Adam Boche, Aaron Rudkin, and Luke Sonnet. Voteview: Congressional roll-call votes database. *https://voteview.com*, 2020.

Haifeng Li. Improved analysis of SP and CoSaMP under total perturbations. *EURASIP Journal on Advances in Signal Processing*, 2016(1):112, 2016.

Yixuan Li, Kun He, David Bindel, and John E. Hopcroft. Uncovering the small community structure in large networks: A local spectral approach. In *Proceedings of the 24th international conference on world wide web*, pages 658–668, 2015.

Deanna Needell and Joel A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002.

References IV

Lorenzo Orecchia and Zeyuan Allen-Zhu. Flow-based algorithms for local graph clustering. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1267–1286. SIAM, 2014.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the STOC*, volume 4, 2004.

Daniel A. Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, 2013.

References V

Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

Joel A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

Nate Veldt, David Gleich, and Michael Mahoney. A simple and strongly-local flow-based method for cut improvement. In *International Conference on Machine Learning*, pages 1938–1947, 2016.

Nate Veldt, Christine Klymko, and David F Gleich. Flow-based local graph clustering with better seed set inclusion. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 378–386. SIAM, 2019.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Di Wang, Kimon Fountoulakis, Monika Henzinger, Michael W Mahoney, and Satish Rao. Capacity releasing diffusion for speed and locality. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3598–3607. JMLR. org, 2017.

References VI

Ke Yin and Xue-Cheng Tai. An effective region force for some variational models for learning and clustering. *Journal of Scientific Computing*, 74(1): 175–196, 2018.