A Sublinear-Time Spectral Clustering Oracle with Improved Preprocessing Time

Ranran Shen Pan Peng

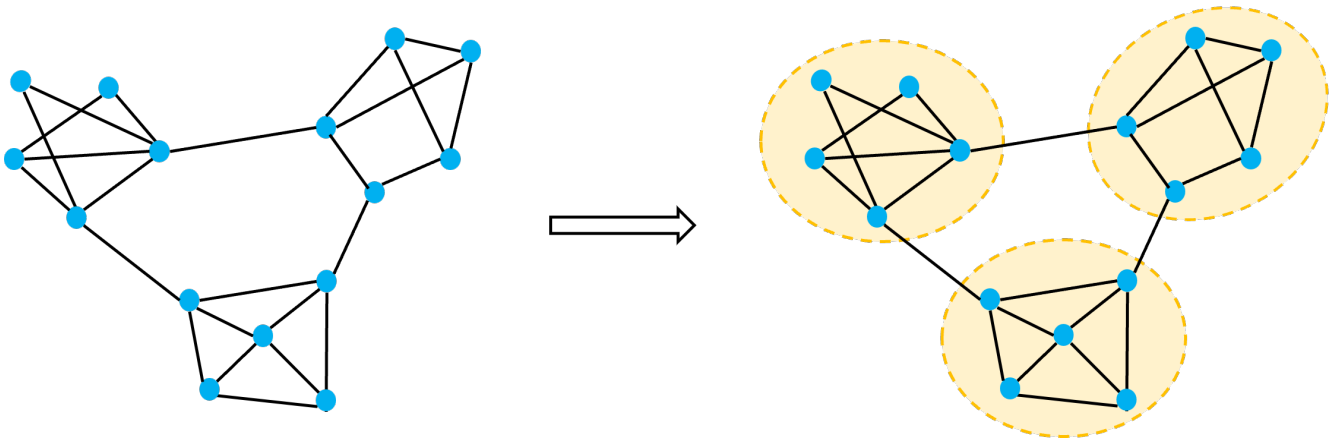
University of Science and Technology of China

December 6, 2023

Outline

Graph Clustering

- Input: $G = (V, E)$ and k $(k \geq 2)$
- Goal: partition V into k disjoint clusters $C_1, \ldots, C_k$, such that each cluster exhibits
  - tight connections inside
  - loose connections outside



Graph clustering $(k = 3)$

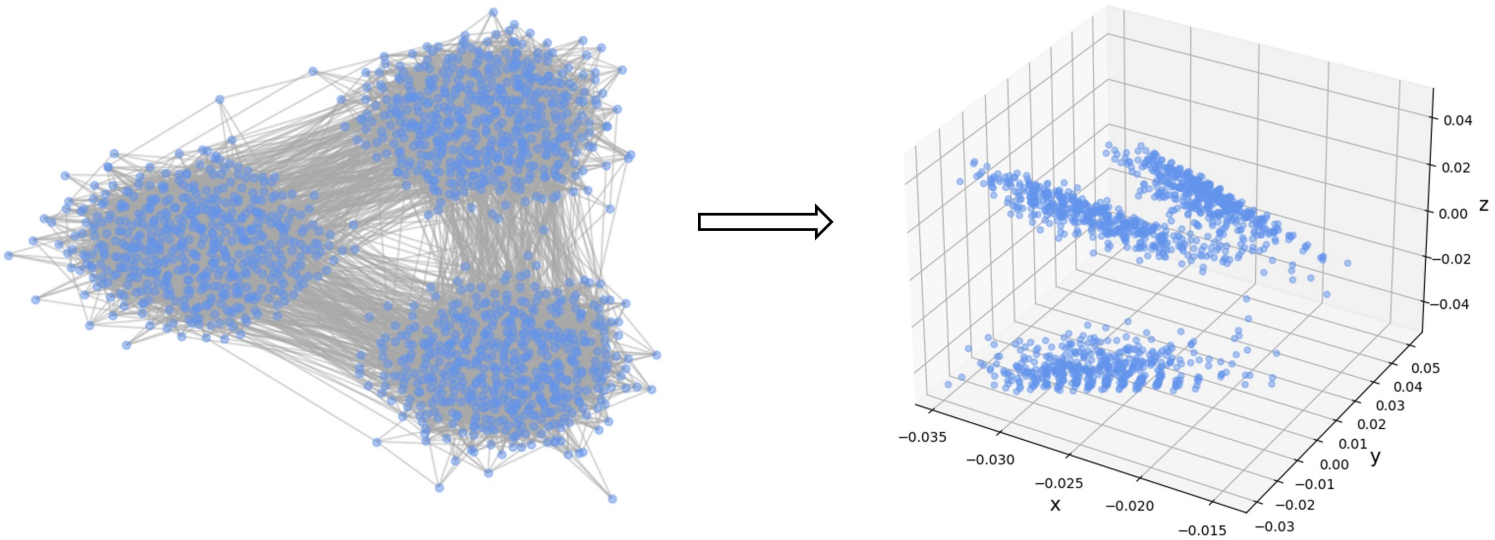
Spectral Clustering

a popular algorithm: spectral clustering (SC)

Spectral Clustering

a popular algorithm: spectral clustering (SC)

- step 1: map vertices to a k-dimensional Euclidean space

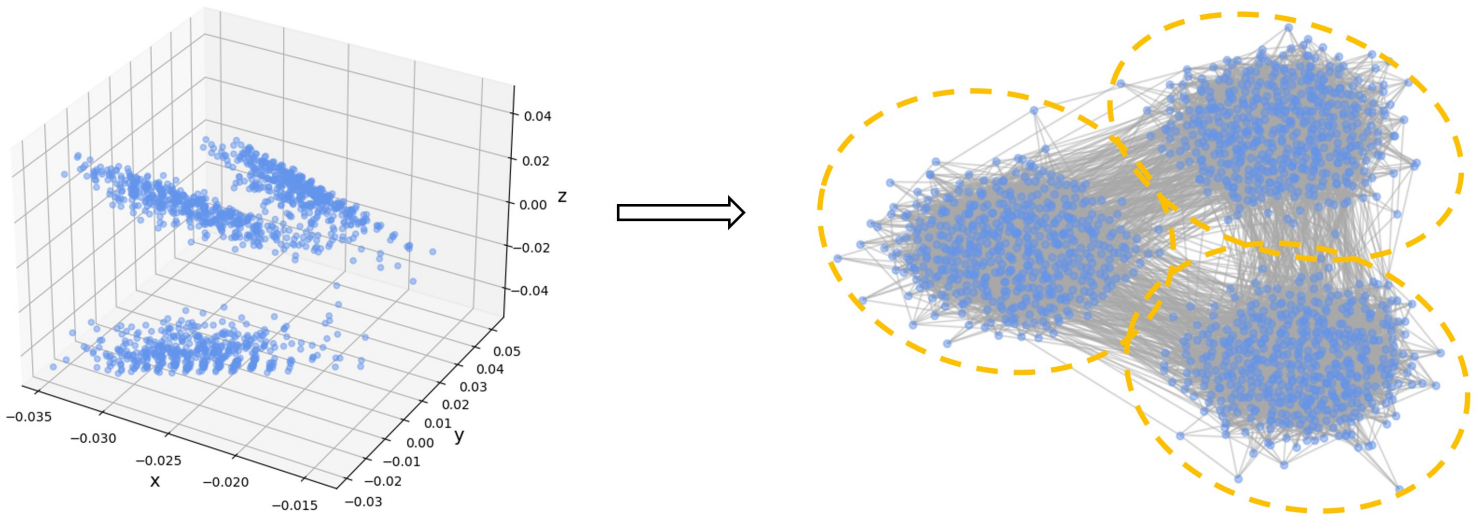


Step 1 of SC

Spectral Clustering

a popular algorithm: spectral clustering (SC)

- step 1: map vertices to a k-dimensional Euclidean space
- step 2: cluster all the resulting points (e.g., k-means)

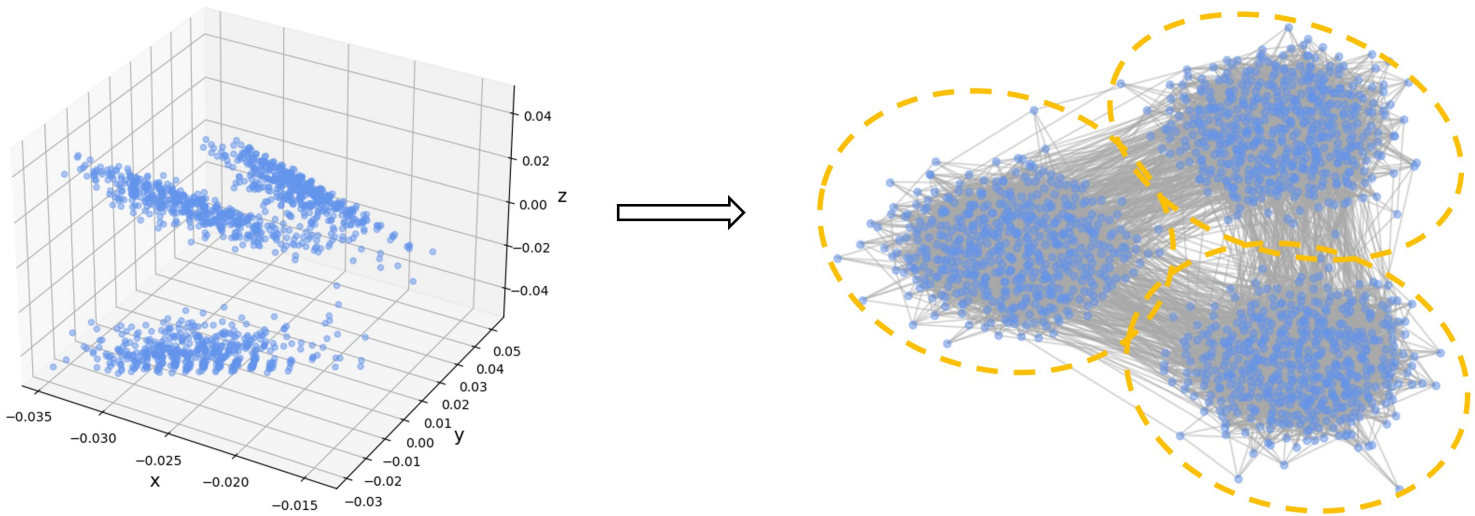


Step 2 of SC

Spectral Clustering

a popular algorithm: spectral clustering (SC)

- step 1: map vertices to a k-dimensional Euclidean space
- step 2: cluster all the resulting points (e.g., k-means)



Step 2 of SC

How to map? Using spectral embeddings.

Spectral Embedding

$G = (V, E)$, $n = |V|$.

Normalized Laplacian Matrix of G

$$L = D^{-1/2}(D - A)D^{-1/2}.$$

- $D \in \mathbb{R}^{n \times n}$: a diagonal matrix where $D(i, i) = \text{degree}(i)$
- $A \in \mathbb{R}^{n \times n}$: adjacency matrix of G

Spectral Embedding

$G = (V, E)$, $n = |V|$.

Normalized Laplacian Matrix of G

$$L = D^{-1/2}(D - A)D^{-1/2}.$$

- $D \in \mathbb{R}^{n \times n}$: a diagonal matrix where $D(i, i) = \text{degree}(i)$
- $A \in \mathbb{R}^{n \times n}$: adjacency matrix of G

Eigen-decomposition:

- eigenvalues of L: $0 \leq \lambda_1 \leq \cdots \leq \lambda_n \leq 2$
- eigenvectors of L: $u_1, \ldots, u_n \in \mathbb{R}^n$

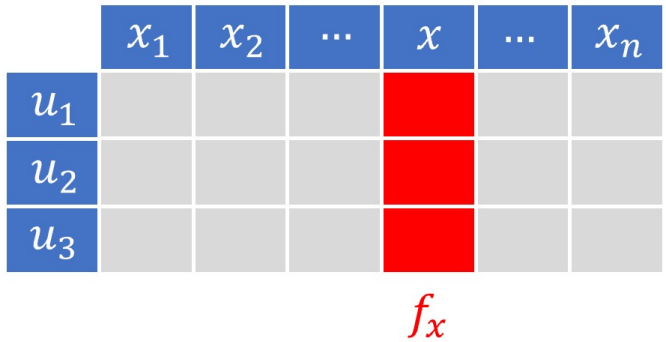
Spectral Embedding

- eigenvectors of L: $u_1, \ldots, u_n \in \mathbb{R}^n$

Spectral Embedding

For a vertex $x \in V$, the spectral embedding of x is defined to be

$$f_x = (u_1(x), \ldots, u_i(x), \ldots, u_k(x))^T, f_x \in \mathbb{R}^k.$$



Spectral embedding ($k = 3$)

Problems of Spectral Clustering

- eigen-decomposition takes $\mathrm{poly}(n)$ time $\Rightarrow$ spectral clustering runs in $\mathrm{poly}(n)$ time, where $n = |V|$

Problems of Spectral Clustering

- eigen-decomposition takes $\mathrm{poly}(n)$ time $\Rightarrow$ spectral clustering runs in $\mathrm{poly}(n)$ time, where $n = |V|$
- n increases $\Rightarrow$ impractical

Problems of Spectral Clustering

- eigen-decomposition takes $\mathrm{poly}(n)$ time $\Rightarrow$ spectral clustering runs in $\mathrm{poly}(n)$ time, where $n = |V|$
- n increases $\Rightarrow$ impractical

- So, sublinear-time algorithms?

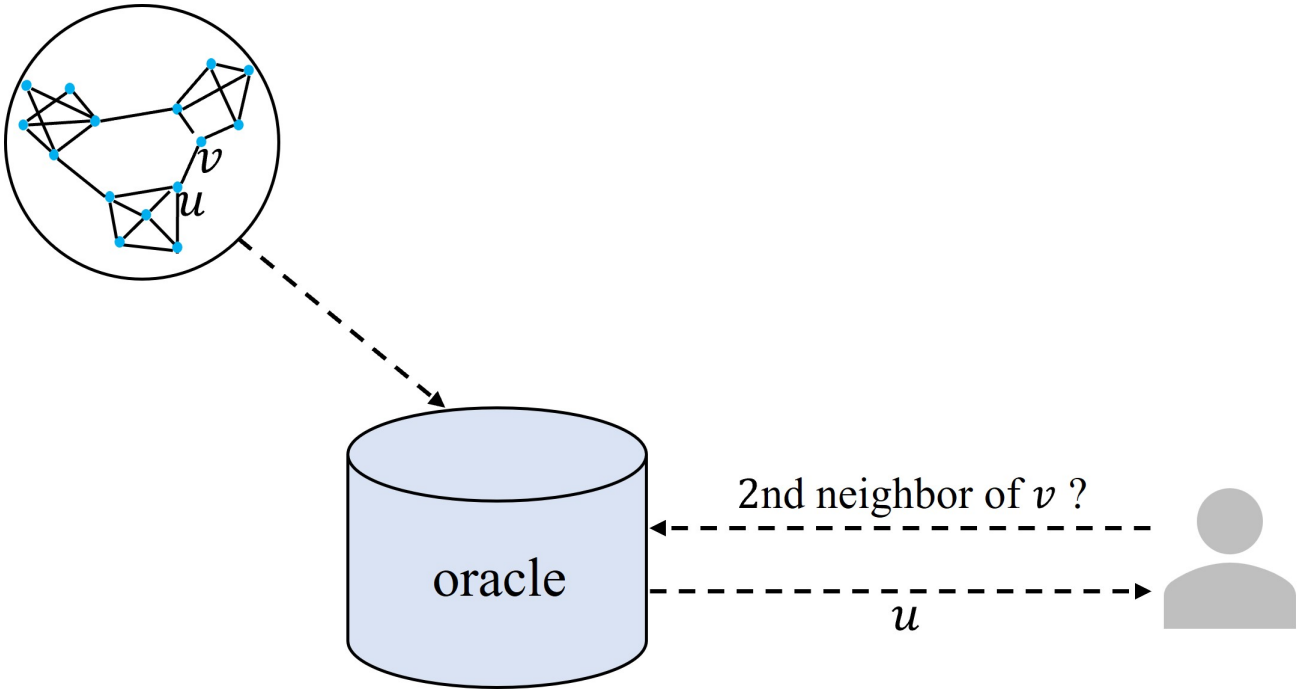
Problems of Spectral Clustering

- eigen-decomposition takes $\mathrm{poly}(n)$ time $\Rightarrow$ spectral clustering runs in $\mathrm{poly}(n)$ time, where $n = |V|$
- n increases $\Rightarrow$ impractical

- So, sublinear-time algorithms?
  - sublinear-time spectral clustering oracles

Query Access

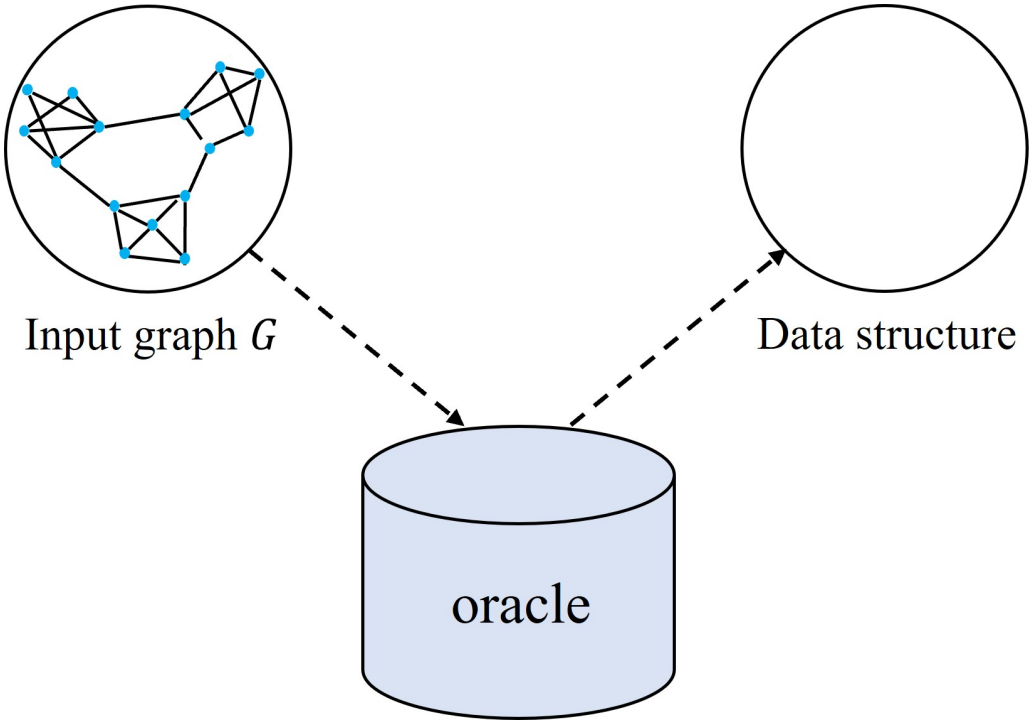
- Has query access to the adjacency list of the input graph G
  - one can query the i-th neighbor of any vertex in constant time



Neighbor query

Two Phases

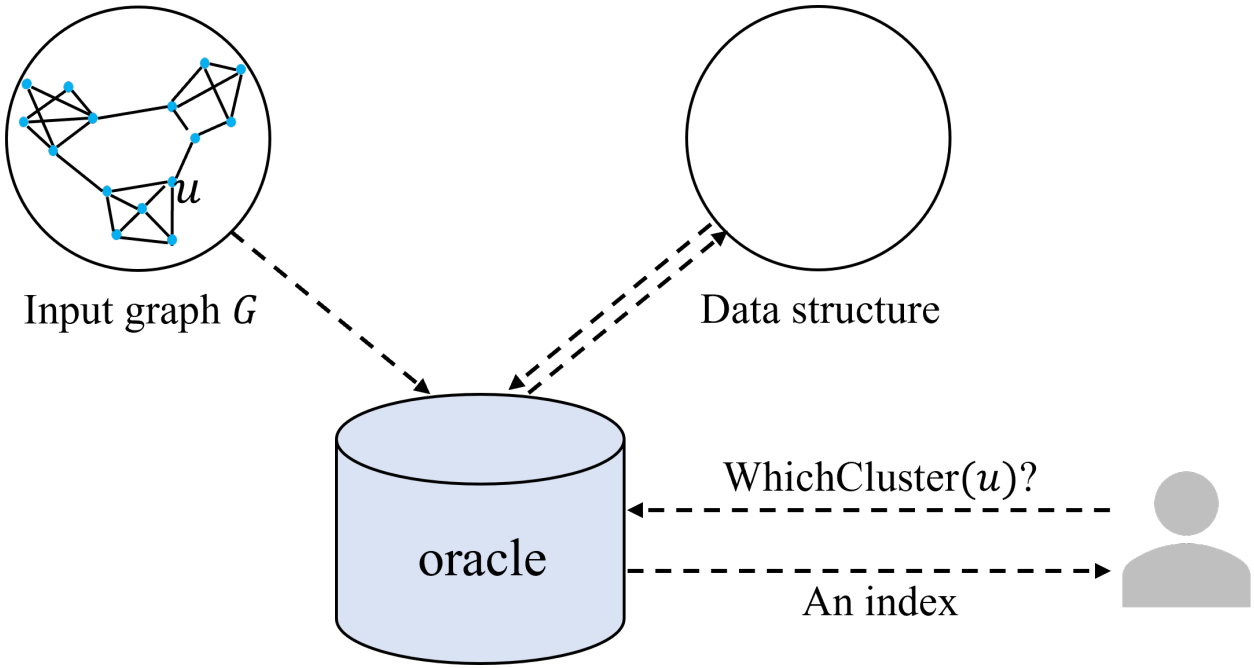
- Preprocessing phase (sublinear-time)
  - build a data structure



Preprocessing phase

Two Phases

- Preprocessing phase (sublinear-time)
  - build a data structure
- Query phase (sublinear-time)
  - answer $\textsc{WhichCluster}(u)$ queries



Query phase

Requirement

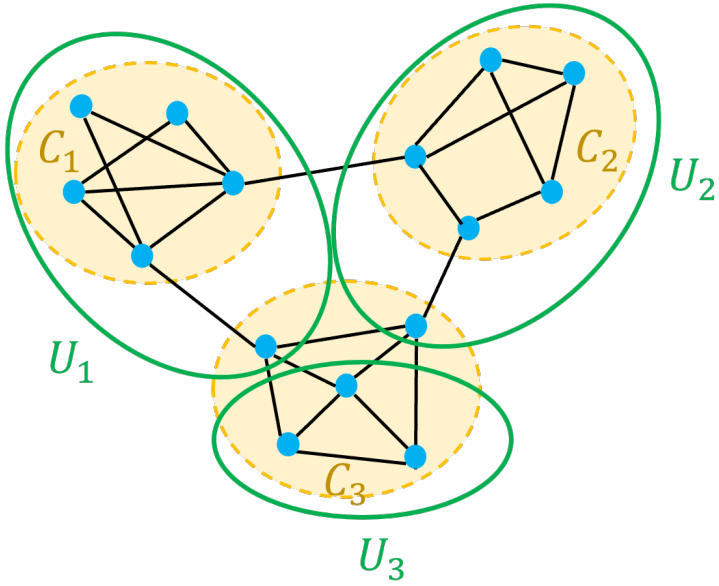
Let $U_i = \{x \in V : \textsc{WhichCluster}(x) = i\}, 1 \leq i \leq k$.

Requirement

Let $U_i = \{x \in V : \textsc{WhichCluster}(x) = i\}, 1 \le i \le k$.

The resulting partition $U_1, \ldots, U_k$ should be

- close to the ground-truth clustering



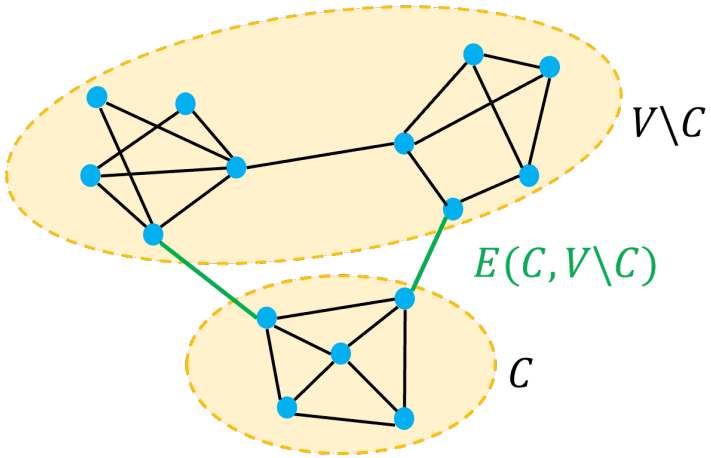
$|U_i \triangle C_i|$ is small

Small misclassification error

Conductance

$G = (V, E)$: a d-regular graph. For a set $C \subseteq V$,

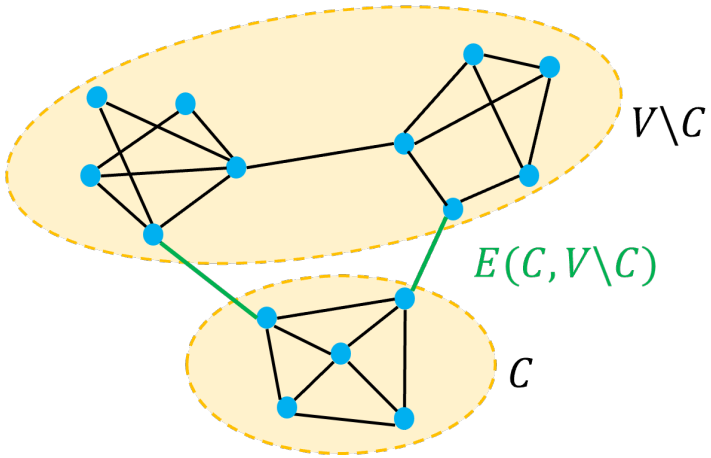
- the outer conductance of C: $\phi_{\text{out}}(C, V) = \frac{|E(C, V \setminus C)|}{d \cdot |C|}$



Conductance

$G = (V, E)$: a d-regular graph. For a set $C \subseteq V$,

- the outer conductance of C: $\phi_{\text{out}}(C, V) = \frac{|E(C, V \setminus C)|}{d \cdot |C|}$



- the inner conductance of C: $\phi_{\text{in}}(C) = \min_{S \subseteq C, 0 < |S| \leq \frac{|C|}{2}} \phi_{\text{out}}(S, C)$

A Good Cluster

A good cluster C has:

- $\phi_{\text{in}}(C) \geq \varphi_{\text{in}} \Leftrightarrow$ tight connection inside
- $\phi_{\text{out}}(C, V) \leq \varphi_{\text{out}} \Leftrightarrow$ loose connection outside ($\varphi_{\text{out}} \ll \varphi_{\text{in}}$)

More Formally About Input Graph G

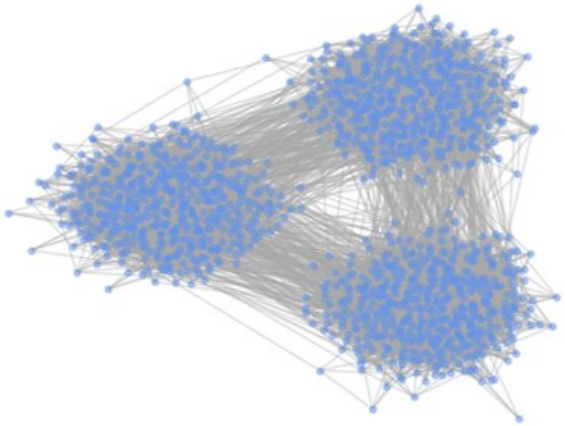
- d-regular graphs

More Formally About Input Graph G

- d-regular graphs
  - can be generalized to d-bounded graphs: maximum degree $\leq d$

More Formally About Input Graph G

- d-regular graphs
  - can be generalized to d-bounded graphs: maximum degree $\leq d$

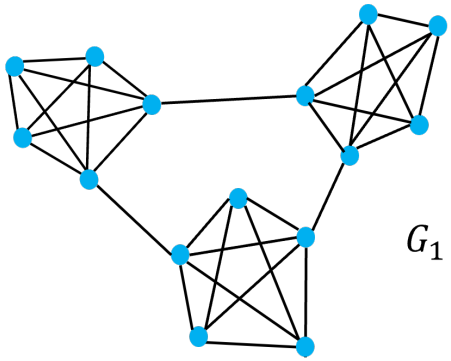


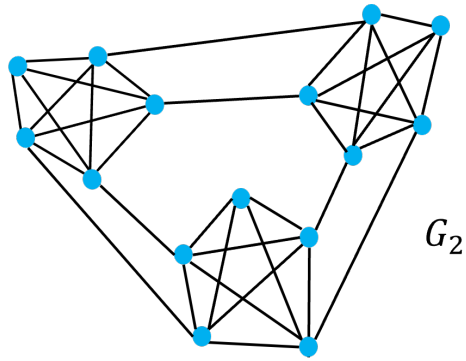
A clusterable graph

- $(k, \varphi, \varepsilon)$-clusterable graphs ($\varepsilon \ll \varphi$)
  - has a k-partition of V, denoted by $C_1, \ldots, C_k$, $\frac{|C_i|}{|C_j|} \in O(1)$
  - tight connections inside: $\phi_{\text{in}}(C_i) \geq \varphi$
  - loose connections outside: $\phi_{\text{out}}(C_i, V) \leq \varepsilon$

Intuition of $(k, \varphi, \varepsilon)$-Clusterable Graphs

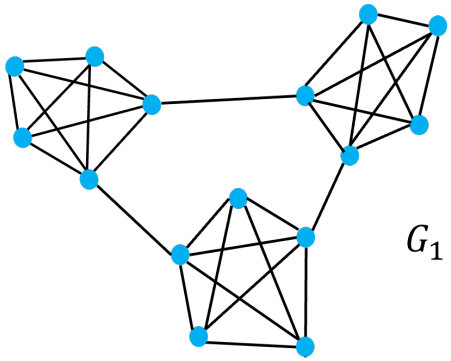
- G_1: a $(k, \varphi, \varepsilon_1)$-clusterable graph
- G_2: a $(k, \varphi, \varepsilon_2)$-clusterable graph, $\varepsilon_2 > \varepsilon_1$

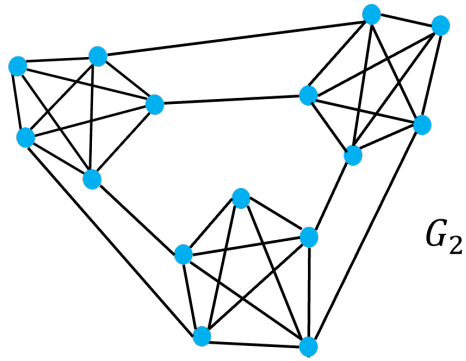




Intuition of $(k, \varphi, \varepsilon)$-Clusterable Graphs

- G_1: a $(k, \varphi, \varepsilon_1)$-clusterable graph
- G_2: a $(k, \varphi, \varepsilon_2)$-clusterable graph, $\varepsilon_2 > \varepsilon_1$

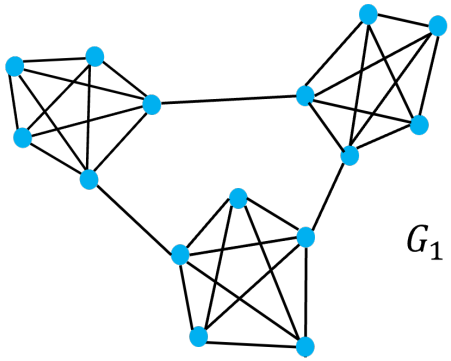


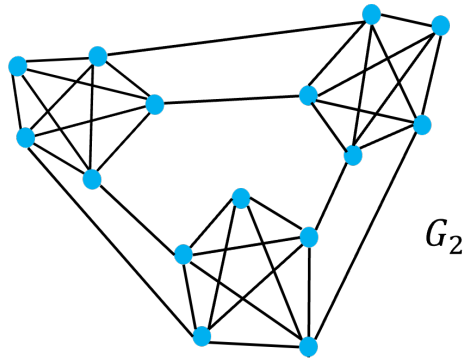


Intuitively, G_2 is more difficult to cluster.

Intuition of $(k, \varphi, \varepsilon)$-Clusterable Graphs

- G_1: a $(k, \varphi, \varepsilon_1)$-clusterable graph
- G_2: a $(k, \varphi, \varepsilon_2)$-clusterable graph, $\varepsilon_2 > \varepsilon_1$





Intuitively, G_2 is more difficult to cluster.

Oracle that can handle a clusterable graph with the bigger ε is the better.

Outline

Previous Work

	[Pen20][1]	[GKL+21][2]
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)\cdot\log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$
preprocessing time	$O\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})\right)$	$2^{\mathrm{poly}(\frac{k}{\varepsilon})}\cdot n^{1/2+O(\varepsilon)}\cdot\mathrm{poly}(\log n)$
query time	$O\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})\right)$	$n^{1/2+O(\varepsilon)}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})$
misclassification error (fraction)	$O\left(k\sqrt{\varepsilon}\right)$	$O\left(\log k\cdot\varepsilon\right)$ per cluster

[1] Peng P. Robust clustering oracle and local reconstructor of cluster structure of graphs. SODA 2020.

[2] Gluch G, Kapralov M, Lattanzi S, Mousavifar A and Sohler C. Spectral clustering oracles in sublinear time. SODA 2021.

Previous Work

	[Pen20][1]	[GKL+21][2]
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)\cdot\log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$
preprocessing time	$O\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})\right)$	$2^{\mathrm{poly}(\frac{k}{\varepsilon})}\cdot n^{1/2+O(\varepsilon)}\cdot\mathrm{poly}(\log n)$
query time	$O\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})\right)$	$n^{1/2+O(\varepsilon)}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})$
misclassification error (fraction)	$O\left(k\sqrt{\varepsilon}\right)$	$O\left(\log k\cdot\varepsilon\right)$ per cluster

- conductance gap: [GKL+21] is better than [Pen20]

[1] Peng P. Robust clustering oracle and local reconstructor of cluster structure of graphs. SODA 2020.

[2] Gluch G, Kapralov M, Lattanzi S, Mousavifar A and Sohler C. Spectral clustering oracles in sublinear time. SODA 2021.

Previous Work

	[Pen20][1]	[GKL+21][2]
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)\cdot\log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$
preprocessing time	$O\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})\right)$	$2^{\mathrm{poly}(\frac{k}{\varepsilon})}\cdot n^{1/2+O(\varepsilon)}\cdot\mathrm{poly}(\log n)$
query time	$O\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})\right)$	$n^{1/2+O(\varepsilon)}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})$
misclassification error (fraction)	$O\left(k\sqrt{\varepsilon}\right)$	$O\left(\log k\cdot\varepsilon\right)$ per cluster

- conductance gap: [GKL+21] is better than [Pen20]
- preprocessing time: $2^{\mathrm{poly}(\frac{k}{\varepsilon})}$ in [GKL+21]
  - hard to implement

[1] Peng P. Robust clustering oracle and local reconstructor of cluster structure of graphs. SODA 2020.

[2] Gluch G, Kapralov M, Lattanzi S, Mousavifar A and Sohler C. Spectral clustering oracles in sublinear time. SODA 2021.

Motivation

	[Pen20]	[GKL+21]
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)\cdot\log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$
preprocessing time	$O\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})\right)$	$2^{\mathrm{poly}(\frac{k}{\varepsilon})}\cdot n^{1/2+O(\varepsilon)}\cdot\mathrm{poly}(\log n)$
query time	$O\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})\right)$	$n^{1/2+O(\varepsilon)}\cdot\mathrm{poly}(\frac{k\log n}{\varepsilon})$
misclassification error (fraction)	$O\left(k\sqrt{\varepsilon}\right)$	$O\left(\log k\cdot\varepsilon\right)$ per cluster

Can we get a spectral clustering oracle with

- better conductance gap than [Pen20] and
- better preprocessing time than [GKL+21]?

Outline

Our Results

	[Pen20]	[GKL+21]	**this work**
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)\cdot\log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)}$
preprocessing time	$\widetilde{O}\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k}{\varepsilon})\right)$	$\widetilde{O}\big(2^{\mathrm{poly}(\frac{k}{\varepsilon})}\cdot n^{1/2+O(\varepsilon)}\big)$	$\widetilde{O}\big(\mathrm{poly}(k)\cdot n^{1/2+O(\varepsilon)}\big)$
query time	$\widetilde{O}\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k}{\varepsilon})\right)$	$\widetilde{O}\big(n^{1/2+O(\varepsilon)}\cdot \mathrm{poly}(\frac{k}{\varepsilon})\big)$	$\widetilde{O}\big(n^{1/2+O(\varepsilon)}\cdot \mathrm{poly}(k)\big)$
misclassification error (fraction)	$O\left(k\sqrt{\varepsilon}\right)$	$O\left(\log k\cdot\varepsilon\right)$ per cluster	$O\left(\mathrm{poly}(k)\cdot\varepsilon^{1/3}\right)$ per cluster

Our Results

	[Pen20]	[GKL+21]	**this work**
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)\cdot\log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)}$
preprocessing time	$\widetilde{O}\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k}{\varepsilon})\right)$	$\widetilde{O}\big(2^{\mathrm{poly}(\frac{k}{\varepsilon})}\cdot n^{1/2+O(\varepsilon)}\big)$	$\widetilde{O}\big(\mathrm{poly}(k)\cdot n^{1/2+O(\varepsilon)}\big)$
query time	$\widetilde{O}\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k}{\varepsilon})\right)$	$\widetilde{O}\big(n^{1/2+O(\varepsilon)}\cdot \mathrm{poly}(\frac{k}{\varepsilon})\big)$	$\widetilde{O}\big(n^{1/2+O(\varepsilon)}\cdot \mathrm{poly}(k)\big)$
misclassification error (fraction)	$O\left(k\sqrt{\varepsilon}\right)$	$O\left(\log k\cdot\varepsilon\right)$ per cluster	$O\left(\mathrm{poly}(k)\cdot\varepsilon^{1/3}\right)$ per cluster

- Conductance gap: $\mathrm{poly}(k)$
  - better than $\mathrm{poly}(k)\cdot\log n$ in [Pen20]
  - slightly worse than $\log k$ in [GKL+21]

Our Results

	[Pen20]	[GKL+21]	**this work**
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)\cdot\log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)}$
preprocessing time	$\widetilde{O}\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k}{\varepsilon})\right)$	$\widetilde{O}\big(2^{\mathrm{poly}(\frac{k}{\varepsilon})}\cdot n^{1/2+O(\varepsilon)}\big)$	$\widetilde{O}\big(\mathrm{poly}(k)\cdot n^{1/2+O(\varepsilon)}\big)$
query time	$\widetilde{O}\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k}{\varepsilon})\right)$	$\widetilde{O}\big(n^{1/2+O(\varepsilon)}\cdot \mathrm{poly}(\frac{k}{\varepsilon})\big)$	$\widetilde{O}\big(n^{1/2+O(\varepsilon)}\cdot \mathrm{poly}(k)\big)$
misclassification error (fraction)	$O\left(k\sqrt{\varepsilon}\right)$	$O\left(\log k\cdot\varepsilon\right)$ per cluster	$O\left(\mathrm{poly}(k)\cdot\varepsilon^{1/3}\right)$ per cluster

- Conductance gap: $\mathrm{poly}(k)$
  - better than $\mathrm{poly}(k)\cdot\log n$ in [Pen20]
  - slightly worse than $\log k$ in [GKL+21]
- Preprocessing time: polynomial in k, better than exponential in k in [GKL+21]

Theorem 2

Theorem 2 (Informal)

Let $G_0 = (V, E)$ be a $(k, \varphi, \varepsilon)$-clusterable graph, where $\frac{\varepsilon}{\varphi^4} \ll \frac{1}{\mathrm{poly}(k)}$.

- $G_0 \xrightarrow{\text{deleting at most } O(d\varphi^2) \text{ edges in each cluster}} G$, or
- $G_0 \xrightarrow{\text{randomly deleting at most } O\left(\frac{kd^2}{d+\log k}\right) \text{ edges}} G$

w.h.p., there exists a clustering oracle for G with the same guarantees as presented in Theorem 1.

Remark: Our oracle is robust against a few edge deletions.

Outline

Recall: Proof Outline of [GKL+21]

$f_x = (u_1(x), \ldots, u_k(x))^T$: spectral embedding of vertex $x \in V$

Cluster Center of C_i

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} f_x$$

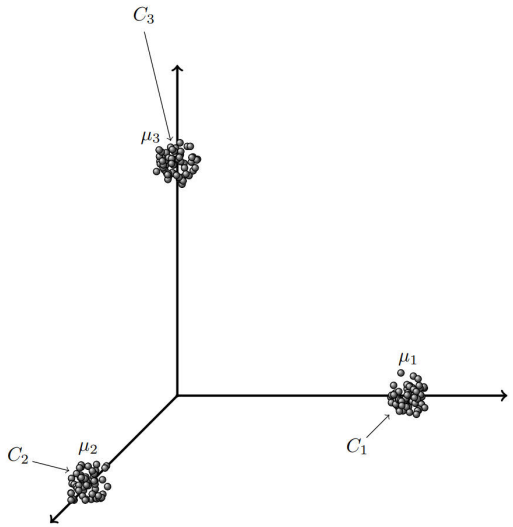
Recall: Proof Outline of [GKL+21]

$f_x = (u_1(x), \ldots, u_k(x))^T$: spectral embedding of vertex $x \in V$

Cluster Center of C_i

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} f_x$$

- μ's are close to orthogonal
- if $x \in C_i$, then f_x is well-concentrated around μ_i



Cluster centers ($k = 3$, [GKL+21])

Recall: Proof Outline of [GKL+21]

$\langle f_x, f_y \rangle$: dot product of f_x and f_y

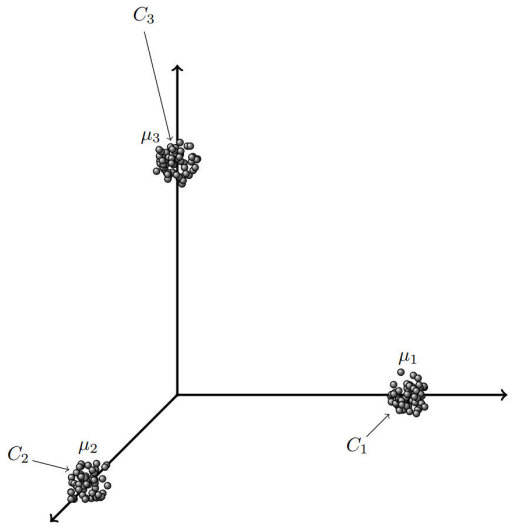
Theorem 3 ([GKL+21], Informal)

There exists a procedure that outputs $\langle f_x, f_y \rangle_{\mathrm{apx}}$ such that w.h.p.

$$|\langle f_x, f_y \rangle_{\mathrm{apx}} - \langle f_x, f_y \rangle| < \frac{1}{n^6}$$

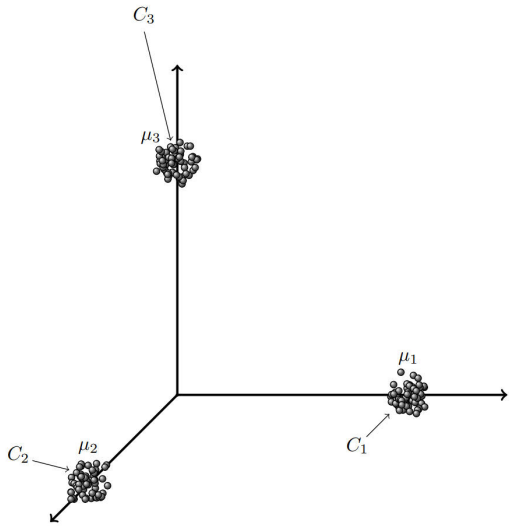
in $n^{1/2+O(\varepsilon)} \cdot \mathrm{poly}(k \log n)$ time.

Recall: Proof Outline of [GKL+21]



Cluster centers ($k = 3$, [GKL+21])

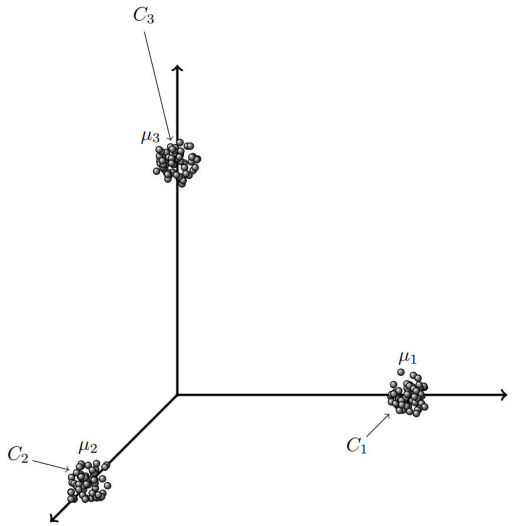
Recall: Proof Outline of [GKL+21]



Cluster centers ($k = 3$, [GKL+21])

- Preprocessing phase:
  - sample some vertices
  - find k cluster centers $\hat{\mu}_1, \ldots, \hat{\mu}_k$ to approximate $\mu_1, \ldots, \mu_k$

Recall: Proof Outline of [GKL+21]



Cluster centers ($k = 3$, [GKL+21])

- Preprocessing phase:
  - sample some vertices
  - find k cluster centers $\hat{\mu}_1, \ldots, \hat{\mu}_k$ to approximate $\mu_1, \ldots, \mu_k$
- Query phase (query x):
  - assign x to cluster C_i if f_x is close to $\hat{\mu}_i$ (i.e., $\langle f_x, \hat{\mu}_i \rangle$ is large)

Recall: Proof Outline of [GKL+21]

Preprocessing phase:

- sample a set of $O(\frac{k^4 \log k}{\varepsilon})$ vertices
- guess in exponential time (i.e., $2^{\mathrm{poly}(\frac{k}{\varepsilon})}$) to get $\hat{\mu}_1, \ldots, \hat{\mu}_k$

Recall: Proof Outline of [GKL+21]

Preprocessing phase:

- sample a set of $O(\frac{k^4 \log k}{\varepsilon})$ vertices
- guess in exponential time (i.e., $2^{\mathrm{poly}(\frac{k}{\varepsilon})}$) to get $\hat{\mu}_1, \ldots, \hat{\mu}_k$

Resulting in $2^{\mathrm{poly}(\frac{k}{\varepsilon})} \cdot n^{1/2+O(\varepsilon)} \cdot \mathrm{poly}(\log n)$ preprocessing time.

Recall: Proof Outline of [GKL+21]

Preprocessing phase:

- sample a set of $O(\frac{k^4 \log k}{\varepsilon})$ vertices
- guess in exponential time (i.e., $2^{\mathrm{poly}(\frac{k}{\varepsilon})}$) to get $\hat{\mu}_1, \ldots, \hat{\mu}_k$

Resulting in $2^{\mathrm{poly}(\frac{k}{\varepsilon})} \cdot n^{1/2+O(\varepsilon)} \cdot \mathrm{poly}(\log n)$ preprocessing time.

This work

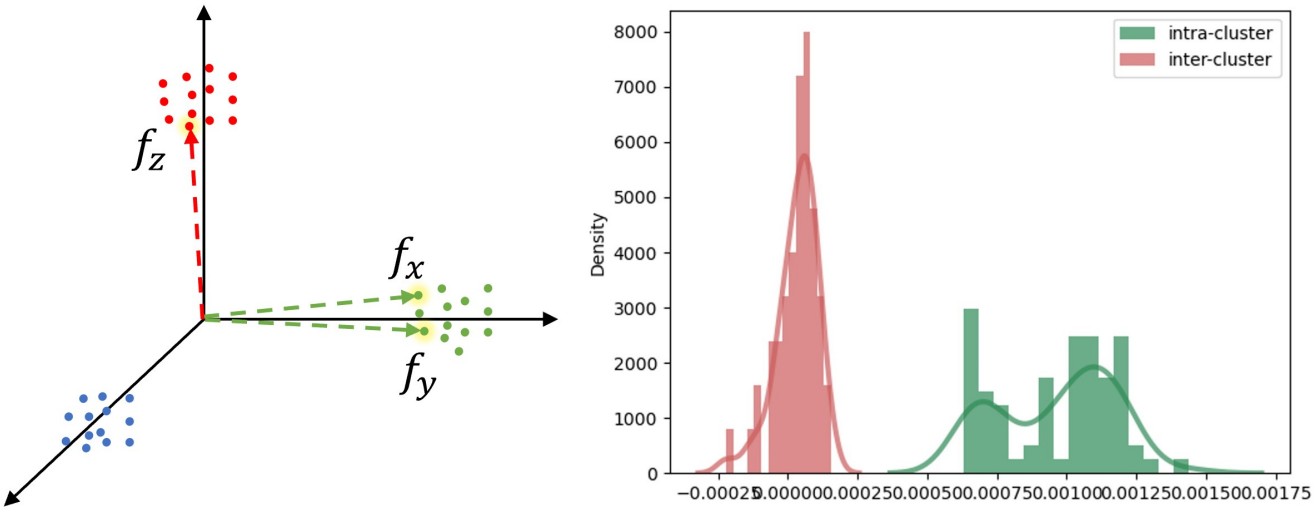
- use cluster centers? No
- use pairwise dot product of spectral embeddings

Our Technique: A Nice Dot Product Gap

Lemma 1 (Informal)

For most vertices in a $(k, \varphi, \varepsilon)$-clusterable graph,

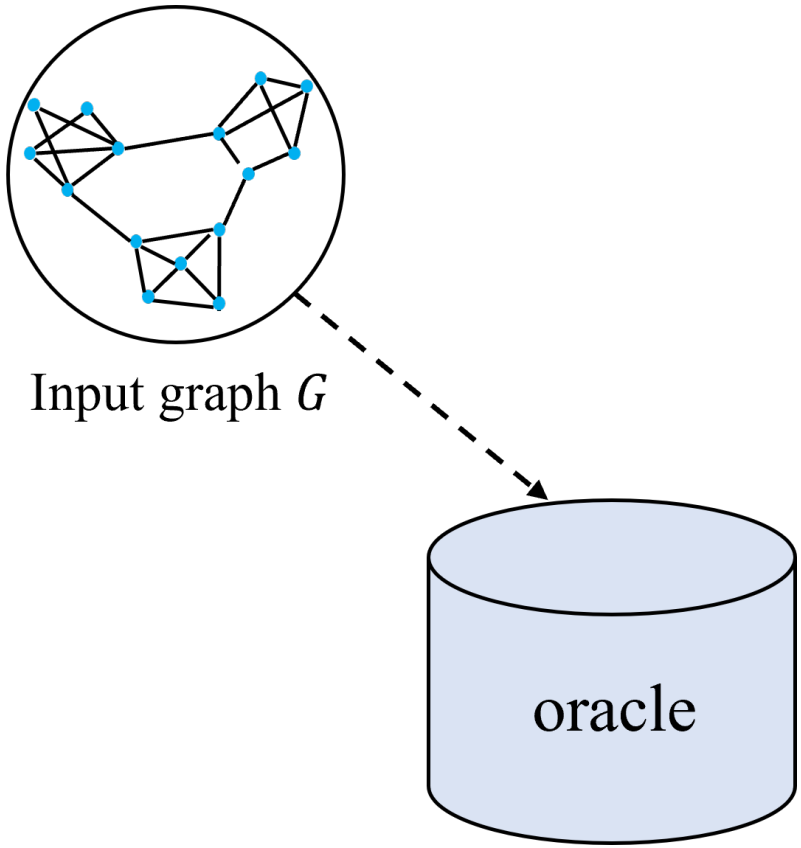
- if x and y are in the same cluster, then $\langle f_x, f_y \rangle$ is close to $\frac{k}{n}$
- if x and z are in the different clusters, then $\langle f_x, f_z \rangle$ is close to 0.



A nice dot product gap

Our Oracle: Preprocessing Phase

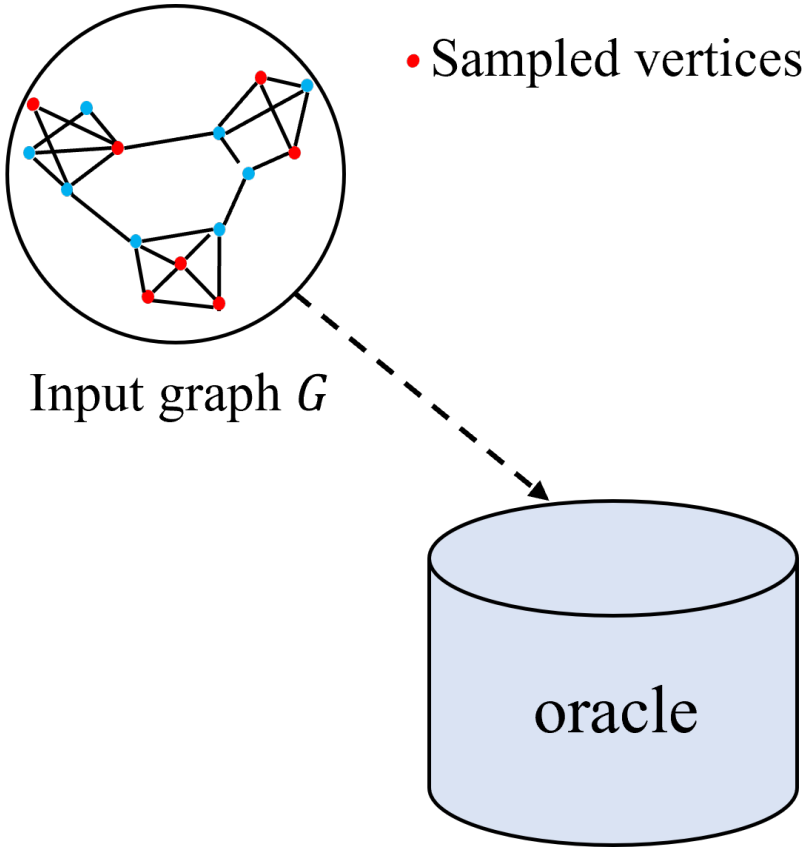
Input graph G: $(k, \varphi, \varepsilon)$-clusterable, $\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)}$



Our clustering oracle

Our Oracle: Preprocessing Phase

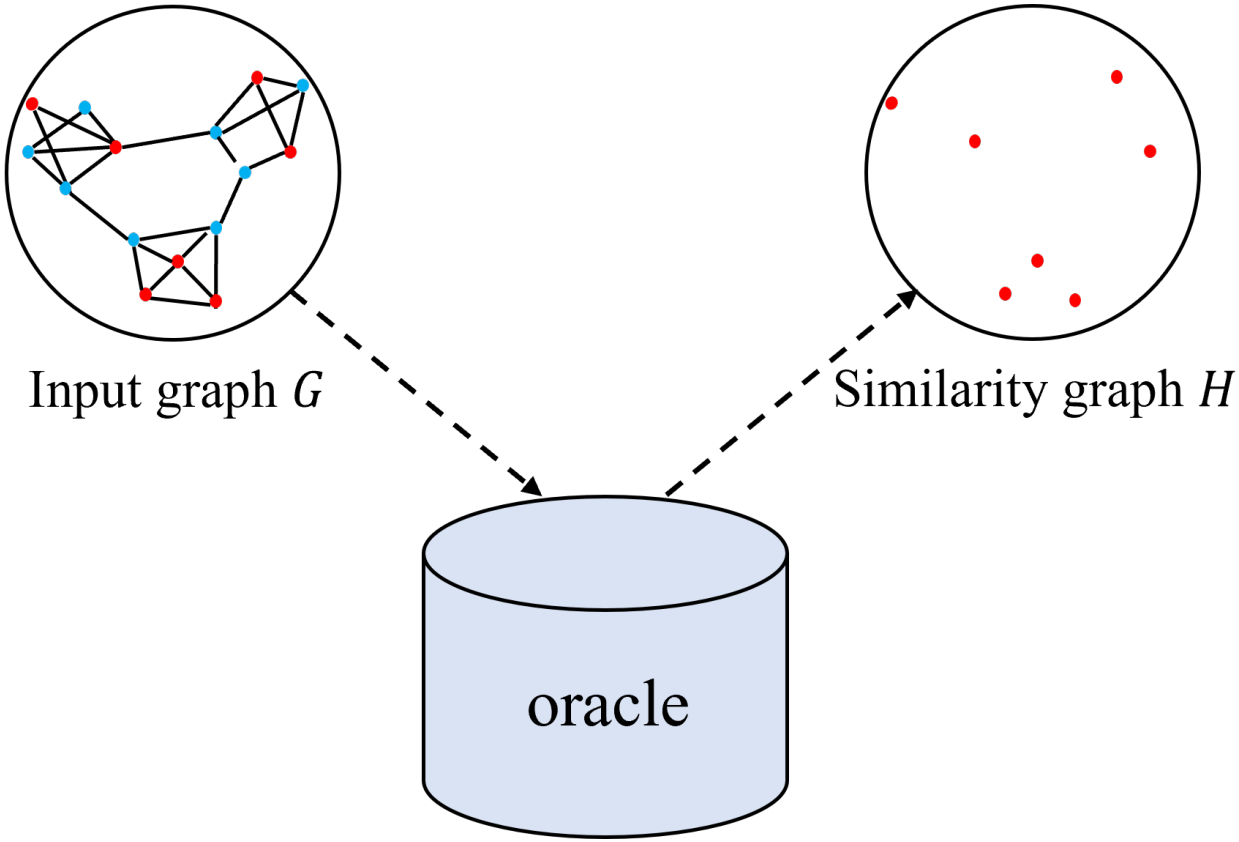
step 1: sample a set S of $O(\mathrm{poly}(k))$ vertices



Sample a set S

Our Oracle: Preprocessing Phase

step 2: generate a similarity graph $H = (S, \emptyset)$



Generate a similarity graph H

Our Oracle: Preprocessing Phase

Lemma 1:

- x, y in the same cluster $\Rightarrow \langle f_x, f_y \rangle$ is large
- x, y in the different clusters $\Rightarrow \langle f_x, f_y \rangle$ is close to 0

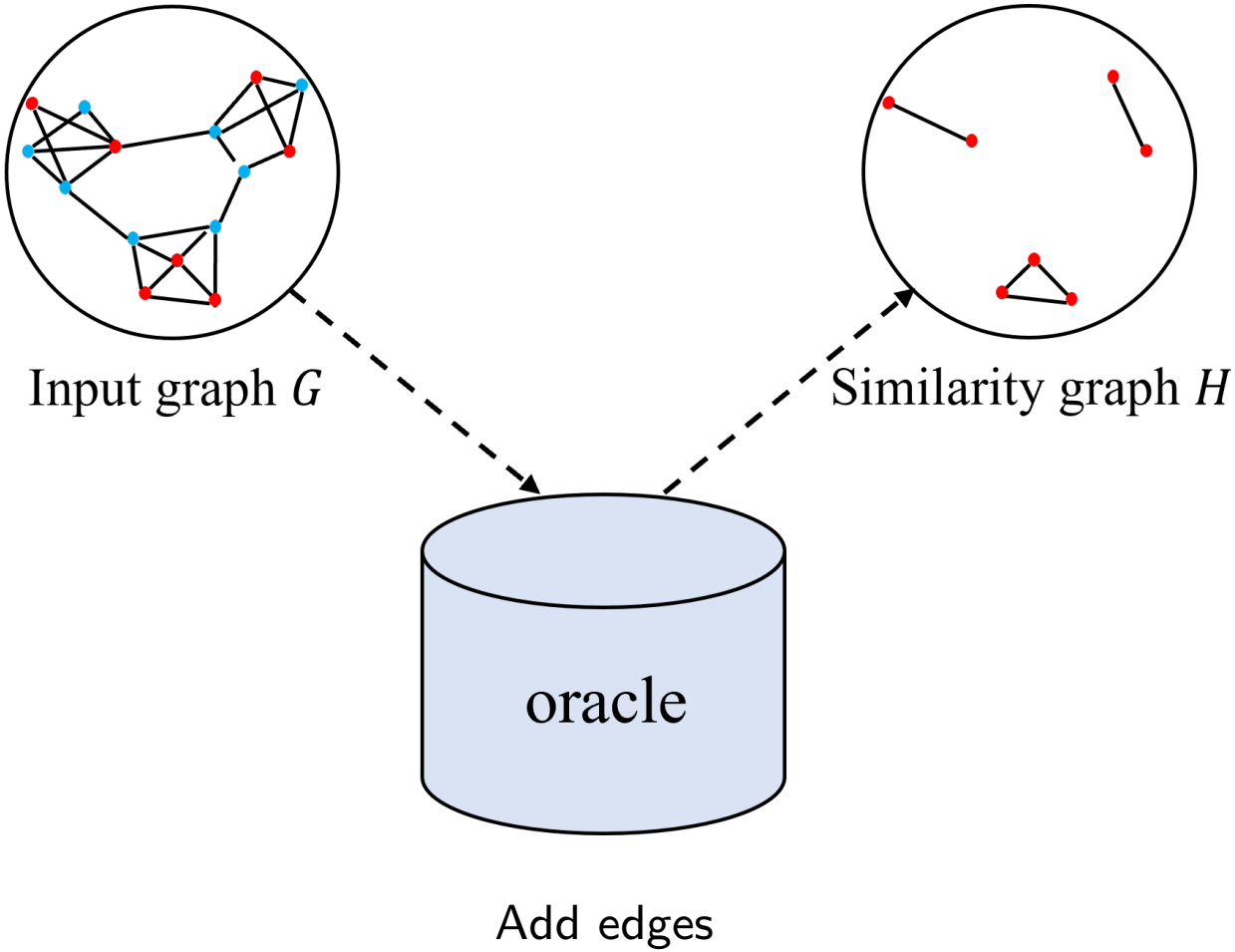
Intuition:

- $\langle f_x, f_y \rangle$ is large $\Rightarrow x, y$ in the same cluster
- $\langle f_x, f_y \rangle$ is close to 0 $\Rightarrow x, y$ in the different clusters

Our Oracle: Preprocessing Phase

step 3: for any pairs $x, y \in S$

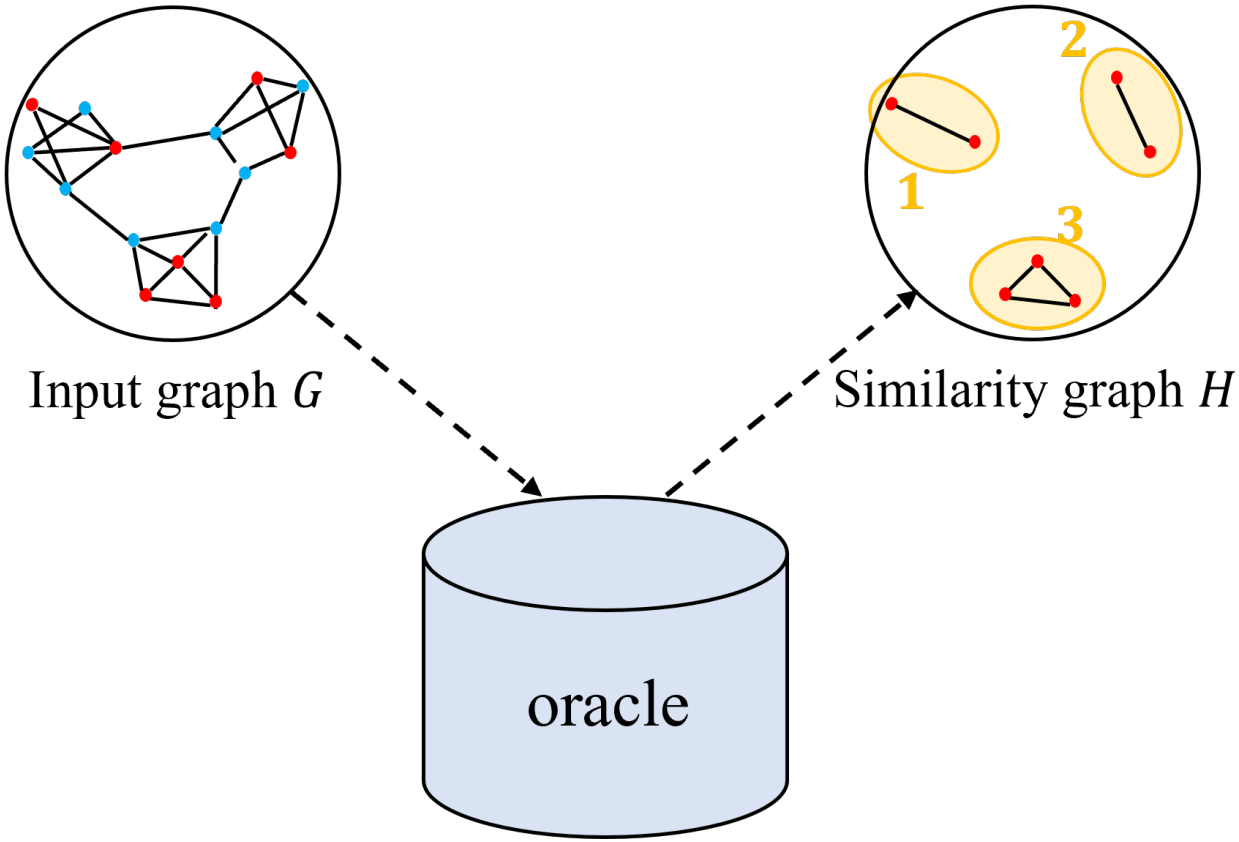
- if $\langle f_x, f_y \rangle_{\text{apx}} \geq \theta$, then add an edge (x, y) to H



Our Oracle: Preprocessing Phase

step 4: find connected components (CCs) in H

- if there are exactly k CCs, then label them with $1, \ldots, k$ $(S_1, \ldots, S_k)$

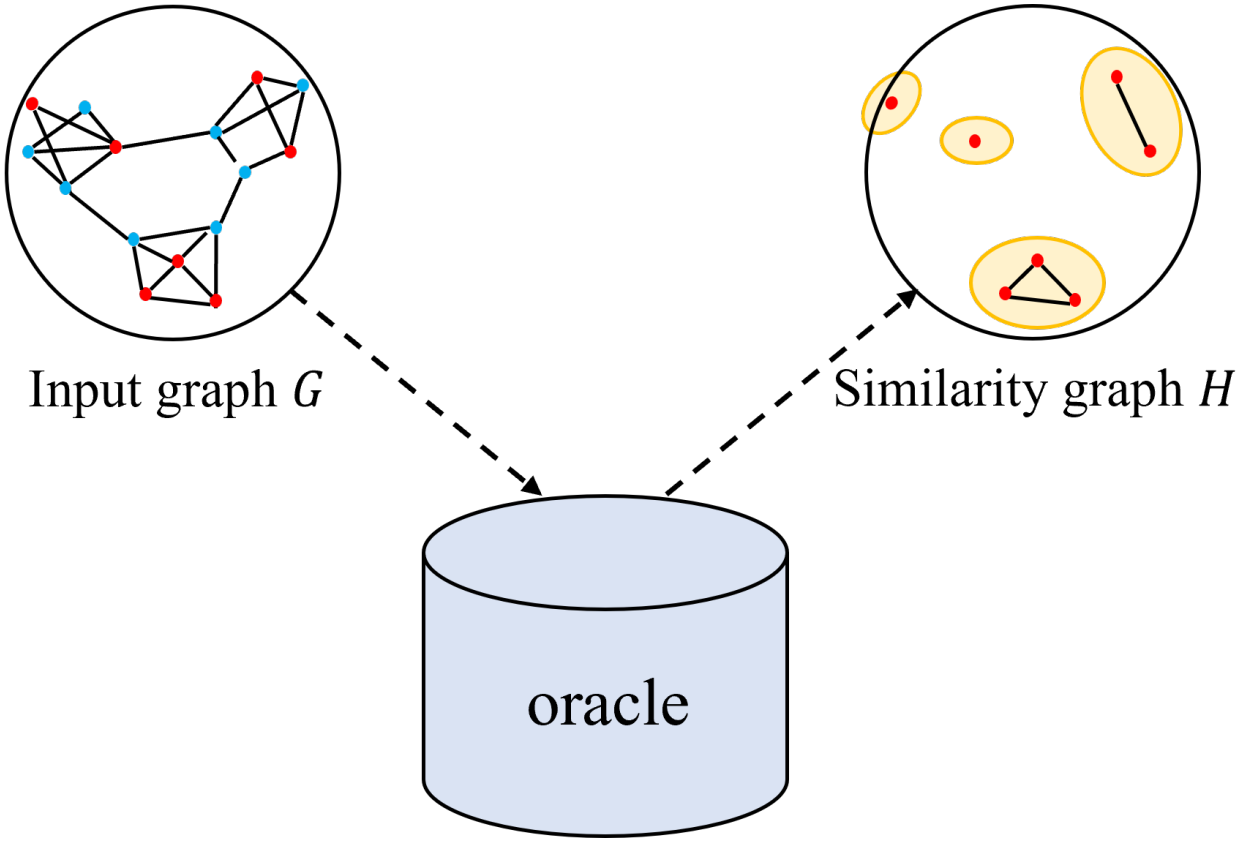


Find k CCs and label them

Our Oracle: Preprocessing Phase

step 4: find connected components (CCs) in H

- if $\#$ of CCs is not k, then fail



Our Oracle: Preprocessing Phase

By

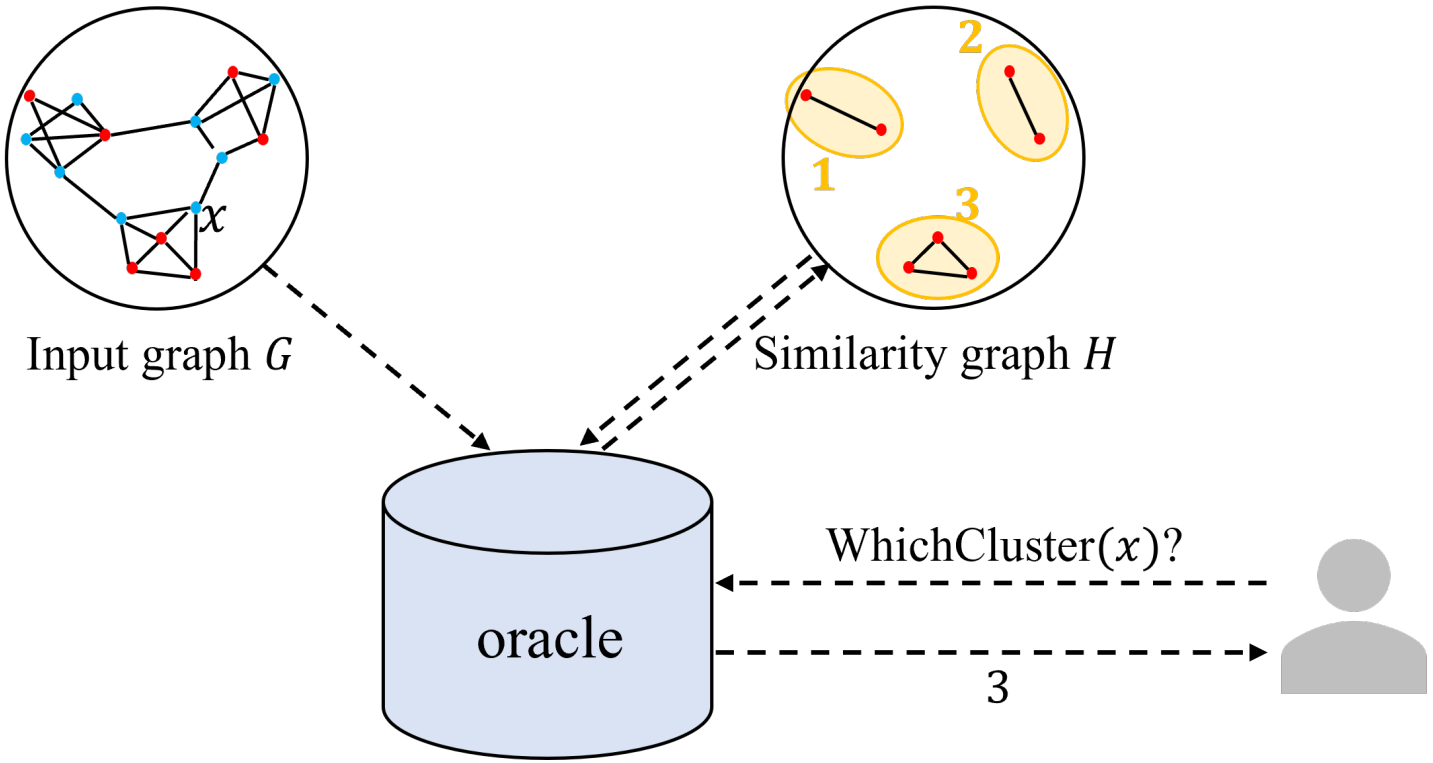
- sampling $O(\mathrm{poly}(k))$ vertices and
- setting θ appropriately,

we can prove that w.h.p. (i.e., 0.95), preprocessing phase will succeed.

Our Oracle: Query Phase (query x)

If there exists a unique S_i, such that for all $u \in S_i$, $\langle f_x, f_u \rangle_{\text{apx}} \geq \theta$

- return i
- else return a random index $\in \{1, \ldots, k\}$



WhichCluster(G, x)

Recall: Our Results

	[Pen20]	[GKL+21]	**this work**
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)\cdot\log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$	$\varepsilon \ll \frac{\varphi^2}{\mathrm{poly}(k)}$
preprocessing time	$\widetilde{O}\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k}{\varepsilon})\right)$	$\widetilde{O}\big(2^{\mathrm{poly}(\frac{k}{\varepsilon})}\cdot n^{1/2+O(\varepsilon)}\big)$	$\widetilde{O}\big(\mathrm{poly}(k)\cdot n^{1/2+O(\varepsilon)}\big)$
query time	$\widetilde{O}\left(\sqrt{n}\cdot\mathrm{poly}(\frac{k}{\varepsilon})\right)$	$\widetilde{O}\big(n^{1/2+O(\varepsilon)}\cdot \mathrm{poly}(\frac{k}{\varepsilon})\big)$	$\widetilde{O}\big(n^{1/2+O(\varepsilon)}\cdot \mathrm{poly}(k)\big)$
misclassification error (fraction)	$O\left(k\sqrt{\varepsilon}\right)$	$O\left(\log k\cdot\varepsilon\right)$ per cluster	$O\left(\mathrm{poly}(k)\cdot\varepsilon^{1/3}\right)$ per cluster

- preprocessing time: polynomial in k, better than exponential in k in [GKL+21]
- conductance gap and misclassification error: slightly worse than [GKL+21]

Outline

Experiments

Input graphs: generated by SBM

Experiments

Input graphs: generated by SBM

- n: $|V|$
- k: number of clusters
- q: $\Pr[(x, y) \in E]$, x, y in different clusters
- p: $\Pr[(x, y) \in E]$, x, y in same cluster

Experiments

Input graphs: generated by SBM

- n: $|V|$
- k: number of clusters
- q: $\Pr[(x, y) \in E]$, x, y in different clusters
- p: $\Pr[(x, y) \in E]$, x, y in same cluster

fix n, k, q: smaller p, more difficult to cluster

Experiments

Input graphs: generated by SBM

- can handle graphs with a smaller conductance gap than [CPS15][3]

p	0.02	0.025	0.03	0.035	0.04	0.05	0.06	0.07
error([CPS15])	-	0.6208	0.4970	0.1996	0.0829	0.0168	0.0030	0.0003
error(this work)	0.3887	0.0030	0.0004	0	0	0	0	0

[3]The oracle is implicit in [CPS15]. Czumaj A, Peng P, Sohler C. Testing cluster structure of graphs. STOC 2015.

Experiments

Input graphs: generated by SBM

- can handle graphs with a smaller conductance gap than [CPS15][3]

p	0.02	0.025	0.03	0.035	0.04	0.05	0.06	0.07
error([CPS15])	-	0.6208	0.4970	0.1996	0.0829	0.0168	0.0030	0.0003
error(this work)	0.3887	0.0030	0.0004	0	0	0	0	0

- robust against a few adversarial edge deletions in each cluster

delNum	0	25	32	40	45	50	55	60	65
error	0.0	0.00007	0.00007	0.00013	0.00047	0.00080	0.00080	0.00080	0.00087

[3]The oracle is implicit in [CPS15]. Czumaj A, Peng P, Sohler C. Testing cluster structure of graphs. STOC 2015.

Experiments

Input graphs: generated by SBM

- can handle graphs with a smaller conductance gap than [CPS15][3]

p	0.02	0.025	0.03	0.035	0.04	0.05	0.06	0.07
error([CPS15])	-	0.6208	0.4970	0.1996	0.0829	0.0168	0.0030	0.0003
error(this work)	0.3887	0.0030	0.0004	0	0	0	0	0

- robust against a few adversarial edge deletions in each cluster

delNum	0	25	32	40	45	50	55	60	65
error	0.0	0.00007	0.00007	0.00013	0.00047	0.00080	0.00080	0.00080	0.00087

Experimental results are consistent with our theoretical results.

[3]The oracle is implicit in [CPS15]. Czumaj A, Peng P, Sohler C. Testing cluster structure of graphs. STOC 2015.

Conclusions

Summary

There exists a spectral clustering oracle that takes a $\left(k, \varphi, O(\frac{\varphi^2}{\mathrm{poly}(k)})\right)$-clusterable graph as input and has

- preprocessing time: $O\left(n^{1/2+O(\varepsilon)} \cdot \mathrm{poly}(k \log n)\right)$
- query time: $O\left(n^{1/2+O(\varepsilon)} \cdot \mathrm{poly}(k \log n)\right)$
- misclassification error: $O\left(\mathrm{poly}(k) \cdot \varepsilon^{1/3}\right)$ per cluster

Conclusions

Summary

There exists a spectral clustering oracle that takes a $\left(k, \varphi, O(\frac{\varphi^2}{\mathrm{poly}(k)})\right)$-clusterable graph as input and has

- preprocessing time: $O\left(n^{1/2+O(\varepsilon)} \cdot \mathrm{poly}(k \log n)\right)$
- query time: $O\left(n^{1/2+O(\varepsilon)} \cdot \mathrm{poly}(k \log n)\right)$
- misclassification error: $O\left(\mathrm{poly}(k) \cdot \varepsilon^{1/3}\right)$ per cluster

Open Question

- How to get a clustering oracle with both $O(\frac{1}{\log k})$ conductance gap and $\widetilde{O}\left(n^{1/2+O(\varepsilon)} \cdot \mathrm{poly}(k)\right)$ preprocessing time?

Thanks!