Optimal Clustering with Noisy Queries via Multi-Armed Bandit

Jinghui Xia[1], Zengfeng Huang[12]

[1]School of Data Science, Fudan University
[2]Shanghai Key Lab of Intelligent Information Processing

The 39th International Conference on Machine Learning, 2022

Clustering with Noisy Queries

- A.k.a. clustering with a faulty oracle.

Clustering with Noisy Queries

- A.k.a. clustering with a faulty oracle.
- A set V of n vertices (or items).

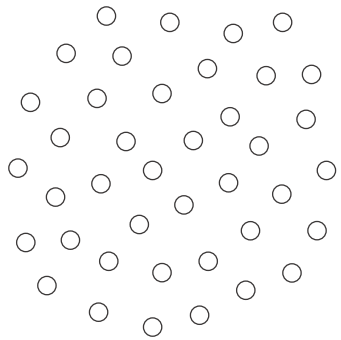

Figure: $n = 40$ vertices.

Clustering with Noisy Queries

- A.k.a. clustering with a faulty oracle.
- A set V of n vertices (or items).
- Its hidden partition $\{V_1, ..., V_k\}$.

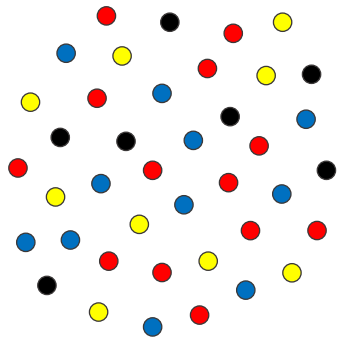

Figure: $k = 4$ hidden clusters.

Clustering with Noisy Queries

- A.k.a. clustering with a faulty oracle.
- A set V of n vertices (or items).
- Its hidden partition $\{V_1, ..., V_k\}$.

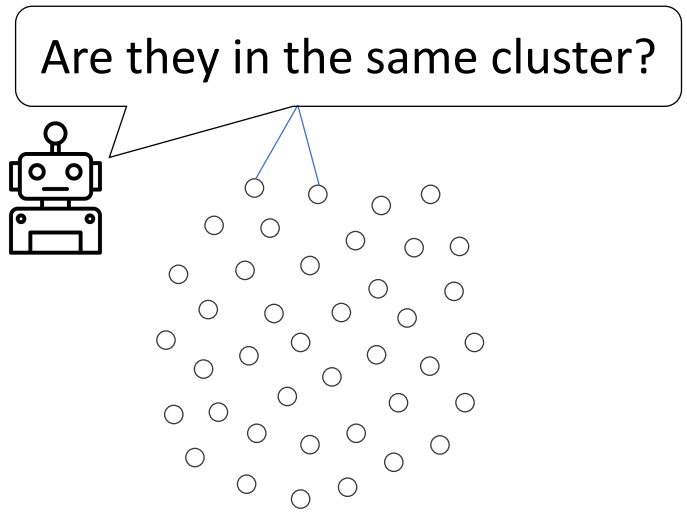

Figure: The algorithm makes a query.

Clustering with Noisy Queries

- A.k.a. clustering with a faulty oracle.
- A set V of n vertices (or items).
- Its hidden partition $\{V_1, ..., V_k\}$.
- A (faulty) oracle that answers whether two vertices belong to the same cluster.

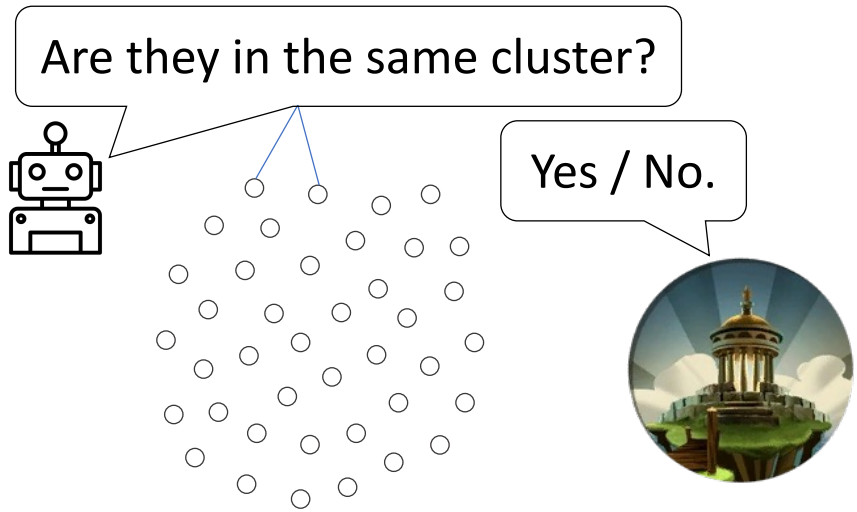

Figure: The oracle answers the query.

Clustering with Noisy Queries

- A.k.a. clustering with a faulty oracle.
- A set V of n vertices (or items).
- Its hidden partition $\{V_1, ..., V_k\}$.
- A (faulty) oracle that answers whether two vertices belong to the same cluster.
- $\mathbb{P}(\text{answer is correct}) = \frac{1}{2} + \frac{\delta}{2}$.

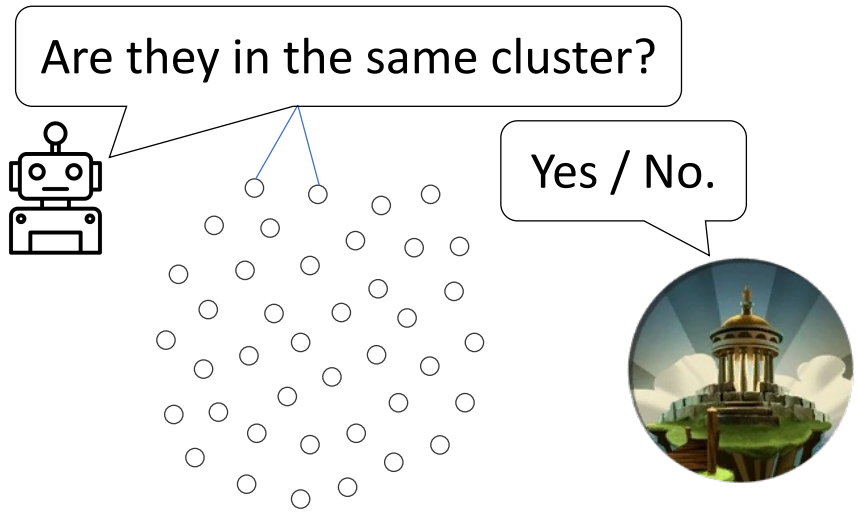

Figure: The oracle answers the query.

Clustering with Noisy Queries

- A.k.a. clustering with a faulty oracle.
- A set V of n vertices (or items).
- Its hidden partition $\{V_1, ..., V_k\}$.
- A (faulty) oracle that answers whether two vertices belong to the same cluster.
- $\mathbb{P}(\text{answer is correct}) = \frac{1}{2} + \frac{\delta}{2}$.
- Goal: recover the hidden clusters with minimum number of queries.

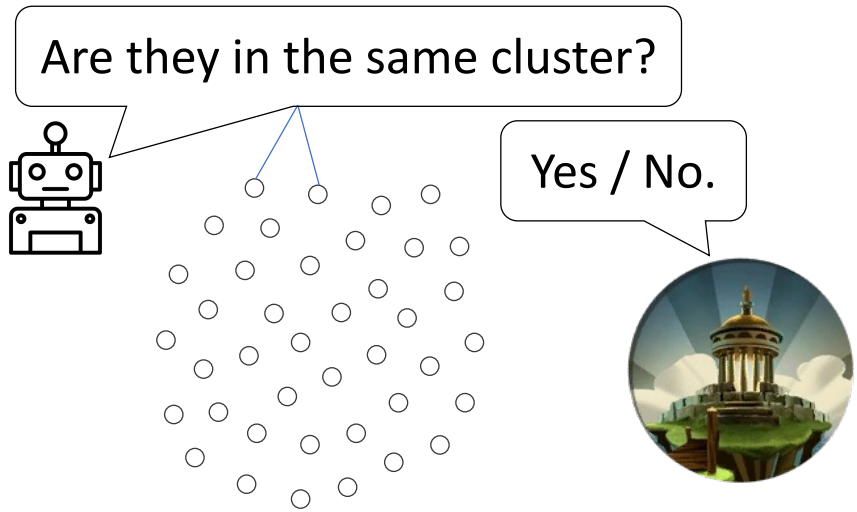

Figure: The oracle answers the query.

Clustering with Noisy Queries (cont.)

Define function $\tau : V \times V \to \pm 1$ s.t.

$$\tau(u, v) = \begin{cases} +1, & V_{c(u)} = V_{c(v)}, \\ -1, & V_{c(u)} \neq V_{c(v)}. \end{cases}$$

The oracle is equivalent to a noisy version of τ, denoted as $\tilde{\tau}$:

$$\tilde{\tau} = \sigma_{u,v} \cdot \tau(u, v),$$

where $\sigma_{u,v}$ is a $\{\pm 1\}$ random variable attaining $+1$ with probability $\frac{1}{2} + \frac{\delta}{2}$, and -1 with probability $\frac{1}{2} - \frac{\delta}{2}$.

We assume that $\sigma_{u,v}$ is independent across different pairs and the oracle always returns the same answer for queries to the same pair.

Previous Results

Paper	#Clusters	Query Complexity	Time Complexity
[MS'17][1]	k	$O(\frac{nk \log n}{\delta^2})$	Quasi-polynomial
		$O(\frac{nk \log n}{\delta^2} + \frac{k^5 \log^2 n}{\delta^8})$	Polynomial
		$\Omega(\frac{nk}{\delta^2})$	
[LMT'20][2]	2	$O(\frac{n \log n}{\delta^2} + \frac{\log^2 n}{\delta^6})$	Polynomial
[PZ'21][3]	k	$O(\frac{nk \log n}{\delta^2} + \frac{k^{10} \log^2 n}{\delta^4})$	Polynomial
This work	k	$O(\frac{n(k+\log n)}{\delta^2} + \frac{k^8 \log^3 n}{\delta^4})$	Polynomial
		$\Omega(\frac{n \log n}{\delta^2})$	

[1] Mazumdar, A. and Saha, B. Clustering with noisy queries. NeurIPS, 2017.

[2] Larsen, K. G., Mitzenmacher, M., and Tsourakakis, C. Clustering with a faulty oracle. WWW, 2020.

[3] Peng, P. and Zhang, J. Towards a query-optimal and time-efficient algorithm for clustering with a faulty oracle. COLT, 2021

Previous Results

Paper	#Clusters	Query Complexity	Time Complexity
[MS'17][1]	k	$O(\frac{nk \log n}{\delta^2})$	Quasi-polynomial
		$O(\frac{nk \log n}{\delta^2} + \frac{k^5 \log^2 n}{\delta^8})$	Polynomial
		$\Omega(\frac{nk}{\delta^2})$	
[LMT'20][2]	2	$O(\frac{n \log n}{\delta^2} + \frac{\log^2 n}{\delta^6})$	Polynomial
[PZ'21][3]	k	$O(\frac{nk \log n}{\delta^2} + \frac{k^{10} \log^2 n}{\delta^4})$	Polynomial
This work	k	$O(\frac{n(k+\log n)}{\delta^2} + \frac{k^8 \log^3 n}{\delta^4})$	Polynomial
		$\Omega(\frac{n \log n}{\delta^2})$	

[1] Mazumdar, A. and Saha, B. Clustering with noisy queries. NeurIPS, 2017.

[2] Larsen, K. G., Mitzenmacher, M., and Tsourakakis, C. Clustering with a faulty oracle. WWW, 2020.

[3] Peng, P. and Zhang, J. Towards a query-optimal and time-efficient algorithm for clustering with a faulty oracle. COLT, 2021

The Two-Step Framework

1. **Find sub-clusters (or biased sets[4]).**
   E.g.[PZ'21].
2. **Cluster the vertices.**
   Compare each vertex to the sub-clusters (or biased sets) to find the true cluster. Require $O(\frac{nk \log n}{\delta^2})$ queries in previous works.

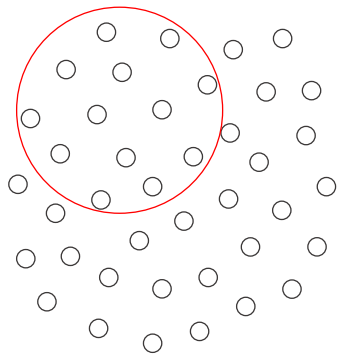

Figure: Sample a subset $T \subset V$ of certain size.

[4] A set of vertices $B \subseteq V$ is called (η, C)-biased if at least $1/2 + \eta$ fraction of the vertices in B belong to a true cluster C, i.e. $|B \cap C| \geq (1/2 + \eta) \cdot |B|$.

The Two-Step Framework

1. **Find sub-clusters (or biased sets[4]).**
   E.g.[PZ'21].
2. **Cluster the vertices.**
   Compare each vertex to the sub-clusters (or biased sets) to find the true cluster. Require $O(\frac{nk \log n}{\delta^2})$ queries in previous works.

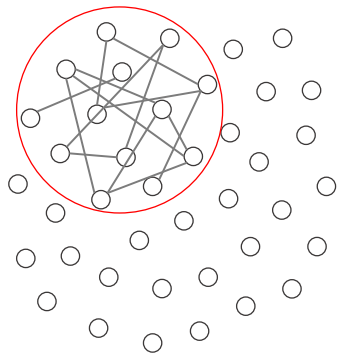

Figure: Query all pairs of vertices in T to construct an SBM-type graph.

[4] A set of vertices $B \subseteq V$ is called (η, C)-biased if at least $1/2 + \eta$ fraction of the vertices in B belong to a true cluster C, i.e. $|B \cap C| \geq (1/2 + \eta) \cdot |B|$.

The Two-Step Framework

1. **Find sub-clusters (or biased sets[4]).**
   E.g.[PZ'21].
2. **Cluster the vertices.**
   Compare each vertex to the sub-clusters (or biased sets) to find the true cluster. Require $O(\frac{nk \log n}{\delta^2})$ queries in previous works.

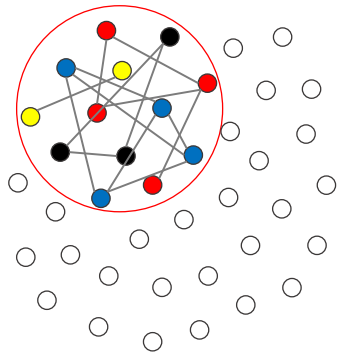

Figure: Partition the graph to get sub-clusters (or biased sets).

[4] A set of vertices $B \subseteq V$ is called (η, C)-biased if at least $1/2 + \eta$ fraction of the vertices in B belong to a true cluster C, i.e. $|B \cap C| \geq (1/2 + \eta) \cdot |B|$.

The Two-Step Framework

1. **Find sub-clusters (or biased sets[4]).**
   E.g.[PZ'21].
2. **Cluster the vertices.**
   Compare each vertex to the sub-clusters (or biased sets) to find the true cluster. Require $O(\frac{nk \log n}{\delta^2})$ queries in previous works.

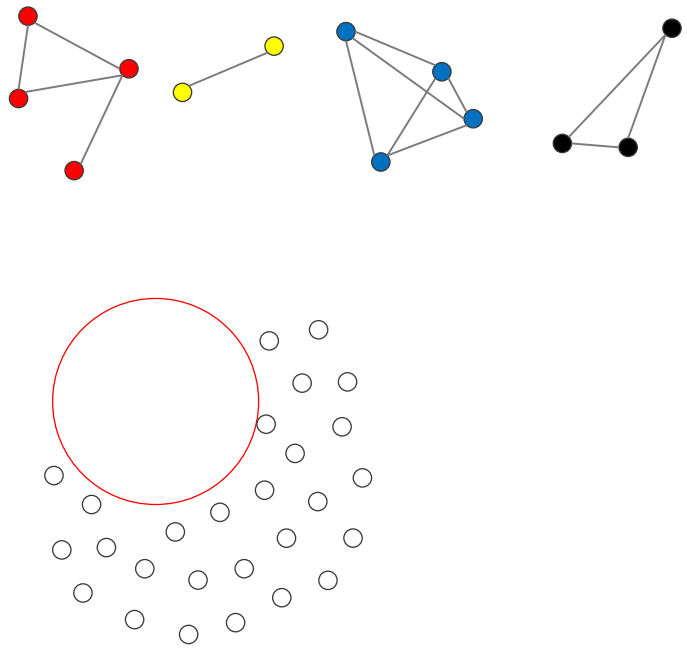

Figure: Compare each vertex to the sub-clusters.

The Two-Step Framework

1. **Find sub-clusters (or biased sets[4]).**
   E.g.[PZ'21].
2. **Cluster the vertices.**
   Consider each sub-cluster as an arm. Identifying the cluster of each vertex can be reduced to a best arm identification problem. $O(\frac{k}{\delta^2}\log\frac{1}{\alpha})$ queries for each vertex to identify the true cluster with probability at least $1-\alpha$.

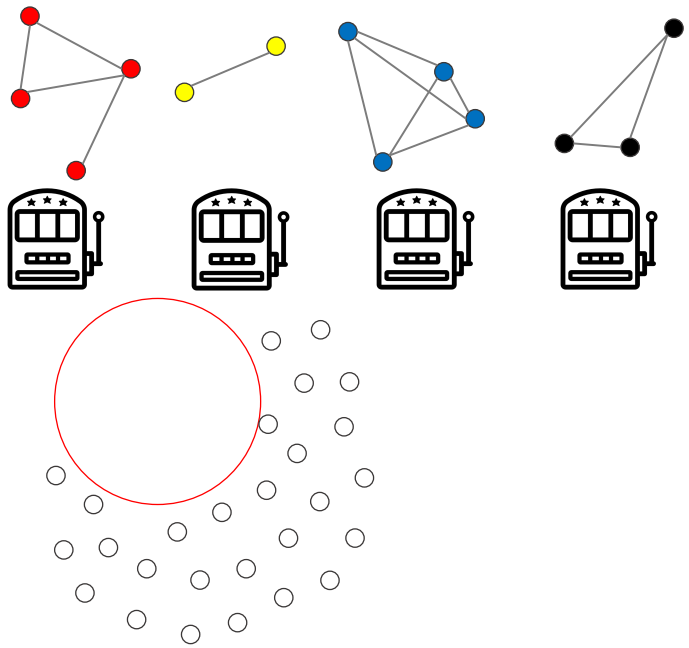

Figure: Consider each sub-cluster as an arm.

Upper Bound

Theorem

There exists a polynomial time algorithm that recovers all the clusters of size $\Omega(\frac{k^4 \log n}{\delta^2})$ *with success probability* $1 - o_n(1)$. *The total number of queries to the faulty oracle is*

$$O\left(\frac{n(k+\log n)}{\delta^2} + \frac{k^8 \log^3 n}{\delta^4}\right).$$

Best Arm Identification with Failure

Define a variant of the best arm identification problem, denoted as (δ, α)-BAIF, where we allow the algorithm to fail (not return any arm) with probability at most $1/8$.

We prove that sample complexity lower bound for (δ, α)-BAIF is still $\Omega(\frac{1}{\delta^2} \log \frac{1}{\alpha})$, same as regular BAI.

Lower Bound

By some reduction from BAIF to clustering with noisy queries, we can prove the following lower bound.

Theorem

For $k \geq 2$, any (randomized) algorithm must make $\Omega(\frac{n \log n}{\delta^2})$ expected number of queries to recover the correct clusters with probability at least $\frac{7}{8}$.

Thus combined with previous $\Omega(\frac{nk}{\delta^2})$ lower bound, we obtain matching upper and lower bounds $\Theta(\frac{n(k+\log n)}{\delta^2})$.

Summary

- A novel polynomial time algorithm for clustering with noisy queries problem.
- Query complexity upper bound $O(\frac{n(k+\log n)}{\delta^2} + \frac{k^8 \log^3 n}{\delta^4})$.
- Query complexity lower bound $\Omega(\frac{n \log n}{\delta^2})$.

For details of our work, please refer to our paper

Optimal Clustering with Noisy Queries via Multi-Armed Bandit