Consistent k-Clustering for General Metrics

Hendrik Fichtenberger, Silvio Lattanzi, Ashkan Norouzi-Fard, Ola Svensson

SODA 2021

Consistent Clustering

input model: stream of point insertions $\langle p_1, p_2, \ldots, p_n \rangle \rightarrow$ sets $P_1, P_2, \ldots$

Consistent Clustering

input model: stream of point insertions $\langle p_1, p_2, \ldots, p_n \rangle \rightarrow$ sets $P_1, P_2, \ldots$

Consistent Clustering

input model: stream of point insertions $\langle p_1, p_2, \dots, p_n \rangle \rightarrow$ sets $P_1, P_2, \dots$

Consistent Clustering

input model: stream of point insertions $\langle p_1, p_2, \ldots, p_n \rangle \rightarrow$ sets $P_1, P_2, \ldots$

classic objective: maintain k centers that minimize clustering cost:

$$\underset{\substack{C_i \subset P_i \\ |C_i| = k}}{\arg\min} \sum_{p \in P_i} d(p, C_i) \qquad \forall i \in [n]$$

↑ general metric

Consistent Clustering

input model: stream of point insertions $\langle p_1, p_2, \ldots, p_n \rangle \rightarrow$ sets $P_1, P_2, \ldots$

classic objective: maintain k centers that minimize clustering cost:

$$\operatorname*{arg\,min}_{\substack{C_i \subset P_i \\ |C_i| = k}} \sum_{p \in P_i} d(p, C_i) \qquad \forall i \in [n]$$

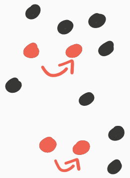
Consistent Clustering

input model: stream of point insertions $\langle p_1, p_2, \dots, p_n \rangle \to$ sets $P_1, P_2, \dots$

classic objective: maintain k centers that minimize clustering cost:

$$\underset{\substack{C_i \subset P_i \\ |C_i| = k}}{\arg\min} \sum_{p \in P_i} d(p, C_i) \qquad \forall i \in [n]$$

Consistent Clustering

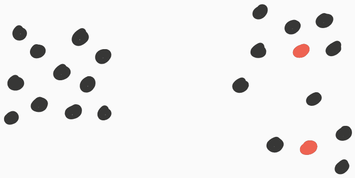


input model: stream of point insertions $\langle p_1, p_2, \dots, p_n \rangle \rightarrow$ sets $P_1, P_2, \dots$

classic objective: maintain k centers that minimize clustering cost:

$$\underset{\substack{C_i \subset P_i \\ |C_i| = k}}{\arg\min} \sum_{p \in P_i} d(p, C_i) \qquad \forall i \in [n]$$

Consistent Clustering

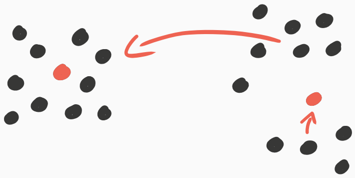


input model: stream of point insertions $\langle p_1, p_2, \ldots, p_n \rangle \rightarrow$ sets $P_1, P_2, \ldots$

classic objective: maintain k centers that minimize clustering cost:

$$\underset{\substack{C_i \subset P_i \\ |C_i| = k}}{\arg\min} \sum_{p \in P_i} d(p, C_i) \qquad \forall i \in [n]$$

Consistent Clustering

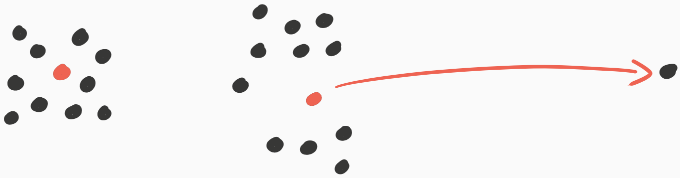


input model: stream of point insertions $\langle p_1, p_2, \ldots, p_n \rangle \rightarrow$ sets $P_1, P_2, \ldots$

classic objective: maintain k centers that minimize clustering cost:

$$\underset{\substack{C_i \subset P_i \\ |C_i| = k}}{\arg\min} \sum_{p \in P_i} d(p, C_i) \qquad \forall i \in [n]$$

Consistent Clustering



input model: stream of point insertions $\langle p_1, p_2, \ldots, p_n \rangle \rightarrow$ sets $P_1, P_2, \ldots$

classic objective: maintain k centers that minimize clustering cost:

$$\operatorname*{arg\,min}_{\substack{C_i \subset P_i \\ |C_i| = k}} \sum_{p \in P_i} d(p, C_i) \qquad \forall i \in [n]$$

Consistent Clustering

input model: stream of point insertions $\langle p_1, p_2, \dots, p_n \rangle \rightarrow$ sets $P_1, P_2, \dots$

classic objective: maintain k centers that minimize clustering cost:

$$\arg\min_{\substack{C_i \subset P_i \\ |C_i| = k}} \sum_{p \in P_i} d(p, C_i) \qquad \forall i \in [n]$$

Consistent Clustering

input model: stream of point insertions $\langle p_1, p_2, \ldots, p_n \rangle \rightarrow$ sets $P_1, P_2, \ldots$

classic objective: maintain k centers that minimize clustering cost:

$$\arg\min_{\substack{C_i \subset P_i \\ |C_i| = k}} \sum_{p \in P_i} d(p, C_i) \qquad \forall i \in [n]$$

consistency objective: approximate solution with as few center swaps as possible

$$\min \sum_{i \in [n]} |C_i \setminus C_{i-1}|$$

Consistent Clustering

Main Result

An insertion-only streaming algorithm that maintains an $O(1)$-approximate k-median solution and swaps at most $O(k \cdot \text{polylog}(n, \Delta))$ centers during the entire execution.

spread of metric

total number of points

number of centers

Lattanzi, Vassilvitskii, ICML'17; Guo et al., preprint '20; Cohen-Addad et al. NeuRIPS'19; Guo et al., APPROX'20

Consistent Clustering

Main Result

An insertion-only streaming algorithm that maintains an $O(1)$-approximate k-median solution and swaps at most $O(k \cdot \mathrm{polylog}(n, \Delta))$ centers during the entire execution.

Lower Bound [LV17]

$\Omega(k \log(n\Delta))$ swaps are neccessary even for offline setting.

Lattanzi, Vassilvitskii, ICML'17; Guo et al., preprint '20; Cohen-Addad et al. NeuRIPS'19; Guo et al., APPROX'20

Consistent Clustering

Main Result

An insertion-only streaming algorithm that maintains an $O(1)$-approximate k-median solution and swaps at most $O(k \cdot \text{polylog}(n, \Delta))$ centers during the entire execution.

Lower Bound [LV17]

$\Omega(k \log(n\Delta))$ swaps are neccessary even for offline setting.

Related Work

Previous result [LV17]: $O(k^2 \log(n\Delta)^4)$

Deterministic with outliers [GKSX20]: $O(k^2 \log(n\Delta)^2)$

Dynamic consistent clustering [CHPSS19, GKLX20, ...]

Lattanzi, Vassilvitskii, ICML'17; Guo et al., preprint '20; Cohen-Addad et al. NeuRIPS'19; Guo et al., APPROX'20

Consistent Clustering

Main Result

An insertion-only streaming algorithm that maintains an $O(1)$-approximate k-median solution and swaps at most $O(k \cdot \text{polylog}(n, \Delta))$ centers during the entire execution.

essentially tight

Lower Bound [LV17]

$\Omega(k \log(n\Delta))$ swaps are neccessary even for offline setting.

Related Work

Previous result [LV17]: $O(k^2 \log(n\Delta)^4)$

Deterministic with outliers [GKSX20]: $O(k^2 \log(n\Delta)^2)$

Dynamic consistent clustering [CHPSS19, GKLX20, ...]

Consistent Clustering

assumption: $\mathrm{OPT}_{\mathrm{guess}} \leq \mathrm{OPT}_k \leq c \cdot \mathrm{OPT}_{\mathrm{guess}}$

↖ recompute solution $O(\log(\Delta n))$ times from scratch

Consistent Clustering

assumption: $\text{OPT}_{\text{guess}} \leq \text{OPT}_k \leq c \cdot \text{OPT}_{\text{guess}}$

recompute solution $O(\log(\Delta n))$ times from scratch

problem: given a stream of $m = O(k \cdot \text{polylog}(n))$ weighted points and $O(n)$ weight updates, maintain solution with at most $O(m)$ swaps

[LV17]: $O(k \cdot m)$

Consistent Clustering

assumption: $\text{OPT}_{\text{guess}} \leq \text{OPT}_k \leq c \cdot \text{OPT}_{\text{guess}}$

recompute solution $O(\log(\Delta n))$ times from scratch

problem: given a stream of $m = O(k \cdot \text{polylog}(n))$ weighted points and $O(n)$ weight updates, maintain solution with at most $O(m)$ swaps

[LV17]: $O(k \cdot m)$

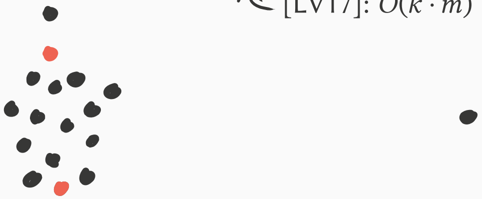
Consistent Clustering

assumption: $\mathrm{OPT}_{\mathrm{guess}} \leq \mathrm{OPT}_k \leq c \cdot \mathrm{OPT}_{\mathrm{guess}}$

recompute solution $O(\log(\Delta n))$ times from scratch

problem: given a stream of $m = O(k \cdot \mathrm{polylog}(n))$ weighted points and $O(n)$ weight updates, maintain solution with at most $O(m)$ swaps

[LV17]: $O(k \cdot m)$



Consistent Clustering

assumption: $\mathrm{OPT}_{\text{guess}} \leq \mathrm{OPT}_k \leq c \cdot \mathrm{OPT}_{\text{guess}}$

recompute solution $O(\log(\Delta n))$ times from scratch

problem: given a stream of $m = O(k \cdot \mathrm{polylog}(n))$ weighted points and $O(n)$ weight updates, maintain solution with at most $O(m)$ swaps

[LV17]: $O(k \cdot m)$

Consistent Clustering

assumption: $\mathrm{OPT}_{\text{guess}} \leq \mathrm{OPT}_k \leq c \cdot \mathrm{OPT}_{\text{guess}}$

↖ recompute solution $O(\log(\Delta n))$ times from scratch

problem: given a stream of $m = O(k \cdot \mathrm{polylog}(n))$ weighted points and $O(n)$ weight updates, maintain solution with at most $O(m)$ swaps

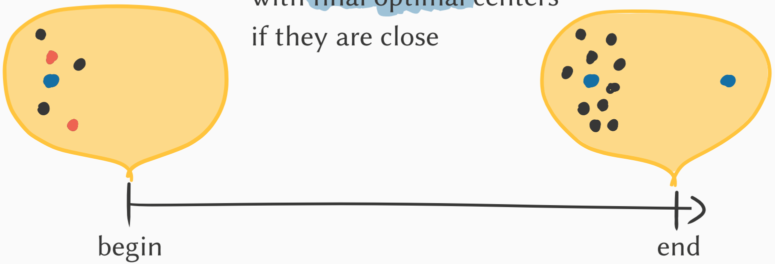
↖ [LV17]: $O(k \cdot m)$

can only change O(1) centers per insertion!

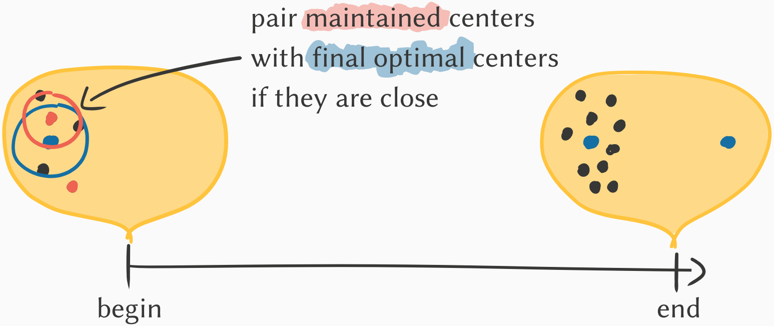
Algorithm

Algorithm

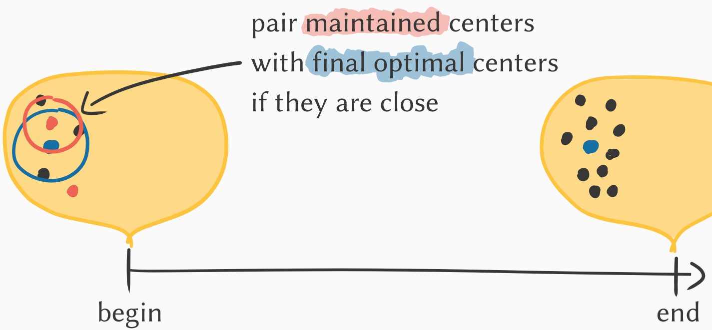
pair maintained centers
with final optimal centers
if they are close



Algorithm


pair maintained centers
with final optimal centers
if they are close
begin
end

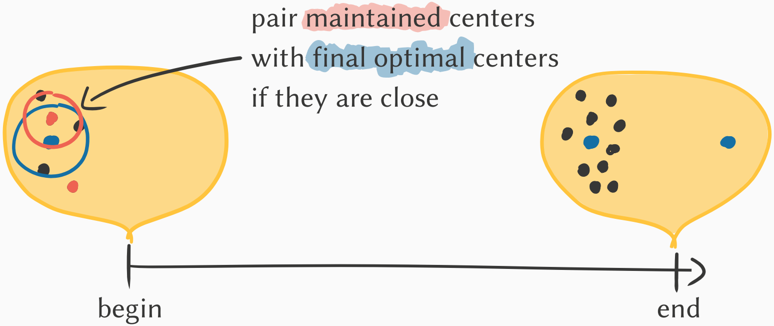
Algorithm



algorithm:

- remove the ℓ unpaired centers but maintain approximation
- open $\ell + 1$ inserted points as centers ($\rightarrow k + 1$ centers)
- swap $O(\ell)$ centers to obtain a good solution with k centers

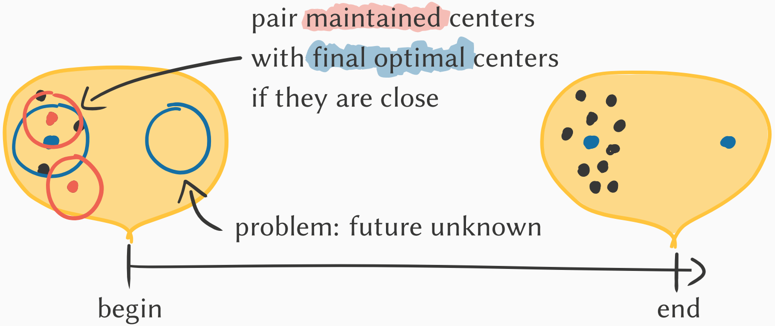
Algorithm



algorithm: "epoch"

- remove the ℓ unpaired centers but maintain approximation
- open $\ell + 1$ inserted points as centers ($\rightarrow k + 1$ centers)
- swap $O(\ell)$ centers to obtain a good solution with k centers
- repeat (until $\mathrm{OPT}_k > c \cdot \mathrm{OPT}_{\mathrm{guess}}$)

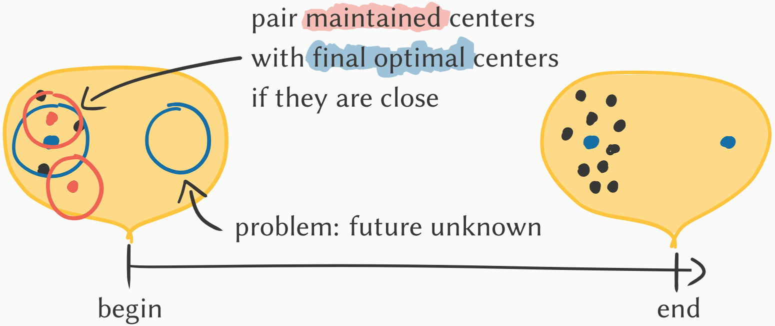
Algorithm



algorithm: "epoch"

- remove the ℓ unpaired centers but maintain approximation
- open $\ell + 1$ inserted points as centers ($\rightarrow k + 1$ centers)
- swap $O(\ell)$ centers to obtain a good solution with k centers
- repeat (until $\mathrm{OPT}_k > c \cdot \mathrm{OPT}_{\mathrm{guess}}$)

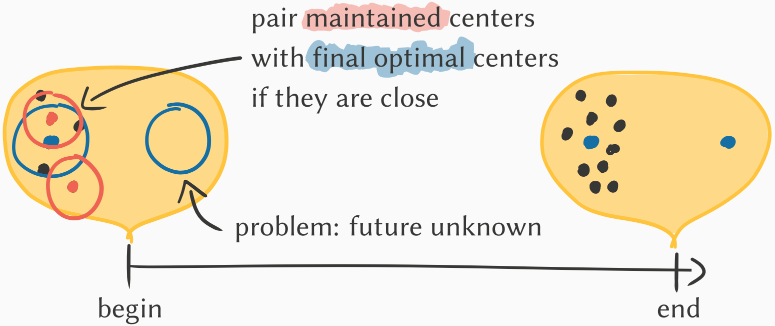
Algorithm



algorithm: "epoch"

- remove ~~the~~ ℓ unpaired centers but maintain approximation
- open $\ell + 1$ inserted points as centers ($\rightarrow k + 1$ centers)
- swap $O(\ell)$ centers to obtain a good solution with k centers
- repeat (until $\mathrm{OPT}_k > c \cdot \mathrm{OPT}_{\text{guess}}$)

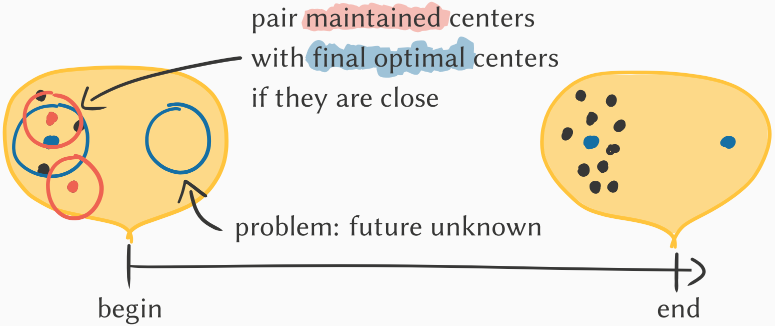
Algorithm



algorithm: "epoch"

- remove $\ell = \Theta(\#\text{unpaired centers})$ centers but maintain approximation
- open $\ell + 1$ inserted points as centers ($\to k + 1$ centers)
- swap $O(\ell)$ centers to obtain a good solution with k centers
- repeat (until $\text{OPT}_k > c \cdot \text{OPT}_{\text{guess}}$)

Algorithm

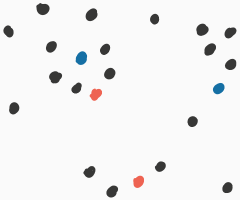


algorithm: "epoch"

- remove $\ell = \Theta(\#\text{unpaired centers})$ centers but maintain approximation
- open $\ell + 1$ inserted points as [needs proof] 1 centers)
- swap $O(\ell)$ centers to obtain a good solution with k centers
- repeat (until $\text{OPT}_k > c \cdot \text{OPT}_{\text{guess}}$)

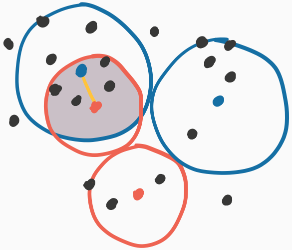
Key Elements of the Analysis

well-separated pairs



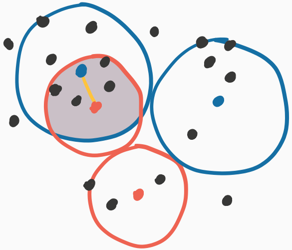
Key Elements of the Analysis

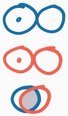
well-separated pairs



Key Elements of the Analysis

well-separated pairs





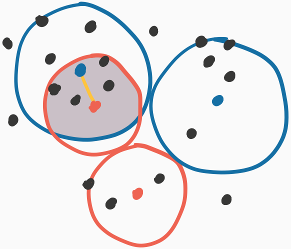
isolated optimal center and

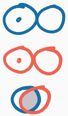
isolated maintained center

that are close to each other

Key Elements of the Analysis

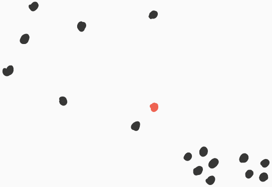
well-separated pairs





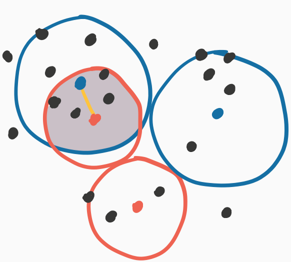
isolated optimal center and
isolated maintained center
that are close to each other

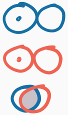
robust centers



Key Elements of the Analysis

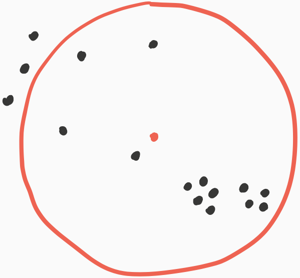
well-separated pairs





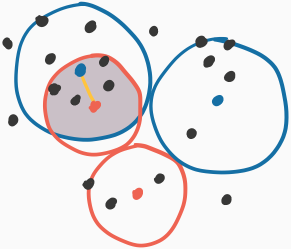
isolated optimal center and
isolated maintained center
that are close to each other

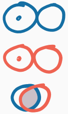
robust centers



Key Elements of the Analysis

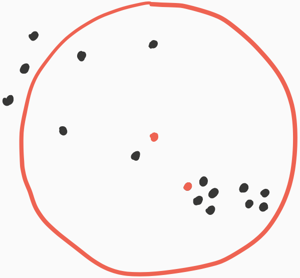
well-separated pairs





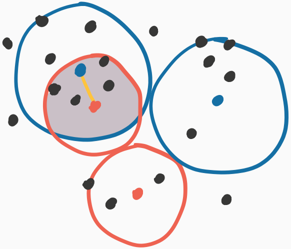
isolated optimal center and
isolated maintained center
that are close to each other

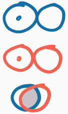
robust centers



Key Elements of the Analysis

well-separated pairs



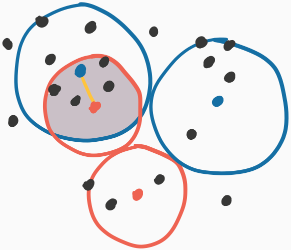


isolated optimal center and
isolated maintained center
that are close to each other

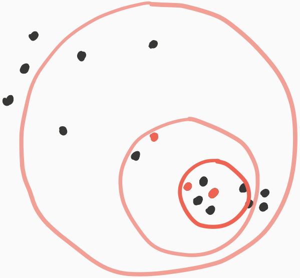
robust centers

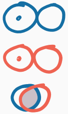
Key Elements of the Analysis

well-separated pairs



robust centers

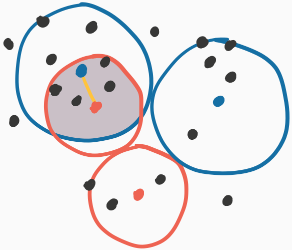


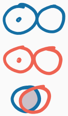


isolated optimal center and
isolated maintained center
that are close to each other

Key Elements of the Analysis

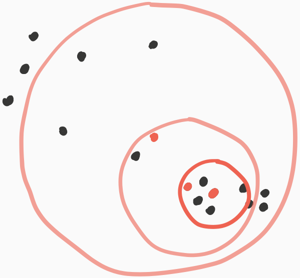
well-separated pairs





isolated optimal center and
isolated maintained center
that are close to each other

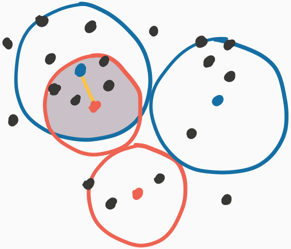
robust centers

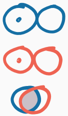


robustify each center
against future insertions
at the cluster's border

Key Elements of the Analysis

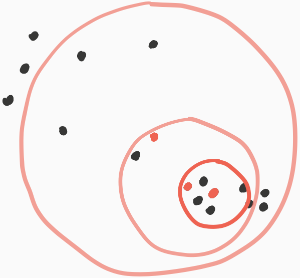
well-separated pairs





isolated optimal center and
isolated maintained center
that are close to each other

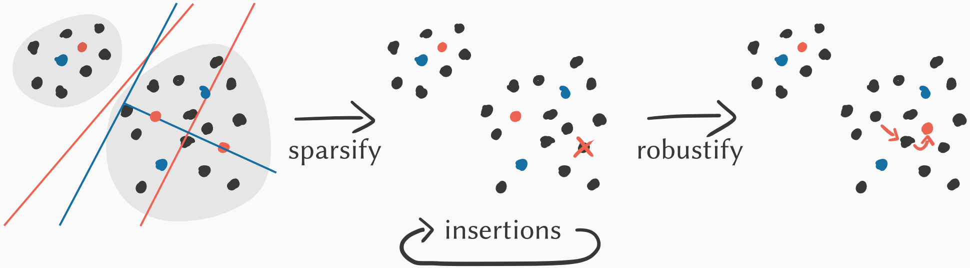
robust centers



robustify each center
against future insertions
at the cluster's border

well-separated, robust centers are approximately optimal
not well-separated centers can be removed

Summary



- $O(1)$-approximate k-clustering with $O(k \cdot \text{polylog}(n, \Delta))$ consistency
- tight up to polylogarithmic factors (even for offline setting)
- analysis exploits structural properties, algorithm is based on epochs
- is there a simpler approach, e.g., by local search?