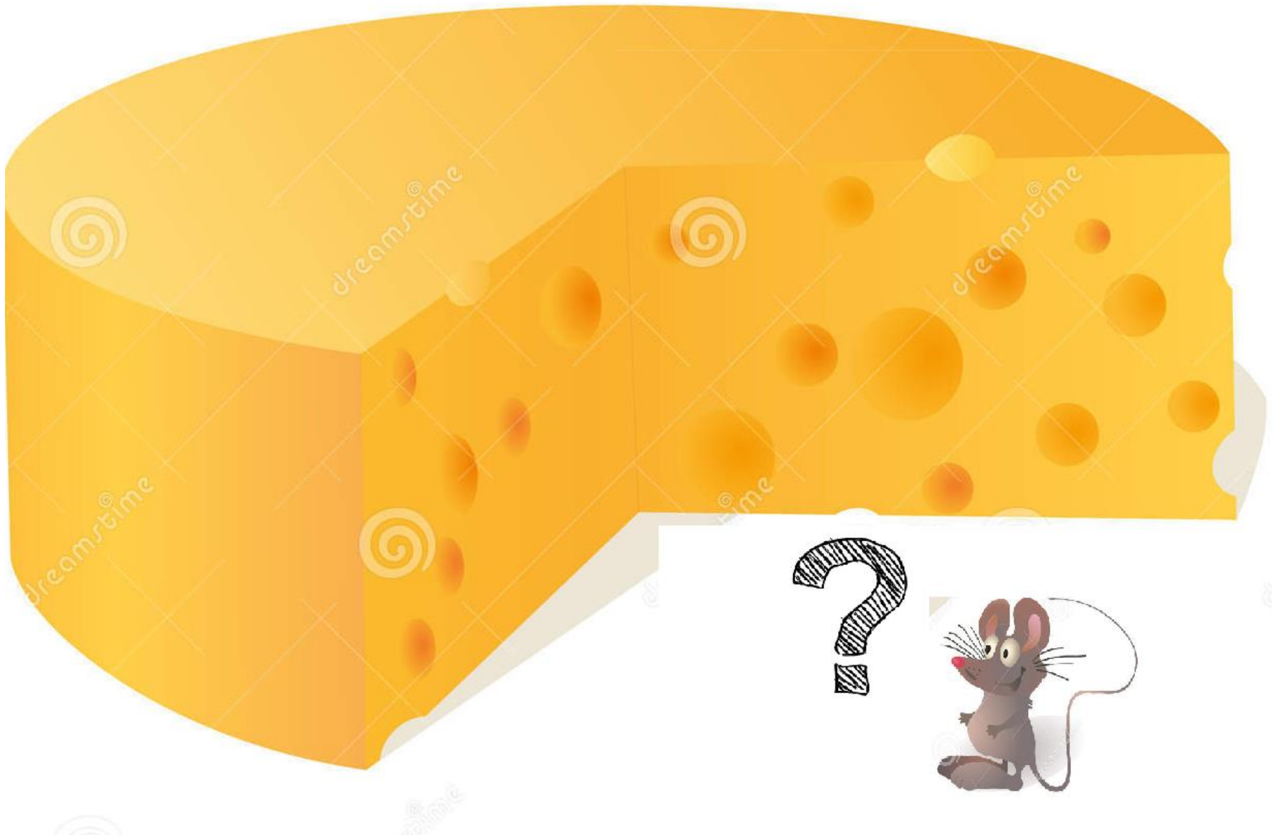
Big Data Class



LECTURER: DAN FELDMAN

TEACHING ASSISTANTS:

IBRAHIM JUBRAN

ALAA MAALOUF

אוניברסיטת חיפה
University of Haifa
جامعة حيفا

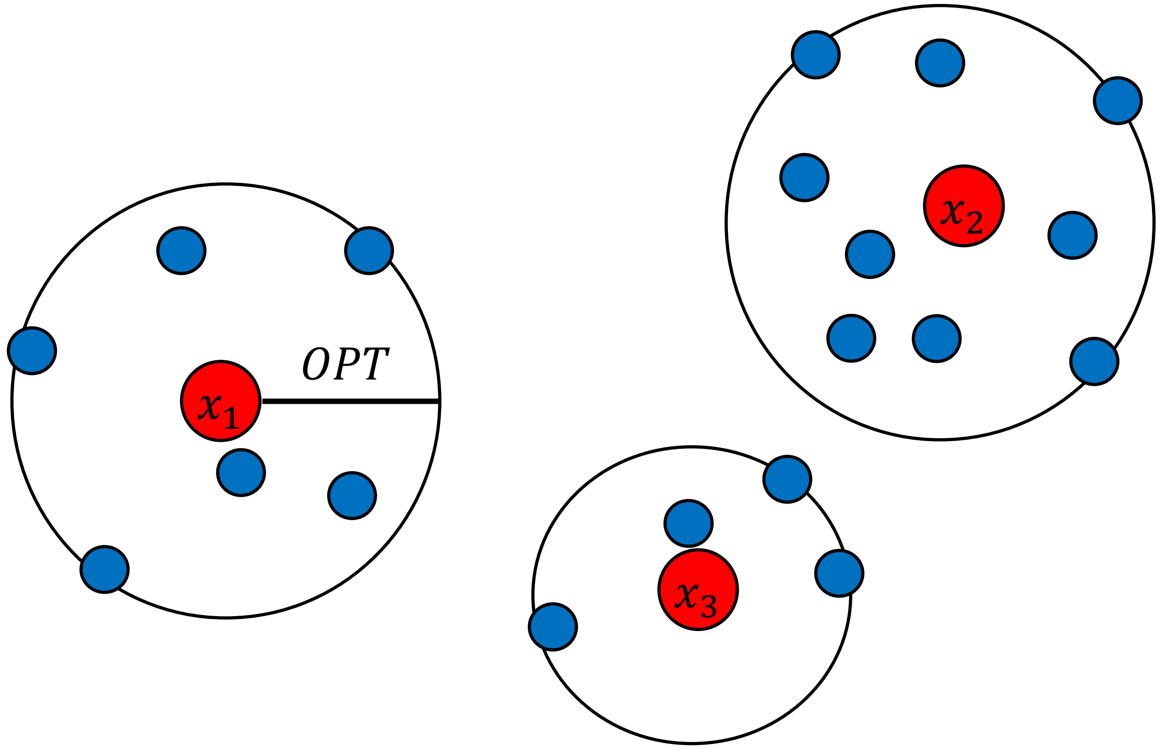
k-Center / k-Minimum Enclosing Balls

- Given a set of n points P in R^d and an integer k, find a set of k points (centers) $X = \{x_1, \dots, x_k\} \subseteq R^d$ that minimizes:

$$\boldsymbol{far_k(P, X) = \max_{p \in P} \|p - X(p)\|}$$

Closest point in X to p

$$X(p) \in arg \min_{x \in X} \|x - p\|$$



k-Center / k-Minimum Enclosing Balls

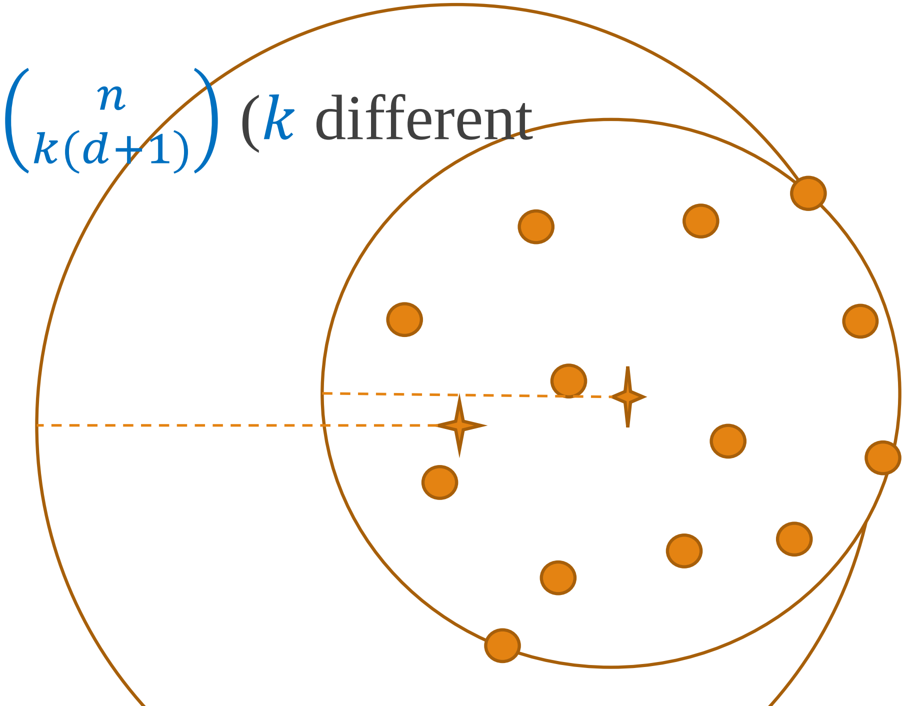
<u>Optimal solution in R^d:</u>

Claim 1: A sphere in R^d is determined by $d + 1$ points.

Claim 2: A sphere with minimal radius enclosing a set of points in R^d passes through $d + 1$ points from the set.

Algorithm: Exhaustive search over all possible tuples $\binom{n}{k(d+1)}$ (k different circles, each determined by $d + 1$ points).

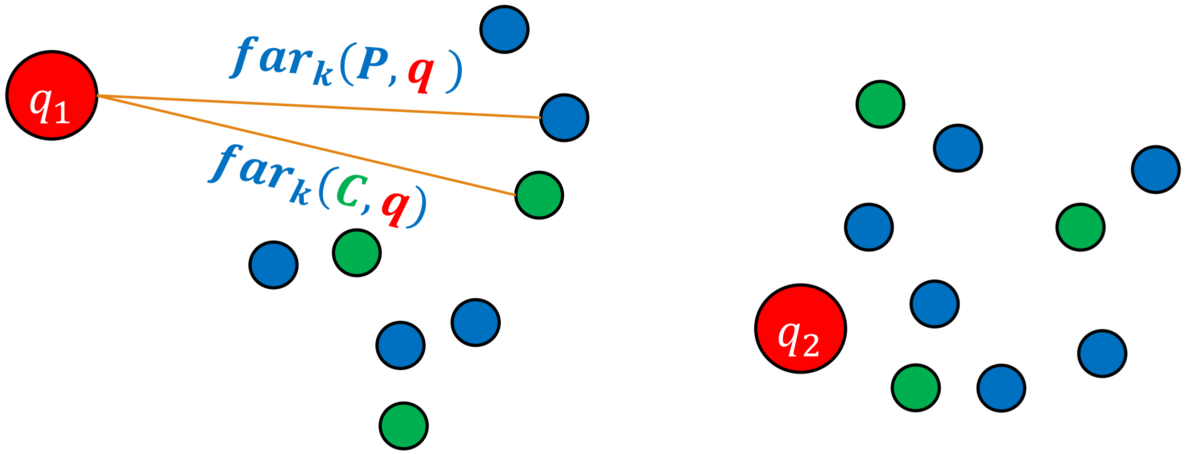
Running time: $n^{O(dk)}$.



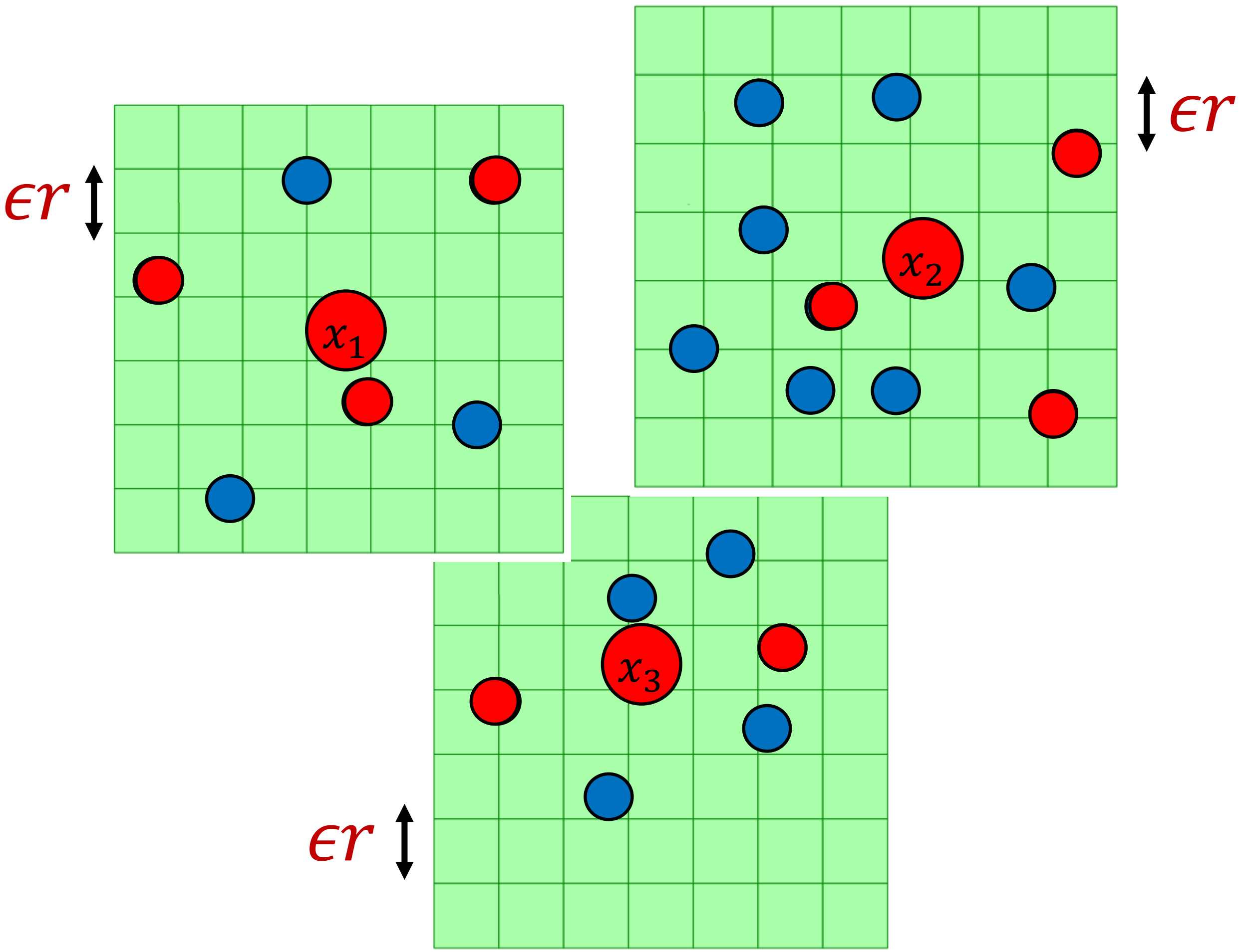
Coreset for k-Center

- <u>Input:</u> (P, k, Q) where $P \subseteq R^d, k$ is an integer and $Q \subseteq \left(R^d\right)^k$.
- <u>Output:</u> $\mathrm{C} \subseteq P, |C| = k\left(\frac{1}{\epsilon}\right)^{O(d)}$ $s.t.$ for every $q \in Q$:

$$far_k(P, q) - far_k(C, q) \leq O(\epsilon) \cdot far_k(P, q)$$



Coreset for k-Center



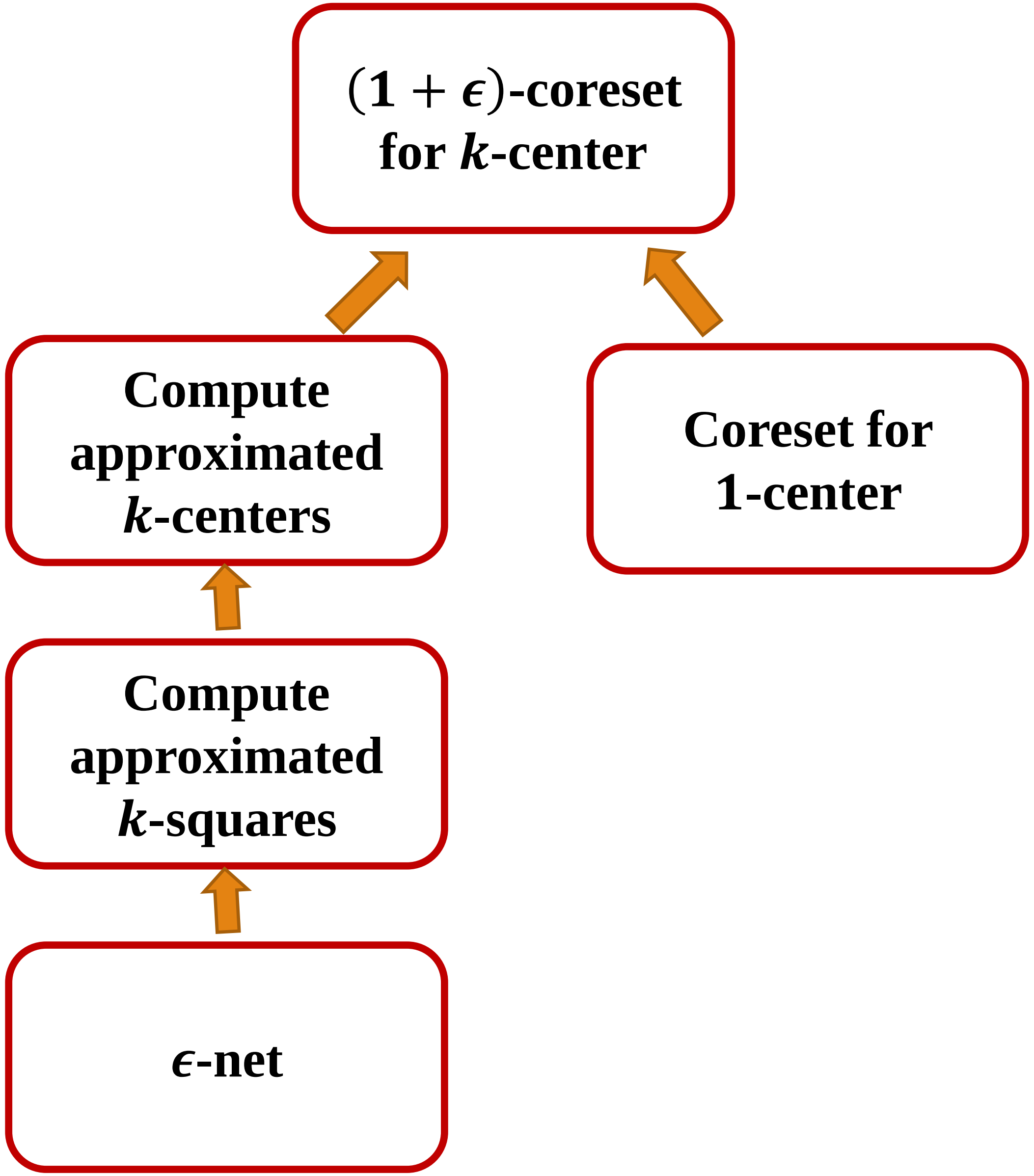
$(1+\epsilon)$-Coreset Algorithm:

1) Find optimal k-centers (To find k "clusters" and the optimal radius OPT).
2) Compute 1-center coreset for each cluster where $r = OPT$.

Total time: $\boldsymbol{n^{O(dk)}}$

Coreset size: $\boldsymbol{k \cdot \left(\frac{1}{\epsilon}\right)^{O(d)}}$

Coreset for k-Center



$(1+\epsilon)$-Coreset Algorithm:

1) Find optimal k-centers
   (To find k “clusters” and the optimal radius OPT).
2) Compute 1-center coreset for each cluster where $r = OPT$.

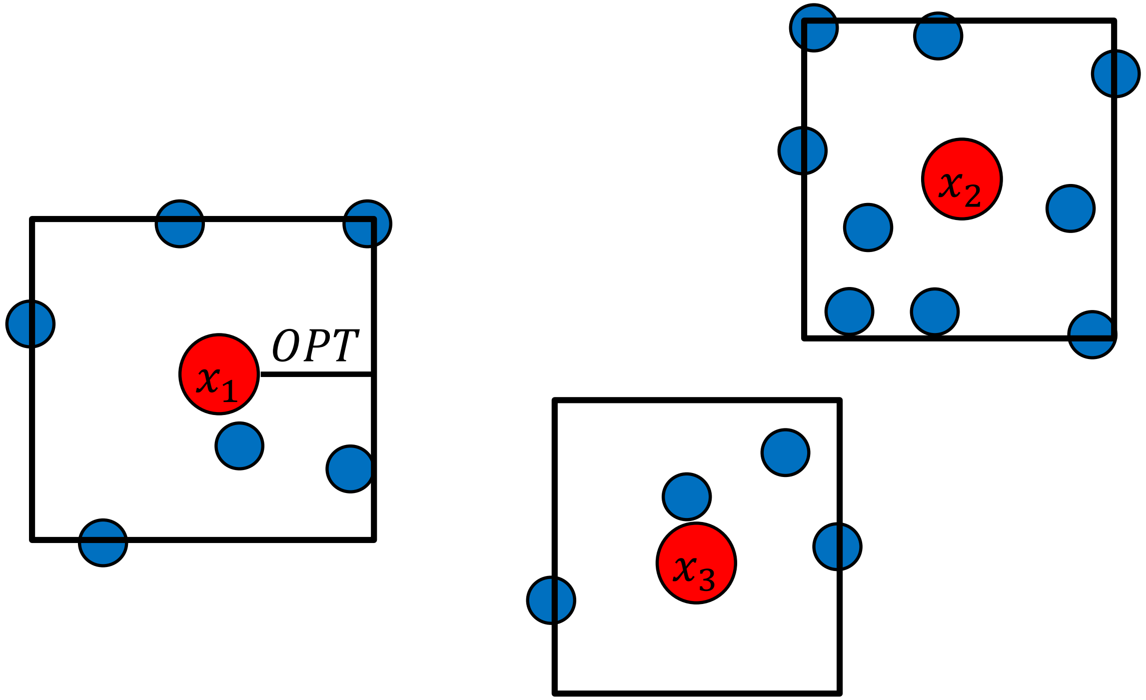
K-Minimum Enclosing Squares

- Given a set of n points P in R^d, find a set of k points (centers) $X = \{x_1, \ldots, x_k\} \subseteq R^d$ that minimizes:

$$far_k(P, X) = \max_{p \in P} \|p - X(p)\|_\infty$$

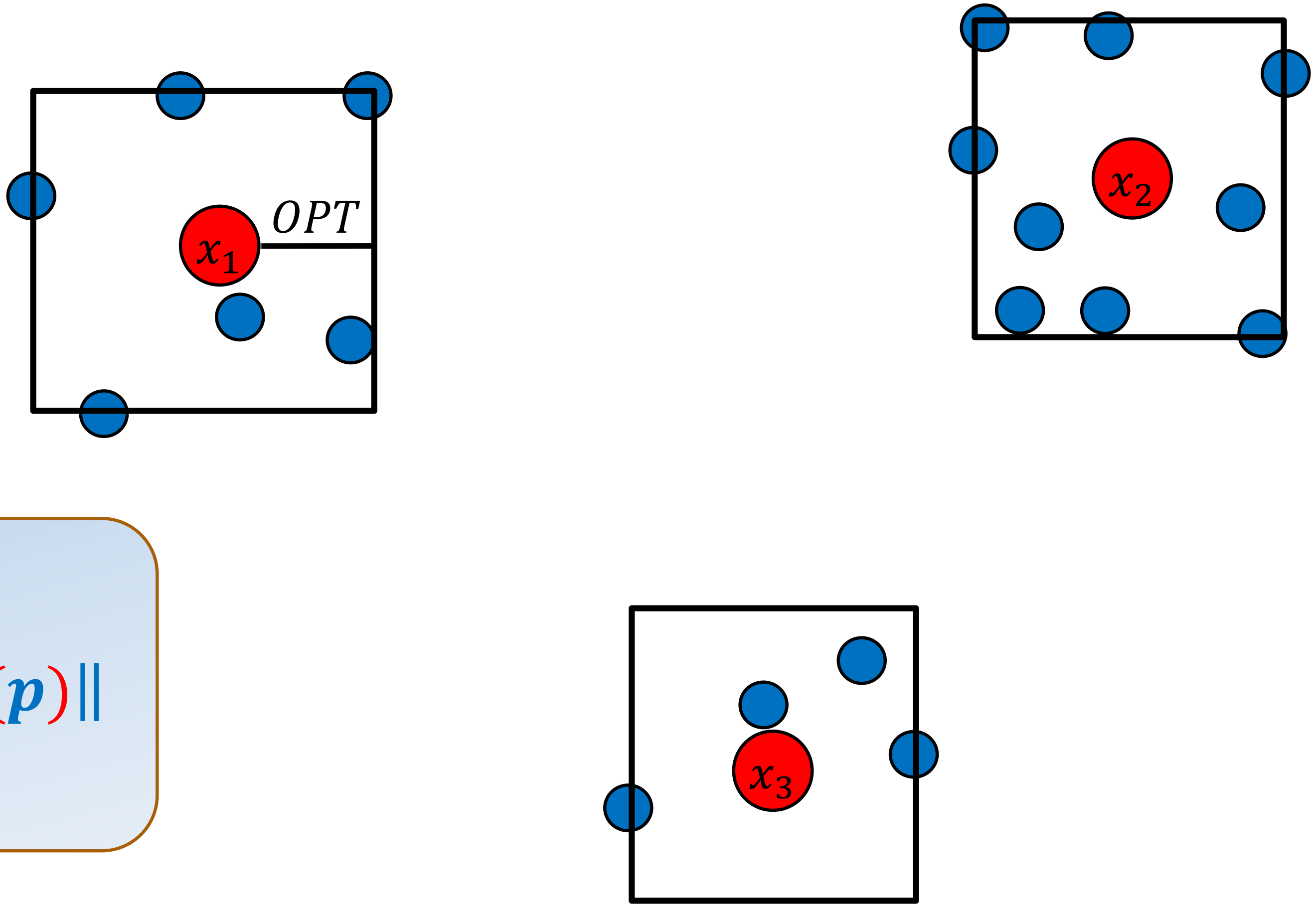
Closest point in X to p

$$X(p) \in arg\min_{x \in X} \|x - p\|_\infty$$



K-Minimum Enclosing Squares → K-Center

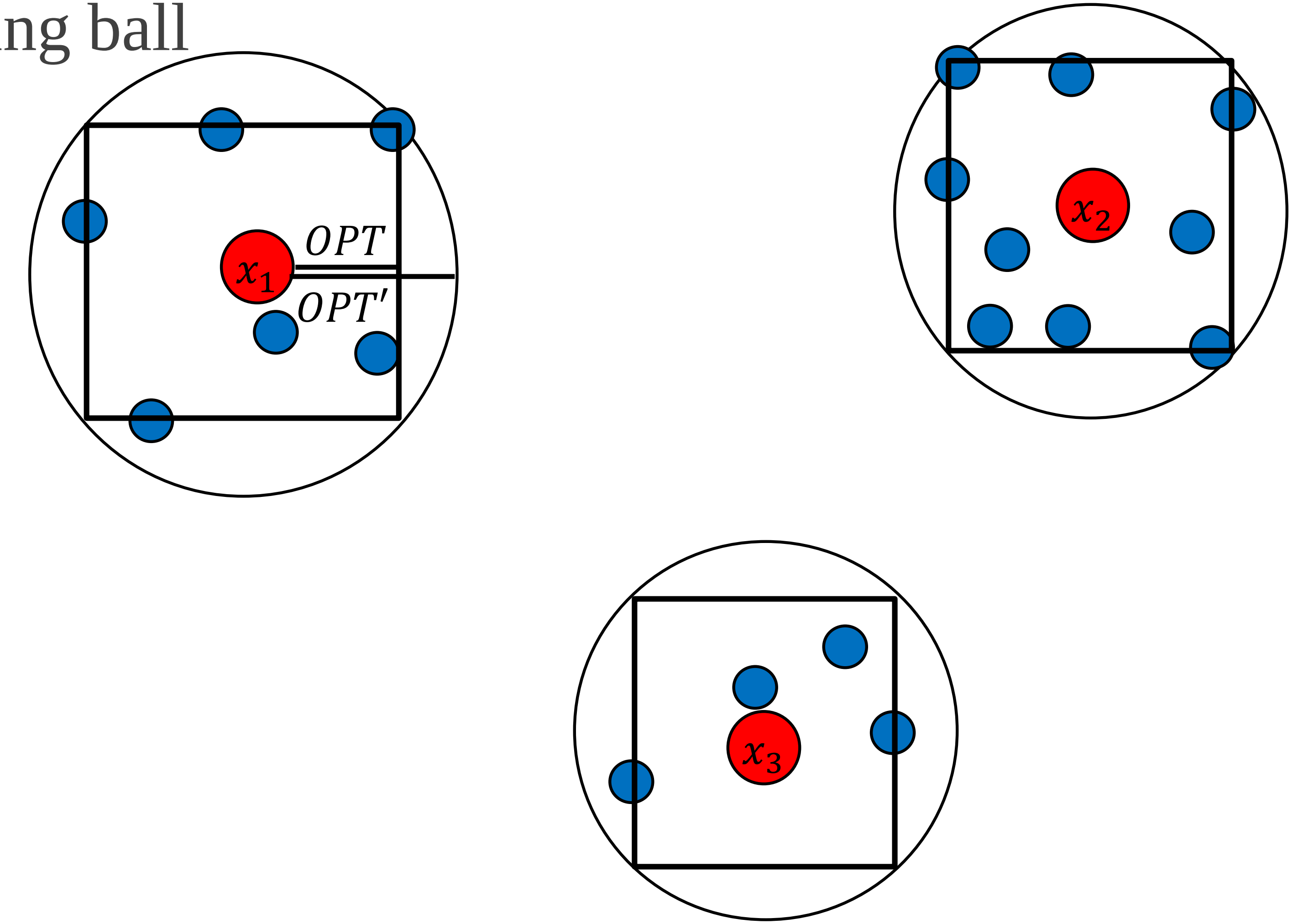
- Given K-minimum Enclosing Squares
  where $OPT = \boldsymbol{far_k}(\boldsymbol{P}, \boldsymbol{X}) = \max_{\boldsymbol{p} \in \boldsymbol{P}} \|\boldsymbol{p} - \boldsymbol{X}(\boldsymbol{p})\|_\infty$



We want to compute:
$\widehat{OPT} = \boldsymbol{far_k}(\boldsymbol{P}, \boldsymbol{X}) = \max_{\boldsymbol{p} \in \boldsymbol{P}} \|\boldsymbol{p} - \boldsymbol{X}(\boldsymbol{p})\|$

K-Minimum Enclosing Squares → K-Center

- Given K-minimum Enclosing Squares
- For each square, draw an enclosing ball



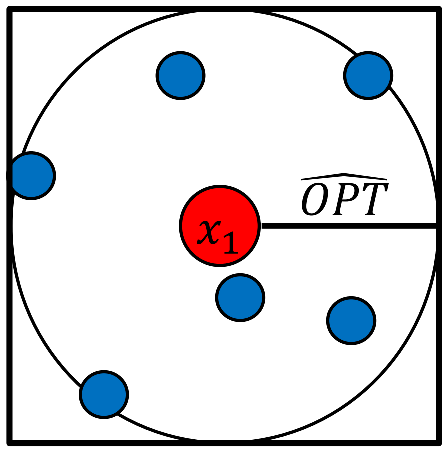
Claim: $OPT' \leq \sqrt{d} \cdot \widehat{OPT}$

K-Minimum Enclosing Squares → K-Center

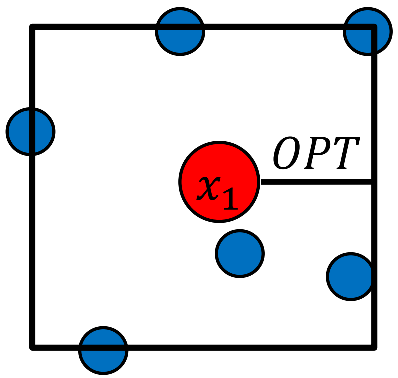
<u>Proof:</u> $OPT' \leq \sqrt{d} \cdot \widehat{OPT}$

1) $OPT \leq \widehat{OPT}$ (by definition)

$\widehat{OPT} = \max_{p \in P} \|p - X(p)\|$



$OPT = \max_{p \in P} \|p - X(p)\|_\infty$

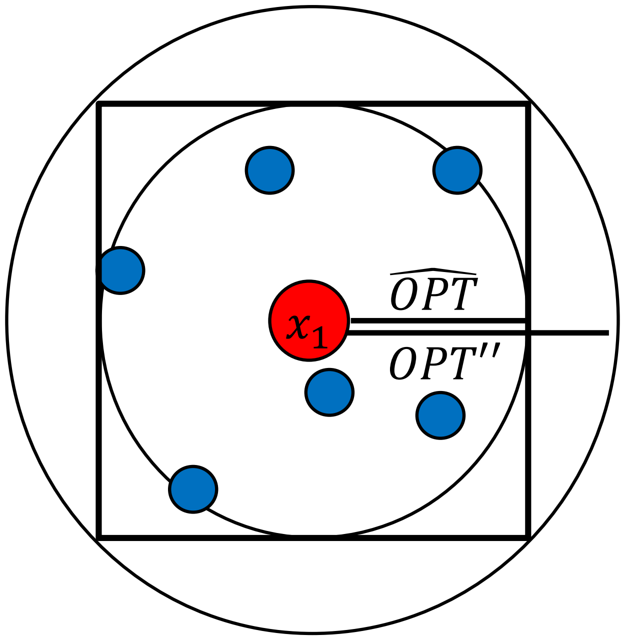


K-Minimum Enclosing Squares → K-Center

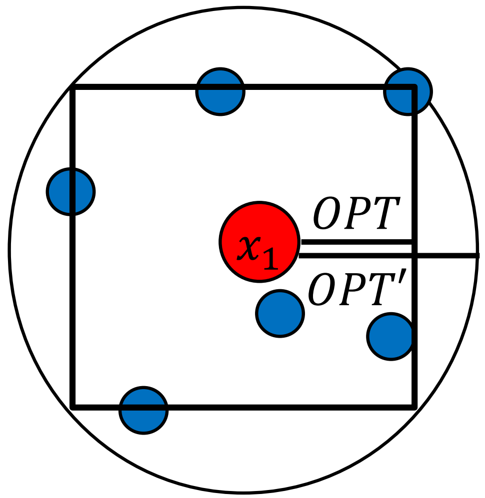
Proof: $OPT' \leq \sqrt{d} \cdot \widehat{OPT}$

1) $OPT \leq \widehat{OPT}$ (by definition)

$\widehat{OPT} = \max_{p \in P} \|p - X(p)\|$



$OPT = \max_{p \in P} \|p - X(p)\|_\infty$



K-Minimum Enclosing Squares → K-Center

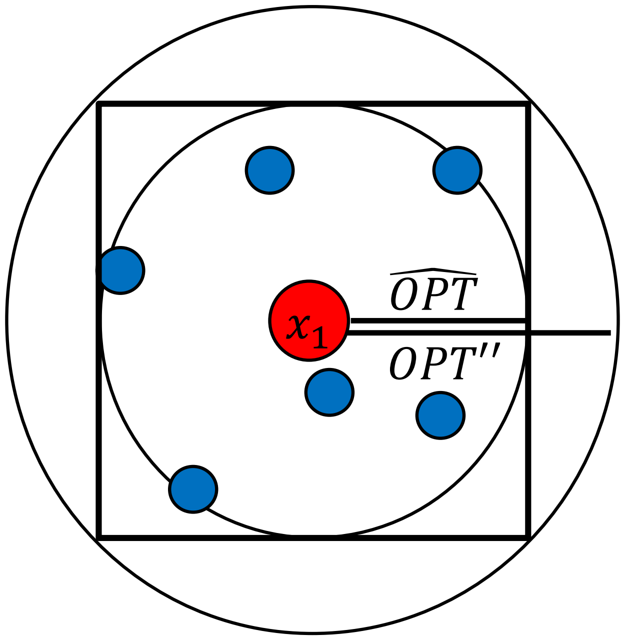
<u>Proof:</u> $OPT' \leq \sqrt{d} \cdot \widehat{OPT}$

1) $OPT \leq \widehat{OPT}$ (by definition)

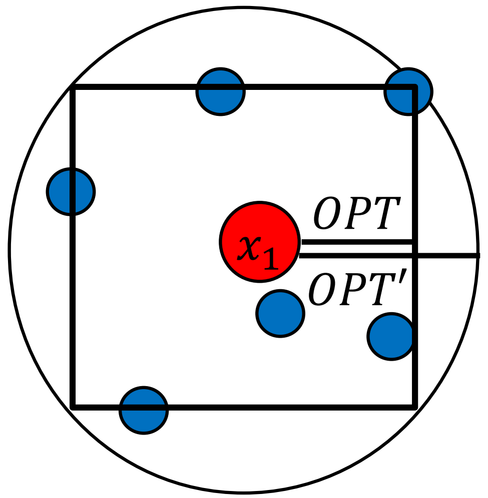
2) $OPT' \leq OPT''$

<u>Claim:</u> $OPT'' \leq \sqrt{d} \cdot \widehat{OPT}$

$$\widehat{OPT} = \max_{p \in P} \|p - X(p)\|$$



$$OPT = \max_{p \in P} \|p - X(p)\|_\infty$$

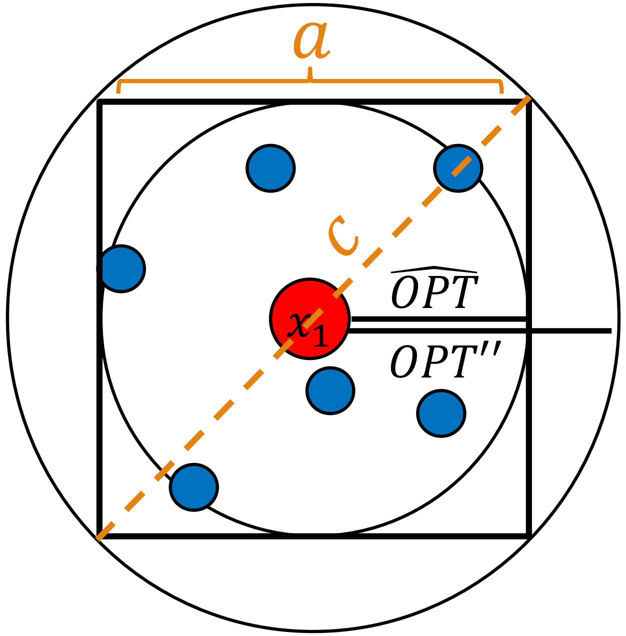


K-Minimum Enclosing Squares → K-Center

<u>Proof:</u> $OPT' \leq \sqrt{d} \cdot \widehat{OPT}$

<u>Proof:</u> $OPT'' \leq \sqrt{d} \cdot \widehat{OPT}$

$$\widehat{OPT} = \max_{p \in P} \|p - X(p)\|$$



$$a = 2 \cdot \widehat{OPT}$$

$$c = \sqrt{a^2 + \cdots + a^2} = \sqrt{da^2}$$
$$= \sqrt{d}\, a = 2\sqrt{d} \cdot \widehat{OPT}$$

$$OPT'' = \frac{c}{2} = \sqrt{d} \cdot \widehat{OPT}$$

■

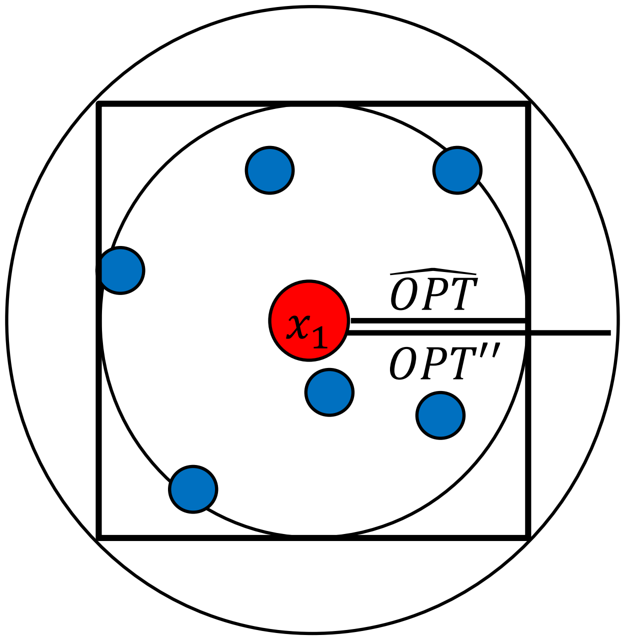
K-Minimum Enclosing Squares → K-Center

<u>Proof:</u> $OPT' \leq \sqrt{d} \cdot \widehat{OPT}$

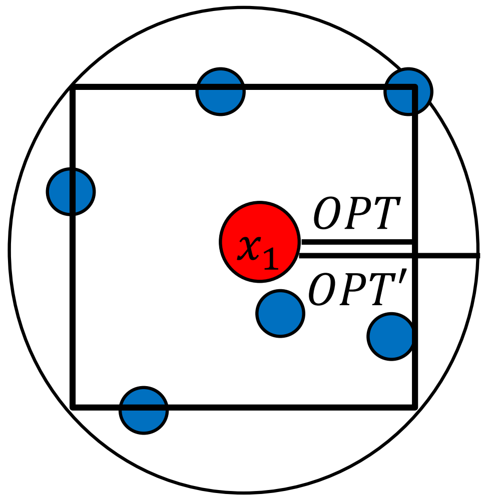
1) $OPT \leq \widehat{OPT}$ (by definition)

2) $OPT' \leq OPT''$

$$\widehat{OPT} = \max_{p \in P} \|p - X(p)\|$$



$$OPT = \max_{p \in P} \|p - X(p)\|_\infty$$



K-Minimum Enclosing Squares → K-Center

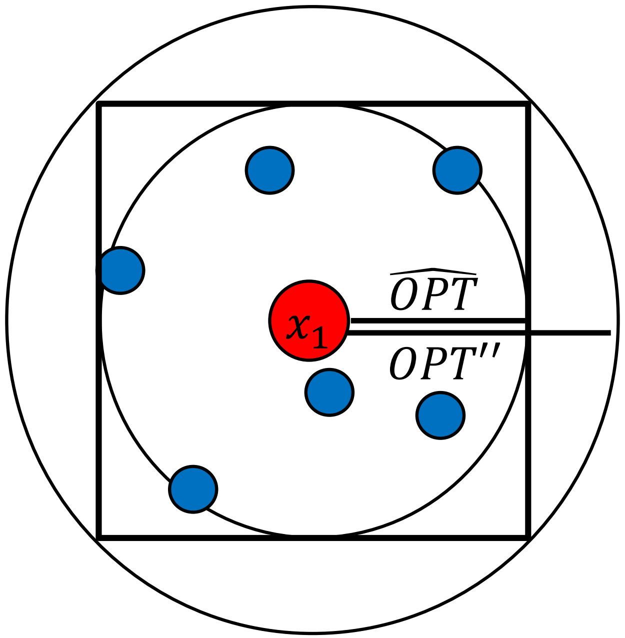
Proof: $OPT' \leq \sqrt{d} \cdot \widehat{OPT}$

1) $OPT \leq \widehat{OPT}$ (by definition)

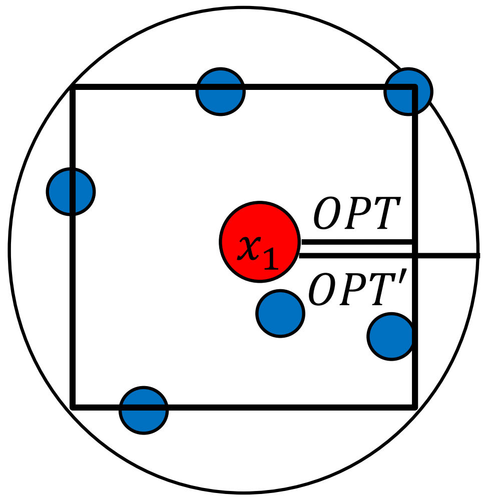
2) $OPT' \leq OPT''$

3) $OPT'' \leq \sqrt{d} \cdot \widehat{OPT}$

$\widehat{OPT} = \max_{p \in P} \|p - X(p)\|$



$OPT = \max_{p \in P} \|p - X(p)\|_\infty$



K-Minimum Enclosing Squares → K-Center

Proof: $OPT' \leq \sqrt{d} \cdot \widehat{OPT}$

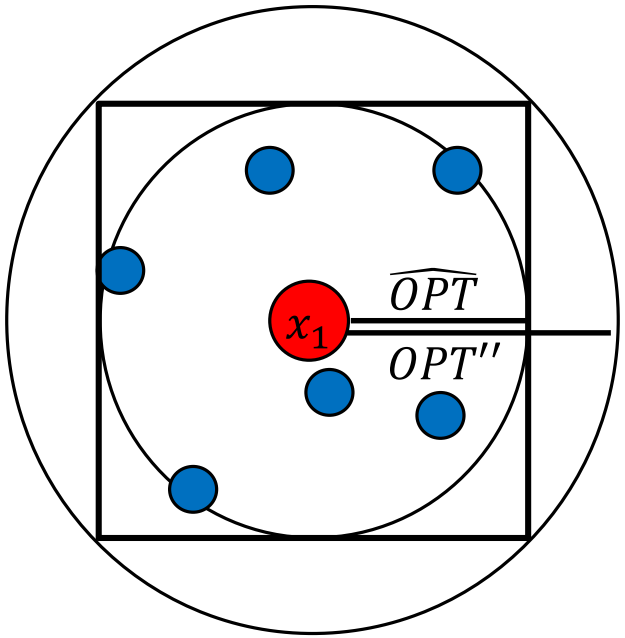
1) $OPT \leq \widehat{OPT}$ (by definition)

2) $OPT' \leq OPT''$

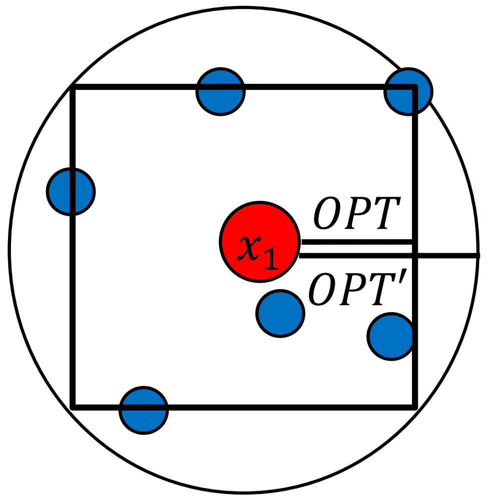
3) $OPT'' \leq \sqrt{d} \cdot \widehat{OPT}$

→ $\boxed{OPT' \leq \sqrt{d} \cdot \widehat{OPT}}$

$\widehat{OPT} = \max_{p \in P} \|p - X(p)\|$



$OPT = \max_{p \in P} \|p - X(p)\|_\infty$



More Formal: k-Squares → k-Centers

Claim 1:

$far_\infty(P,q) \le far_2(P,q) \le \sqrt{d} \cdot far_\infty(P,q)$

Proof of claim 1:

$$far_\infty(P,q) = \max_{p\in P} \|p-q\|_\infty$$

$$\le \max_{p\in P} \|p-q\|_2 = far_2(P,q)$$

$$= \max_{p\in P} \sqrt{(p(1)-q(1))^2 + \cdots + (p(d)-q(d))^2}$$

$$\le \max_{p\in P} \sqrt{d \cdot \max_i (p(i)-q(i))^2}$$

$$= \sqrt{d} \cdot \max_{p\in P} \left(\max_i |p(i)-q(i)|\right)$$

$$= \sqrt{d} \cdot \max_{p\in P} (\|p-q\|_\infty) = \sqrt{d} \cdot far_\infty(P,q)$$

Definitions:

$$far_2(P,q) = \max_{p\in P} \|p-q\|_2$$

$$OPT_2 = \operatorname*{argmin}_{q\in Q} far_2(P,q)$$

$$far_\infty(P,q) = \max_{p\in P} \|p-q\|_\infty$$

$$OPT_\infty = \operatorname*{argmin}_{q\in Q} far_\infty(P,q)$$

More Formal: k-Squares → k-Centers

Claim 2:

An α-approximation for k-squares is an $O(\alpha \cdot \sqrt{d})$-approximation for k-centers.

Proof of claim 2:

Let a_∞ be the α-approximation for k-squares → $far_\infty(P, a_\infty) \le \alpha \cdot far_\infty(P, OPT_\infty)$.

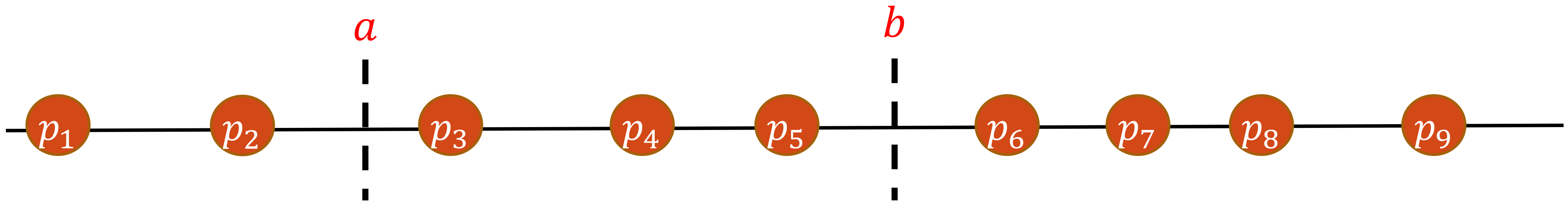
$$
\begin{aligned}
\rightarrow far_2(P, a_\infty) &\le \sqrt{d} \cdot far_\infty(P, a_\infty) && \text{(Right side of Claim 1)} \\
&\le \alpha\sqrt{d} \cdot far_\infty(P, OPT_\infty) && \text{(Definition of } a_\infty\text{)} \\
&\le \alpha\sqrt{d} \cdot far_\infty(P, OPT_2) && \text{(Definition of } OPT_\infty\text{)} \\
&\le \alpha\sqrt{d} \cdot far_2(P, OPT_2) && \text{(Left side of Claim 1)}
\end{aligned}
$$

ϵ-net (d=1)

- Input: $P \subseteq R, Q = \{[a,b] |\ a,b \in R, a \leq b\}$
- Output: $C \subseteq P, |C| = \frac{1}{\epsilon}$ $s.t.$ for every $q \in Q$:

$$\left|\frac{|\boldsymbol{P} \cap \boldsymbol{q}|}{|\boldsymbol{P}|} - \frac{|\boldsymbol{C} \cap \boldsymbol{q}|}{|\boldsymbol{C}|}\right| \leq \boldsymbol{\epsilon}$$

$\boldsymbol{q} = [\boldsymbol{a}, \boldsymbol{b}]$



$$\frac{|\boldsymbol{P} \cap \boldsymbol{q}|}{|\boldsymbol{P}|} = \frac{3}{9} = \frac{1}{3}$$

How should we pick the subset C?
We need C to have the same percentage of points in $[a,b]$ as P.
→ we want to “sample points” uniformly from P.

ϵ-net (d=1)

Algorithm:

Assume $p_1 \le p_2 \le \cdots \le p_n$

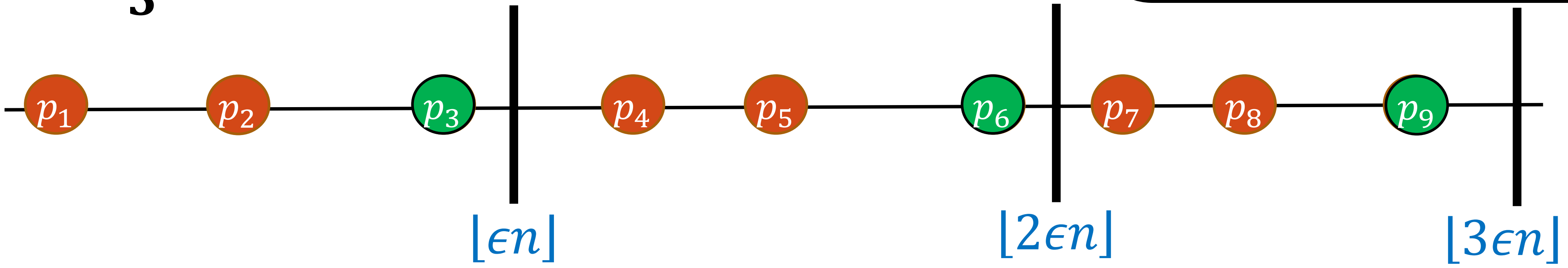
$I = \{\lfloor \epsilon n \rfloor, \lfloor 2\epsilon n \rfloor, \dots, \lfloor n \rfloor\}$

$\mathrm{C} = \{\mathrm{p_i} | \mathrm{i} \in I\}$.

Example:

$\boldsymbol{\epsilon = \frac{1}{3}, n = 9}$

$|\boldsymbol{C}| = \frac{\mathbf{1}}{\boldsymbol{\epsilon}}$



Claim 1: Assume $|P \cap q| = k$. By the construction of C we get that: $\left\lfloor \frac{k}{\epsilon n} \right\rfloor \le |C \cap q| \le \left\lceil \frac{k}{\epsilon n} \right\rceil$.

Claim 2: $\frac{|C \cap q|}{|C|} \le \frac{\left\lceil \frac{k}{\epsilon n} \right\rceil}{|C|} = \epsilon \left\lceil \frac{k}{\epsilon n} \right\rceil \le \frac{k}{n} + \epsilon$

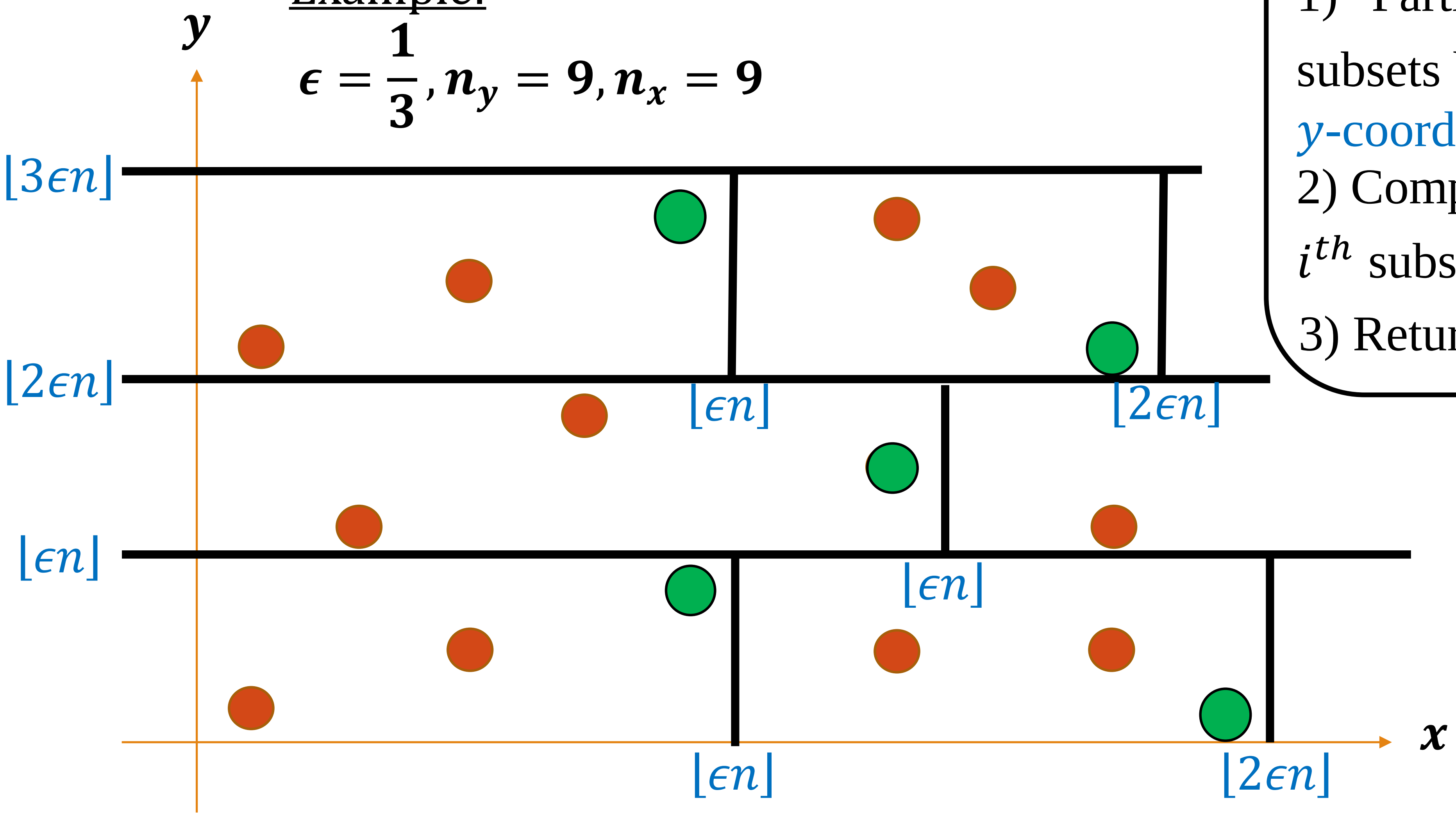
Claim 3: $\frac{|C \cap q|}{|C|} \ge \frac{\left\lfloor \frac{k}{\epsilon n} \right\rfloor}{|C|} = \epsilon \left\lfloor \frac{k}{\epsilon n} \right\rfloor \ge \frac{k}{n} - \epsilon$

$$\rightarrow \left| \underbrace{\frac{|\boldsymbol{P} \cap \boldsymbol{q}|}{|\boldsymbol{P}|}}_{\frac{k}{n}} - \frac{|\boldsymbol{C} \cap \boldsymbol{q}|}{|\boldsymbol{C}|} \right| \le \boldsymbol{\epsilon}$$

ϵ-net (d=2)

<u>Example:</u>

$$\boldsymbol{\epsilon = \frac{1}{3}, n_y = 9, n_x = 9}$$



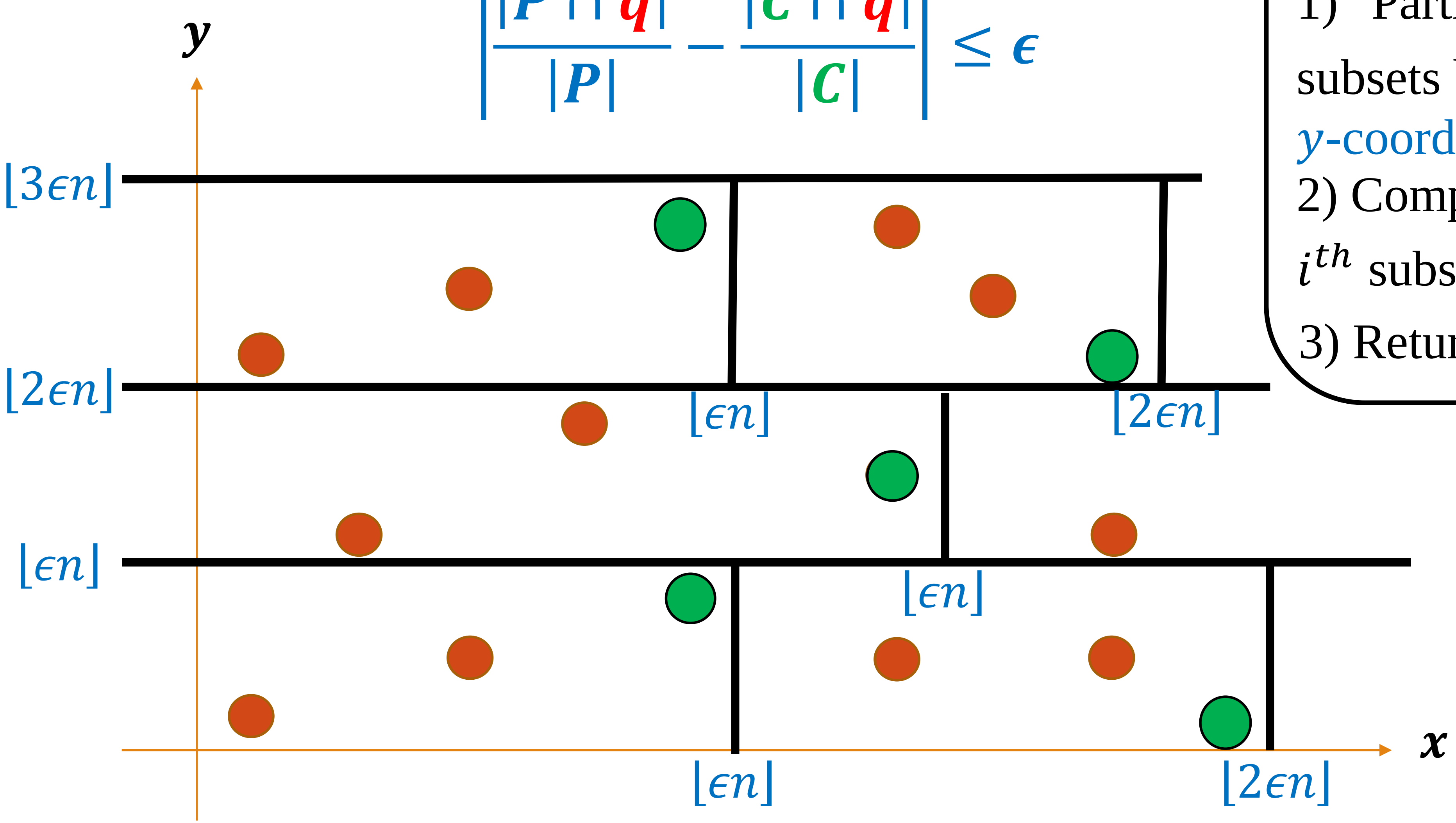
<u>Algorithm:</u>

1) Partition the points into $\left\lfloor\frac{1}{\epsilon}\right\rfloor$ subsets by their sorted y-coordinate
2) Compute an ϵ-net C_i for the i^{th} subset for every $i \in \left\lfloor\frac{1}{\epsilon}\right\rfloor$.
3) Return $\mathrm{C} = \bigcup_i C_i$

$$|\boldsymbol{C}| = \boldsymbol{O}\left(\left(\frac{\mathbf{1}}{\boldsymbol{\epsilon}}\right)^{\boldsymbol{d}}\right)$$

ϵ-net (d=2)

$$\left| \frac{|P \cap q|}{|P|} - \frac{|C \cap q|}{|C|} \right| \leq \epsilon$$



Algorithm:

1) Partition the points into $\left\lfloor \frac{1}{\epsilon} \right\rfloor$ subsets by their sorted y-coordinate
2) Compute an ϵ-net C_i for the i^{th} subset for every $i \in \left\lfloor \frac{1}{\epsilon} \right\rfloor$.
3) Return $C = \cup_i C_i$

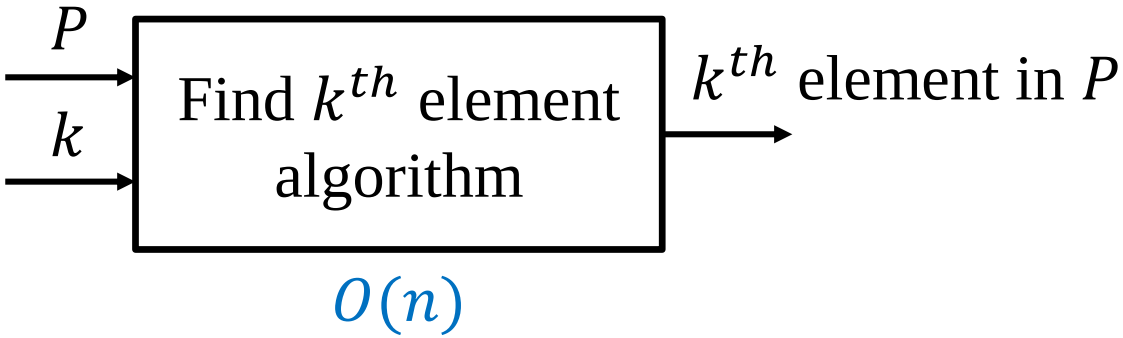
$$|C| = O\left(\left(\frac{1}{\epsilon}\right)^d\right)$$

$$\text{Time} = O\left(\frac{nd}{\epsilon}\right)$$

ϵ-net time analysis

Algorithm for computing the ϵ-net:

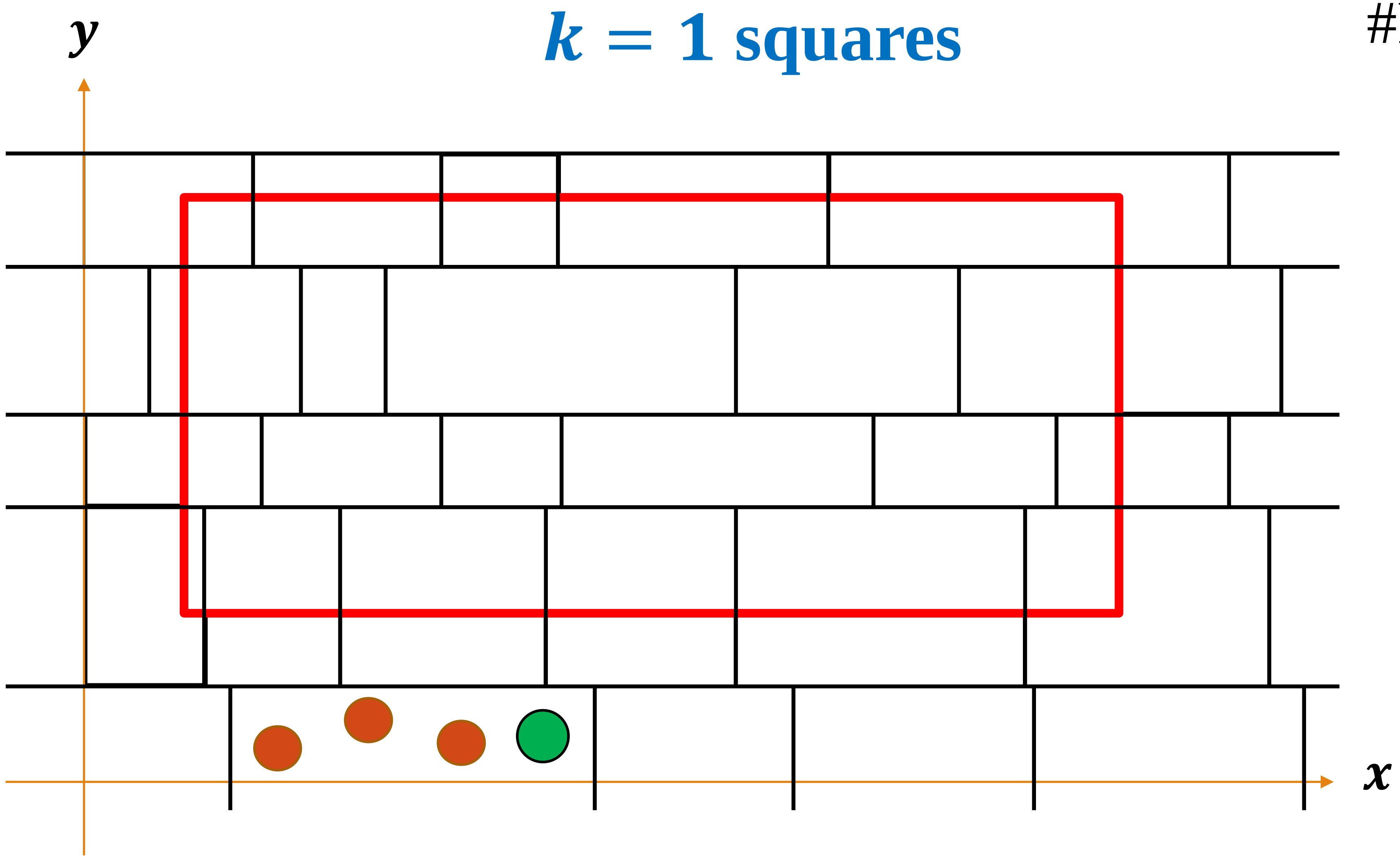
Assuming there is an algorithm for finding the element with rank=k in an unsorted set of n points:



- Find the $\lfloor \epsilon n \rfloor^{th}$ point using this algorithm in $O(n)$ time.
- Repeat $\left\lfloor \frac{1}{\epsilon} \right\rfloor$ times.
- Repeat for every dimension.

Total time: $O\left(\frac{nd}{\epsilon}\right)$.

ϵ-net (d=2)



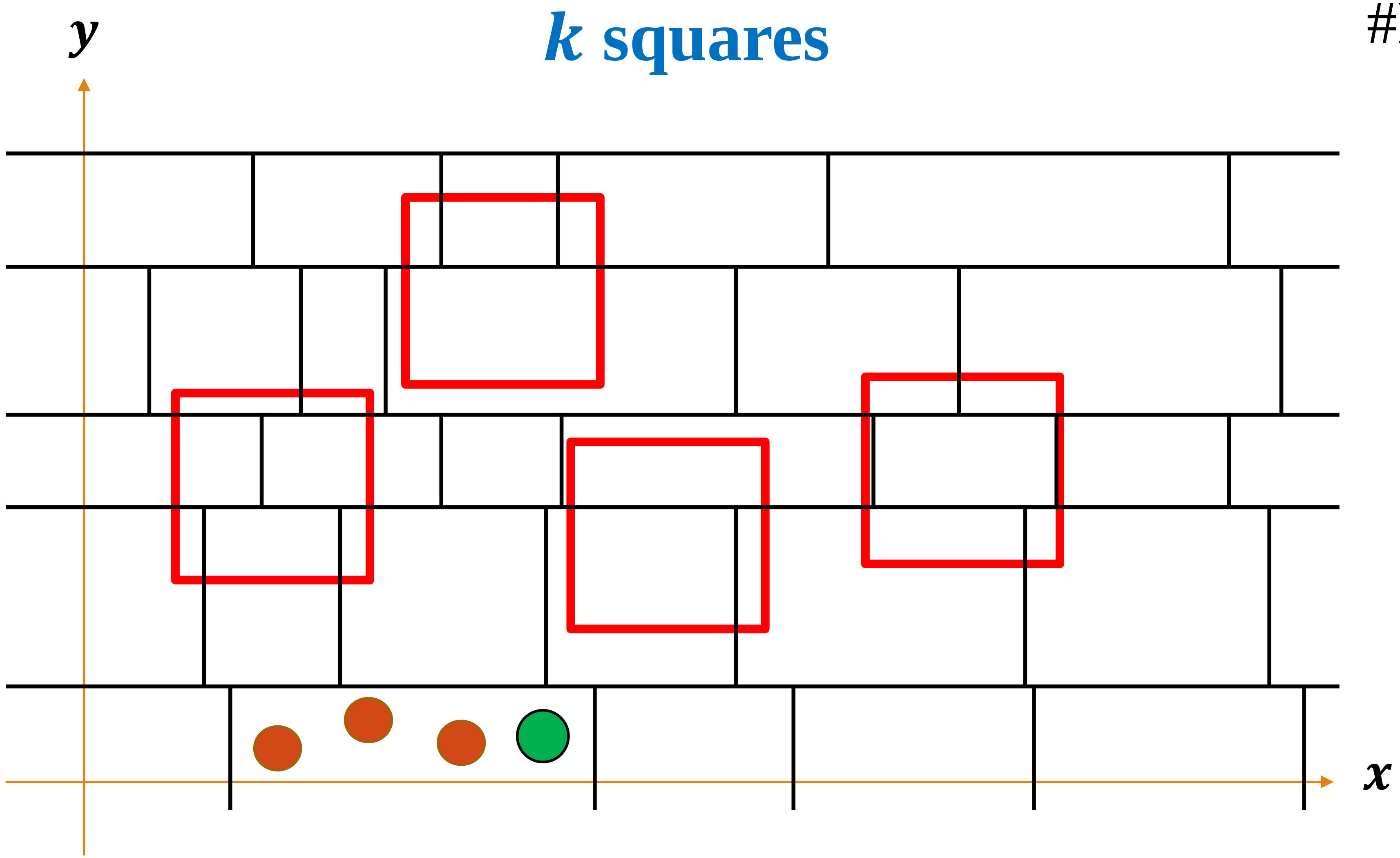
#Points in block $< \epsilon^2 n$

#Bad blocks $< \frac{4}{\epsilon}$

$\Downarrow$

#Bad points $< \epsilon n$

ϵ-net (d=2)



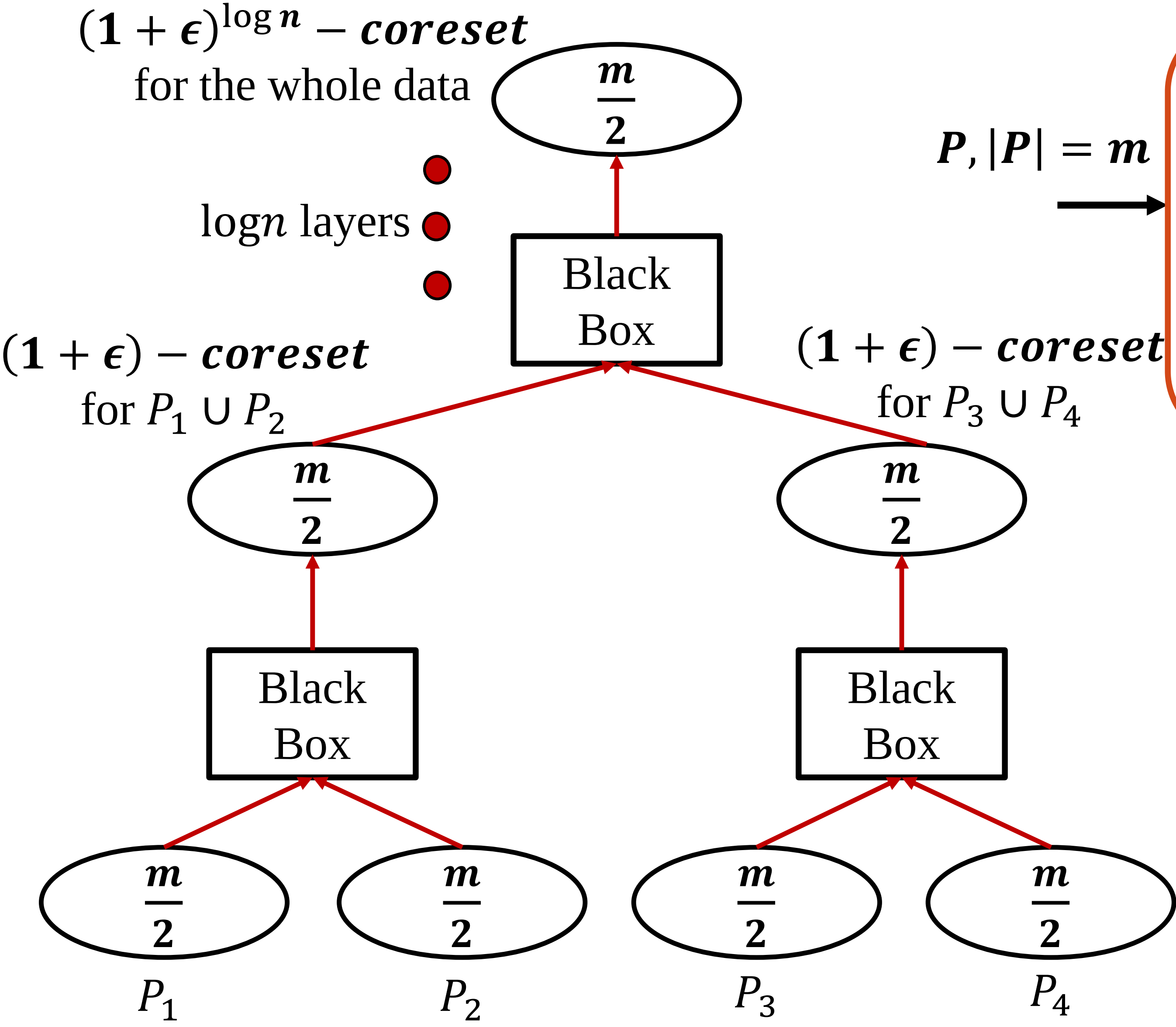
#Points in block $< \epsilon^2 n$

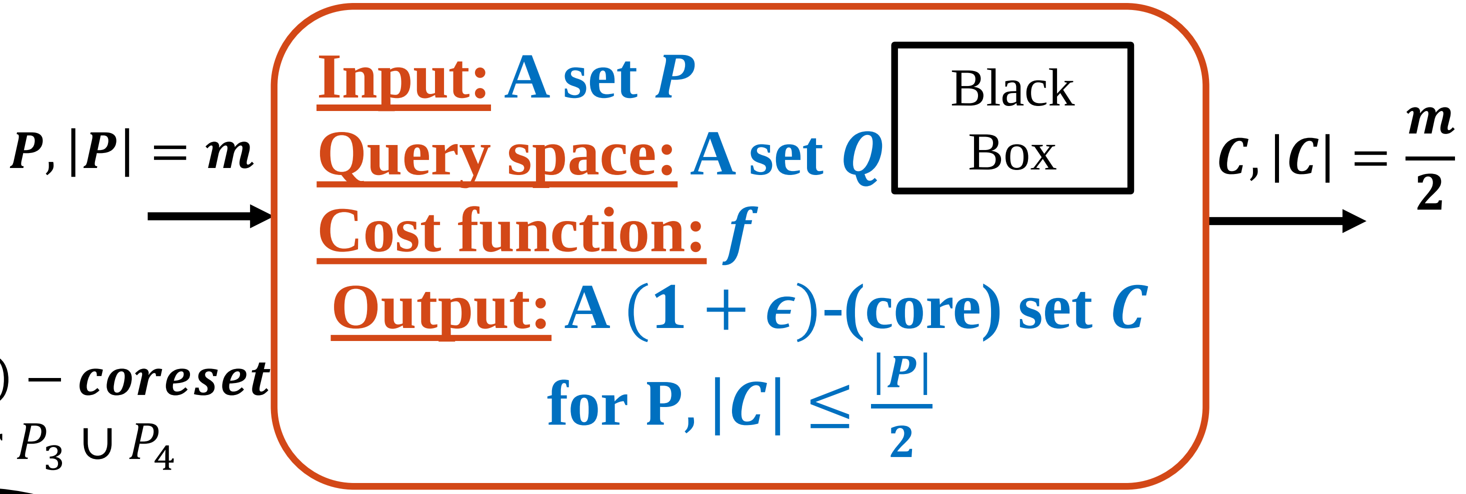
#Bad blocks $< k\frac{4}{\epsilon}$

$\Downarrow$

#Bad points $< k\epsilon n$

Coreset for k-Center - Streaming





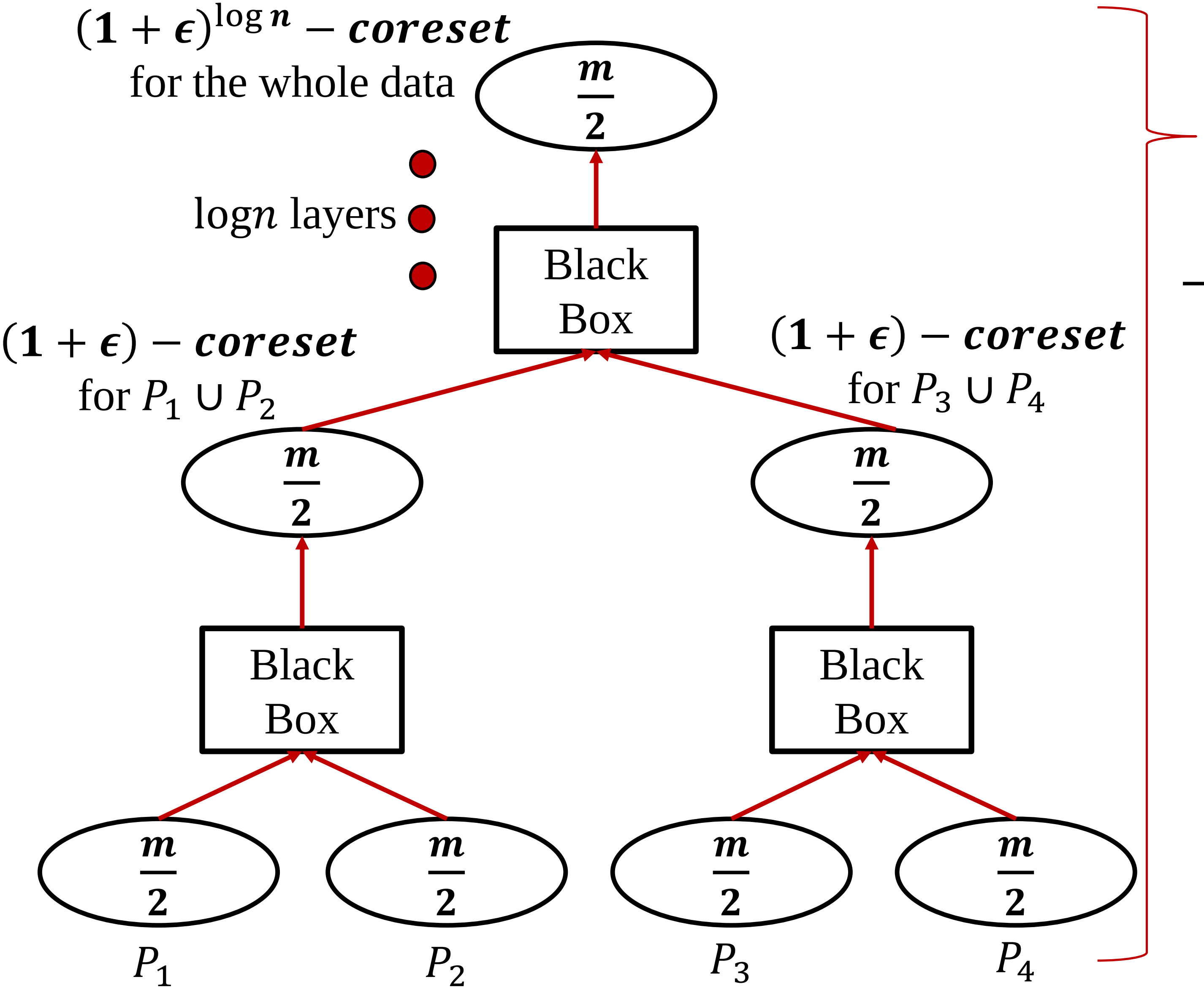
Smallest input size: $\mathbf{m} = 2k\left(\frac{1}{\epsilon}\right)^{O(d)}$

Output (coreset) size: $\frac{m}{2} = k\left(\frac{1}{\epsilon}\right)^{O(d)}$

Black box running time:

$$\mathrm{T_{BB}} = m^{O(dk)} = \left(k\left(\frac{1}{\epsilon}\right)\right)^{O(d^2k)}$$

Coreset for k-Center - Streaming



<u>Final error:</u>

$$(\mathbf{1}+\boldsymbol{\epsilon})^{\log n} \cong \mathbf{1}+\boldsymbol{\epsilon}\cdot \log n$$

→ Define $\epsilon' = \frac{\epsilon}{\log n}$ and run the process with ϵ' instead of ϵ to get a $(1+\epsilon)$-approximation.

<u>Total running time:</u>

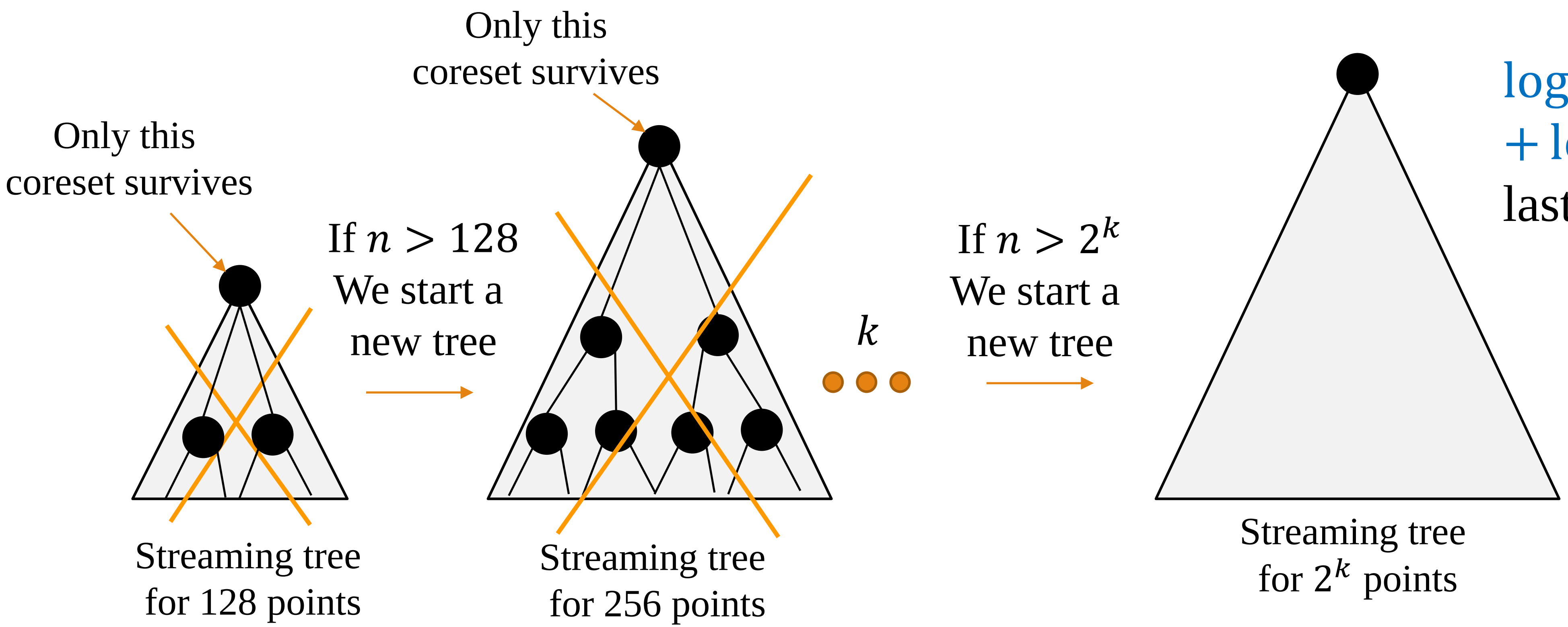
$$\mathbf{n}\cdot T_{BB} = n\cdot\left(k\left(\frac{1}{\epsilon'}\right)\right)^{O(d^2k)}$$

$$= n\cdot\left(k\left(\frac{\log n}{\epsilon}\right)\right)^{O(d^2k)}$$

Coreset for k-Center - Streaming

Problem: What if n (the number of input data) is unknown or infinite?

Solution: Doubling. (start with a small tree, then double the size).



$\log n$ coresets (**roots**) $+ \log n$ coresets in the last tree.