Clustering (Part 2)

OUTLINE

1. k-center clustering
   - Sequential Farthest-First Traversal.
   - Coreset technique.
   - MapReduce Farthest-First Traversal.
2. Case study: diameter of a pointset.

k-center clustering

Farthest-First Traversal: algorithm

- Popular 2-approximation sequential algorithm developed by T.F. Gonzalez [Theoretical Computer Science, 38:293-306, 1985]
- Simple and, somewhat fast, implementation when the input fits in main memory.
- Powerful primitive for extracting samples for further analyses

Input Set P of N points from a metric space (M, d), integer $k > 1$

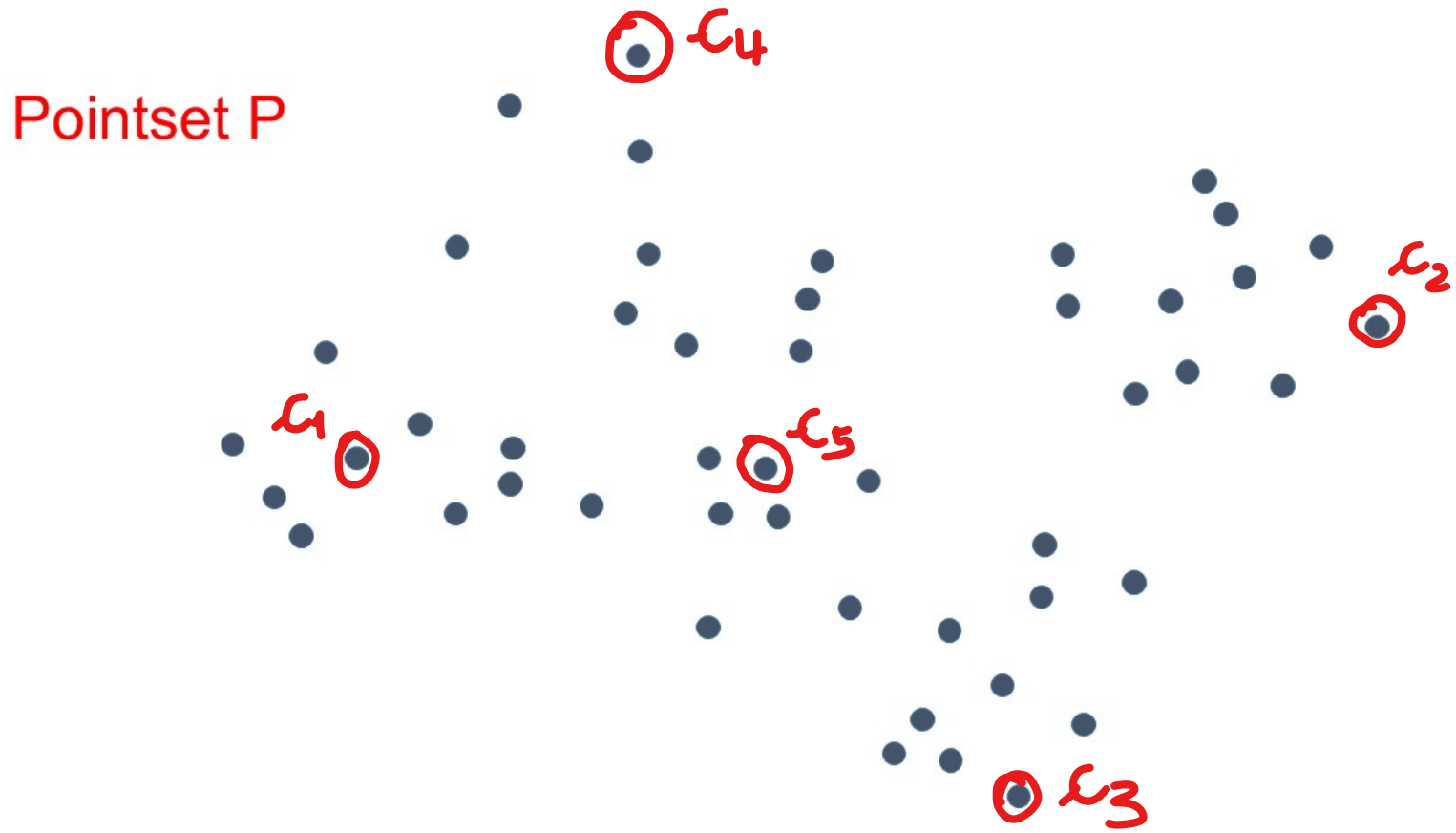
Output A set S of k centers which is a good solution to the k-center problem on P (i.e., $\Phi_{\text{kcenter}}(P, S)$ "close" to $\Phi^{\text{opt}}_{\text{kcenter}}(P, k)$)

Farthest-First Traversal: algorithm

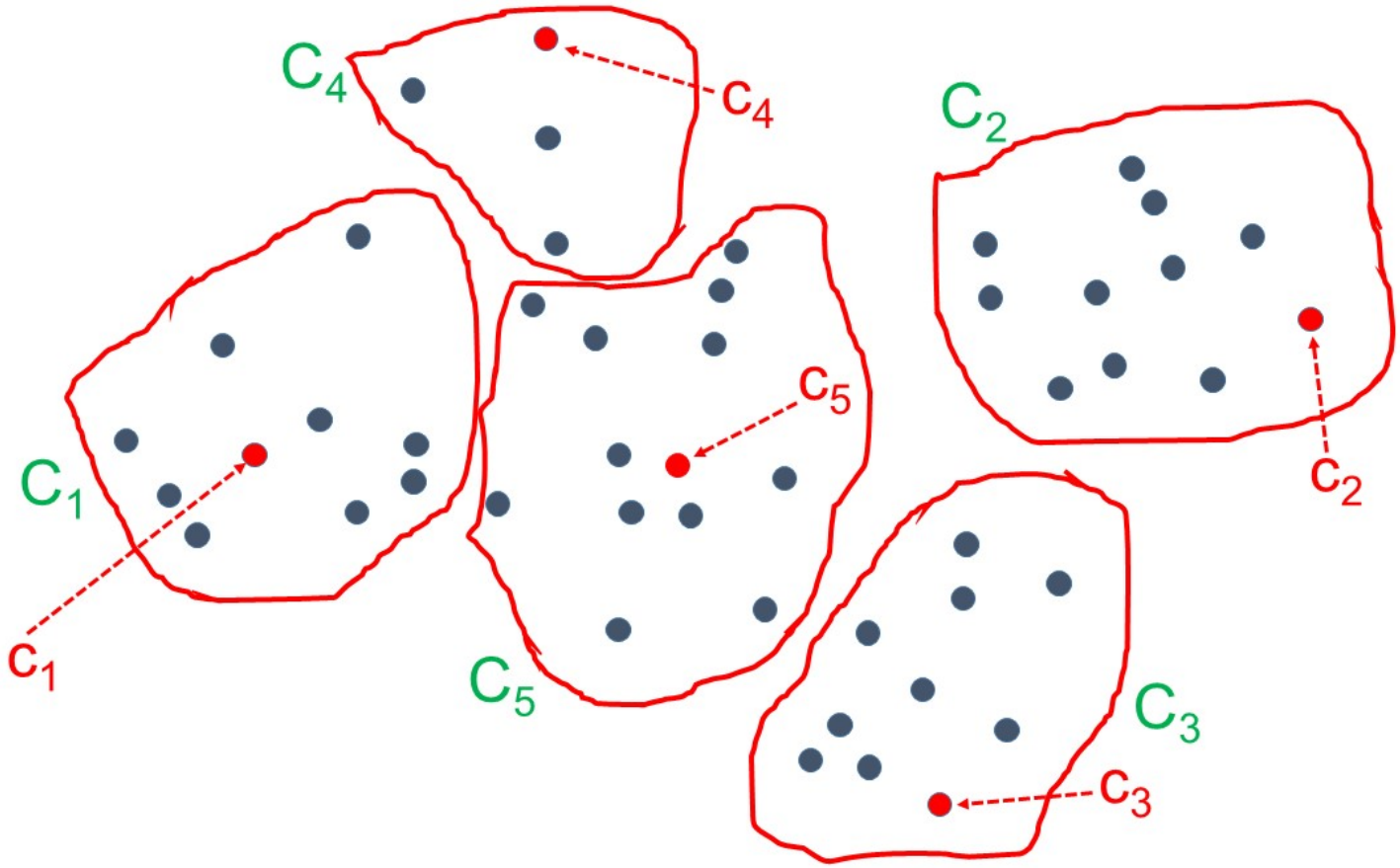
$S \leftarrow \{c_1\}$ // $c_1 \in P$ *arbitrary point*
for $i \leftarrow 2$ **to** k **do**
  `Find the point` $c_i \in P - S$ `that maximizes` $d(c_i, S)$
  $S \leftarrow S \cup \{c_i\}$
return S

Observation: The best clustering around the centers of S can be computed by invoking Assign(P, S). In fact, *the assignment of each point to the closest center can be easily maintained in every iteration of the for-loop*.

Farthest-First Traversal: example ($k = 5$)



Farthest-First Traversal: example ($k = 5$)



Exercise

Show that the Farthest-First Traversal algorithm can be implemented to run in $O(N \cdot k)$ time.

Hint: make sure that in each iteration i of the for-loop each point $p \in P - S$ knows its closest center among $c_1, c_2, \ldots, c_{i-1}$ and the distance from such a center.

Farthest-First Traversal: analysis

Theorem

Let S be the set of centers returned by running Farthest-First Traversal on P. Then:

$$\Phi_{\text{kcenter}}(P, S) \leq 2 \cdot \Phi^{\text{opt}}_{\text{kcenter}}(P, k).$$

That is, Farthest-First Traversal is a 2-approximation algorithm.

Proof of Theorem

Let $S = \{c_1, c_2, \ldots, c_k\}$ $\quad c_i \equiv$ i-th center discovered

Obs * $c_2 \equiv$ farthest point of P from c_1

* $c_3 \equiv$ farthest point of P from $\{c_1, c_2\}$

We have: $d(c_2, c_1) \geq d(c_3, c_1) \geq d(c_3, \{c_1, c_2\})$

$\uparrow$ by first obs.

$d(c_3, \{c_1, c_2\}) = \min\{d(c_3, c_1), d(c_3, c_2)\}$

Similarly, we have

$$d(c_3, \{c_1, c_2\}) \geq d(c_4, \{c_1, c_2\}) \geq d(c_4, \{c_1, c_2, c_3\})$$

Proof of Theorem

If we iterate the argument, we get

$$d(c_1, c_2) \geq$$

$$d(c_3, \{c_1, c_2\}) \geq$$

$$d(c_4, \{c_1, c_2, c_3\}) \geq$$

$$\vdots$$

$$d(c_i, \{c_1, c_2, \ldots, c_{i-1}\} \geq$$

$$\vdots$$

$$d(c_k, \{c_1, c_2, \ldots, c_{k-1}\}) \geq$$

$$d(q, \{c_1, c_2, \ldots, c_k\}) = d(q, S)$$

where q = farthest point of P from S

Proof of Theorem

Consequently

(a) $d(q, S) = \phi_{\text{kcenter}}(P, S)$

(b) If we call $q \triangleq c_{k+1}$ then $\forall i, j \quad 1 \le i < j \le k+1$

$$d(c_j, c_i) \ge d(c_j, \{c_1, \ldots, \underbrace{c_i}, \ldots, c_{j-1}\}) \ge d(q, S) = \phi_{\text{kcenter}}(P, S)$$

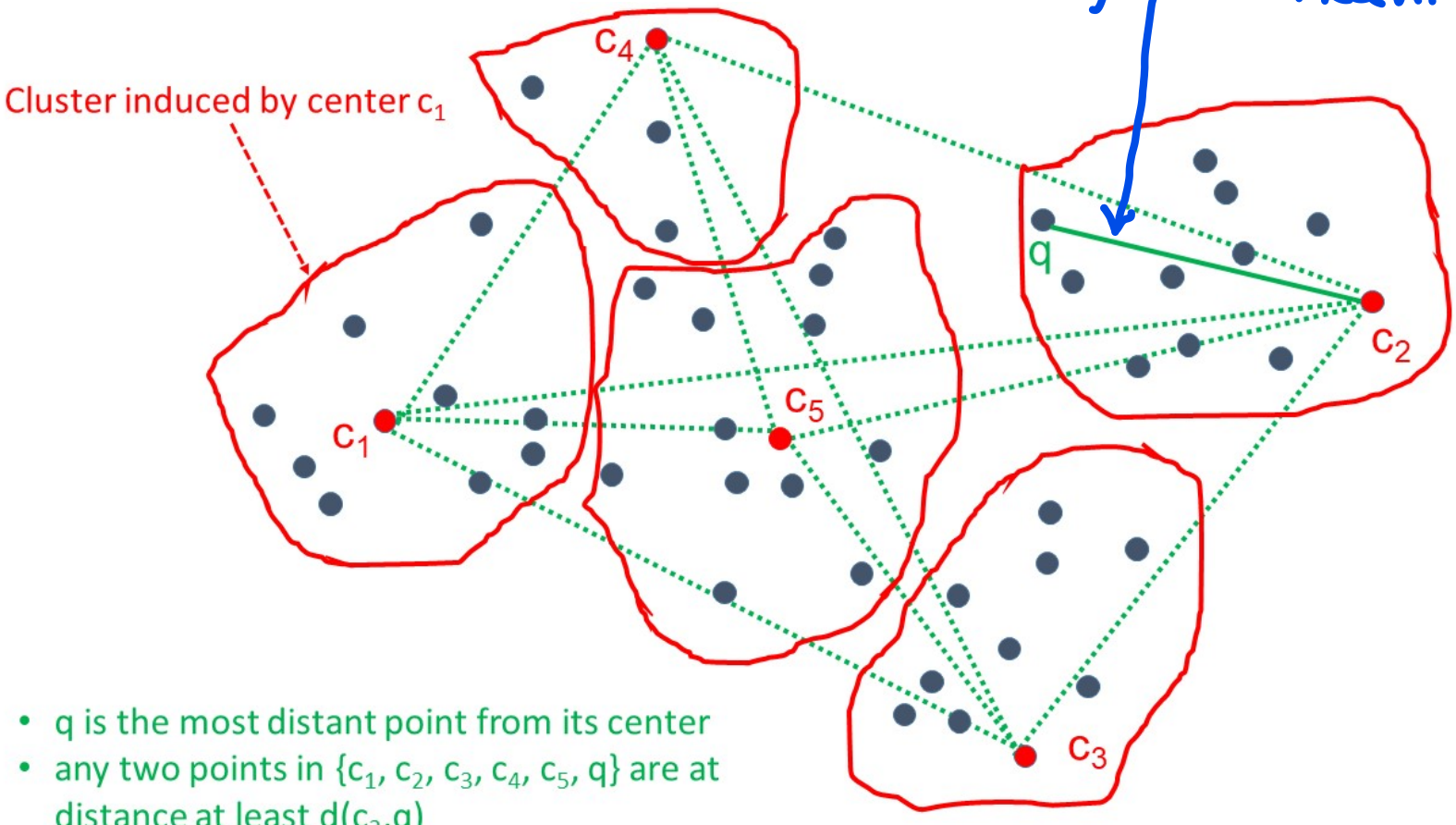
$\Rightarrow$ we have $k+1$ points $(\{c_1, c_2, \ldots, c_k, q\})$ at distance $\ge \phi_{\text{kcenter}}(P, S)$ from one another

Proof of Theorem

Observation. The fact that the centers discovered by Farthest-First Traversal are "well spaced" to one another is a crucial property that turns out useful for other applications: e.g. diversity maximization

Proof of Theorem

$d(q, c_2) = \phi_{kcenter}(P, S)$



- q is the most distant point from its center
- any two points in $\{c_1, c_2, c_3, c_4, c_5, q\}$ are at distance at least $d(c_2,q)$

Proof of Theorem

Let S^* be the optimal set of centers

$S^* = \{c_1^*, c_2^*, \ldots, c_k^*\}$

- $\phi^{OPT}_{k\text{center}}(P,k) = \phi_{k\text{center}}(P, S^*) \leq \phi_{k\text{center}}(P,S)$

- $\forall x \in P \quad d(x, S^*) \leq \phi^{OPT}_{k\text{center}}(P,k)$

Define:

$C_i^* = \{x \in P : c_i^* \text{ is the center of } S^* \text{ closest to } x\}$

$1 \leq i \leq k$ with ties broken arbitrarily

Proof of Theorem

$\{C_1^*, \ldots, C_k^*\}$ — k clusters, $\bigcup C_i^* = P$

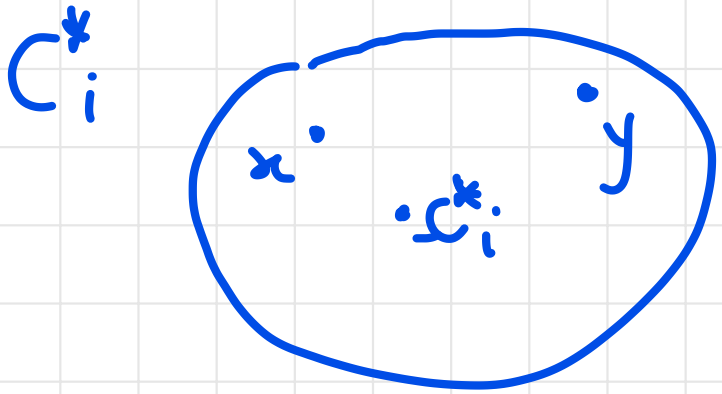
$\{c_1, c_2, \ldots, c_k, q\}$ — $k+1$ points of P

$\Rightarrow$ by pigeonhole principle, two points in $\{c_1, c_2, \ldots, c_k, q\}$ must fall in the same C_i^*

Call them x, y and recall that

$d(x, y) \geq \Phi_{\text{karate}}(P, S)$

Proof of Theorem



Thus $\phi_{\text{kcenter}}(P,S) \le d(x,y)$

$\le d(x,c_i^*) + d(c_i^*,y)$ by triangle inequality

$\le 2\,\phi_{\text{kcenter}}(P,S^*)$

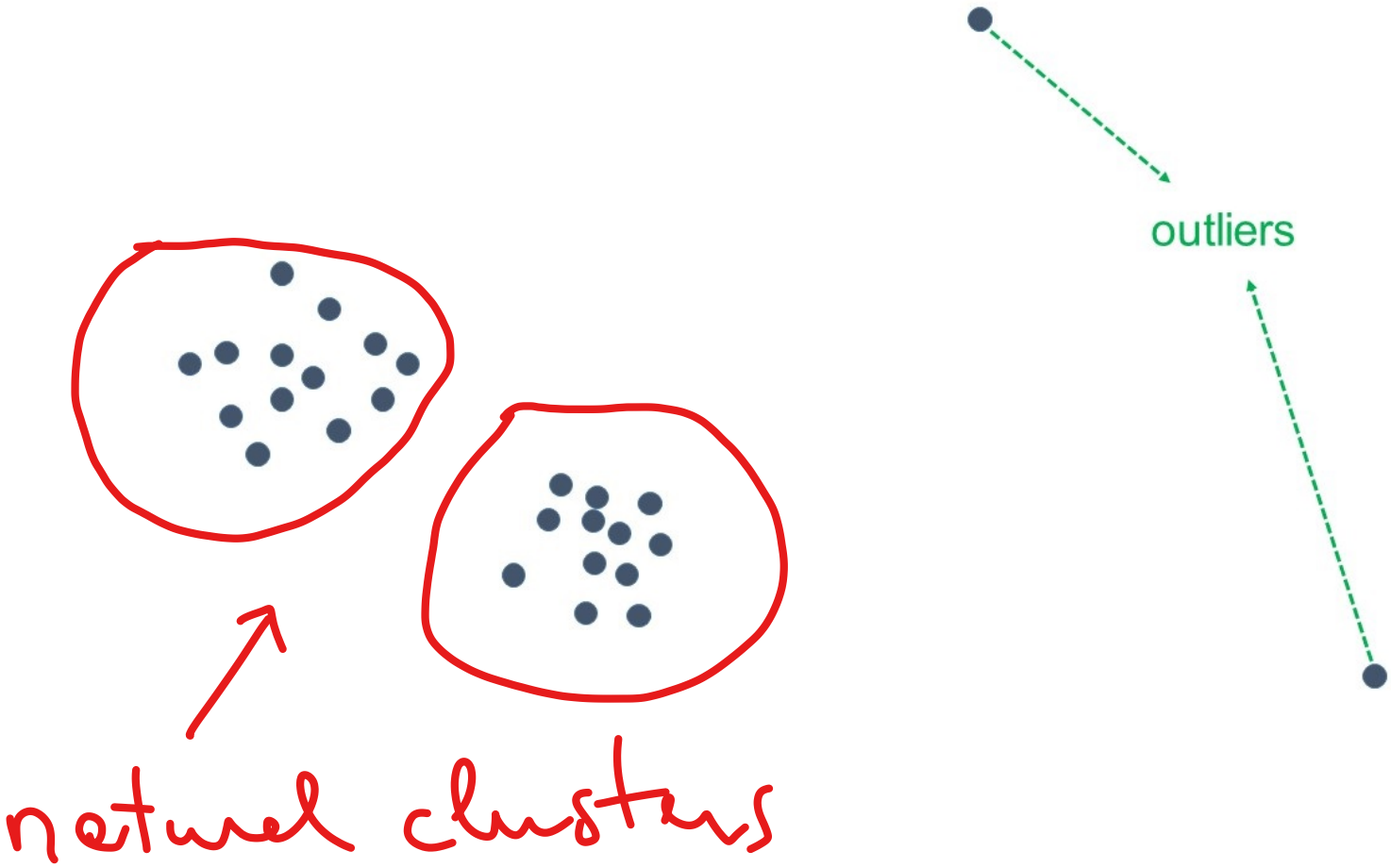
$= 2\,\phi^{\text{OPT}}_{\text{kcenter}}(P,k)$ by optimality of S^*

□

Observations on k-center clustering

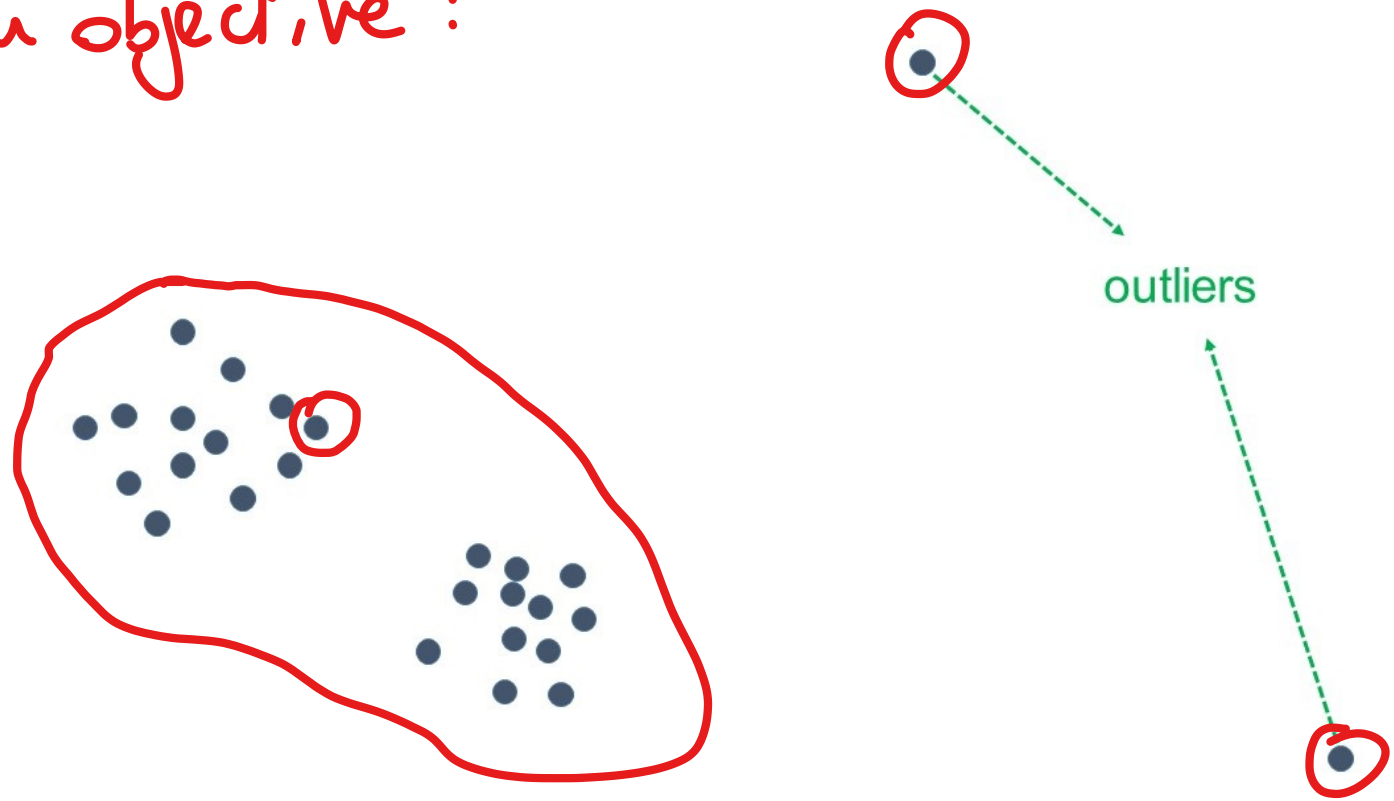
- The k-center objective focuses on worst-case distance of points from their closest centers.
- Farthest-First Traversal's approximation guarantees are almost the best one can obtain in practice. It was proved that computing a c-approximate solution to k-center is NP-hard for any fixed $c < 2$.
- The k-center objective is very sensitive to noise. For noisy datasets (e.g., with outliers), the clustering which optimizes the k-center objective may obfuscate some "natural" clustering inherent in the data. See example in the next slide.

Example: noisy pointset



Example: noisy pointset

Best 3 centers for this pointset w.r.t. the k-center objective:



⇒ Solving k-center with $k=3$ does not separate the 2 natural clusters

k-center clustering for big data

Observation. Farthest-First Traversal requires $k-1$ scans of the pointset P with P stored in main memory: impractical for massive P and k not so small.

How can we compute a "good" solution to k-center for a pointset P which is too large for a single machine?

Coreset technique

Suppose that we want to solve a problem Π for a massive input P.

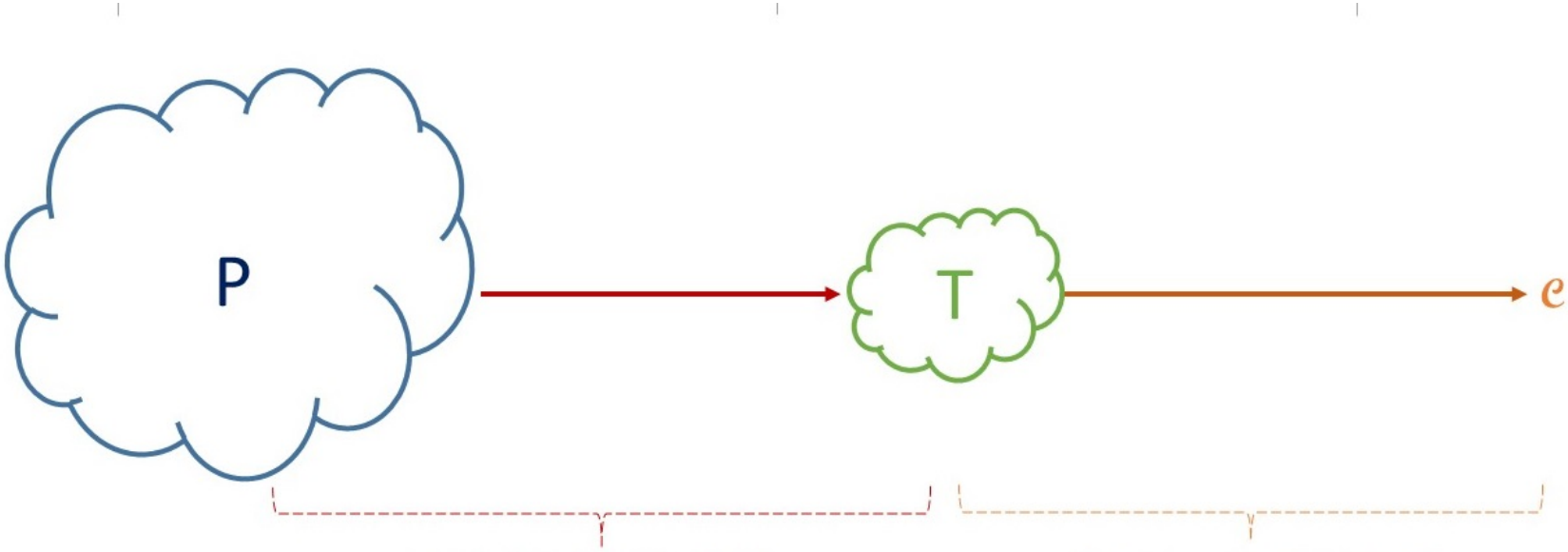
Coreset technique

1. Extract a "small" subset T from P (dubbed coreset), making sure that it represents P well. w.r.t. solutions to Π
2. Run best known (possibly slow) sequential algorithm for Π on the small coreset T, rather than on the entire input P.

The technique is effective if

- T can be extracted efficiently by processing P either on a distributed platform or as a stream.
- The solution computed on T is a good solution for Π w.r.t. the entire input P.

Coreset technique



Composable Coreset technique

Composable Coreset technique

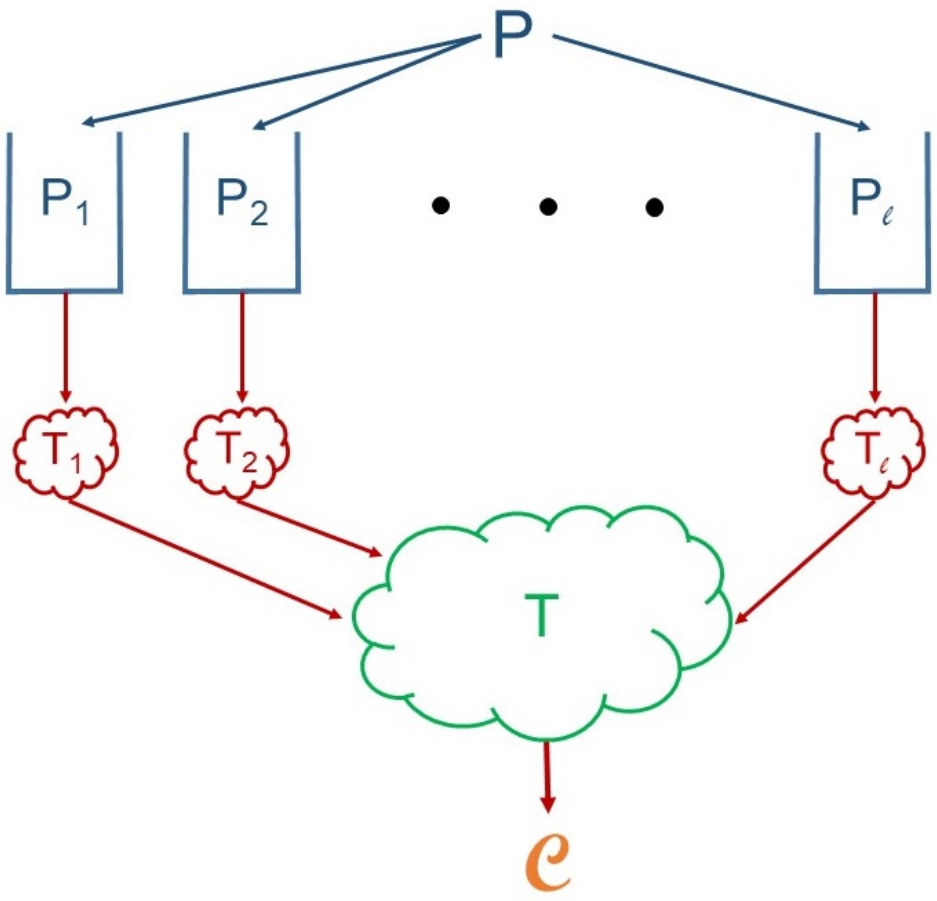
1. Partition P into ℓ subsets $P_1, P_2, \ldots, P_\ell$, and extract a "small" coreset T_i from each P_i, making sure that it represents P_i well.
2. Run best known (sequential) algorithm for Π, possibly expensive, on $T = \cup_{i=1,\ell} T_i$, rather than on P.

$T \equiv$ coupletion of T_i

The technique is effective if

- Each T_i can be extracted efficiently from P_i (in parallel for all i's).
- The final coreset T is still small and the solution computed on T is a good solution for Π w.r.t. the entire input P.

Composable coreset technique



- Extract a *small coreset* T_i from each P_i
- Apply best known sequential algorithm to final coreset T = union of the T_i's, to compute final solution $\mathcal{C}$

Application to k-center clustering

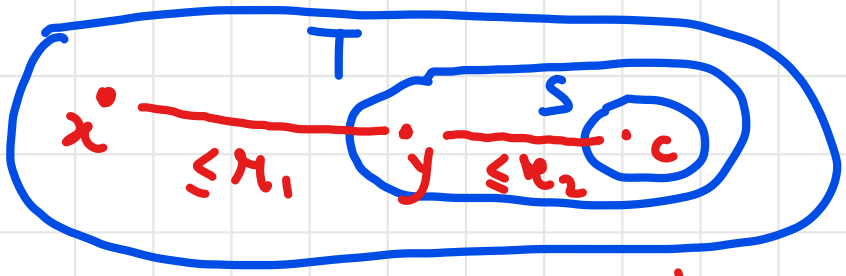
Main ideas

* input instance: (P, k) $\quad |P| = N$
* $P \longrightarrow P_1, P_2 \ldots P_\ell \quad |P_i| \approx N/\ell$
* Extract T_i from P_i using Farthest-First Traversal $\Rightarrow |T_i| = k \quad (\text{if } k \leq N/\ell)$
* let $T = \bigcup_{i=1}^{k} T_i \Rightarrow |T| = k \cdot \ell$
* Extract final solution S from T using again Farthest-First Traversal

Application to k-center clustering

Let $r_1 = \max_{1 \le i \le l} \phi_{\text{kcenter}}(P_i, T_i)$

$r_2 = \phi_{\text{kcenter}}(T, S)$ $\quad T = \bigcup_{i=1}^{l} T_i$



$\forall x \in P \;\; \exists y \in T : d(x,y) \le r_1$

$\forall y \in T \;\; \exists c \in S : d(y,c) \le r_2$

$\Rightarrow \forall x \in P \;\; \exists c \in S : d(x,c) \le r_1 + r_2$ → 4-approx.

We will show: $r_1, r_2 \le 2\,\phi^{OPT}_{\text{kcenter}}(P,k)$

MapReduce-Farthest-First Traversal

Let P be a set of N points (N large!) from a metric space (M, d), and let $k > 1$ be an integer.

Exercise: fill-in details of key-value pairs

Algorithm MR-Farthest-First Traversal

- **Round 1:**
  - Map Phase: Partition P arbitrarily into ℓ subsets of equal size $P_1, P_2, \ldots, P_\ell$.
  - Reduce Phase: for every $i \in [1, \ell]$ separately, run Farthest-First Traversal on P_i to determine a set $T_i \subseteq P_i$ of k centers.
- **Round 2:**
  - Map Phase: empty.
  - Reduce Phase: gather the coreset $T = \cup_{i=1}^{\ell} T_i$ (of size $\ell \cdot k$) and run, using a single reducer, Farthest-First Traversal on T to determine a set $S = \{c_1, c_2, \ldots, c_k\}$ of k centers, and return S as output.

Analysis of MR-Farthest-First Traversal

We analyze first M_A and M_L

input : $\{(ID_x, x) : x \in P \text{ and } ID_x \in [0, N)\}$

Output : $\{(o, c) : c \in S\}$

Also assume ℓ is selected so that $k \leq N/\ell$

M_A : $\Theta(N)$ since in each round the number of input, output and intermediate pairs is $O(N)$

M_L {
- R1: $O(N/\ell)$ needed to store a partition P_i.
- R2: $O(\ell \cdot k)$ needed to store T

Analysis of MR-Farthest-First Traversal

$\Rightarrow M_L : O\left(\max\left\{\frac{N}{\ell}, \ell k\right\}\right)$

$\frac{N}{\ell} = \ell k \Leftrightarrow \ell^2 = \frac{N}{k} \Leftrightarrow \ell = \sqrt{N/k}$

$\Rightarrow M_L = O\left(\sqrt{N \cdot k}\right)$ observe that: if $k = o(N)$ then $M_L = o(N)$

In particular: if k is constant w.r.t. N then $M_L = O(\sqrt{N})$

Analysis of MR-Farthest-First Traversal

Let T be the union of the coresets T_i computed by MR-Farthest-First Traversal on input P. The following lemma establishes the quality of T.

Lemma

For every $x \in P$ we have

$$d(x, T) \leq 2 \cdot \Phi^{\text{opt}}_{\text{kcenter}}(P, k).$$

$\Rightarrow$ T is a good representative for P with respect to the k-center objective

Proof of Lemma

Consider an arbitrary partition P_j for some $1 \le j \le l$

* $T_j \equiv$ set of k centers extracted from P_j with FFT
* $q_j \equiv$ point of P_j farthest from T_j

$$d(q_j, T_j) = \max_{x \in P_j} d(x, T_j)$$

By repeating the same argument used in the analysis of FFT we can show that $T_j \cup \{q_j\}$ is a set of $k+1$ points of P_j such that any 2 of them are at distance $\ge d(q_j, T_j)$ from one another

Proof of Lemma

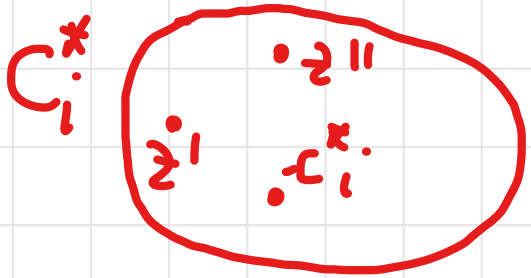
Let $S^* = \{c_1^*, c_2^*, \dots, c_k^*\}$ be the optimal solution of k-center for P, and let C_i^* be the optimal cluster around c_i^*, with $1 \le i \le k$

Hence, $d(x, c_i^*) \le \Phi^{OPT}_{kcenter}(P, k) \quad \forall x \in C_i^*$

Since $P_j \subseteq P$ and $T_j \cup \{q_j\}$ is a set of $k+1$ points there must exist an optimal cluster C_i^* where 2 points of $T_j \cup \{q_j\}$ fall

Call z', z'' these 2 points

Proof of Lemma



Hence,

$$d(q_j, T_j) \leq d(z', z'') \leq d(z', c_i^*) + d(c_i^*, z'')$$
$$\leq 2\, \Phi^{OPT}_{kcenter}(P, k)$$

$$\Rightarrow \forall x \in P_j : d(x, T_j) \leq d(q_j, T_j) \leq 2\, \Phi^{OPT}_{kcenter}(P, k)$$

and this is true for every partition P_j.

$$\Rightarrow \forall x \in P \quad \text{if } x \in P_j \quad \text{we have that}$$

$$d(x, T = \bigcup_j T_j) \leq d(x, T_j) \leq 2\, \Phi^{OPT}_{kcenter}(P, k) \qquad \square$$

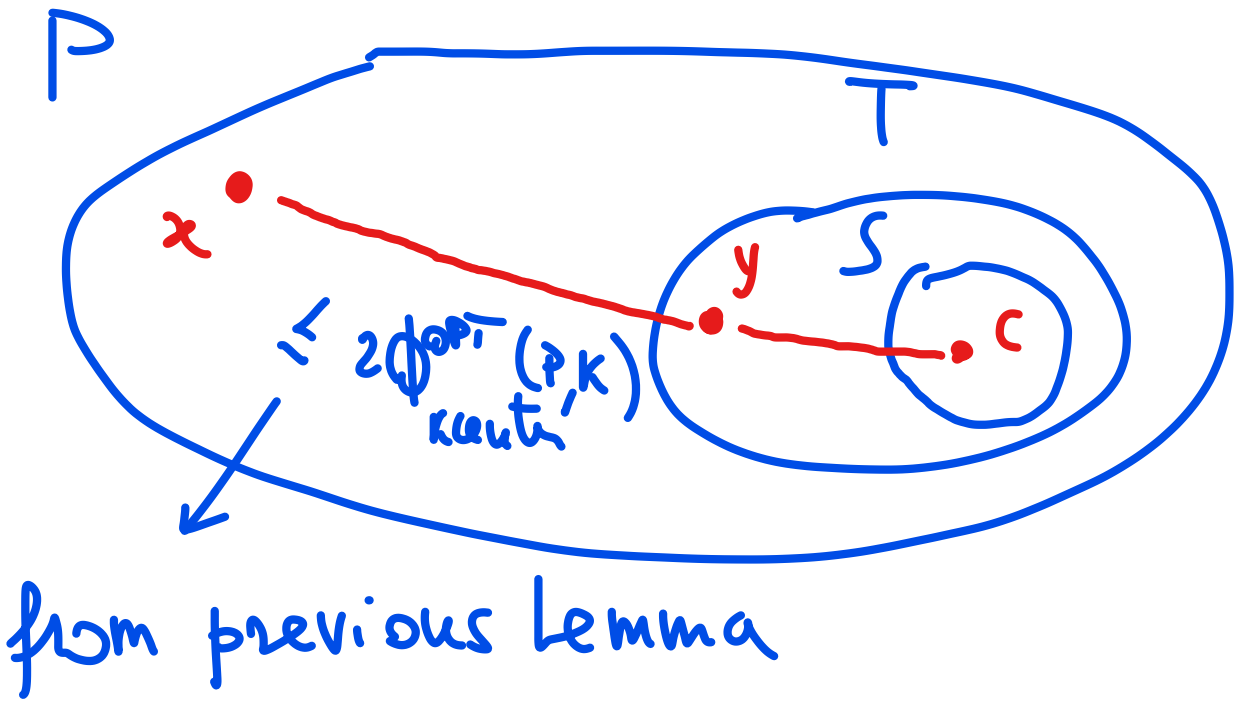
Analysis of MR-Farthest-First Traversal

Theorem

Let S be the set of k centers returned by running MR-Farthest-First Traversal on P. Then:

$$\Phi_{\text{kcenter}}(P, S) \leq 4 \cdot \Phi^{\text{opt}}_{\text{kcenter}}(P, k).$$

That is, MR-Farthest-First Traversal is a 4-approximation algorithm.



Proof of Theorem

We are left to show that for each $y \in T$ $\exists c \in S$ such that $d(y,c) \leq 2\phi^{OPT}_{\text{kcenter}}(P,k)$

If this is true, then $\forall x \in P$, letting y be the point of T closest to x and letting c be the point of S closest to y, we have

$$d(x,S) \leq d(x,c) \leq d(x,y) + d(y,c) \leq 4\phi^{OPT}_{\text{kcenter}}(P,k)$$

and the theorem will follow.

Proof of Theorem

$S \equiv$ set of k centers computed by FFT on T

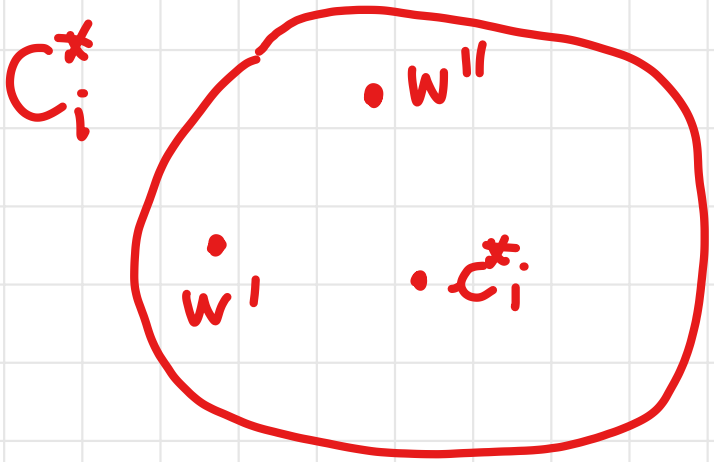
$q \equiv$ point of T farthest from S

$\Rightarrow \forall y \in T \quad d(y, S) \leq d(q, S)$

By the same argument used in the analysis of FFT we can show that the set $S \cup \{q\}$ contains $k+1$ points which are at distance $\geq d(q, S)$ from one another

Proof of Theorem

2 of these points (say w', w'') will fall in the same optimal cluster C_i^* (of the optimal solution for $\mathcal{T}$)



$$d(q,s) \le d(w',w'') \le$$
$$\le d(w', c_i^*) + d(c_i^*, w'')$$
$$\le 2\,\phi^{OPT}_{k\text{center}}(P,k)$$

$$\Rightarrow \forall y \in T \quad d(y,S) \le d(q,S) \le 2\phi^{OPT}_{k\text{center}}(P,k)$$

□

Observations on MR-Farthest-First Traversal

- Farthest-First Traversal provides good coresets T_i's, hence a good final coreset T since it ensures that any point not belonging to T is well represented by some coreset point.
- MR-Farthest-First Traversal is able to handle very large pointsets and the the final approximation is not too far from the best achievable one.

Low-dimensional pointsets

When P has low dimensionality the quality of the solution returned by MR-Farthest-First Traversal can be made arbitrarily close to 2 by selecting a slightly larger coreset, while still ensuring sublinear local space and linear aggregate space.

When P has low dimensionality, the distance between the next center selected by FFT and the previously selected centers decreases somewhat sharply. Based on this property, if in each partition P_j we select $k' > k$ centers instead of k, we get a much better coreset even if k' is not much larger than k

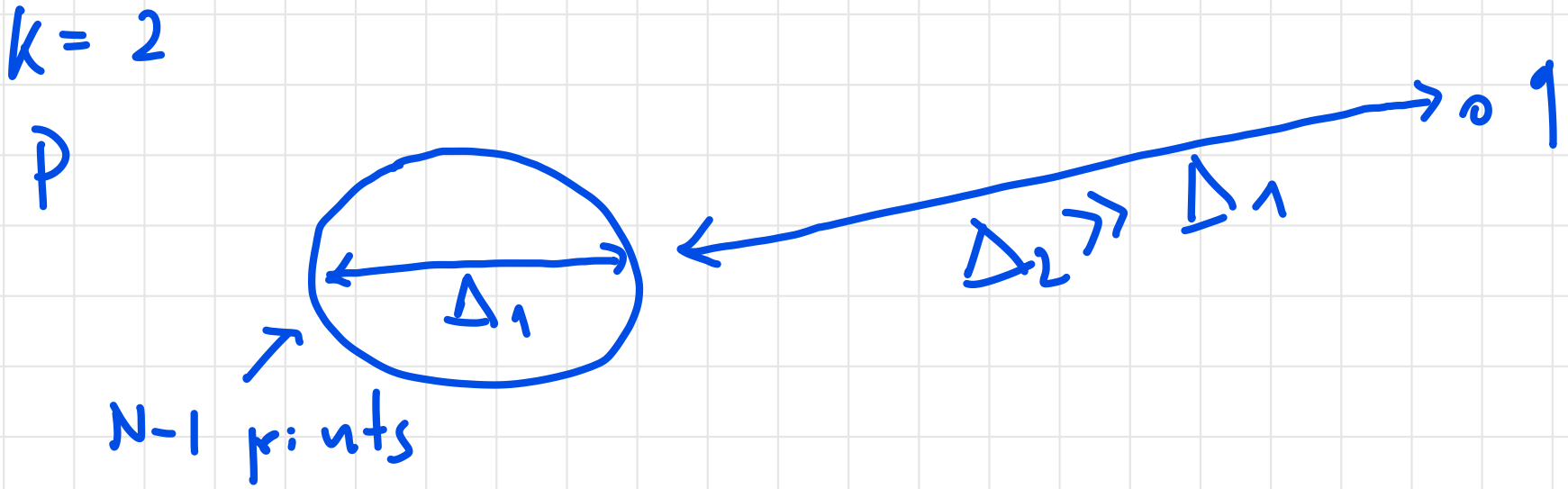
Does a random sample provide a good coreset?

Let $T \subseteq P$ be a coreset of $|T| = \sqrt{Nk}$ points, selected at random from P independently, with replacement and with uniform probability. Consider a set S of k centers computed by running Farthest-First Traversal on T.

Is S is ~~not~~ a good solution to k-center on P?

Does a random sample provide a good coreset?

$k = 2$



$$Prob(q \in T) = \sqrt{N \cdot k}\ \frac{1}{N} = \sqrt{\frac{k}{N}} \xrightarrow[N \to \infty]{} 0$$

No matter how we select a set S of 2 centers from T, we have that, with high probability,

Does a random sample provide a good coreset?

$\phi_{kcenter}(P,S) \approx \Delta_2$ while $\phi^{opt}_{kcenter}(P,2) \approx \Delta_1$

hence, the approximation ratio can be arbitrarily large ($\approx \Delta_2/\Delta_1$)

In general, if the natural clusters in the data (considering also outliers) are not well balanced, then random sampling may not work well

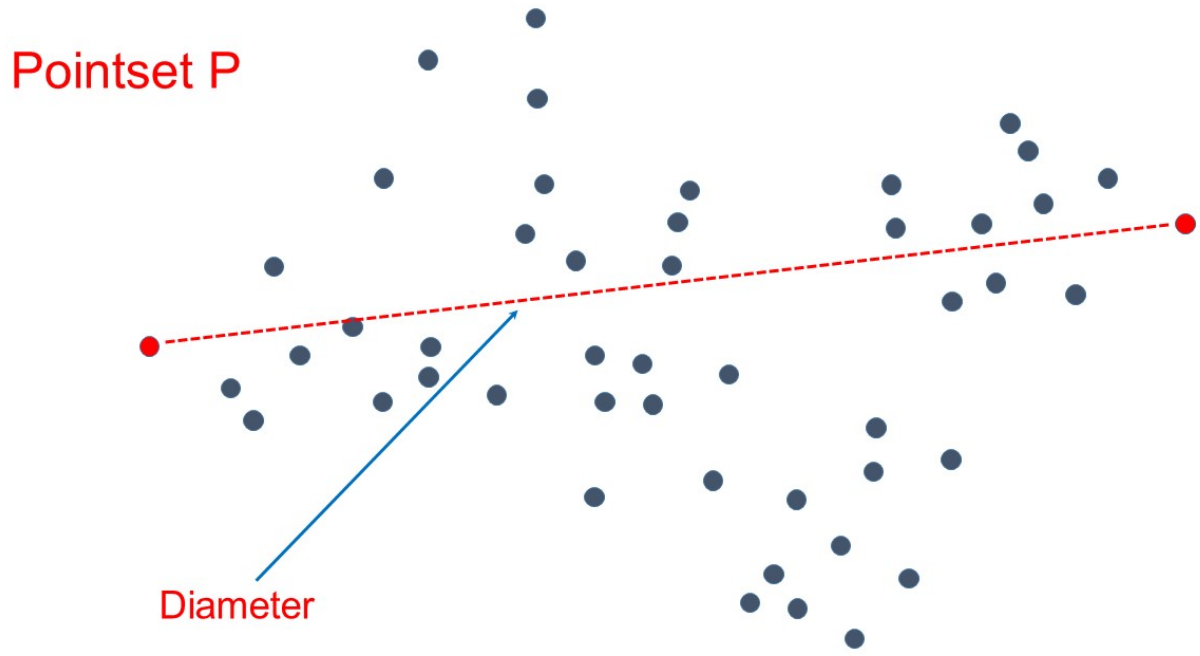
Case study: diameter of a pointset

Diameter

Problem: given a set P of N points from a metric space (M, d) determine its diameter

$$d_{\max} = \max_{x,y \in P} d(x, y),$$

i.e., the maximum distance between two points.



Exact diameter computation

in $\mathbb{R}^3$ it can be computed in time $O(N \log^2 N)$

Sadly, the computation of the exact diameter requires almost qudratic operations (except for special cases such as the low-dimensional Euclidean spaces), hence it is impractical for very large pointsets.

Exercise

Design an MR algorithm to compute the exact diameter of a set P of N points, which requires $R = O(1)$, $M_L = O\left(\sqrt{N}\right)$ and $M_A = O\left(N^2\right)$. Assume that P is initially provided as the set of pairs (i, x_i), for $0 \leq i < N$, where the x_i's are the points.

2-approximation to the diameter

For an arbitrary $x_i \in P$ define

$$d_{\max}(i) = \max\{d(x_i, x_j) \ : \ 0 \leq j < N\}.$$

Lemma

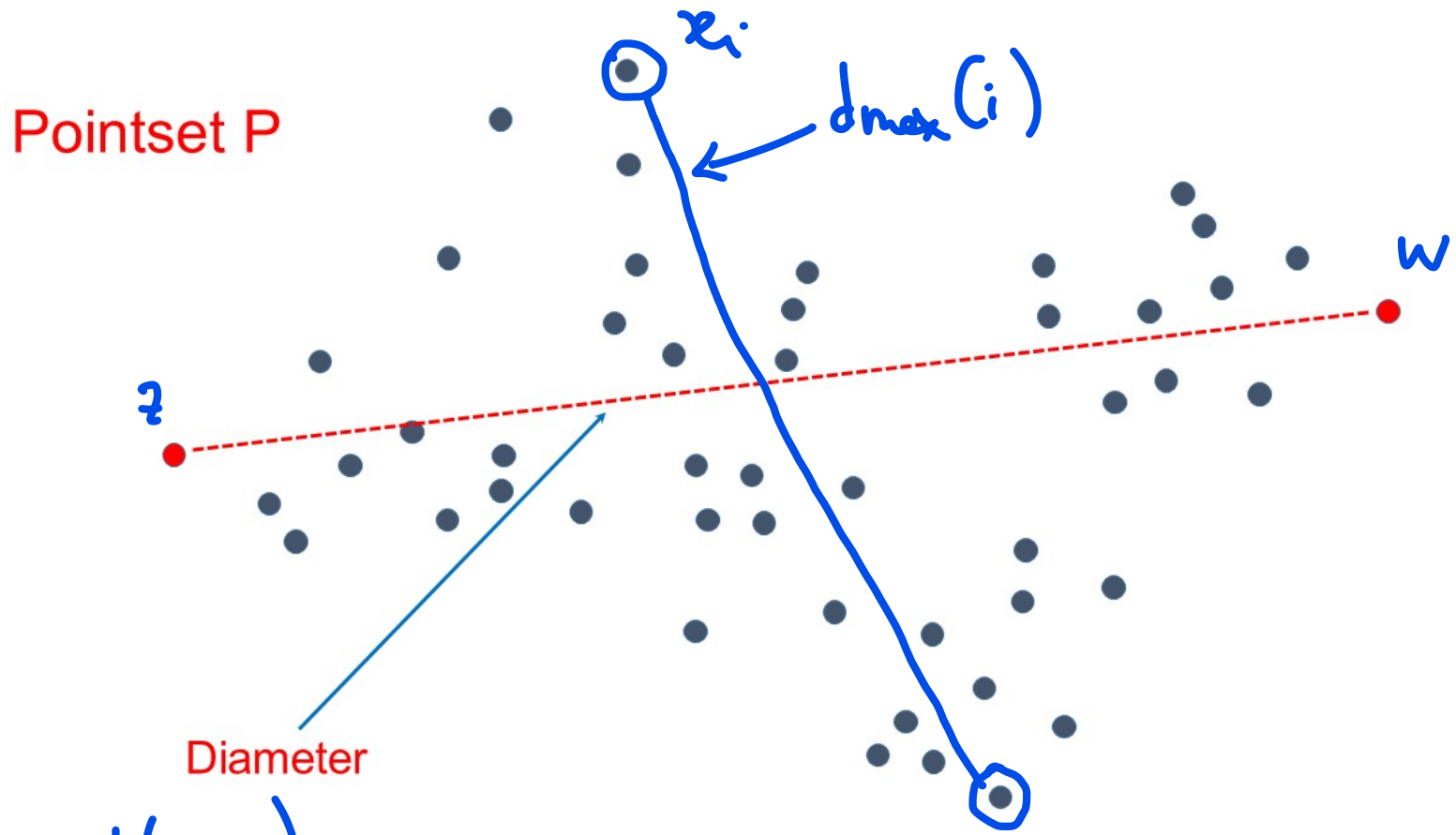
For any $0 \leq i < N$ we have $d_{\max} \in [d_{\max}(i), 2d_{\max}(i)]$.

exact diameter

Exercise

Show that for any arbitrary $0 \leq i < N$, the value $d_{\max}(i)$ can be computed in MapReduce using 2 rounds, $M_L = O\left(\sqrt{N}\right)$ and $M_A = O(N)$.

Proof of Lemma



$d_{max} = d(z, w)$

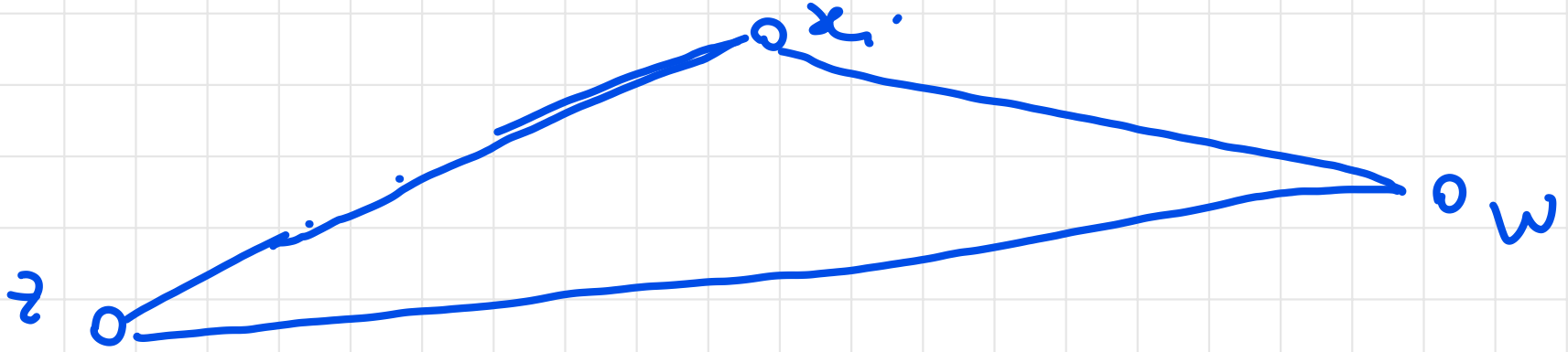
Proof of Lemma

* $d_{max} \geq d_{max}(i)$ Trivial since

$$d_{max} = \max_{x,y \in P} d(x,y)$$

$$d_{max}(i) = \max_{y \in P} d(x_i, y)$$

* $d_{max} \leq 2 d_{max}(i)$



Proof of Lemma

$$d_{max} = d(z,w)$$
$$\leq d(z,x_i) + d(x_i,w)$$
$$\leq 2\, d_{max}(i)$$

□

Better, coreset-based diameter approximation

Coreset-based approximation:

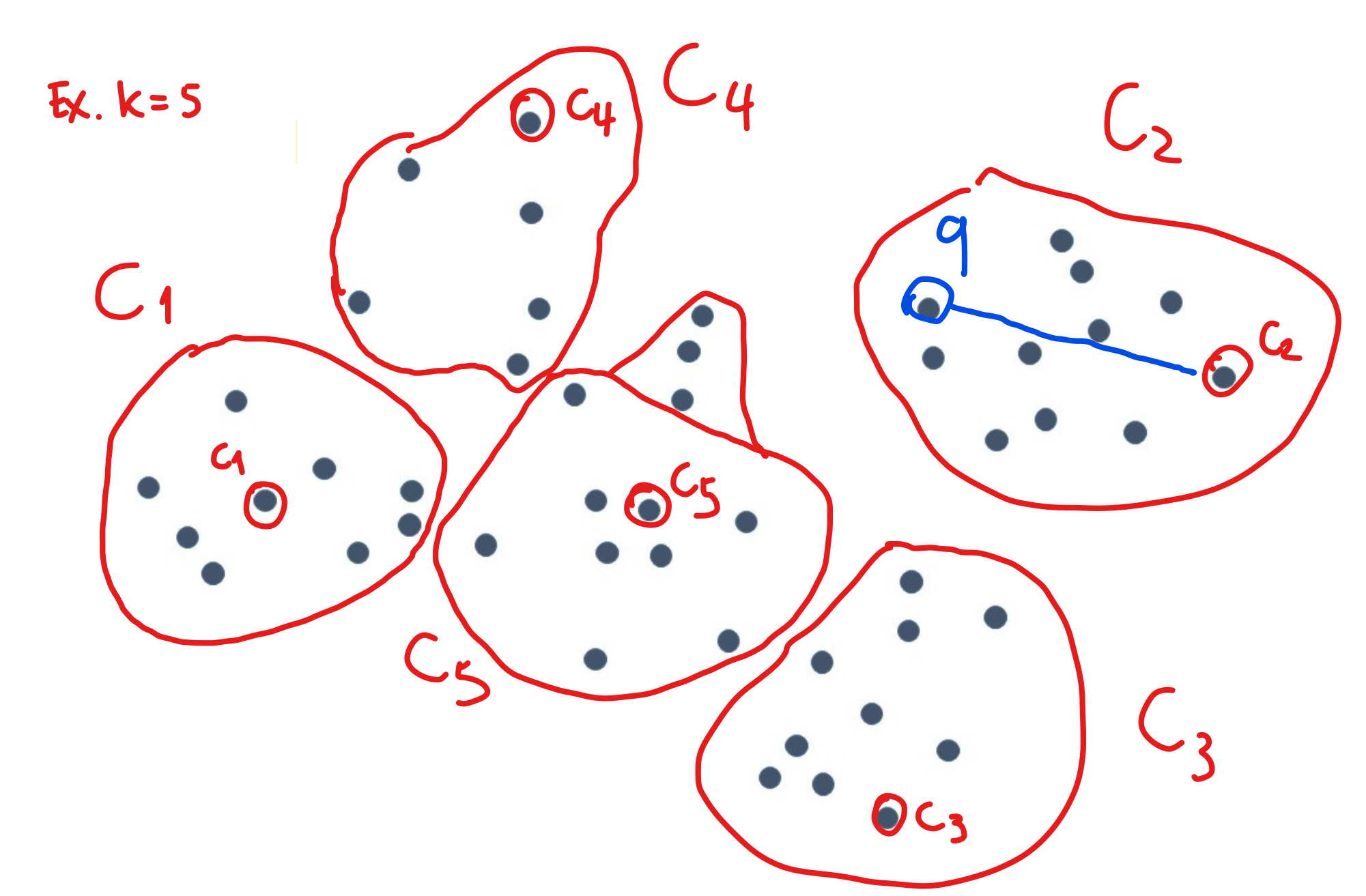
- Fix a suitable $k \geq 2$.
- Extract a coreset $S \subset P$ of size k by running a k-center clustering algorithm on P and taking the k cluster centers as set S.
- Return $d_S = \max_{x,y \in S} d(x,y)$ as an approximation of $d_{\max}$.

Can the approximation be computed efficiently?

If $k = O(1)$, d_S can be computed

- Sequentially: in $O(N)$ time (using Farthest-First Traversal)
- In MapReduce: in 2 rounds, with local space $M_L = O\left(\sqrt{N}\right)$ and aggregate space $M_L = O(N)$ (through MR-Farthest-First Traversal)

Is d_S a good approximation of $d_{\max}$?


Ex. k=5
C4
C4
C2
C1
q
c2
c1
c5
C5
C3
c3

$S = \{c_1, c_2, \ldots, c_k\}$

$q \equiv$ point of P furthest from S

defn $R \triangleq d(q, S)$

As before, let z, w be the points such that $d_{max} = d(z, w)$

c_i = closest center to z

c_j = closest center to w

By iterating the triangle inequality, you get

$$d(z,w) \leq d(z,c_i) + d(c_i,c_j) + d(c_j,w)$$

$$d_{max} \leq 2R + d_S$$

$$\Rightarrow d_S \geq d_{max} - 2R$$

$\Rightarrow d_{max} - 2R \leq d_S \leq d_{max}$

Obs. The value R decreases as k grows longer and when it becomes negligible with respect to d_{max} then d_S becomes a very tight estimate of the diameter

When P belongs to a metric space of low dimensionality, a constant k is sufficient to get an estimate of dmax very close to the actual value

Exercise

Let P be a set of N points in a metric space (M, d), and let $T \subseteq P$ be a coreset of $|T| > k$ points such that for each $x \in P$ we have $d(x, T) \leq \epsilon \Phi^{\text{opt}}_{\text{kcenter}}(P, k)$, for some $\epsilon \in (0, 1)$. Let S be the set of k centers obtained by running the Farthest-First Traversal algorithm on T. Prove an upper bound to $\Phi_{\text{kcenter}}(P, S)$ as a function of ϵ and $\Phi^{\text{opt}}_{\text{kcenter}}(P, k)$.

Exercise

Let P be a set of points in a metric space (M, d), and let $T \subseteq P$. For any $k < |T|, |P|$, show that $\Phi^{\text{opt}}_{\text{kcenter}}(T, k) \leq 2\Phi^{\text{opt}}_{\text{kcenter}}(P, k)$. Is the bound tight?

Exercise

Let P be a set of N points in a metric space (M, d), and let $\mathcal{C} = (C_1, C_2, \ldots, C_k; c_1, c_2, \ldots, c_k)$ be a k-clustering of P. Initially, each point $q \in P$ is represented by a pair $(\mathrm{ID}(q), (q, c(q)))$, where $\mathrm{ID}(q)$ is a distinct key in $[0, N-1]$ and $c(q) \in \{c_1, \ldots, c_k\}$ is the center of the cluster of q.

1. Design a 2-round MapReduce algorithm that for each cluster center c_i determines the most distant point among those belonging to the cluster C_i (ties can be broken arbitrarily).
2. Analyze the local and aggregate space required by your algorithm. Your algorithm must require $o(N)$ local space and $O(N)$ aggregate space.

Exercise

Let P be a set of N *bicolored points* from a metric space, partitioned into k clusters $C_1, C_2, \ldots, C_k$. Each point $x \in P$ is initially represented by the key-value pair $(\mathrm{ID}_x, (x, i_x, \gamma_x))$, where ID_x is a distinct key in $[0, N-1]$, i_x is the index of the cluster which x belongs to, and $\gamma_x \in \{0, 1\}$ is the color of x.

1. Design a 2-round MapReduce algorithm that for each cluster C_i checks whether all points of C_i have the same color. The output of the algorithm must be the k pairs (i, b_i), with $1 \leq i \leq k$, where $b_i = -1$ if C_i contains points of different colors, otherwise b_i is the color common to all points of C_i.
2. Analyze the local and aggregate space required by your algorithm. Your algorithm must require $o(N)$ local space and $O(N)$ aggregate space.

References

LRU14 J. Leskovec, A. Rajaraman and J. Ullman. Mining Massive Datasets. Cambridge University Press, 2014. Sections 3.1.1 and 3.5, and Chapter 7

BHK18 A. Blum, J. Hopcroft, and R. Kannan. Foundations of Data Science. Manuscript, June 2018. Chapter 7

CPP19 M. Ceccarello, A. Pietracaprina and G. Pucci Solving k-center Clustering (with Outliers) in MapReduce and Streaming, almost as Accurately as Sequentially. Proc. VLDB Endow. 12(7): 766-778 (2019)