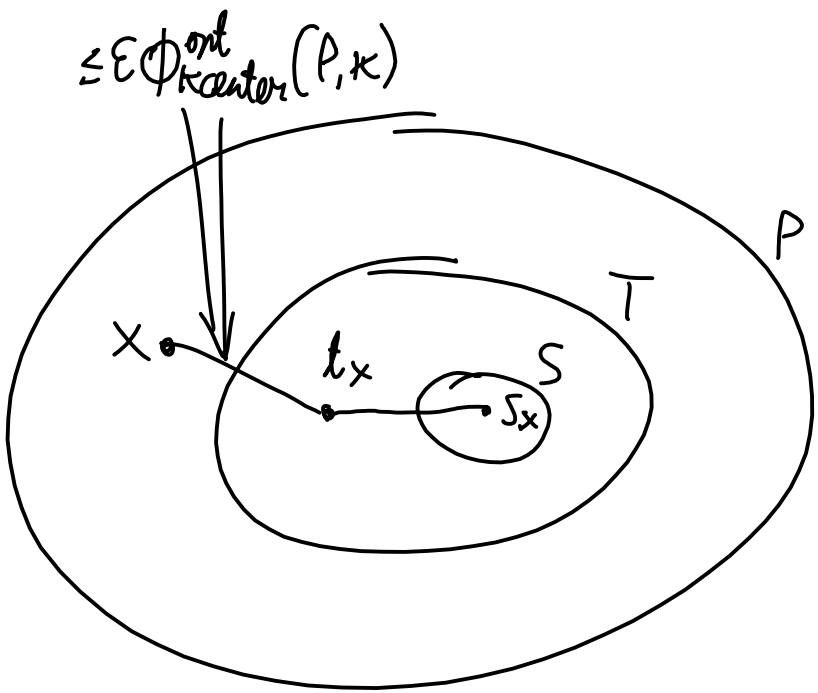
Coreset Technique

(Part 1 - Exercises)

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 $\Phi^{\text{opt}}_{\text{kcenter}}(P, k)$.



$$\Phi_{kcenter}(P,S) = f(\epsilon) \cdot \Phi^{opt}_{kcenter}(P,k)$$

$\forall x \in P$: $d(x,S) \leq f(\varepsilon) \cdot \phi^{opt}(P,k)$

t_x: punto + vicino a x in T
s_x: // // // a t_x in S

$$d(x,S) \leq d(x,s_x) \leq d(x,t_x) + d(t_x,s_x)$$
$$\leq \varepsilon\, \phi^{opt}(P,k) + 2\phi^{opt}(P,k)$$

$d(x,S) \leq (2+\varepsilon)\phi^{opt}(P,k) \Rightarrow (2+\varepsilon)$ approx.

Dimostriamo $\forall t \in i \in S$: $d(x,i) \leq 2\phi^{opt}(P,k)$
$s = \{s_1, \dots, s_k\} \Rightarrow s_i$: punto ricordato a it. i di FFT
q: punto da it. extra di FFT $\Rightarrow q := s_{k+1} \Rightarrow \hat{S} = S \cup \{q\} = \{s_1, \dots, s_{k+1}\}$
$\hat{C}^*$ = cluster ottimali

$\exists r_i, r_j,\ i<j$ | stesso cluster $\hat{C}_\ell^*$ con centro $\hat{c}_\ell^*$

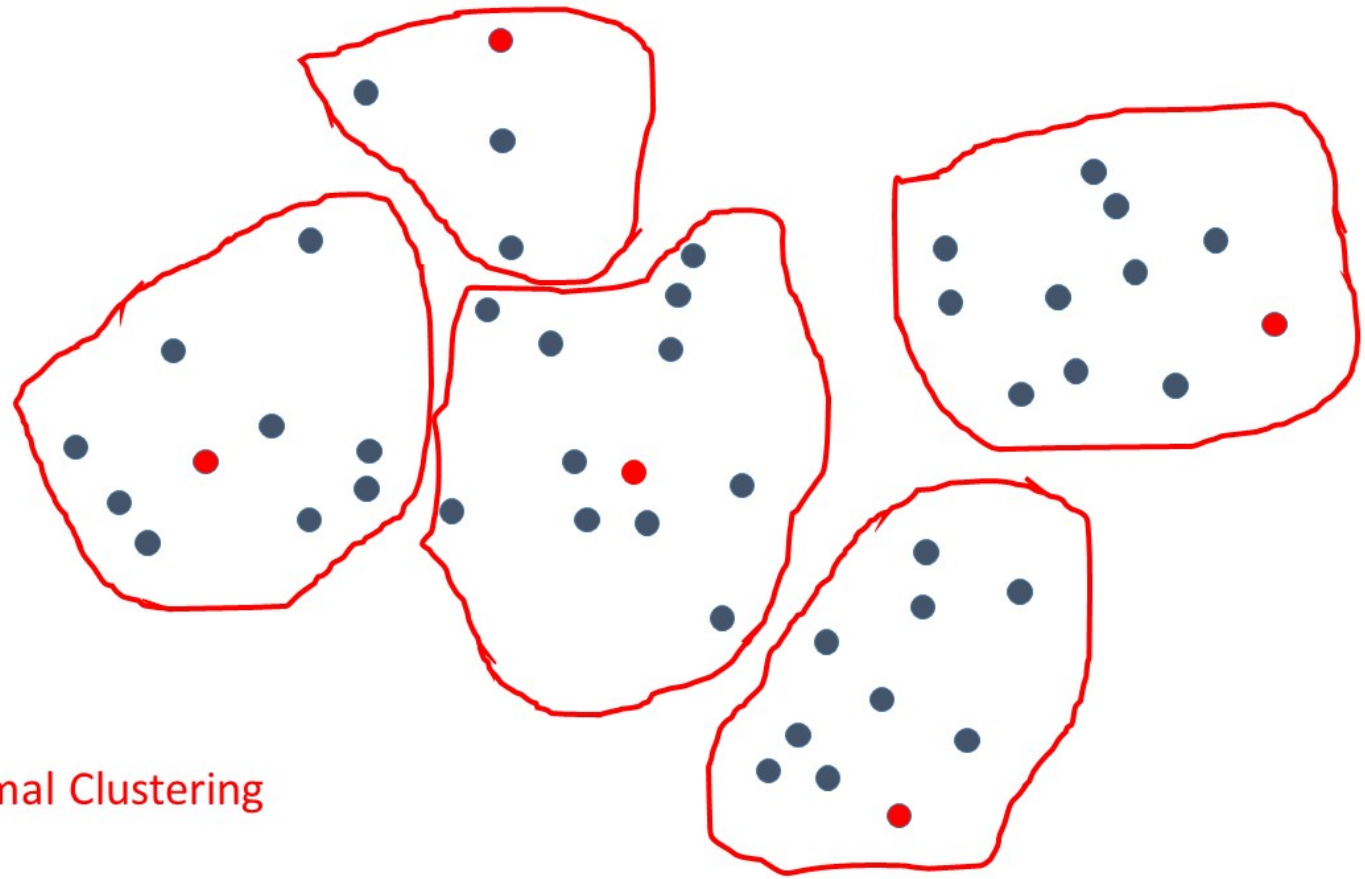
$d(r_i, r_j) \le d(r_i, \hat{c}_\ell^*) + d(r_j, \hat{c}_\ell^*) \le \phi^{opt}(P,k) + \phi^{opt}(P,k) \le 2\phi^{opt}(P,k)$

$\forall x \in T,\ d(x,S) = d(x, \{r_1, \dots, r_k\}) \le d(x, \{r_1, \dots, r_{j-1}\}) \le$

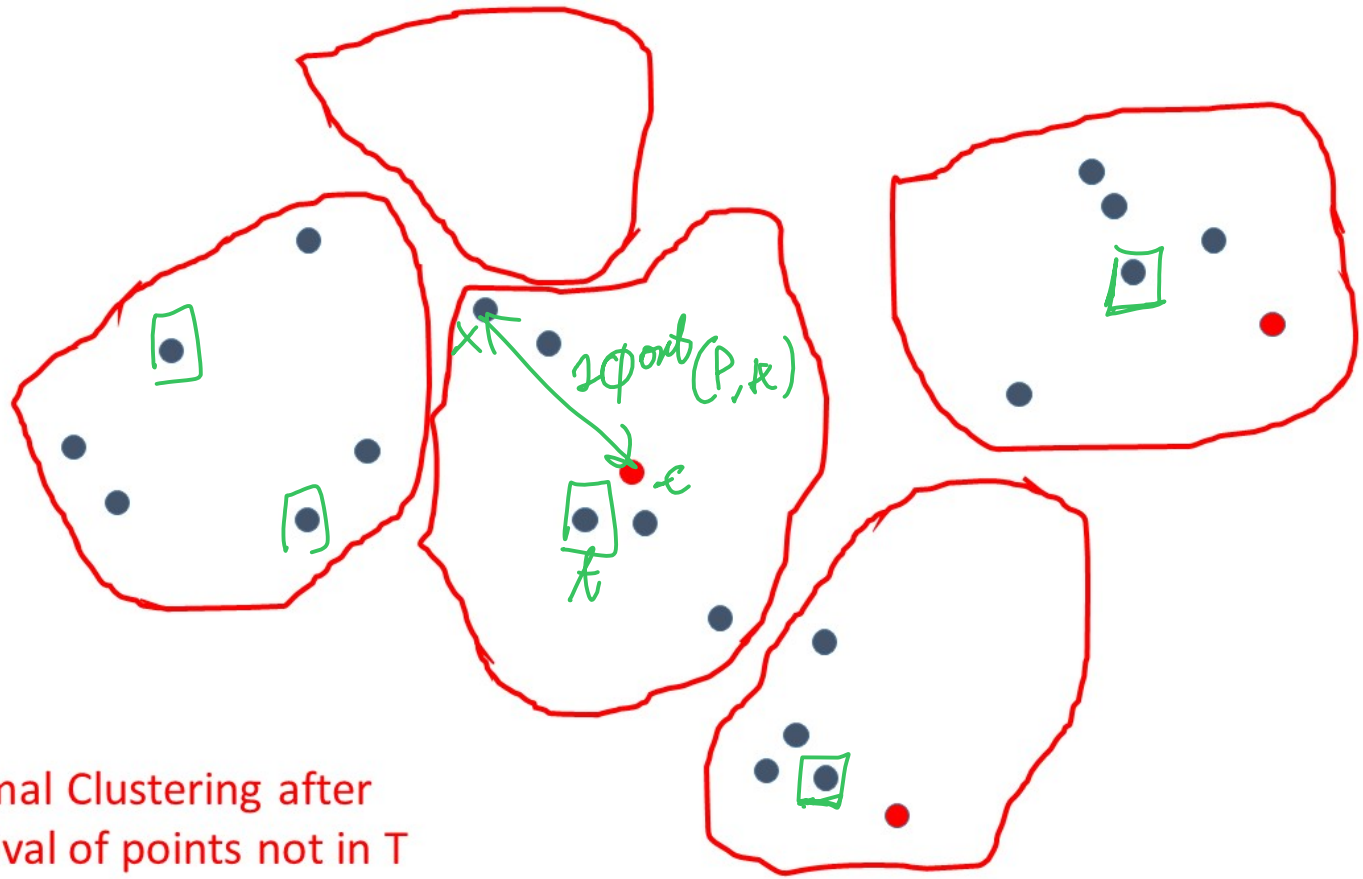
$\le d(r_j, \{r_1, \dots, r_{j-1}\}) \le d(r_j, r_i) \le 2\phi^{opt}(P,k)$

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?



Optimal Clustering



Optimal Clustering after
removal of points not in T

per ogni cluster, prendo punto
se ottengo k' < k punti, prendo altri k-k' } ottengo set T'

$$\phi^{ott}(T,k) \le \phi(T,\hat{T}) \le 2\phi^{ott}(P,k)$$

$\Downarrow$

sol. generata prendendo k punti

1) $C^* = \{C_1^*, \dots, C_k^*\}$ sol. ottima in P (centri $\hat{c}_1^*, \dots, \hat{c}_k^*$)
2) partiziono P usando C^*
3) tolgo tutti punti non in T
4) $\forall C_i^*$ prendo 1 punto in $T \Rightarrow \hat{t}_i$
5) $\hat{T} = \{\hat{t}_1, \dots, \hat{t}_k\}$, tutti punti $T_1 C_i^*$ assegnati a $\hat{t}_i$
6) $\phi^*(T,k) \le \phi(T,\hat{T})$

$$\forall t \in T \quad d(t,\hat{T}) \le d(t,\hat{t}_i) \le d(t,\hat{c}_i^*) + d(\hat{c}_i^*, \hat{t}_i) \le 2\phi^{ott}(P,k)$$

(con $t \in C_i^*$)

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.

si può mandare tutti punti di un cluster a un reducer

no garanzie su cluster: potremmo averne solo uno con tutti punti

$\Downarrow$

$M_L = \Theta(N)$

possiamo usare deterministic partitioning

ROUND 1

- Map: $(ID(q), (q, c(q))) \rightarrow (ID(q) \bmod \sqrt{N}, (q, c(q)))$
- Reduce: riceve L_i: tutte coppie con chiave i = valore di tutte coppie con chiave i
  - siano $\hat{c}_i$ centri presenti in almeno una coppia in L_i
  - $\forall c \in \hat{c}_i$ calcolo punto + lontano da punti in L_i con centro c
  - emetto $(c, q_c \; \forall c \in \hat{C}_i)$

ROUND 2:

- Map: niente
- Reduce: L_c valori da coppie con chiave c
  - calcolo $\max_{q \in L_c} d(q,c) = d_{c,MAX}$
  - emetto $(c, d_{c,MAX})$

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.