Lecture 7

TTIC 41000: Algorithms for Massive Data

Toyota Technological Institute at Chicago

Spring 2021

Instructor: Sepideh Mahabadi

Announcement

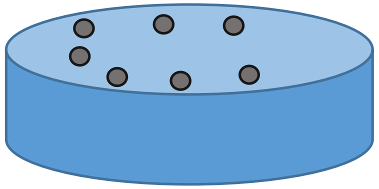
- The schedule has condensed
- Project presentations are May 24 and 26
- First draft of project is due May 24
- Homework 1 will be out this week

This Lecture

- ❑ Core-sets
- ❑ Farthest point
- ❑ Diversity maximization

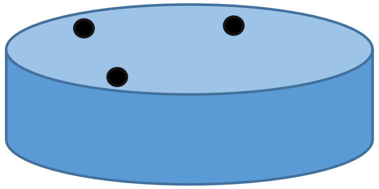
Core-sets [Agarwal, Har-Peled, Varadarajan'05]

Core-sets: a small subset U of the data V that represents it well.



Core-sets [Agarwal, Har-Peled, Varadarajan'05]

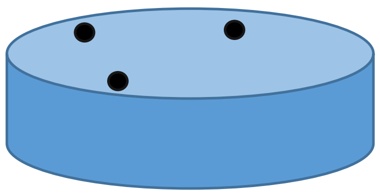
Core-sets: a small subset U of the data V that represents it well.

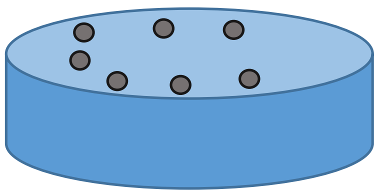


Core-sets [Agarwal, Har-Peled, Varadarajan'05]

Core-sets: a small subset $\boldsymbol{U}$ of the data $\boldsymbol{V}$ that represents it well.

Solving the problem over core-set $\boldsymbol{U}$ → **Solving the problem over dataset $\boldsymbol{V}$ (approximately)**





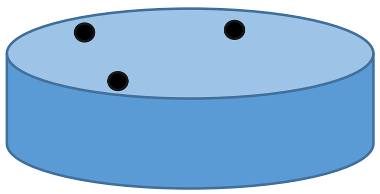
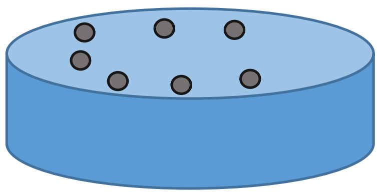
Core-sets [Agarwal, Har-Peled, Varadarajan’05]

Core-sets: a small subset U of the data V that represents it well.

➢ Task specific

Solving the problem over core-set U

Solving the problem over dataset V (approximately)

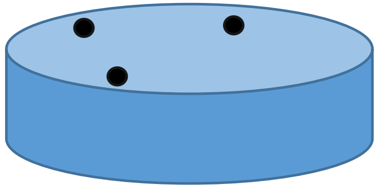
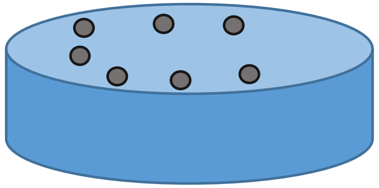
f () $\approx$ f ()

Core-sets [Agarwal, Har-Peled, Varadarajan'05]

α –**Core-sets**: a small subset U of the data V that represents it well.

- ➢ Task specific

Solving the problem over core-set U → **Solving the problem over dataset V (approximately)**

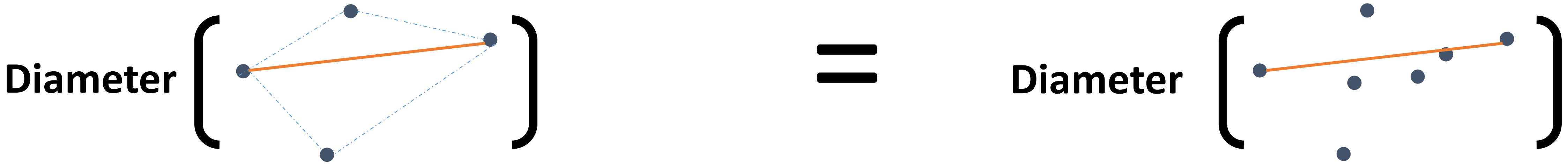
$f\Big($  $\Big) \approx \frac{1}{\alpha} f\Big($  $\Big)$

Core-sets [Agarwal, Har-Peled, Varadarajan'05]

Core-sets: a small subset U of the data V that represents it well.

➢ Task specific

Convex Hull is a 1-core-set for Diameter



Example Applications

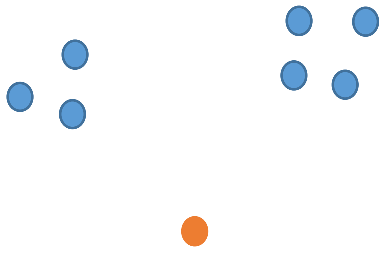
- The algorithm takes too much time to run on the data
- Compress the data, summarization
- Low storage
- Low communication
- Can be used in other massive data models

Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q, P)/\alpha \leq Far(q, S) \leq Far(q, P)$

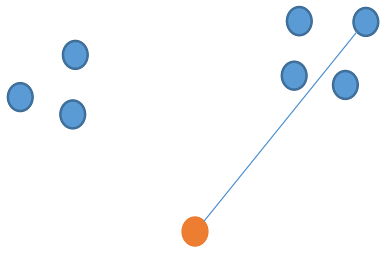
Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q, P)/\alpha \leq Far(q, S) \leq Far(q, P)$



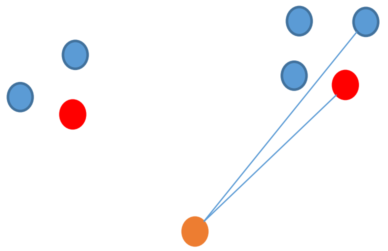
Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q,P)/\alpha \leq Far(q,S) \leq Far(q,P)$



Maintain Distance to Farthest Point (1-center)

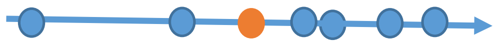
- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q, P)/\alpha \leq Far(q, S) \leq Far(q, P)$



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q, P)/\alpha \leq Far(q, S) \leq Far(q, P)$

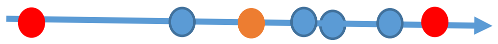
- The points are on one line



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q, P)/\alpha \leq Far(q, S) \leq Far(q, P)$

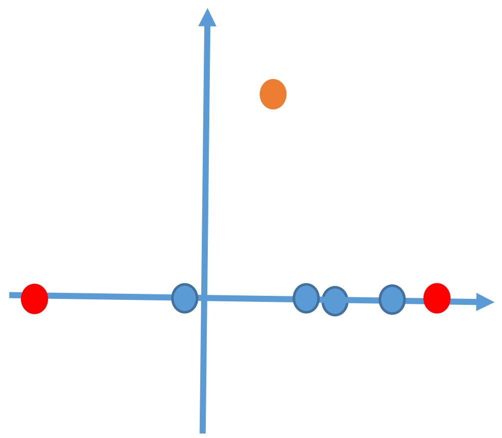
- The points are on one line (two extreme points)



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q,P)/\alpha \leq Far(q,S) \leq Far(q,P)$

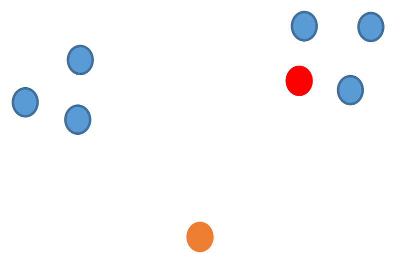
- The points are on one line (two extreme points)
- The query is anywhere (same holds)



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q, P)/\alpha \leq Far(q, S) \leq Far(q, P)$

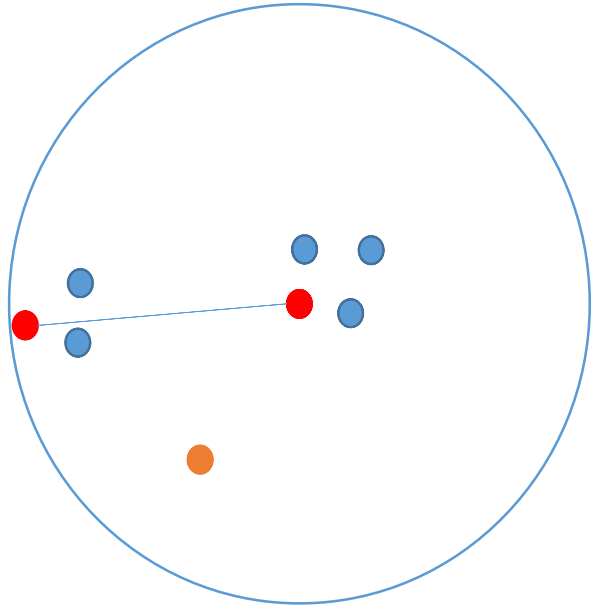
- General setting?
  - O(1)-approximation is easy
    - Take any point $p_1 \in P$ and the farthest to it $p_2 \in P$



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q,P)/\alpha \leq Far(q,S) \leq Far(q,P)$

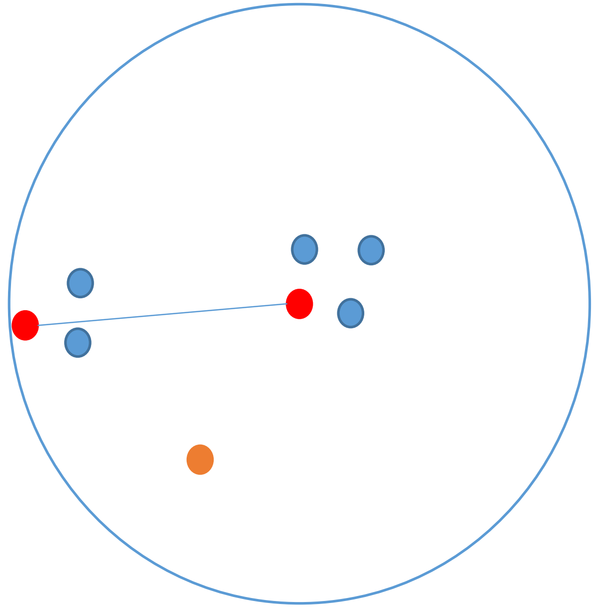
- General setting?
  - O(1)-approximation is easy
    - Take any point $p_1 \in P$ and the farthest to it $p_2 \in P$
    - Let $r = dist(p_1, p_2)$



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q,P)/\alpha \leq Far(q,S) \leq Far(q,P)$

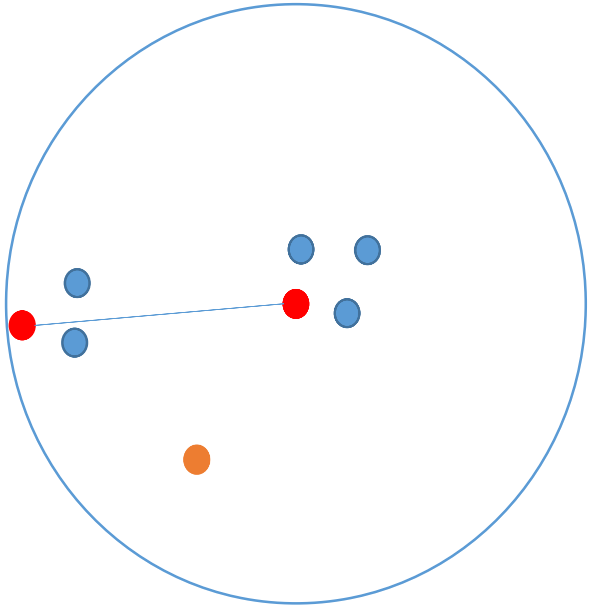
- General setting?
  - O(1)-approximation is easy
    - Take any point $p_1 \in P$ and the farthest to it $p_2 \in P$
    - Let $r = dist(p_1, p_2)$
    - $Far(q,S) \geq r/2$



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q,P)/\alpha \leq Far(q,S) \leq Far(q,P)$

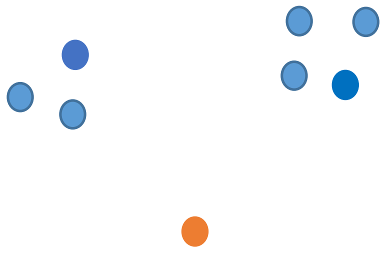
- General setting?
  - O(1)-approximation is easy
    - Take any point $p_1 \in P$ and the farthest to it $p_2 \in P$
    - Let $r = dist(p_1, p_2)$
    - $Far(q,S) \geq r/2$
    - $Far(q,P) \leq dist(q,p_1) + r \leq 2r$



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q,P)/\alpha \leq Far(q,S) \leq Far(q,P)$

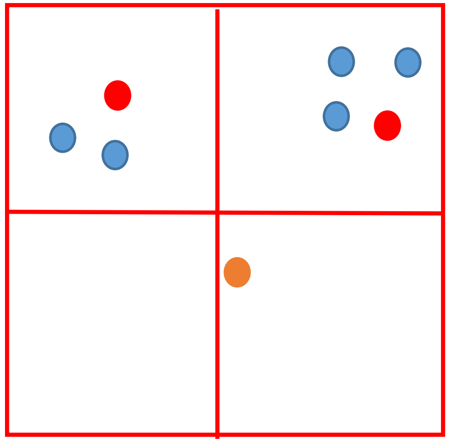
- General setting? Better approximation?



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q,P)/\alpha \leq Far(q,S) \leq Far(q,P)$

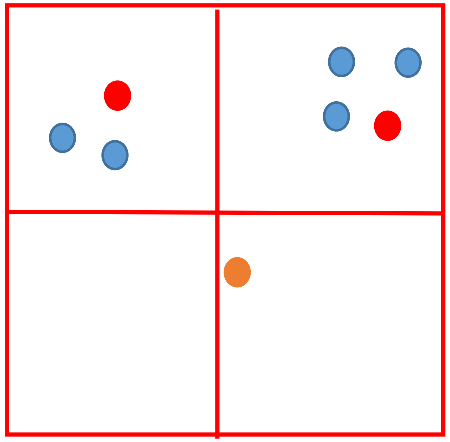
- General setting?
  - Impose a grid of side length ϵr
  - For each non-empty cell, keep one point in the core-set



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q, P)/\alpha \leq Far(q, S) \leq Far(q, P)$

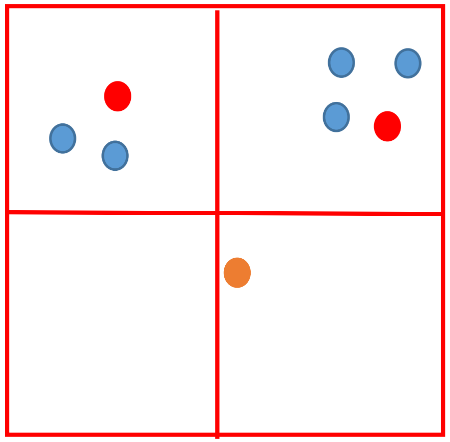
- General setting?
  - Impose a grid of side length ϵr
  - For each non-empty cell, keep one point in the core-set
  - Size of core-set: $\left(\frac{1}{\epsilon}\right)^d$



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q,P)/\alpha \leq Far(q,S) \leq Far(q,P)$

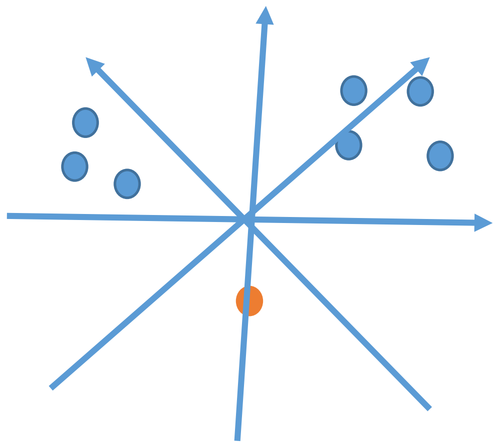
- General setting?
  - Impose a grid of side length ϵr
  - For each non-empty cell, keep one point in the core-set
  - Size of core-set: $\left(\frac{1}{\epsilon}\right)^d$
  - Error: additive $\epsilon r\sqrt{d}$ which is $(1+\epsilon)$ approximation for constant dimension



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q,P)/\alpha \leq Far(q,S) \leq Far(q,P)$

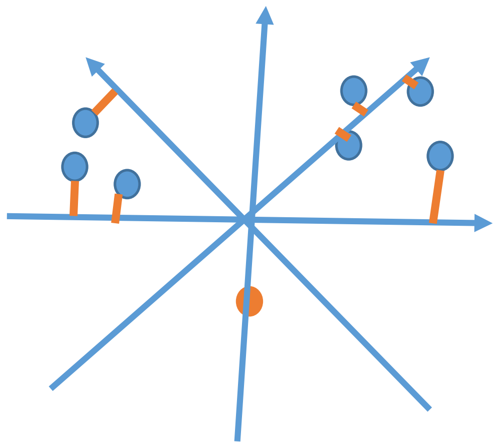
- General setting?
  - Cover the unit sphere with vectors v_i with separation angle at most ϵ
  - Project all points to closest line
  - Use 1-dimensional exact core-set
  - Size $\left(\frac{1}{\epsilon}\right)^{d-1}$
  - Error: each point is dis-located at most $r \sin \epsilon \approx r\epsilon$



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q,P)/\alpha \leq Far(q,S) \leq Far(q,P)$

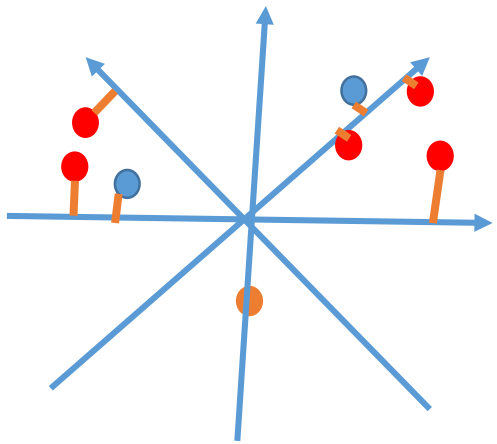
- General setting?
  - Cover the unit sphere with vectors v_i with separation angle at most ϵ
  - Project all points to closest line
  - Use 1-dimensional exact core-set
  - Size $\left(\frac{1}{\epsilon}\right)^{d-1}$
  - Error: each point is dis-located at most $r \sin \epsilon \approx r\epsilon$



Maintain Distance to Farthest Point (1-center)

- Given a point set $P \in \mathbb{R}^d$ find a core-set S, s.t. for any query point q,
- $Far(q, P)/\alpha \leq Far(q, S) \leq Far(q, P)$

- General setting?
  - Cover the unit sphere with vectors v_i with separation angle at most ϵ
  - Project all points to closest line
  - Use 1-dimensional exact core-set
  - Size $\left(\frac{1}{\epsilon}\right)^{d-1}$
  - Error: each point is dis-located at most $r \sin \epsilon \approx r\epsilon$



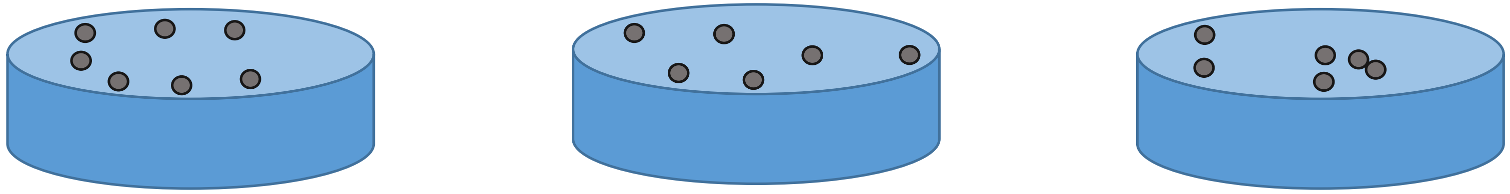
Generic Notion

- Weak Core-set (approximates the optimal solution)
- Strong Core-set (approximates any solution)
- Can be a weighted subset
- Additional information (not necessarily the subset)

Composable Core-sets

Core-sets with composability property:

"The union of core-sets is a core-set for the union"

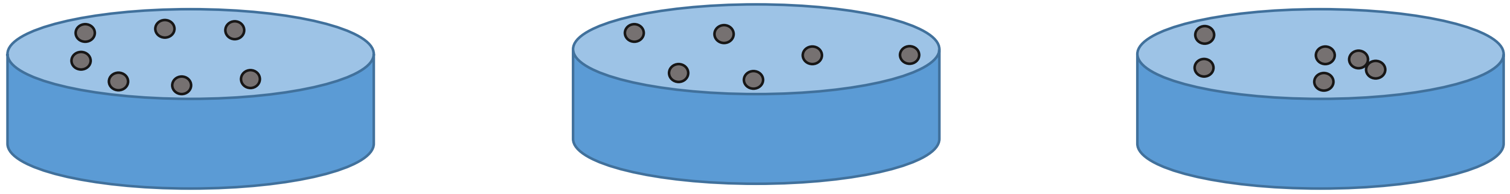


Composable Core-sets

Core-sets with composability property:

"The union of core-sets is a core-set for the union"

- Let $\boldsymbol{f}$ be an optimization function
- Multiple data sets $\boldsymbol{V_1, \cdots, V_m}$

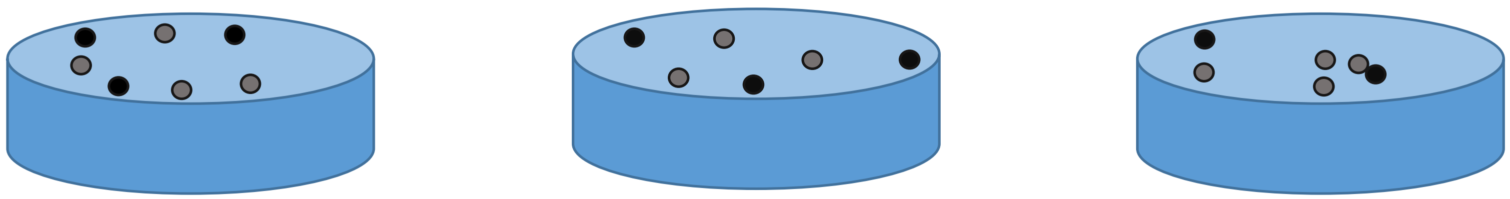


Composable Core-sets

Core-sets with composability property:

“The union of core-sets is a core-set for the union”

- Let $\boldsymbol{f}$ be an optimization function
- Multiple data sets $\boldsymbol{V_1, \cdots, V_m}$ and their core-sets $\boldsymbol{U_1 \subset V_1, \cdots, U_m \subset V_m}$,

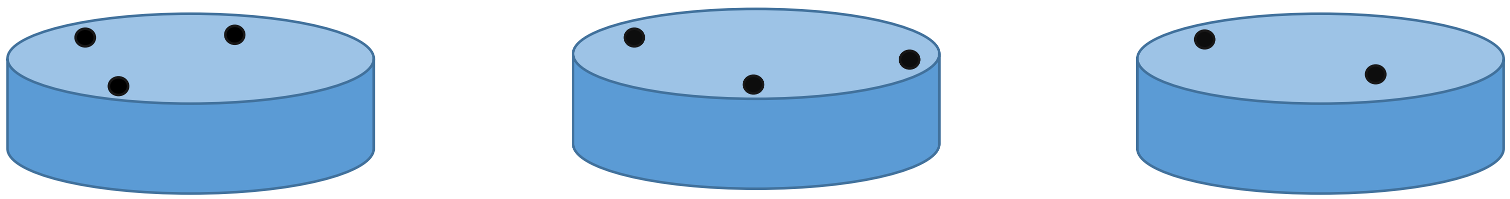


Composable Core-sets

Core-sets with composability property:

"The union of core-sets is a core-set for the union"

- Let $\boldsymbol{f}$ be an optimization function
- Multiple data sets $\boldsymbol{V_1, \cdots, V_m}$ and their core-sets $\boldsymbol{U_1 \subset V_1, \cdots, U_m \subset V_m}$,

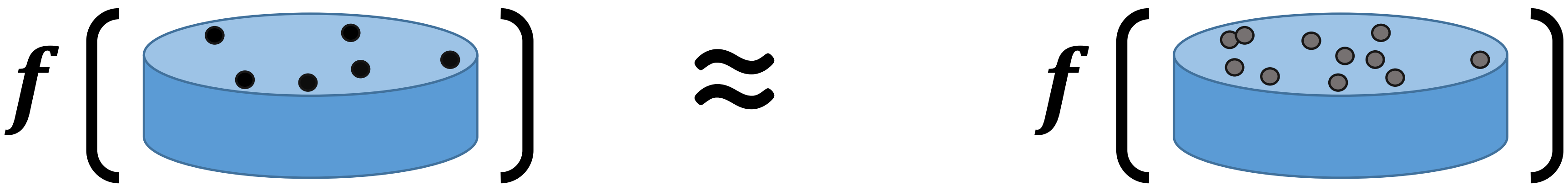


Composable Core-sets

Core-sets with composability property:

"The union of core-sets is a core-set for the union"

- Let $\boldsymbol{f}$ be an optimization function
- Multiple data sets $\boldsymbol{V_1}, \cdots, \boldsymbol{V_m}$ and their core-sets $\boldsymbol{U_1} \subset \boldsymbol{V_1}, \cdots, \boldsymbol{U_m} \subset \boldsymbol{V_m}$,
  - $\boldsymbol{f}(\boldsymbol{U_1} \cup \cdots \cup \boldsymbol{U_m})$ approximates $\boldsymbol{f}(\boldsymbol{V_1} \cup \cdots \cup \boldsymbol{V_m})$ by a factor $\boldsymbol{\alpha}$

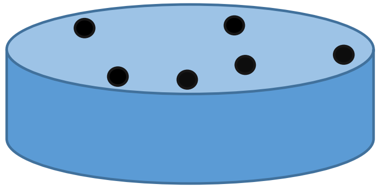
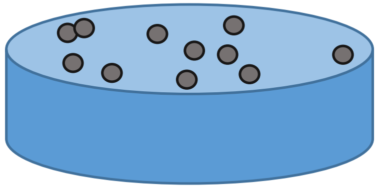


$\boldsymbol{\alpha}$ –Composable Core-sets

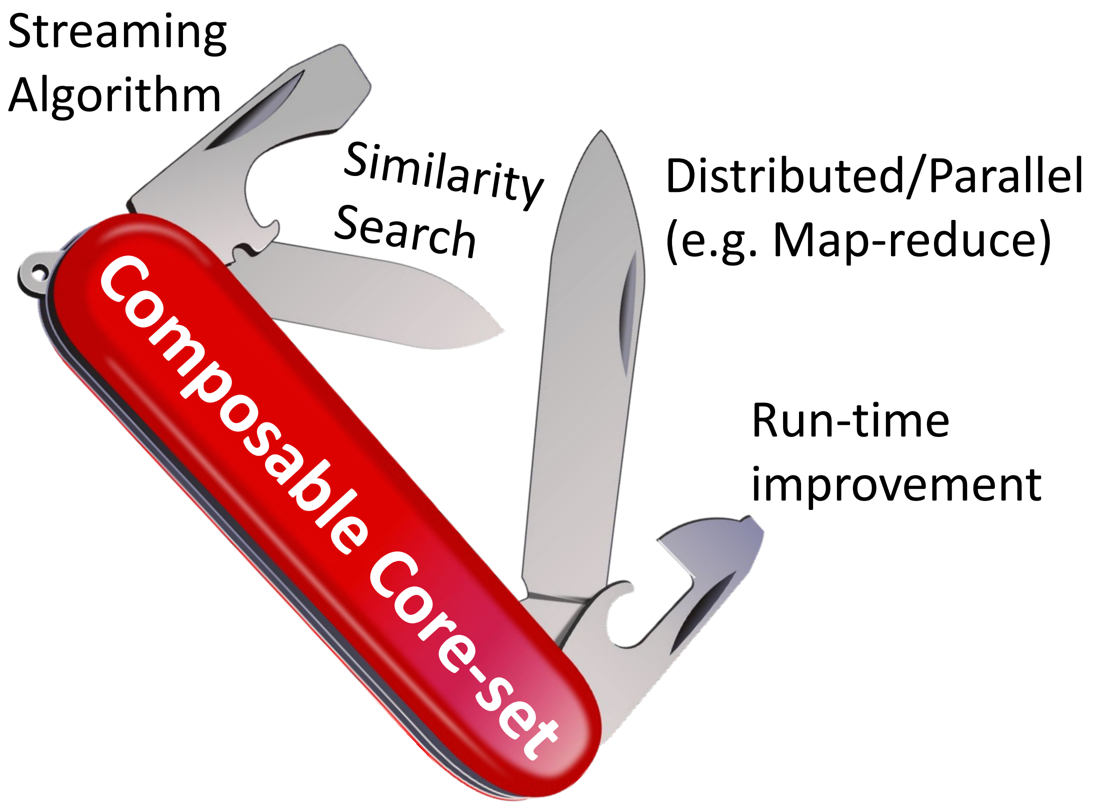
Core-sets with composability property:

"The union of core-sets is a core-set for the union"

- Let $\boldsymbol{f}$ be an optimization function
- Multiple data sets $\boldsymbol{V_1}, \cdots, \boldsymbol{V_m}$ and their core-sets $\boldsymbol{U_1} \subset \boldsymbol{V_1}, \cdots, \boldsymbol{U_m} \subset \boldsymbol{V_m}$,
  - $\boldsymbol{f}(\boldsymbol{U_1} \cup \cdots \cup \boldsymbol{U_m})$ approximates $\boldsymbol{f}(\boldsymbol{V_1} \cup \cdots \cup \boldsymbol{V_m})$ by a factor $\boldsymbol{\alpha}$

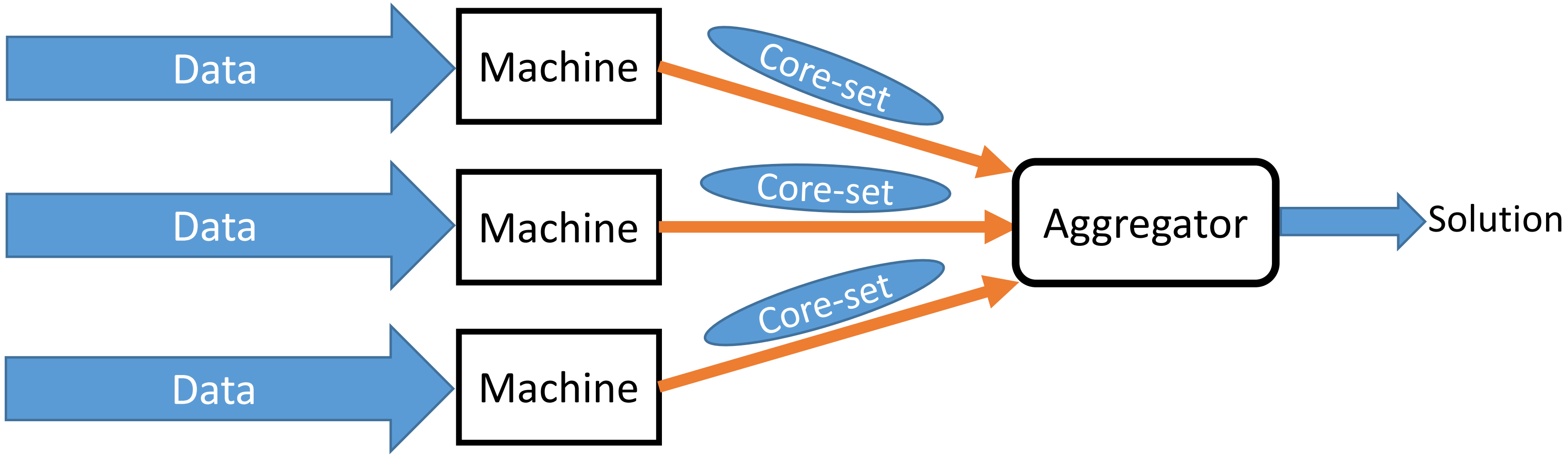
$\boldsymbol{f}\Big($  $\Big) \approx \frac{1}{\alpha}\, \boldsymbol{f}\Big($  $\Big)$

Having a **composable core-set** for a task, **automatically** gives **algorithms** in **several** massive data processing **models** for the same task.



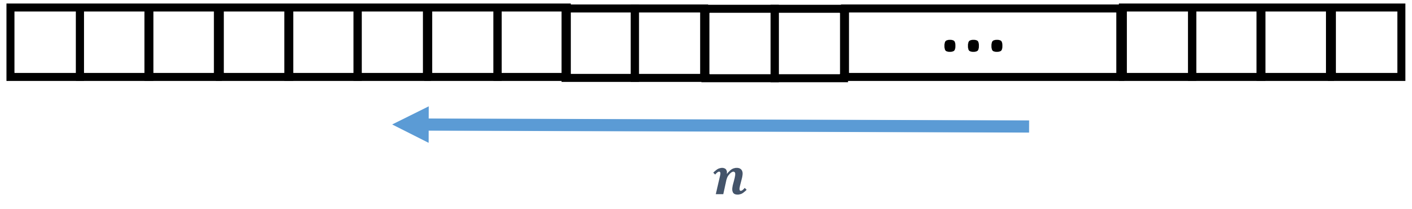
Application: Distributed/Parallel Systems (e.g. Map-Reduce)

- ❑ Multiple Machines
  - Each holding part of the data
- ❑ Each machine computes a composable core-set and sends it to the coordinator
- ➢ Composability guarantees a good solution
- ➢ Total communication is low



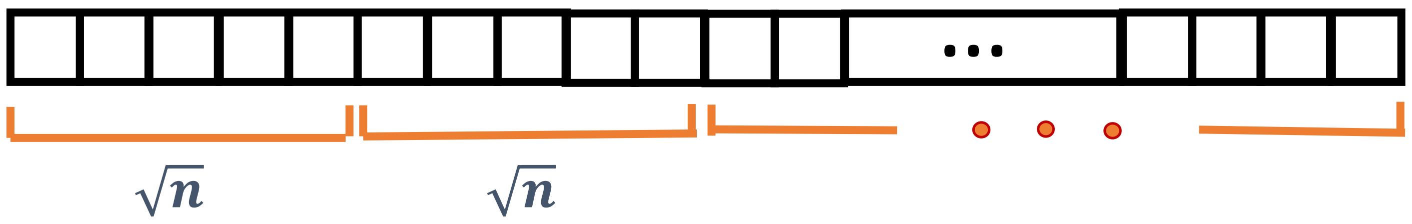
Application to Streaming Computation

- Streaming Computation:
  - Processing a sequence of n data elements “on the fly”
  - Limited storage $o(n)$



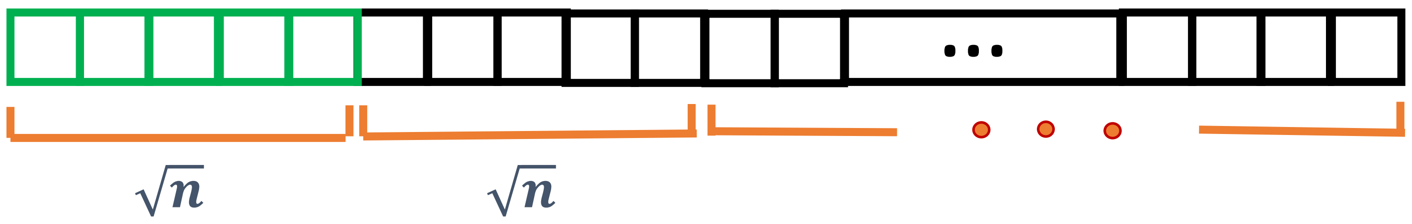
Application to Streaming Computation

- Streaming Computation:
  - Processing a sequence of n data elements "on the fly"
  - Limited storage $o(n)$
- Composable Core-set
  - Divide into chunks



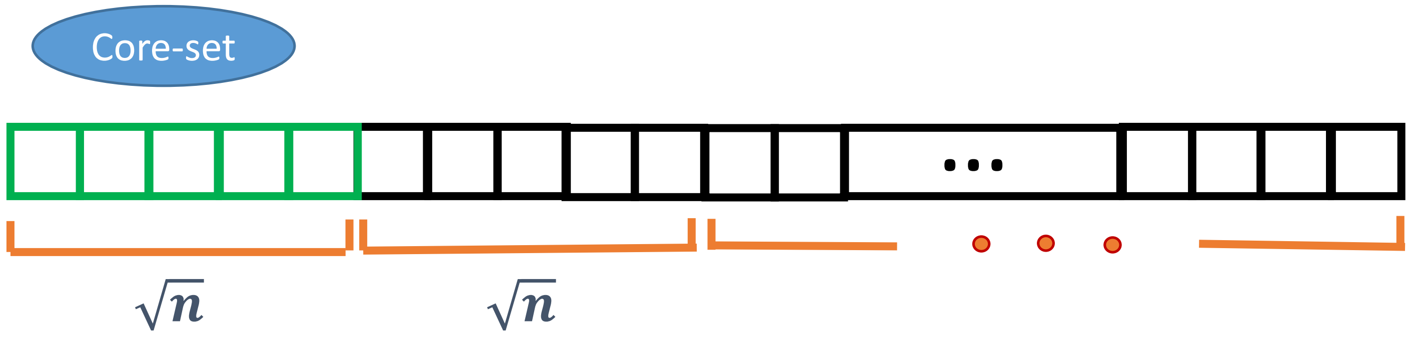
Application to Streaming Computation

- Streaming Computation:
  - Processing a sequence of n data elements "on the fly"
  - Limited storage $o(n)$
- Composable Core-set
  - Divide into chunks



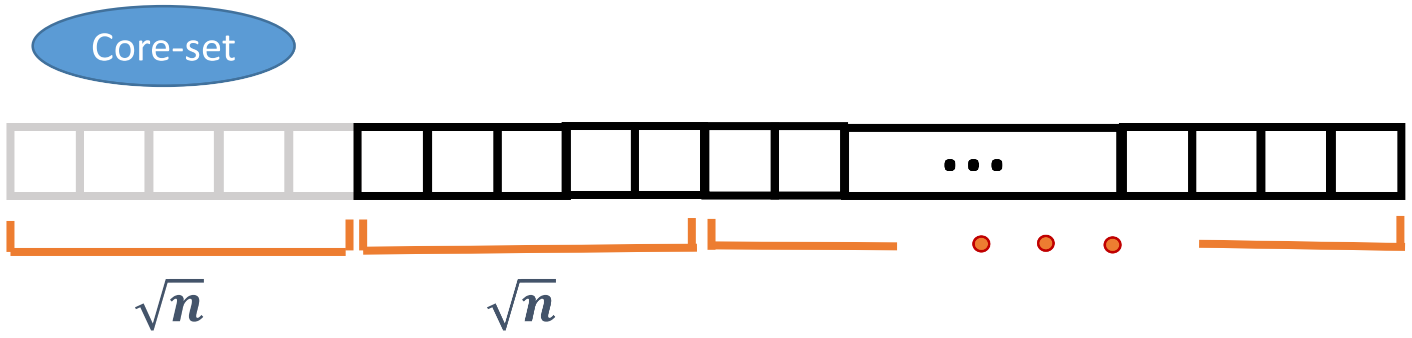
Application to Streaming Computation

- Streaming Computation:
  - Processing a sequence of n data elements "on the fly"
  - Limited storage $o(n)$
- Composable Core-set
  - Divide into chunks
  - Compute Core-set for each chunk as it arrives



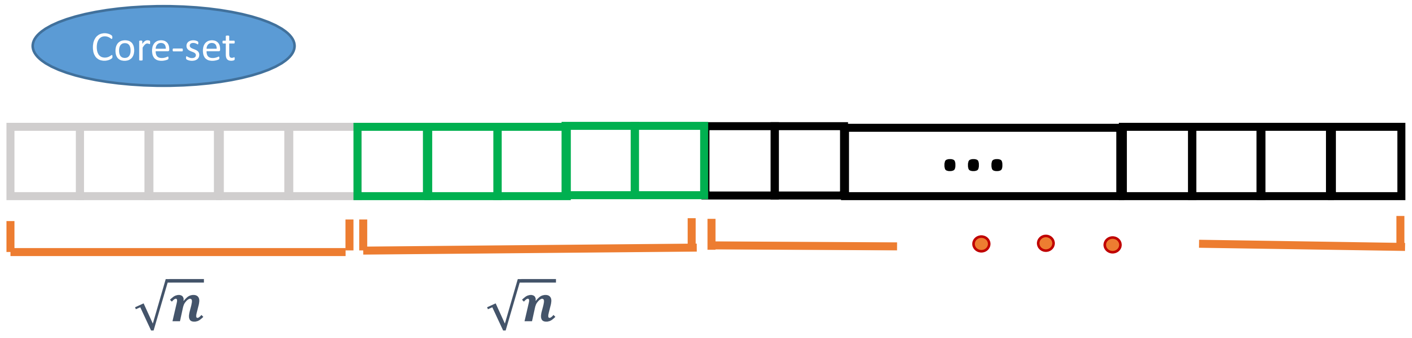
Application to Streaming Computation

- Streaming Computation:
  - Processing a sequence of n data elements "on the fly"
  - Limited storage $o(n)$
- Composable Core-set
  - Divide into chunks
  - Compute Core-set for each chunk as it arrives



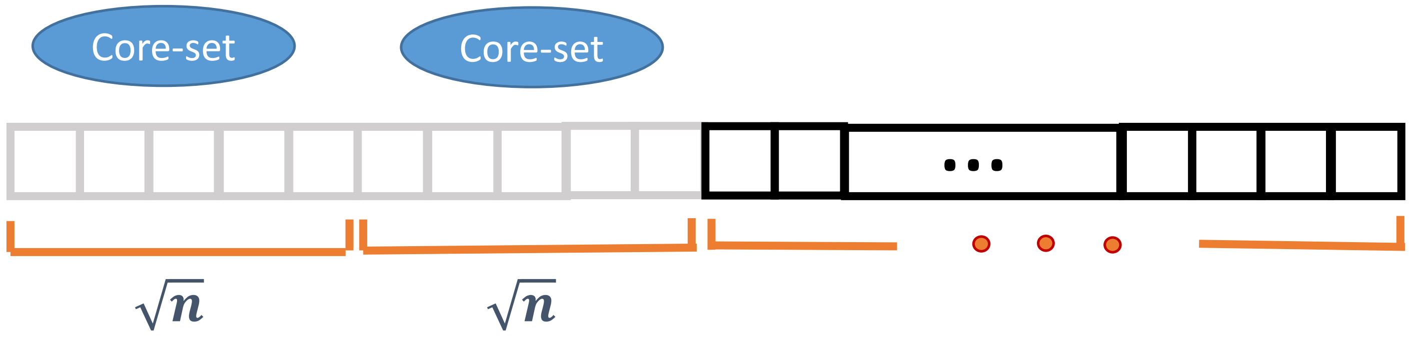
Application to Streaming Computation

- Streaming Computation:
  - Processing a sequence of n data elements "on the fly"
  - Limited storage $o(n)$
- Composable Core-set
  - Divide into chunks
  - Compute Core-set for each chunk as it arrives



Application to Streaming Computation

- Streaming Computation:
  - Processing a sequence of n data elements "on the fly"
  - Limited storage $o(n)$
- Composable Core-set
  - Divide into chunks
  - Compute Core-set for each chunk as it arrives

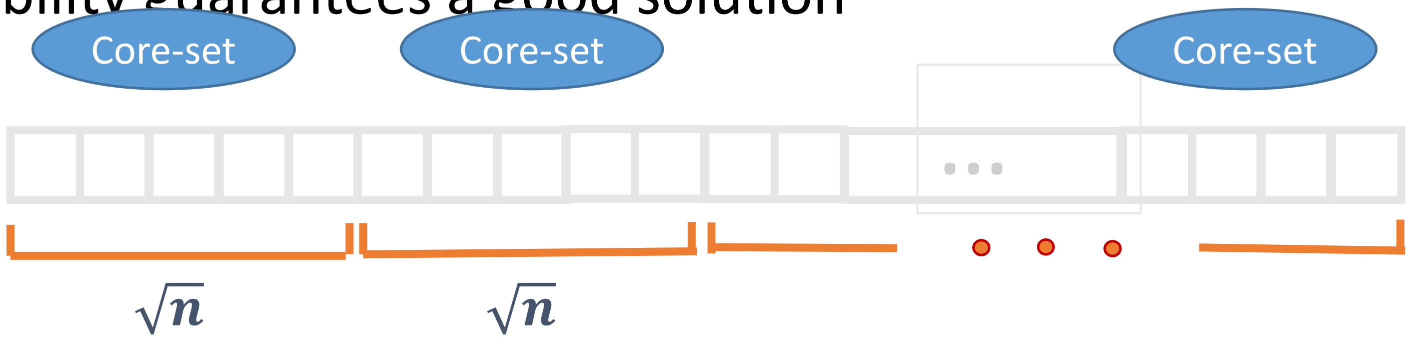


Application to Streaming Computation

- ❑ Streaming Computation:
  - Processing a sequence of n data elements "on the fly"
  - Limited storage $o(n)$
- ❑ Composable Core-set
  - Divide into chunks
  - Compute Core-set for each chunk as it arrives

➢ Space goes down from n to $\approx \sqrt{n}$

➢ Composability guarantees a good solution



Diversity Maximization

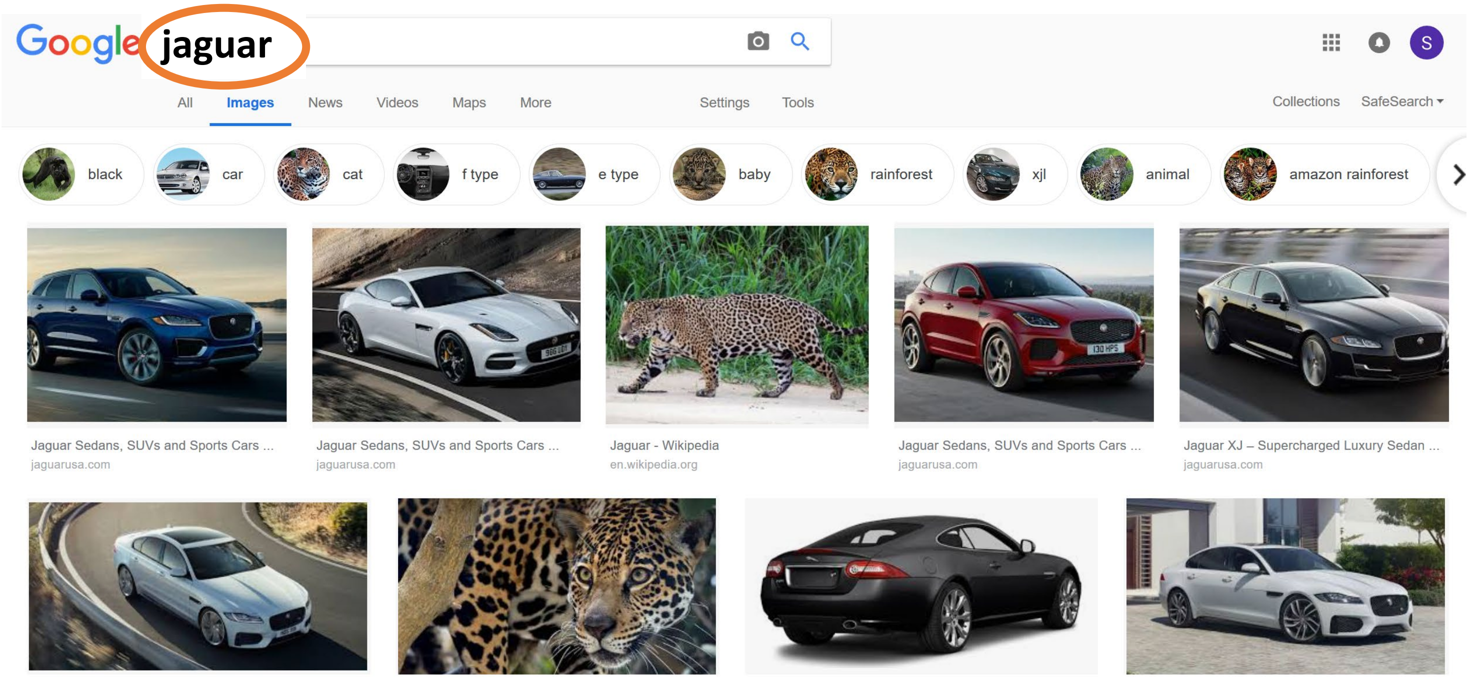
Diversity Maximization

Given a set of objects, how to pick a few of them while maximizing diversity?

Diversity Maximization

Given a set of objects, how to pick a few of them while maximizing diversity?

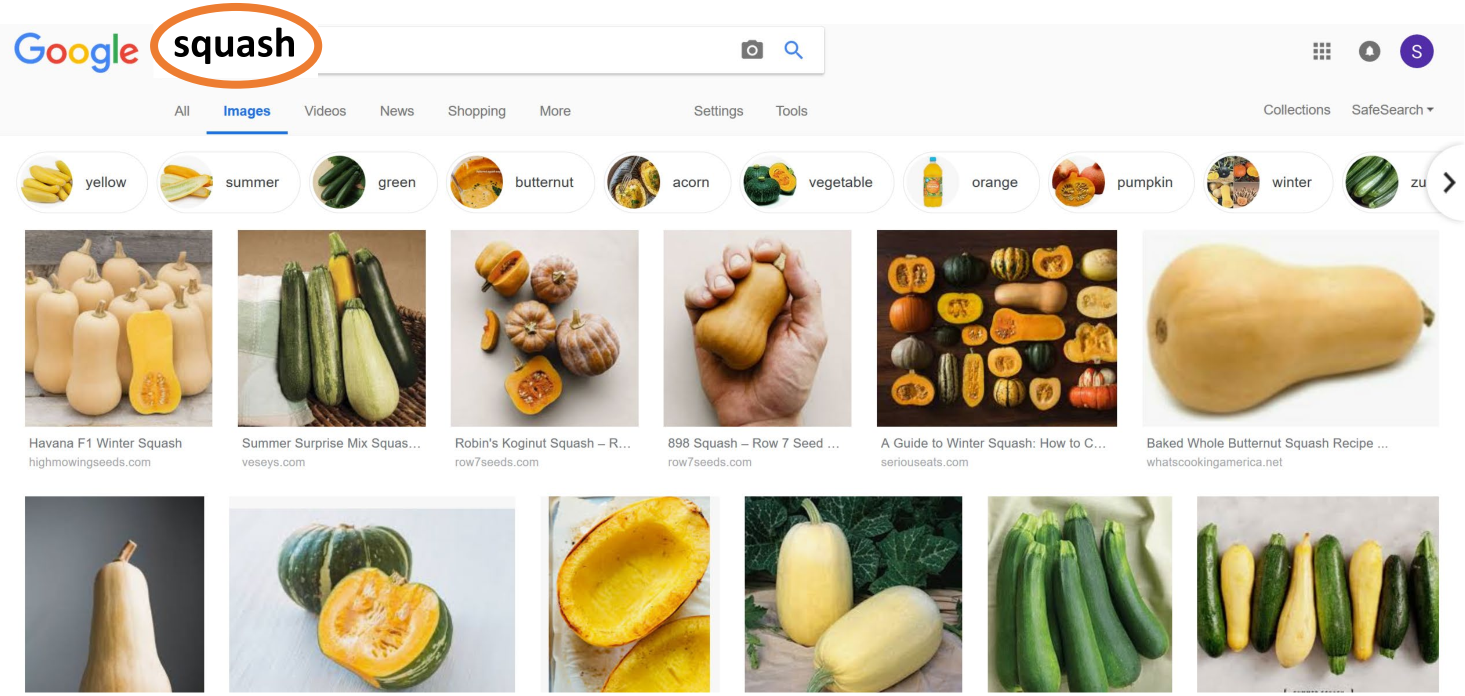
- **Searching**



Diversity Maximization

Given a set of objects, how to pick a few of them while maximizing diversity?

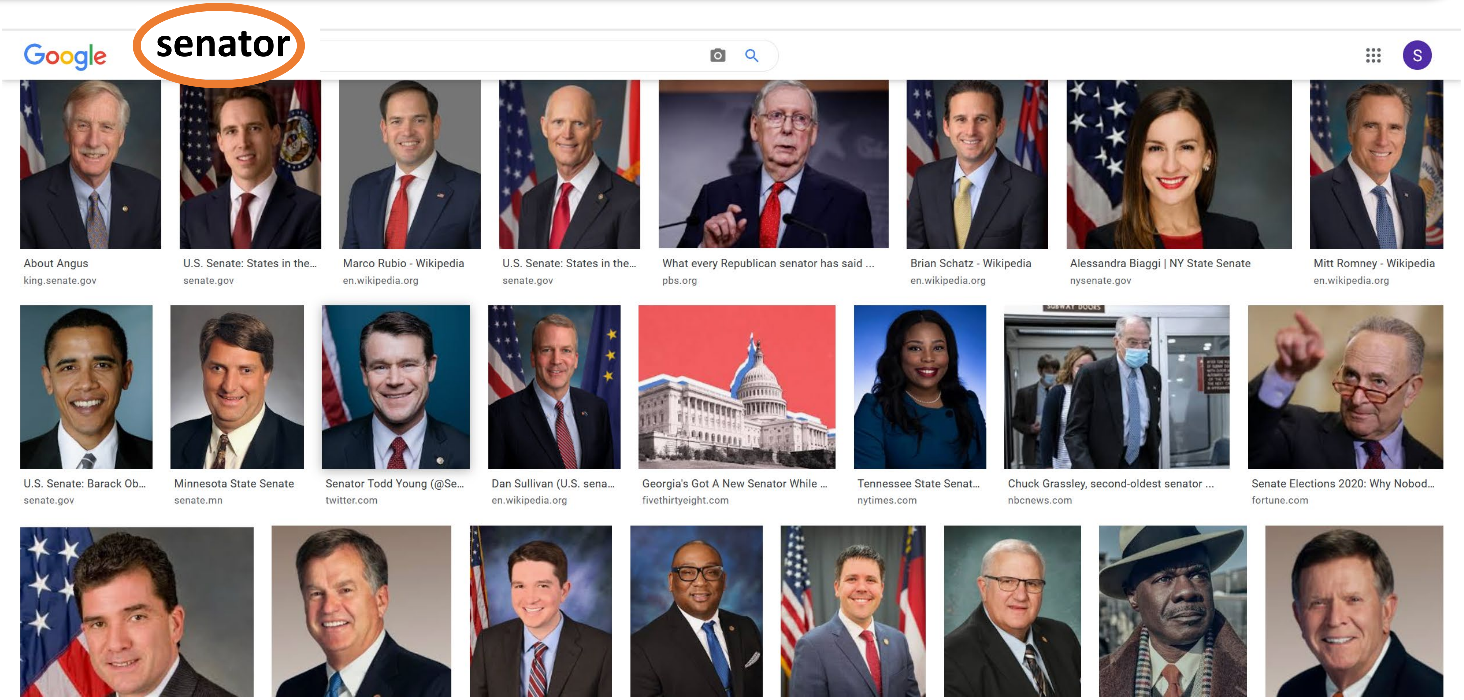
- **Searching**



Diversity Maximization

Given a set of objects, how to pick a few of them while maximizing diversity?

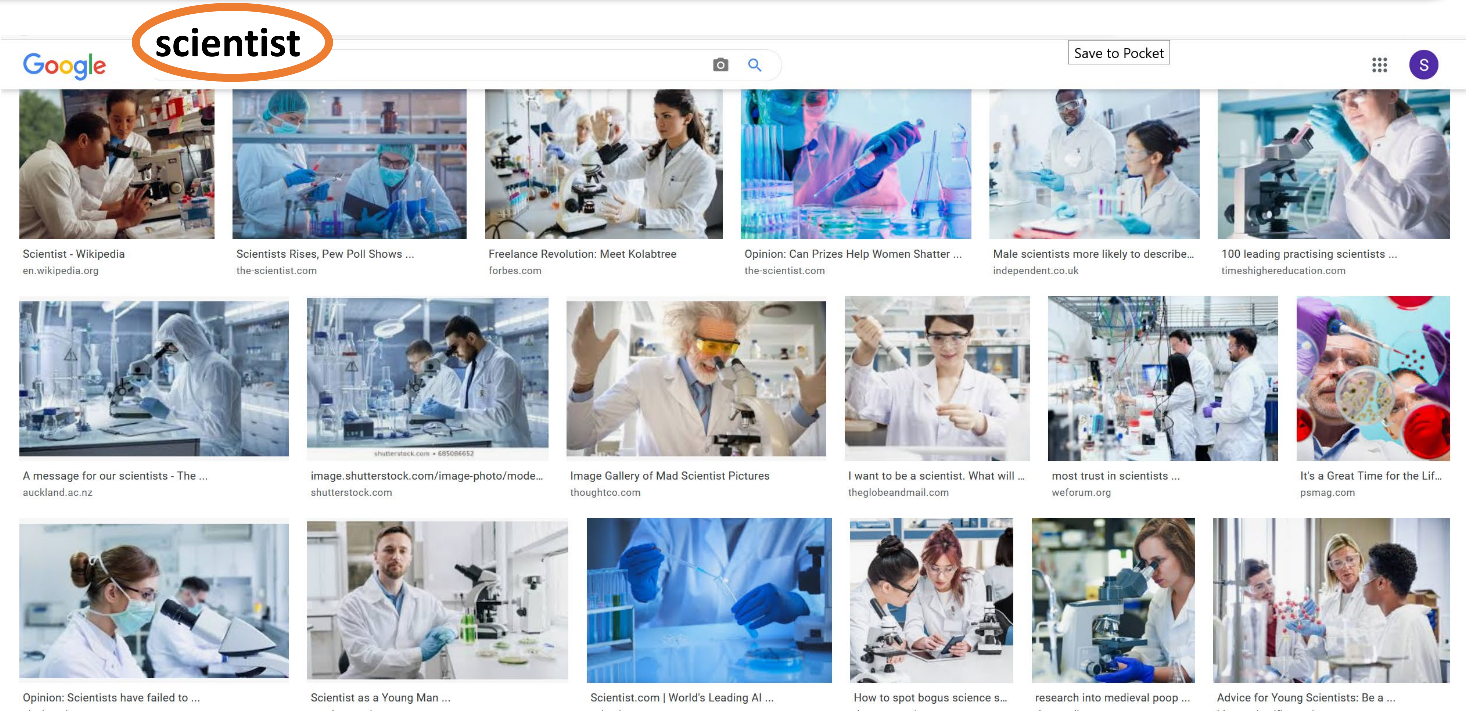
- **Searching**



Diversity Maximization

Given a set of objects, how to pick a few of them while maximizing diversity?

- **Searching**



Diversity Maximization

Given a set of objects, how to pick a few of them while maximizing diversity?

- Searching
- **Recommender Systems**

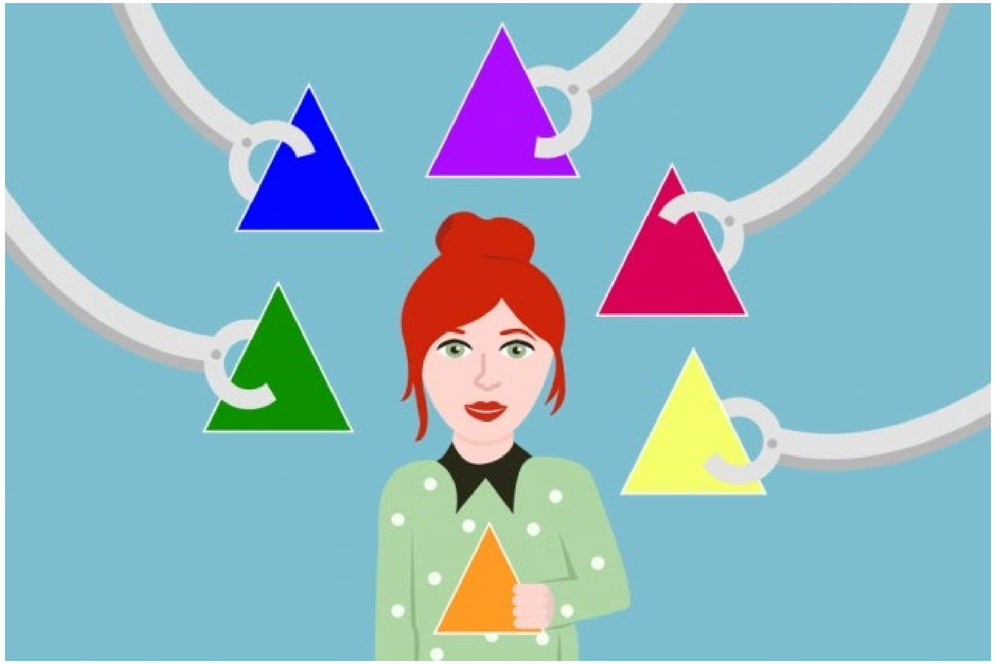


Image from: http://news.mit.edu/2017/better-recommendation-algorithm-1206

Diversity Maximization

Given a set of objects, how to pick a few of them while maximizing diversity?

- Searching
- Recommender Systems
- Summarization
- Object detection, …

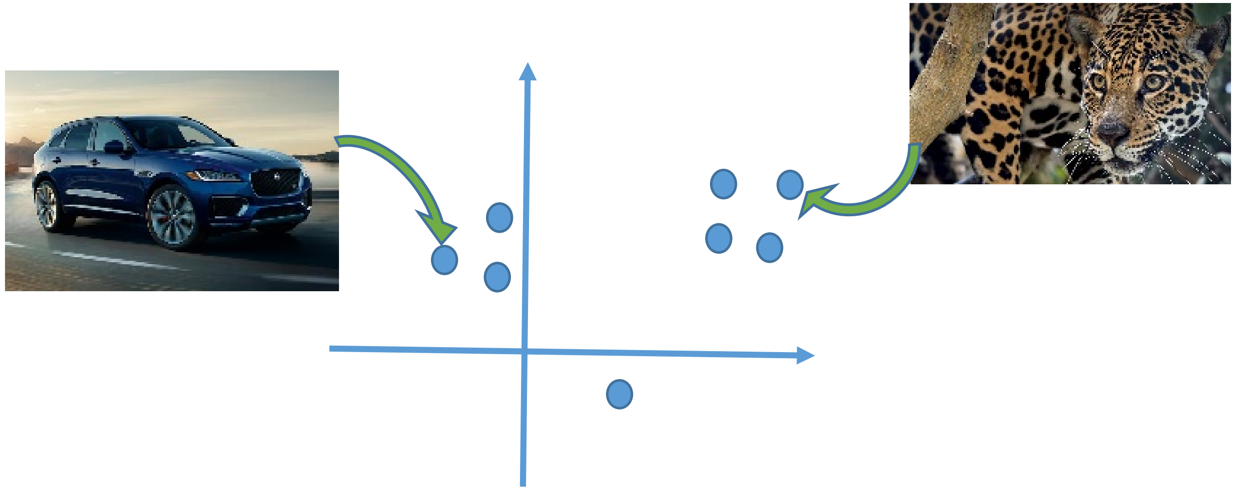
A small subset of items must be selected to represent the larger population

Modeling the Objects

Objects
(documents, images, etc)

Feature Vectors

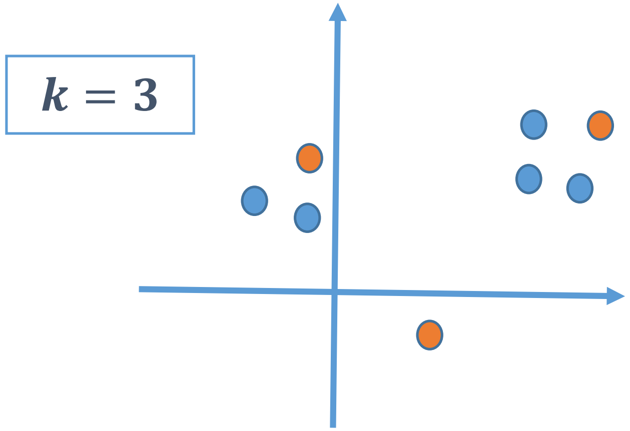
Points in a high dimensional space



Diversity Maximization: The Model

Input: a set of n vectors $\mathrm{V} \subset \mathbb{R}^d$ and a parameter $k \leq d$,

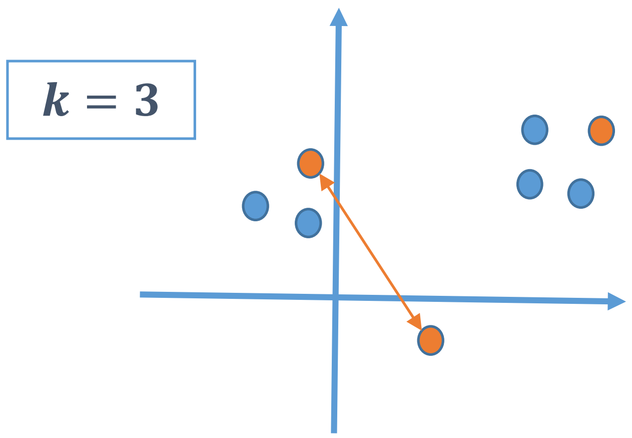
Goal: **pick k points while maximizing "diversity".**



Minimum Pairwise Distance

Input: a set of n vectors $\mathrm{V} \subset \mathbb{R}^d$ and a parameter $k \leq d$,

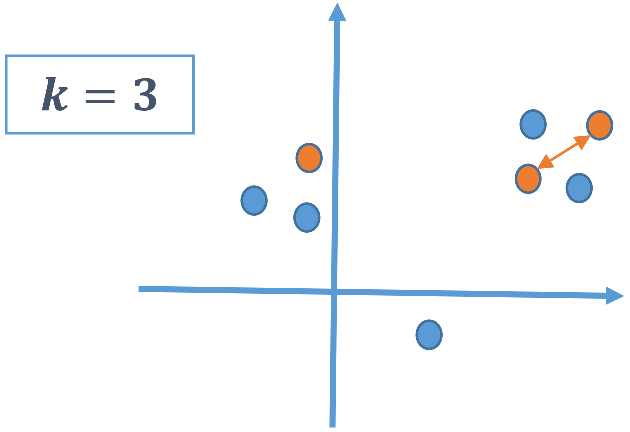
Goal: **pick k points s.t. the minimum pairwise distance of the picked points is maximized.**



Minimum Pairwise Distance

Input: a set of n vectors $\mathrm{V} \subset \mathbb{R}^d$ and a parameter $k \leq d$,

Goal: pick k points s.t. the minimum pairwise distance of the picked points is maximized.



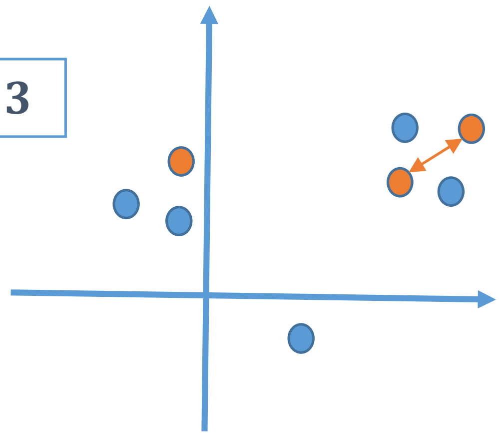
Minimum Pairwise Distance

Input: a set of n vectors $\mathrm{V} \subset \mathbb{R}^d$ and a parameter $k \leq d$,

Goal: pick k points s.t. the minimum pairwise distance of the picked points is maximized.

- NP-hard to approximate better than 2
- Greedy gives a constant approximation

$\boldsymbol{k = 3}$



Maximizing the minimum pairwise distance

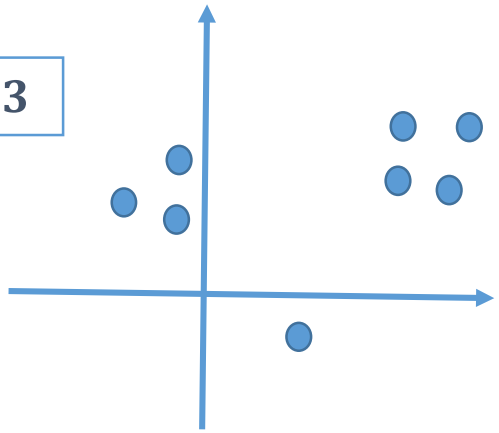
The Greedy Algorithm provides approximation factor $O(1)$

Input: a set V of n points and a parameter k

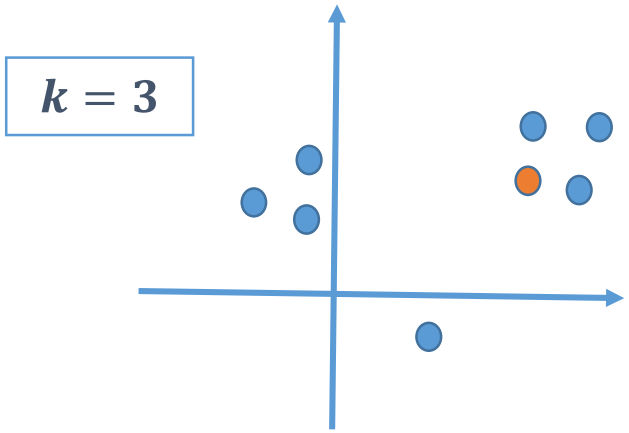
1. Start with an empty set S
2. For k iterations, add the point $p \in V \setminus S$ that is farthest away from S.

Maximizing the minimum pairwise distance

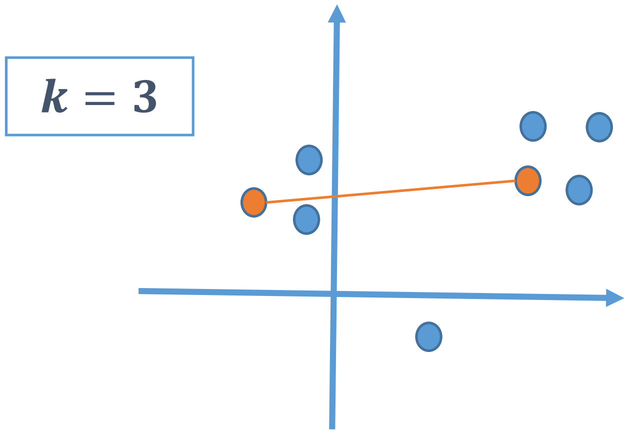
$k = 3$



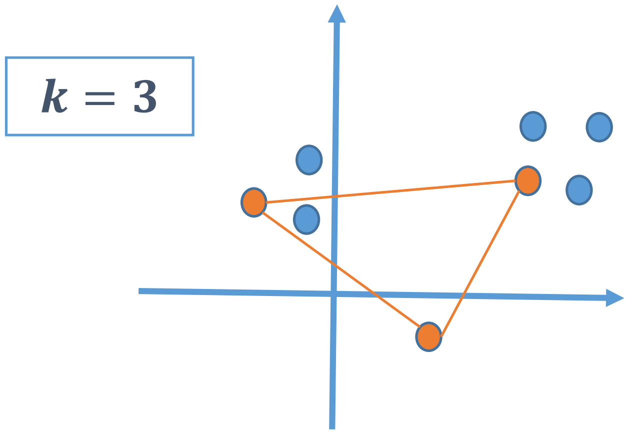
Maximizing the minimum pairwise distance



Maximizing the minimum pairwise distance

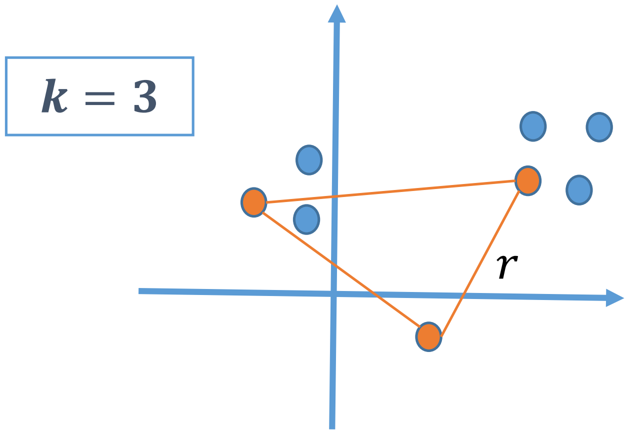


Maximizing the minimum pairwise distance



Maximizing the minimum pairwise distance

Let r be the diversity of S, i.e., $\min_{q_1, q_2 \in S} dist(q_1, q_2)$



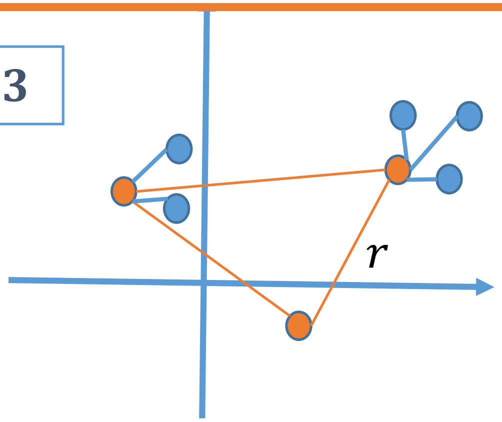
Maximizing the minimum pairwise distance

Let r be the diversity of S, i.e., $\min_{q_1,q_2 \in S} dist(q_1, q_2)$

Observation: For any point $p \in V$, we have $dist(p, S) \leq r$

- $\exists q \in S$ such that $dist(p, q) \leq r$

$\boldsymbol{k = 3}$

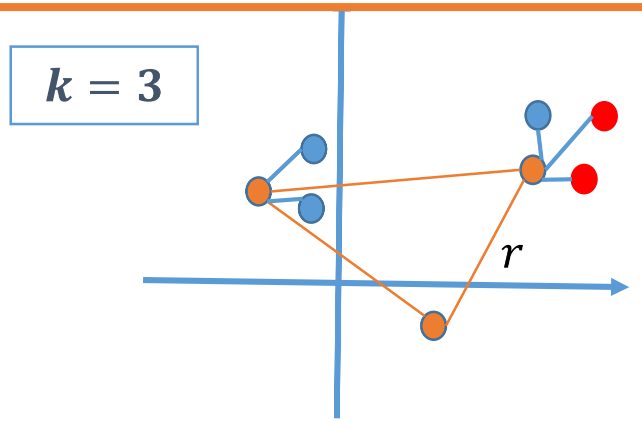


Maximizing the minimum pairwise distance

Let r be the diversity of S, i.e., $\min_{q_1,q_2 \in S} dist(q_1, q_2)$

Observation: For any point $p \in V$, we have $dist(p, S) \le r$

- $\exists q \in S$ such that $dist(p, q) \le r$



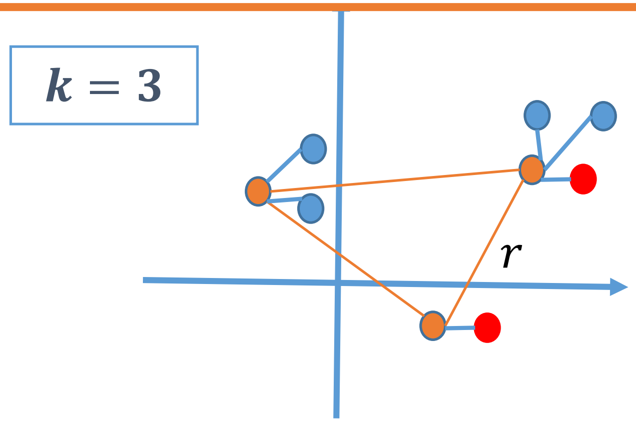
$Opt \le 2r$

Maximizing the minimum pairwise distance

Let r be the diversity of S, i.e., $\min_{q_1,q_2 \in S} dist(q_1, q_2)$

Observation: For any point $p \in V$, we have $dist(p, S) \leq r$

- $\exists q \in S$ such that $dist(p, q) \leq r$



$Opt \leq 3r$

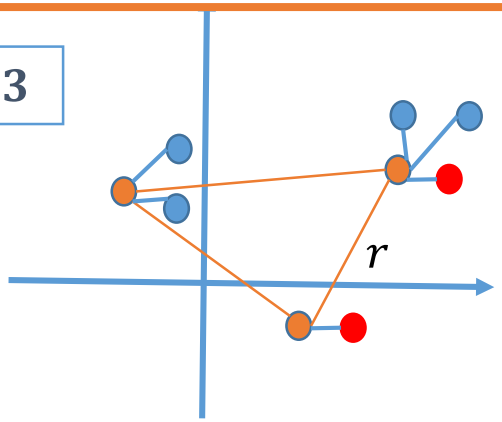
Maximizing the minimum pairwise distance

Let r be the diversity of S, i.e., $\min_{q_1,q_2 \in S} dist(q_1, q_2)$

Observation: For any point $p \in V$, we have $dist(p, S) \leq r$

- $\exists q \in S$ such that $dist(p, q) \leq r$

$\boldsymbol{k = 3}$

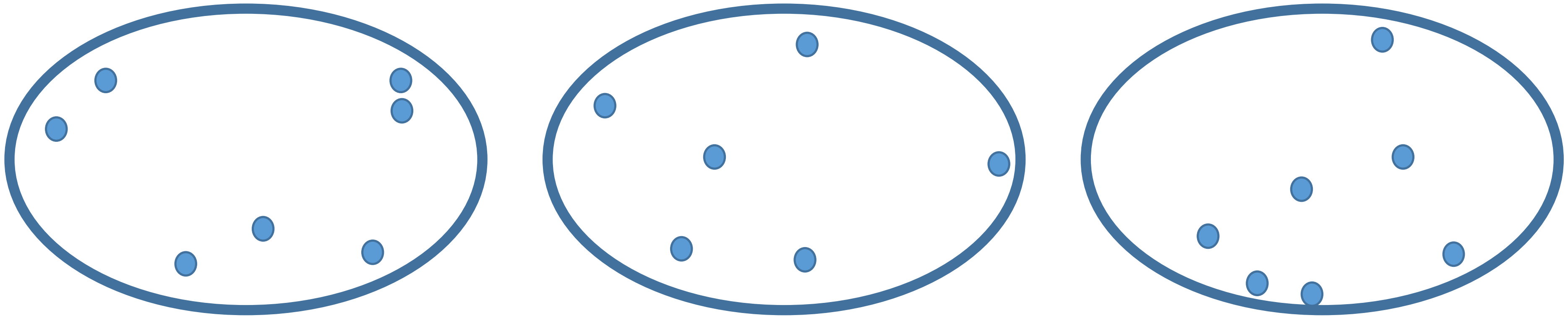


$Opt \leq 2r$

Composable Core-set Setting

The Greedy Algorithm produces a composable core-set of size k with approximation factor $O(1)$

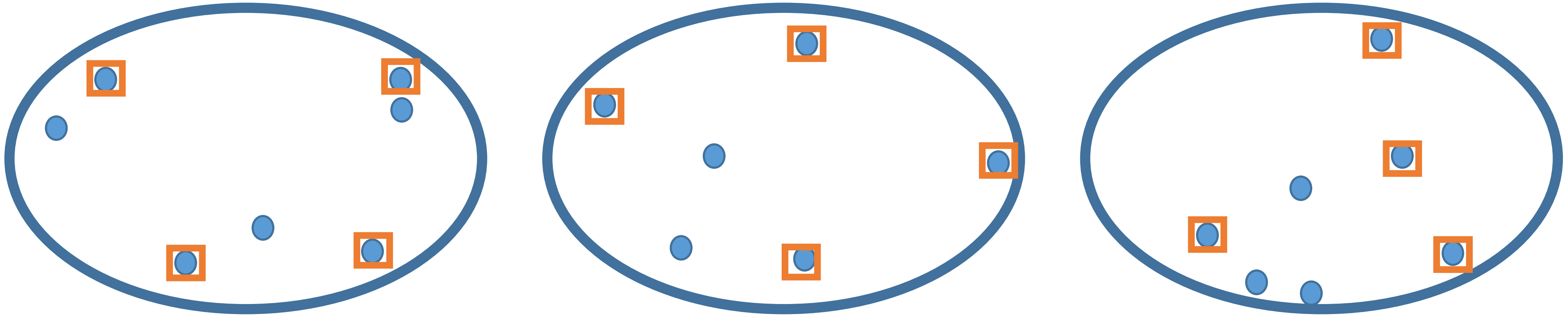
Let $V_1, \ldots, V_m$ be the set of points, $V = \bigcup_i V_i$



Let $V_1, \ldots, V_m$ be the set of points, $V = \cup_i V_i$

Let $S_1, \ldots, S_m$ be their core-sets, $S = \cup_i S_i$

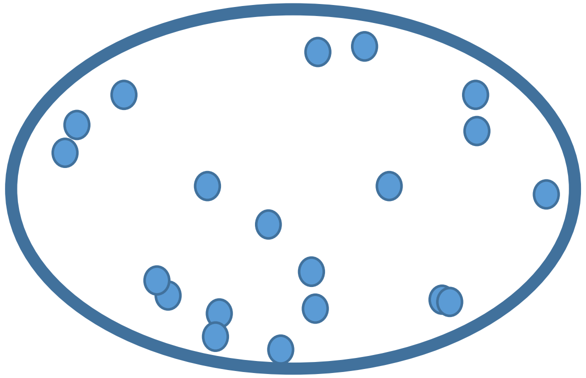
Goal: $div_k(S) \geq div_k(V)/c$



Let $V_1, \ldots, V_m$ be the set of points, $V = \bigcup_i V_i$

Let $S_1, \ldots, S_m$ be their core-sets, $S = \bigcup_i S_i$

Goal: $div_k(S) \geq div_k(V)/c$



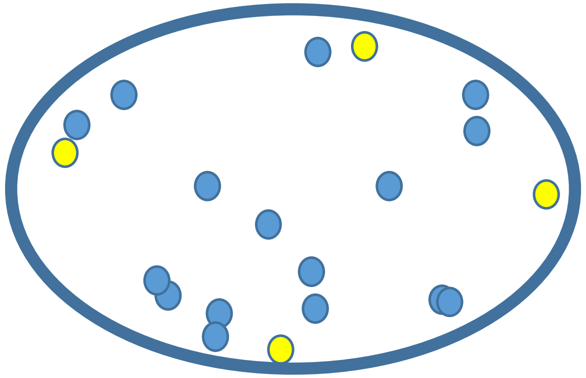
Let $V_1, \dots, V_m$ be the set of points, $V = \cup_i V_i$

Let $S_1, \dots, S_m$ be their core-sets, $S = \cup_i S_i$

Let $Opt = \{o_1, \dots, o_k\}$ be the optimal solution

Goal: $div_k(S) \geq div_k(V)/c$

Goal: $div_k(S) \geq div_k(Opt)/c$



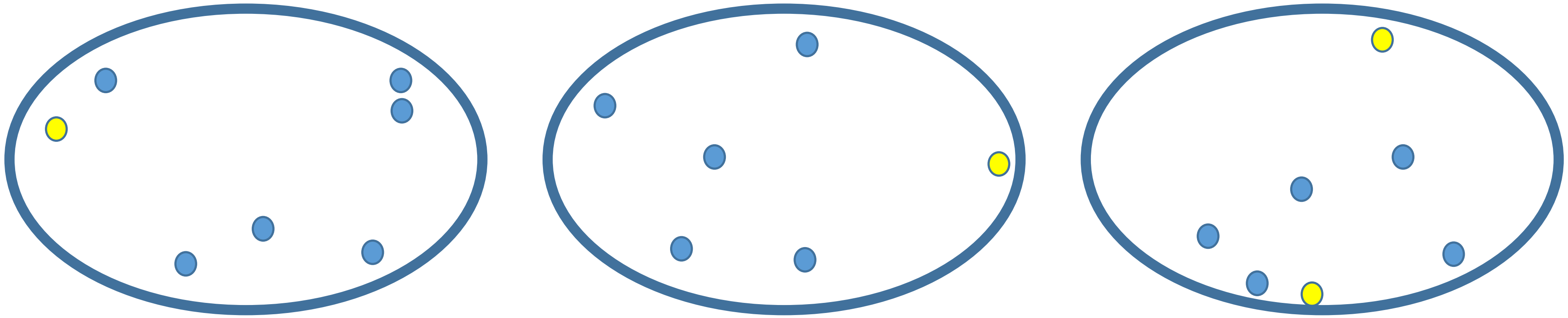
Let $V_1, \dots, V_m$ be the set of points, $V = \cup_i V_i$

Let $S_1, \dots, S_m$ be their core-sets, $S = \cup_i S_i$

Let $Opt = \{o_1, \dots, o_k\}$ be the optimal solution

Goal: $div_k(S) \geq div_k(V)/c$

Goal: $div_k(S) \geq div_k(Opt)/c$



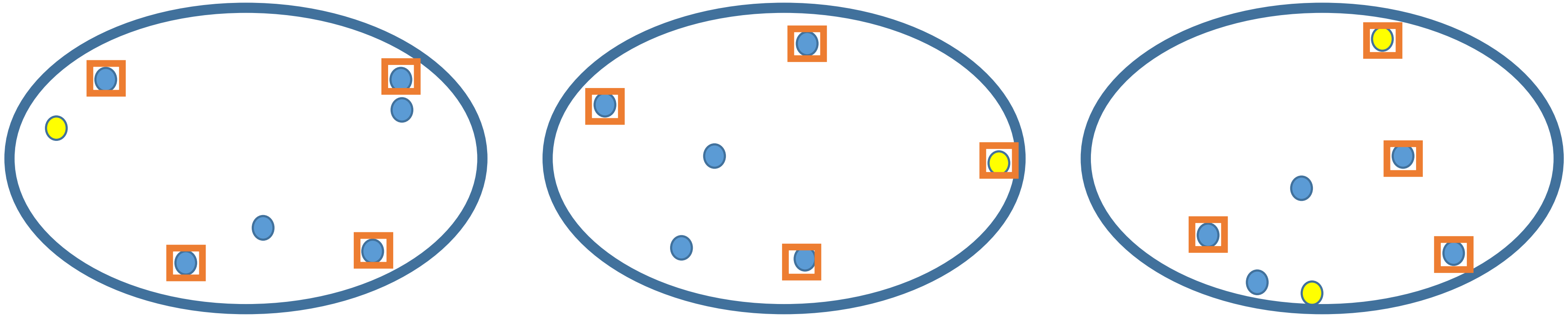
Let $V_1, \dots, V_m$ be the set of points, $V = \bigcup_i V_i$

Let $S_1, \dots, S_m$ be their core-sets, $S = \bigcup_i S_i$

Let $Opt = \{o_1, \dots, o_k\}$ be the optimal solution

Let r be the maximum diversity $r = \max_i div_k(S_i)$

Goal: $div_k(S) \geq div_k(V)/c$
Goal: $div_k(S) \geq div_k(Opt)/c$
Note: $div_k(S) \geq r$



Let $V_1, \dots, V_m$ be the set of points, $V = \cup_i V_i$

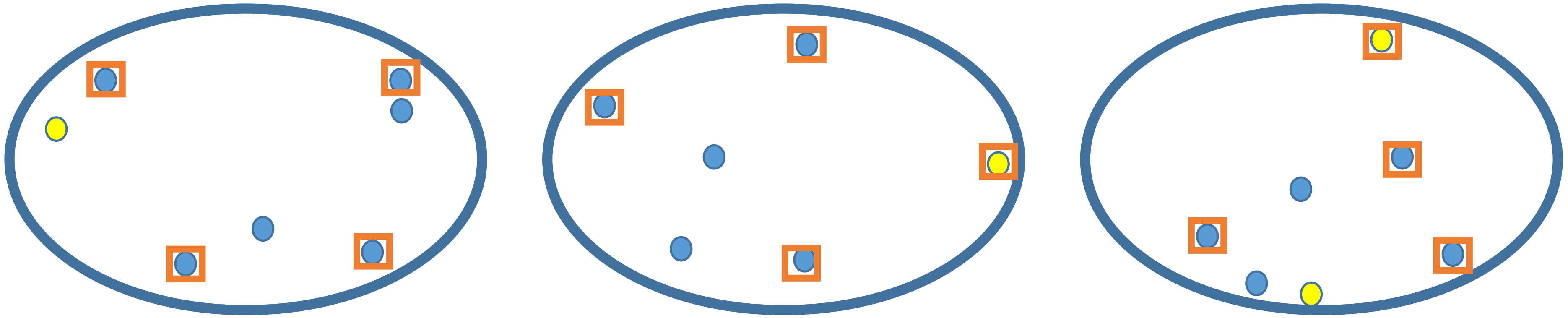
Let $S_1, \dots, S_m$ be their core-sets, $S = \cup_i S_i$

Let $Opt = \{o_1, \dots, o_k\}$ be the optimal solution

Let r be the maximum diversity $r = \max_i div_k(S_i)$

Case 1: one of S_i has diversity as good as the optimum: $r \geq div(Opt)/\text{c}$

Goal: $div_k(S) \geq div_k(V)/c$
Goal: $div_k(S) \geq div_k(Opt)/c$
Note: $div_k(S) \geq r$



Let $V_1, \dots, V_m$ be the set of points, $V = \cup_i V_i$

Let $S_1, \dots, S_m$ be their core-sets, $S = \cup_i S_i$

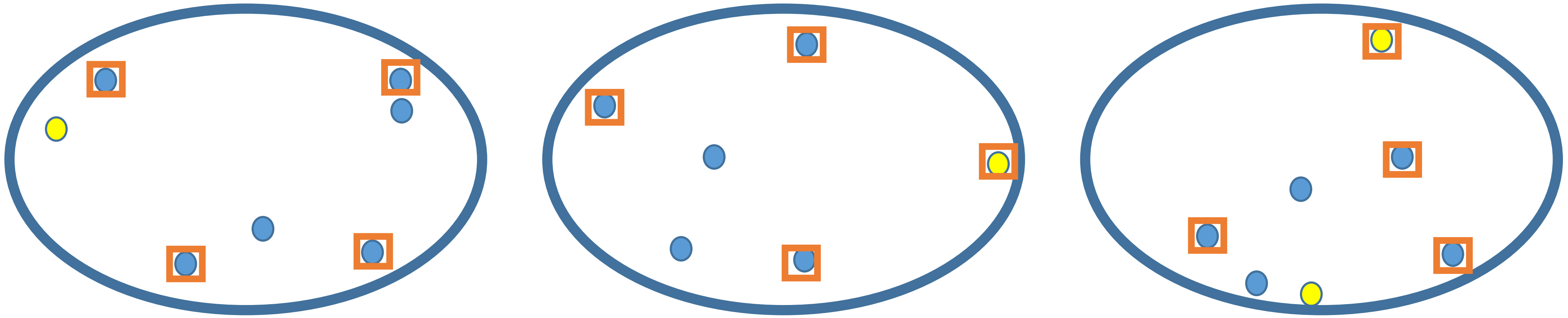
Let $Opt = \{o_1, \dots, o_k\}$ be the optimal solution

Let r be the maximum diversity $r = \max_i div_k(S_i)$

Goal: $div_k(S) \geq div_k(V)/c$
Goal: $div_k(S) \geq div_k(Opt)/c$
Note: $div_k(S) \geq r$

Case 1: one of S_i has diversity as good as the optimum: $r \geq div(Opt)/c$

Case 2: $r \leq div(Opt)/c$



Let $V_1, \dots, V_m$ be the set of points, $V = \cup_i V_i$

Let $S_1, \dots, S_m$ be their core-sets, $S = \cup_i S_i$

Let $Opt = \{o_1, \dots, o_k\}$ be the optimal solution

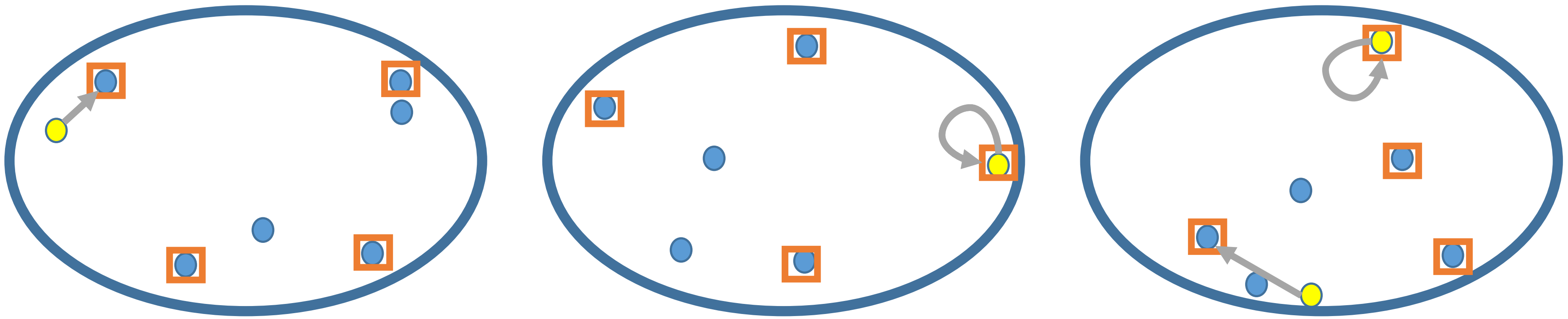
Let r be the maximum diversity $r = \max_i div_k(S_i)$

Goal: $div_k(S) \geq div_k(V)/c$
Goal: $div_k(S) \geq div_k(Opt)/c$
Note: $div_k(S) \geq r$

Case 1: one of S_i has diversity as good as the optimum: $r \geq div(Opt)/\mathrm{c}$

Case 2: $r \leq div(Opt)/\mathrm{c}$

- Define mapping μ from $Opt = \{o_1, \dots, o_k\}$ to S s.t. $dist\big(o_i, \mu(o_i)\big) \leq r$



Let $V_1, \dots, V_m$ be the set of points, $V = \bigcup_i V_i$

Let $S_1, \dots, S_m$ be their core-sets, $S = \bigcup_i S_i$

Let $Opt = \{o_1, \dots, o_k\}$ be the optimal solution

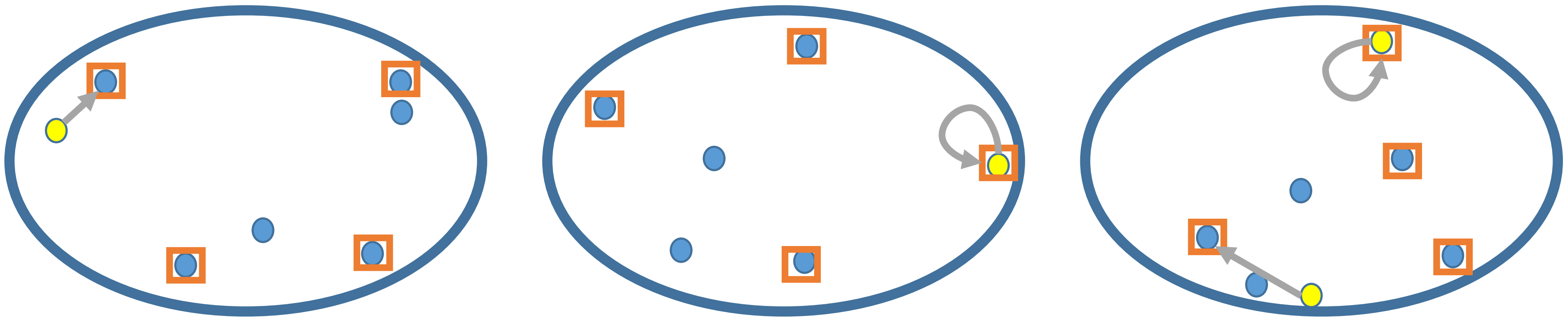
Let r be the maximum diversity $r = \max_i div_k(S_i)$

Goal: $div_k(S) \geq div_k(V)/c$
Goal: $div_k(S) \geq div_k(Opt)/c$
Note: $div_k(S) \geq r$

Case 1: one of S_i has diversity as good as the optimum: $r \geq div(Opt)$/c

Case 2: $r \leq div(Opt)$/c

- Define mapping μ from $Opt = \{o_1, \dots, o_k\}$ to S s.t. $dist\big(o_i, \mu(o_i)\big) \leq r$
- Replacing o_i with $\mu(o_i)$ has still large diversity
- $div(\{\mu(o_i)\})$ is approximately as good ad $div(\{o_i\})$

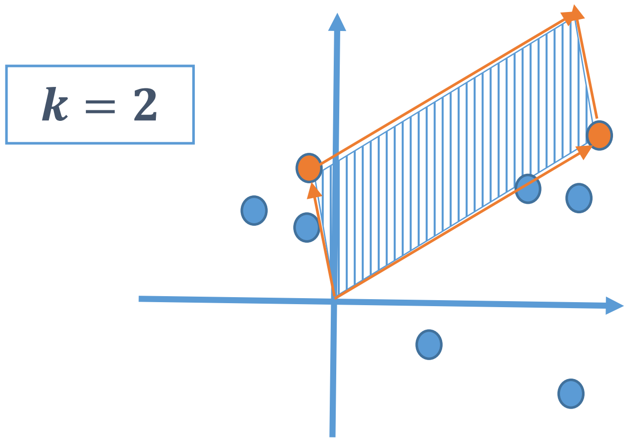


Diversity: Volume

Diversity: Volume

Input: a set of n vectors $\mathrm{V} \subset \mathbb{R}^d$ and a parameter $k \leq d$,

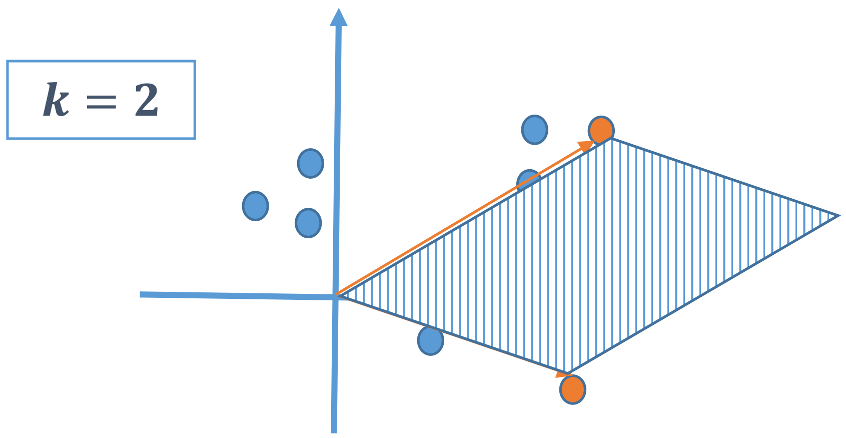
Goal: pick k points s.t. the **volume of the parallelepiped** spanned by the picked points is maximized.



Diversity: Volume

Input: a set of n vectors $\mathrm{V} \subset \mathbb{R}^d$ and a parameter $k \leq d$,

Goal: pick k points s.t. the **volume of the parallelepiped** spanned by the picked points is maximized.

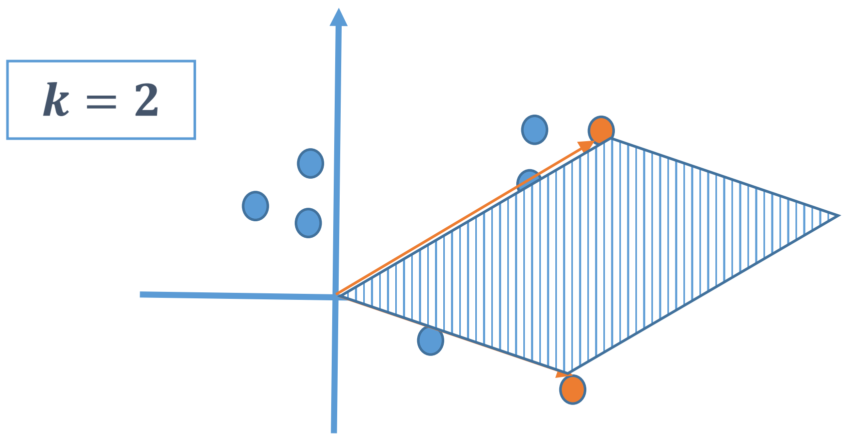


Diversity: Volume

Input: a set of n vectors $V \subset \mathbb{R}^d$ and a parameter $k \leq d$,

Goal: pick k points s.t. the **volume of the parallelepiped** spanned by the picked points is maximized.

- ❑ **Convex optimization + randomized rounding** $O(e^{k/2})$ [Nik'15]
- ❑ Hard to approximate within $\Omega(c^k)$ [CMI'13]
- ❑ Greedy is used in practice, achieves $k!$ [CMI'07]

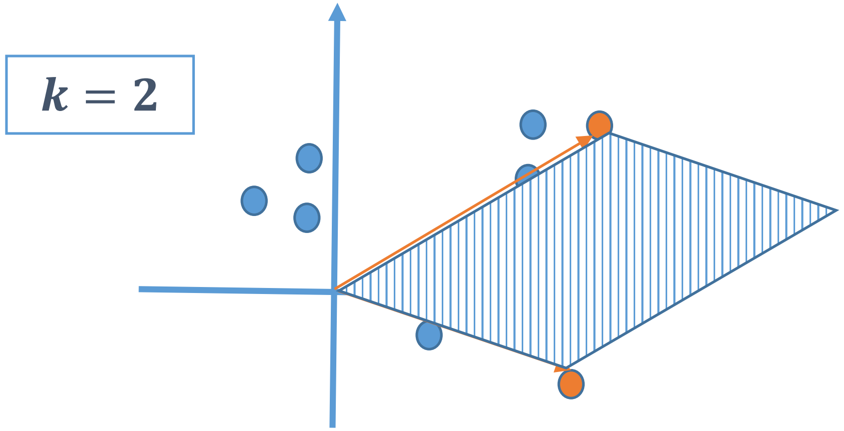


Diversity: Volume

Input: a set of n vectors $\mathrm{V} \subset \mathbb{R}^d$ and a parameter $k \leq d$,

Goal: pick k points s.t. the **volume of the parallelepiped** spanned by the picked points is maximized.

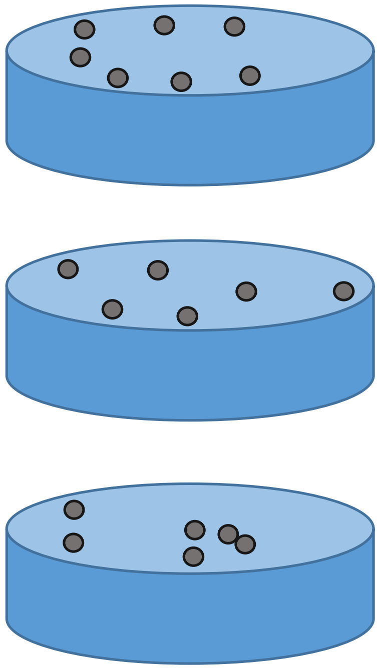
- ❑ **Convex optimization + randomized rounding** $O(e^{k/2})$ [Nik'15]
- ❑ Hard to approximate within $\Omega(c^k)$ [CMI'13]
- ❑ Greedy is used in practice, achieves $k!$ [CMI'07]



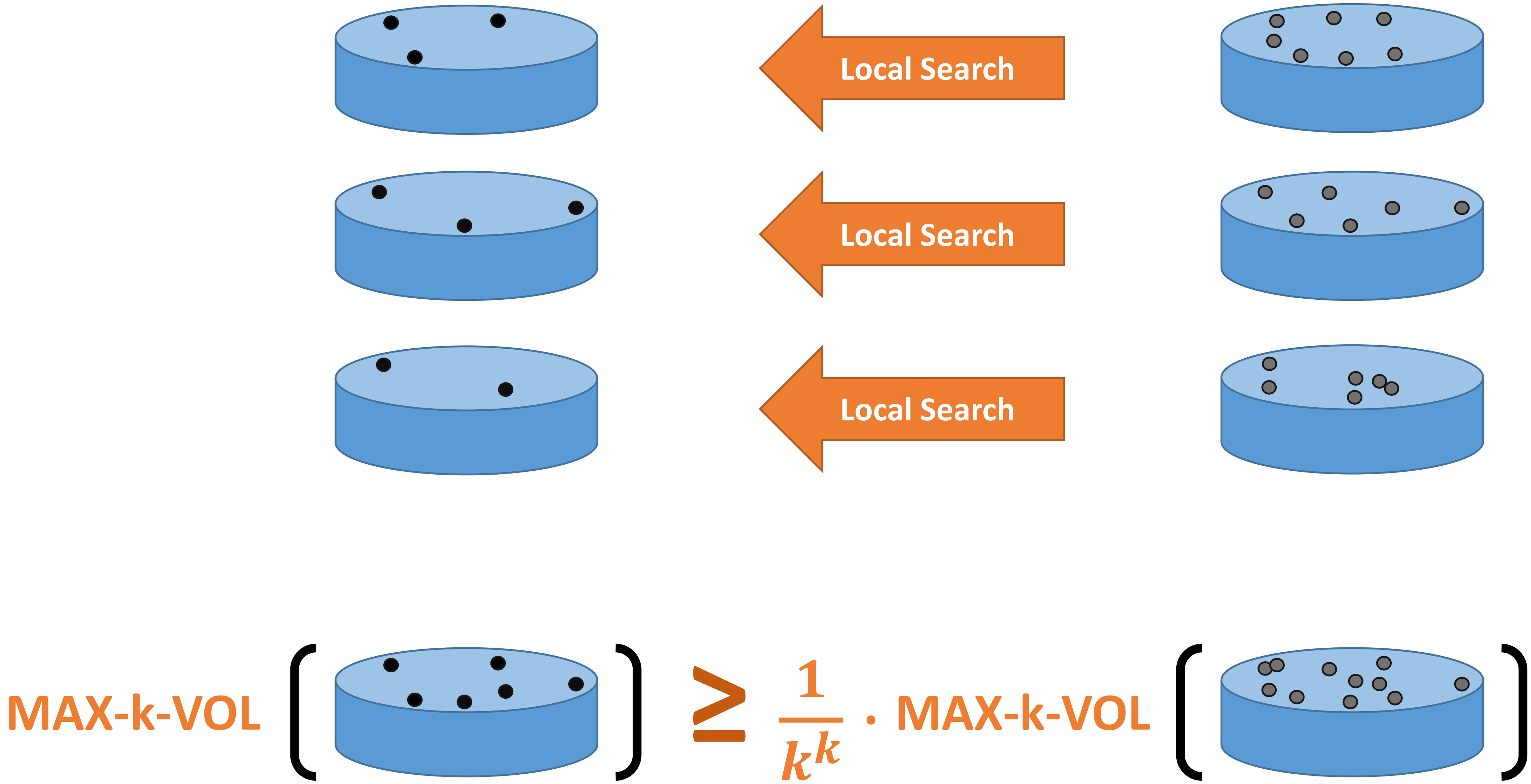
- ❑ Higher order notion of diversity (not based on pairwise distances only)

The Local Search Algorithm

The Local Search Algorithm produces a composable core-set of size k with approximation factor $O(k)^k$ for the volume maximization problem.



Local Search
Local Search
Local Search


Local Search
Local Search
Local Search
MAX-k-VOL
≥
$\frac{1}{k^k}$ ·
MAX-k-VOL

The Local Search Algorithm

Input: a set V of n points and a parameter k

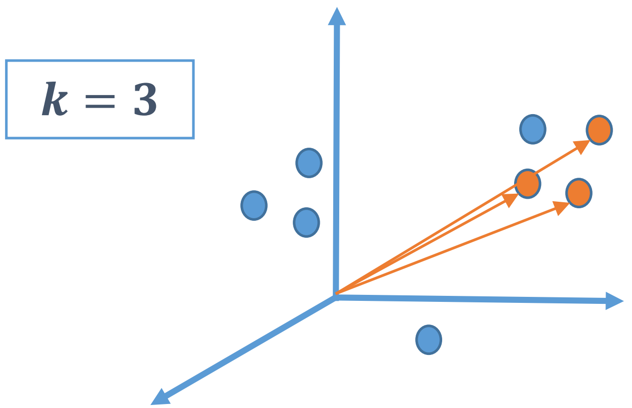
1. Start with an arbitrary subset of k points $S \subseteq V$
2. While there exists a point $p \in V \setminus S$ and $q \in S$ s.t. replacing q with p increases the volume, then swap them, i.e., $S = S \cup \{p\} \setminus \{q\}$

The Local Search Algorithm

Input: a set V of n points and a parameter k

1. Start with an arbitrary subset of k points $S \subseteq V$

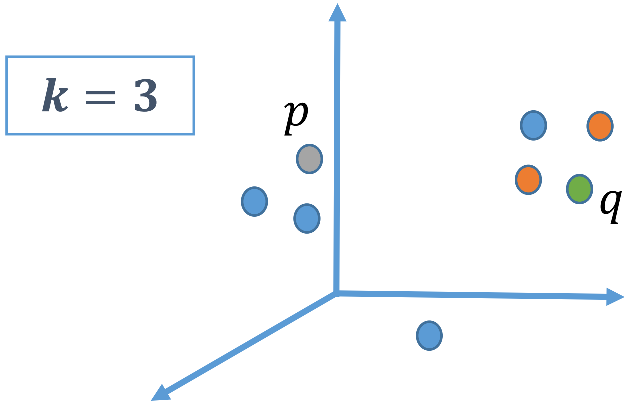
2. While there exists a point $p \in V \setminus S$ and $q \in S$ s.t. replacing q with p increases the volume, then swap them, i.e., $S = S \cup \{p\} \setminus \{q\}$



The Local Search Algorithm

Input: a set V of n points and a parameter k

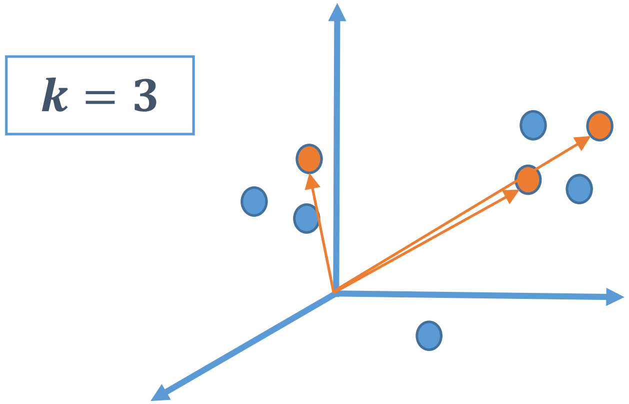
1. Start with an arbitrary subset of k points $S \subseteq V$
2. While there exists a point $p \in V \setminus S$ and $q \in S$ s.t. replacing q with p increases the volume, then swap them, i.e., $S = S \cup \{p\} \setminus \{q\}$



The Local Search Algorithm

Input: a set V of n points and a parameter k

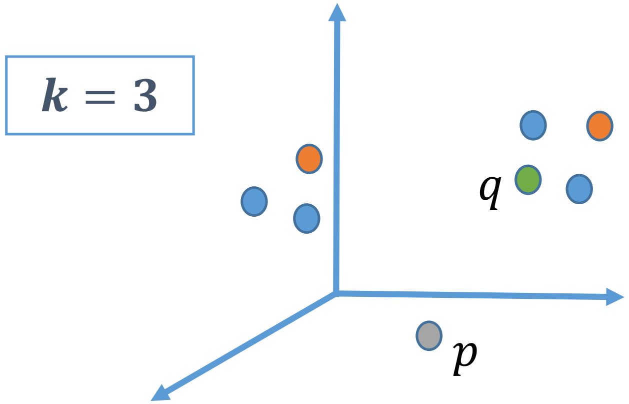
1. Start with an arbitrary subset of k points $S \subseteq V$
2. While there exists a point $p \in V \setminus S$ and $q \in S$ s.t. replacing q with p increases the volume, then swap them, i.e., $S = S \cup \{p\} \setminus \{q\}$



The Local Search Algorithm

Input: a set V of n points and a parameter k

1. Start with an arbitrary subset of k points $S \subseteq V$
2. While there exists a point $p \in V \setminus S$ and $q \in S$ s.t. replacing q with p increases the volume, then swap them, i.e., $S = S \cup \{p\} \setminus \{q\}$

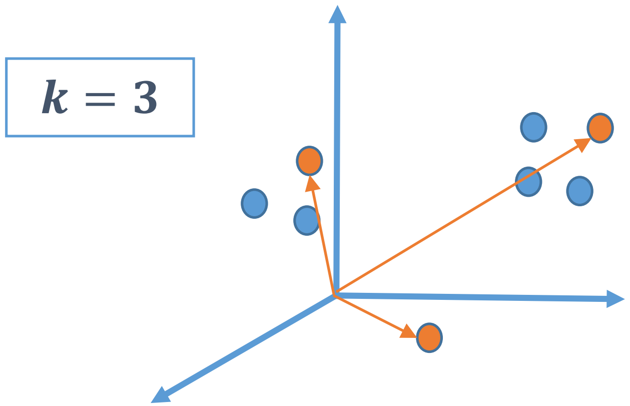


The Local Search Algorithm

Input: a set V of n points and a parameter k

1. Start with an arbitrary subset of k points $S \subseteq V$

2. While there exists a point $p \in V \setminus S$ and $q \in S$ s.t. replacing q with p increases the volume, then swap them, i.e., $S = S \cup \{p\} \setminus \{q\}$



To bound the run time

Input: a set V of n points a

Start with a crude approximation (Greedy algorithm)

1. Start with an **arbitrary** subset of k points $S \subseteq V$
2. While there exists a point $p \in V \setminus S$ and $q \in S$ s.t. replacing q with p **increases** the volume, then swap them, i.e., $S = S \cup \{p\} \setminus \{q\}$

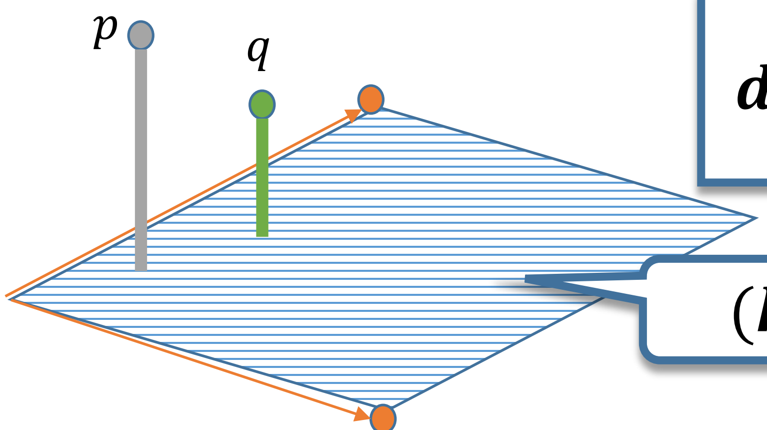
If it increases by at least a factor of $(1 + \epsilon)$

Checking the condition

Input: a set V of n points and a parameter k

1. Start with an arbitrary subset of k points $S \subseteq V$
2. While there exists a point $p \in V \setminus S$ and $q \in S$ s.t. replacing q with p **increases the volume**, then swap them, i.e., $S = S \cup \{p\} \setminus \{q\}$

$$\boldsymbol{dist(p, H_{S\setminus\{q\}}) > dist(q, H_{S\setminus\{q\}})}$$



$(k-1)$-dimensional Subspace

Local Search Lemma

Local Search gives a $2k$ −approximate **core-set** for ***k*-directional height.**

Will define shortly

Local Search Lemma

Local Search gives a $2k$ −approximate **core-set** for ***k*-directional height.**

Height-Volume Lemma

Any α **core-set** for k**-directional height** gives a α^k core-set for **volume maximization**

Local Search Lemma

Local Search gives a $2k-$approximate **core-set** for ***k*-directional height.**

+

Height-Volume Lemma

Any α **core-set** for ***k*-directional height** gives a α^k core-set for **volume maximization**

Local Search Lemma

Local Search gives a $2k$ −approximate **core-set** for ***k*-directional height.**

+

Height-Volume Lemma

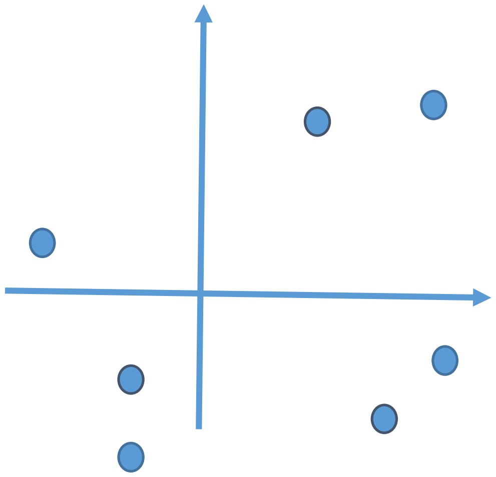
Any α **core-set** for ***k*-directional height** gives a α^k core-set for **volume maximization**

$\boldsymbol{\alpha = 2k}$

Theorem

Local Search produces a $O(k)^k$ core-set for volume maximization.

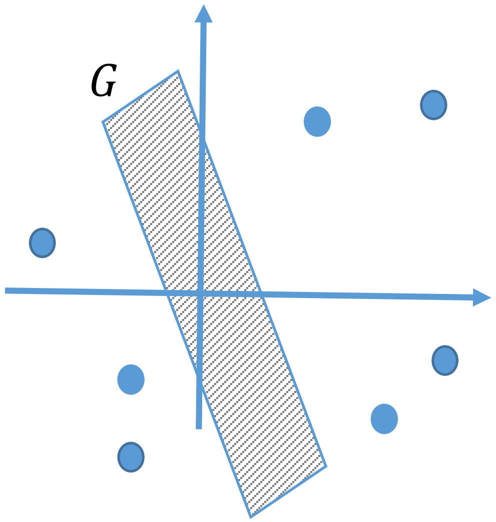
$\boldsymbol{k}$-Directional Height



Given

- a point set P, and

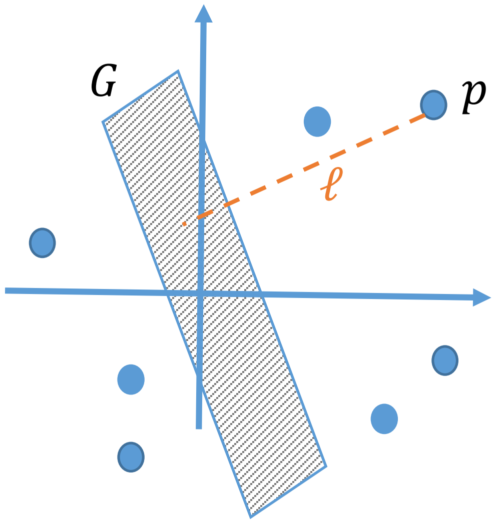
$\boldsymbol{k}$-Directional Height



Given

- a point set P, and
- a $(k-1)$-dimensional subspace $\boldsymbol{G}$ (direction),

$\boldsymbol{k}$-Directional Height



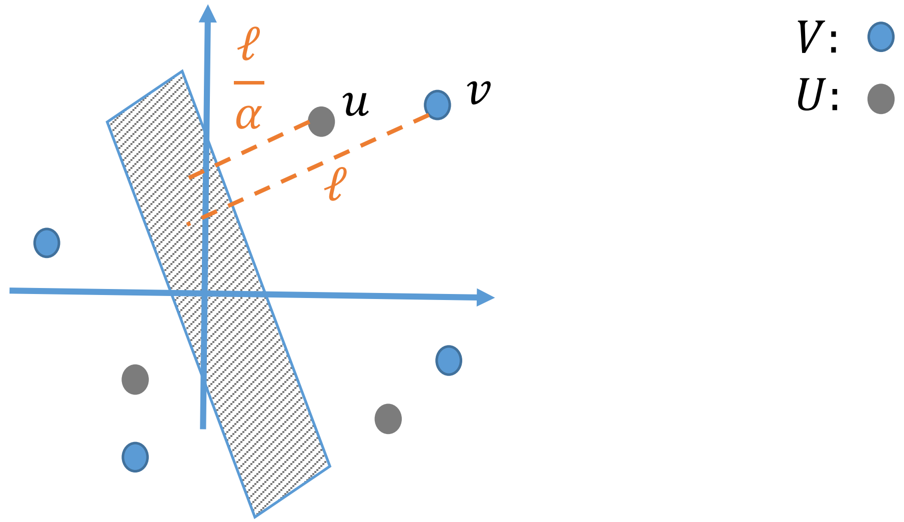
Given

- a point set P, and
- a $(k-1)$-dimensional subspace $\boldsymbol{G}$ (direction),

The $\boldsymbol{k}$-Directional Height of P in the direction of G is defined as

$$\max_{p \in P} dist(p, G)$$

$\boldsymbol{\alpha}$ –Core-set for $\boldsymbol{k}$-Directional Height



A subset of points that preserve the k-directional height for all subspaces G of dimension $k-1$ at the same time upto an approximation factor α.

Local Search Lemma

Local Search gives a $2k$ −approximate **core-set** for ***k*-directional height.**

Local Search Lemma

Local Search gives a $2k$ −approximate **core-set** for ***k*-directional height.**

- V is the point set
- $S = LS(V)$ is the core-set produced by local search

Local Search Lemma

Local Search gives a $2k$ –approximate core-set for k-directional height.

Need to prove:

For any $(k-1)$-dimensional subspace G

$$\max_{q \in S} dist(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} dist(p, G)$$

Local Search Lemma:

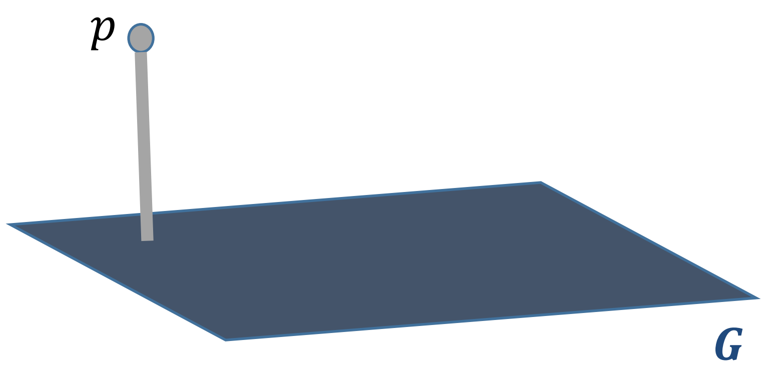
For any $(k-1)$-dimensional subspace G, the maximum distance of the point set to G is approximately preserved

$$\max_{s \in S} dist(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} dist(p, G)$$

Local Search Lemma:

For any $(k-1)$-dimensional subspace G, the maximum distance of the point set to G is approximately preserved

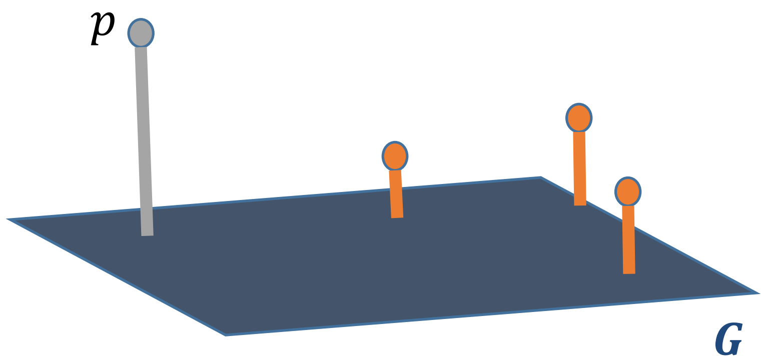
$$\max_{s \in S} dist(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} dist(p, G)$$



Local Search Lemma:

For any $(k-1)$-dimensional subspace G, the maximum distance of the point set to G is approximately preserved

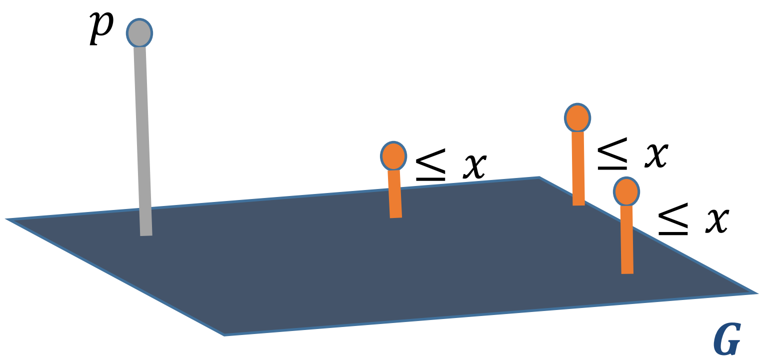
$$\max_{s \in S} dist(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} dist(p, G)$$



Local Search Lemma:

For any $(k-1)$-dimensional subspace G, the maximum distance of the point set to G is approximately preserved

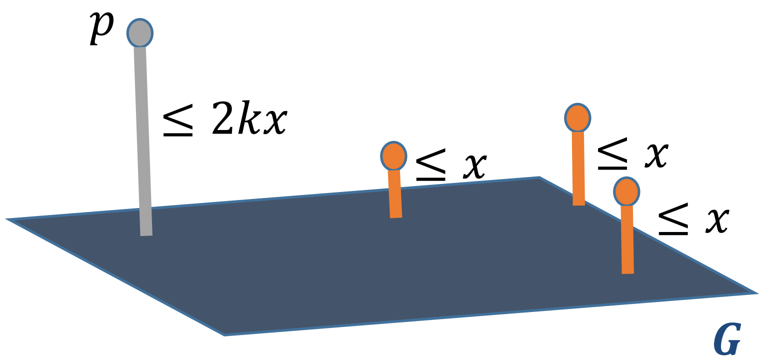
$$\max_{s \in S} dist(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} dist(p, G)$$



Local Search Lemma:

For any $(k-1)$-dimensional subspace G, the maximum distance of the point set to G is approximately preserved

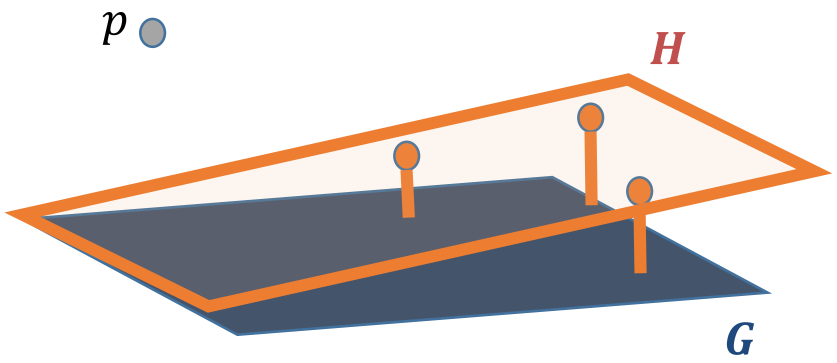
$$\max_{s \in S} dist(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} dist(p, G)$$



Goal: $\boldsymbol{d(p, G) \leq 2kx}$

Local Search Lemma:
For any $(k-1)$-dimensional subspace G, the maximum distance of the point set to G is approximately preserved

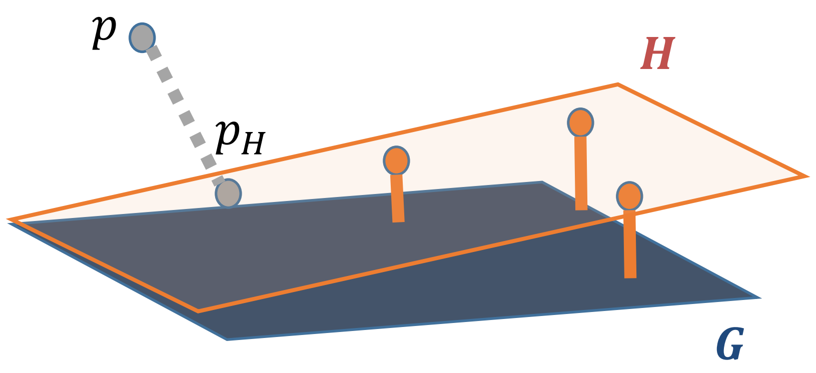
$$\max_{s \in S} dist(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} dist(p, G)$$

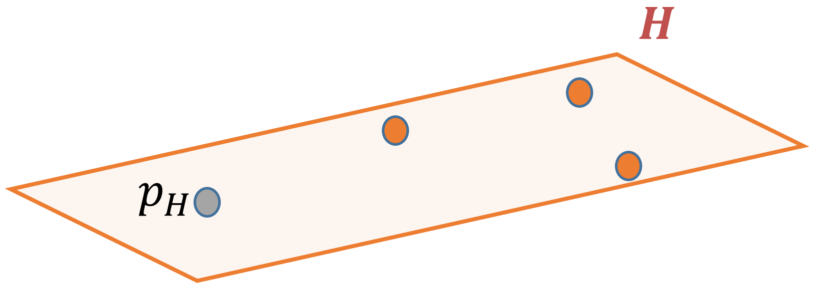


Local Search Lemma:

For any $(k-1)$-dimensional subspace G, the maximum distance of the point set to G is approximately preserved

$$\max_{s \in S} dist(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} dist(p, G)$$

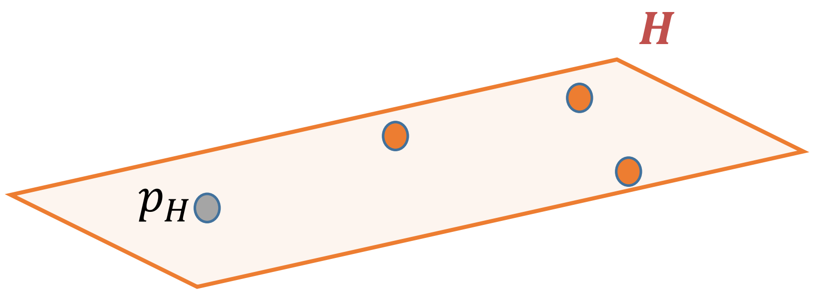




We can write p_H as linear combination of core-set points,

$$p_H = \sum_{i=1}^{k} \alpha_i q_i$$

Properties of Local Search



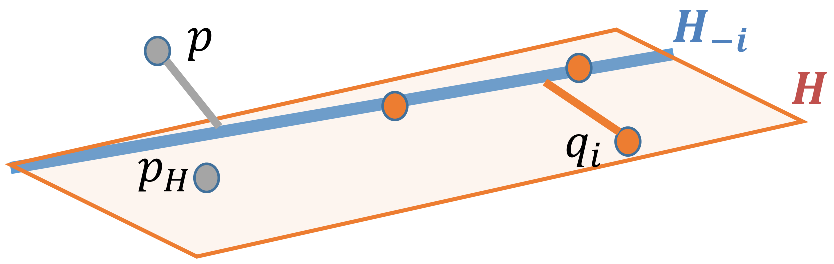
We can write p_H as linear combination of core-set points, with **small coefficient**.

$$p_H = \sum_{i=1}^{k} \alpha_i q_i \quad \text{s.t.} \quad \text{all } |\alpha_i| \le 1$$

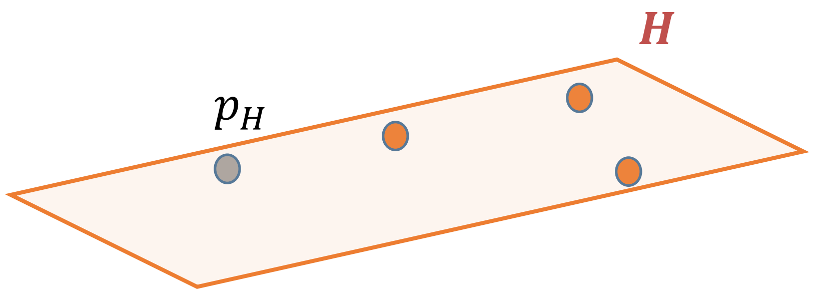
Properties of Local Search

We can write p_H as linear combination of core-set points, with **small coefficient**.

$$p_H = \sum_{i=1}^{k} \alpha_i q_i \quad \text{s.t.} \quad \text{all } |\alpha_i| \leq 1$$



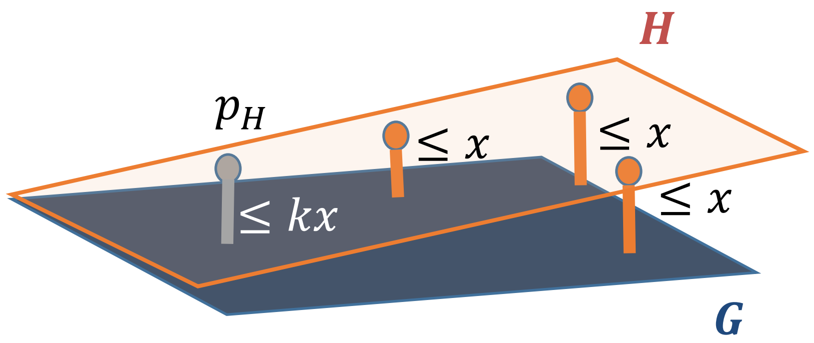
Properties of Local Search



We can write p_H as linear combination of core-set points, with small coefficient.

$$p_H = \sum_{i=1}^{k} \alpha_i q_i \quad \text{s.t.} \quad \text{all } |\alpha_i| \leq 1$$

Properties of Local Search



We can write p_H as linear combination of core-set points, with small coefficient.

$$p_H = \sum_{i=1}^{k} \alpha_i q_i \quad \text{s.t.} \quad \text{all } |\alpha_i| \leq 1$$

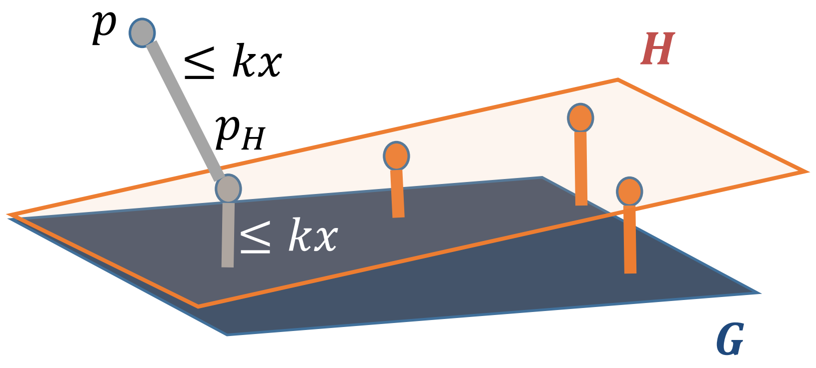
Triangle Inequality

$$\boldsymbol{d(p_H, G) \leq kx}$$

Local Search Lemma:

For any $(k-1)$-dimensional subspace G, the maximum distance of the point set to G is approximately preserved

$$\max_{s \in S} dist(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} dist(p, G)$$



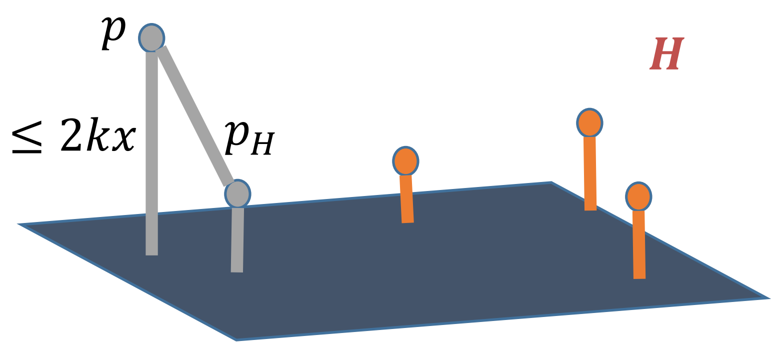
$$\boldsymbol{d(p, p_H) \leq kx}$$

$$\boldsymbol{d(p_H, G) \leq kx}$$

Local Search Lemma:

For any $(k-1)$-dimensional subspace G, the maximum distance of the point set to G is approximately preserved

$$\max_{s \in S} dist(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} dist(p, G)$$



Goal: $\boldsymbol{d(p, G) \leq 2kx}$

Local Search Lemma

Local Search gives a $2k$ −approximate **core-set** for ***k*-directional height.**

Height-Volume Lemma

Any α **core-set** for ***k*-directional height** gives a α^k core-set for **volume maximization**

$$\boldsymbol{\alpha = 2k}$$

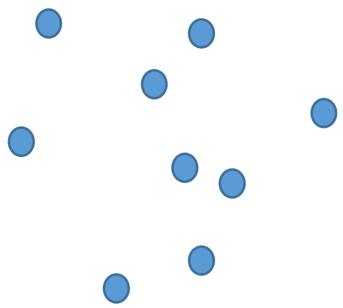
Theorem:

Local Search produces a $O(k)^k$ core-set for volume maximization.

Height-Volume Lemma

Any α **core-set** for k**-directional height** gives a α^k composable core-set for **volume maximization**

Let $V = \bigcup_i V_i$ be the union of the point sets

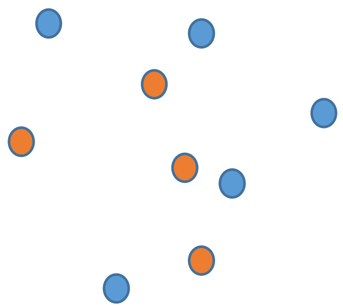


Height-Volume Lemma

Any α **core-set** for **k-directional height** gives a α^k composable core-set for **volume maximization**

Let $V = \bigcup_i V_i$ be the union of the point sets

Let $S = \bigcup_i S_i$ be the union of core-sets



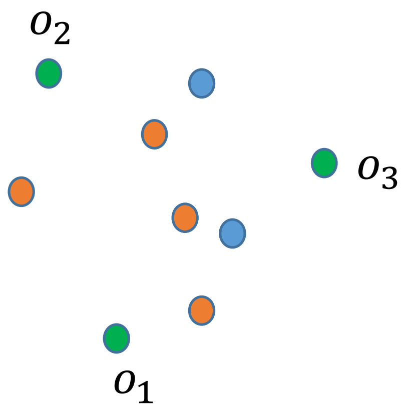
Height-Volume Lemma

Any α **core-set** for k**-directional height** gives a α^k composable core-set for **volume maximization**

Let $\boldsymbol{V} = \cup_i V_i$ be the union of the point sets

Let $\boldsymbol{S} = \cup_i S_i$ be the union of core-sets

Let $\boldsymbol{Opt} = \{\boldsymbol{o_1}, \dots, \boldsymbol{o_k}\} \subset V$ be the optimal subset of points maximizing the volume



Height-Volume Lemma

Any α **core-set** for k**-directional height** gives a α^k composable core-set for **volume maximization**

Let $\boldsymbol{V} = \bigcup_i V_i$ be the union of the point sets

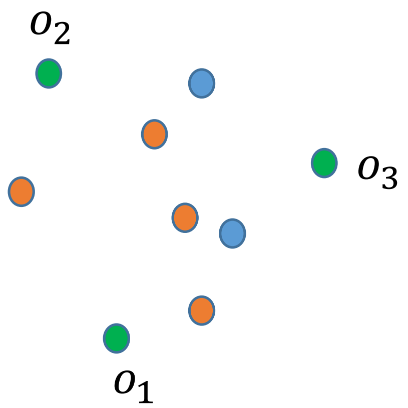
Let $\boldsymbol{S} = \bigcup_i S_i$ be the union of core-sets

Let $\boldsymbol{Opt} = \{\boldsymbol{o_1}, \dots, \boldsymbol{o_k}\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1\ to\ k$

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any α **core-set** for **k-directional height** gives a α^k composable core-set for **volume maximization**

Let $\boldsymbol{V} = \cup_i V_i$ be the union of the point sets

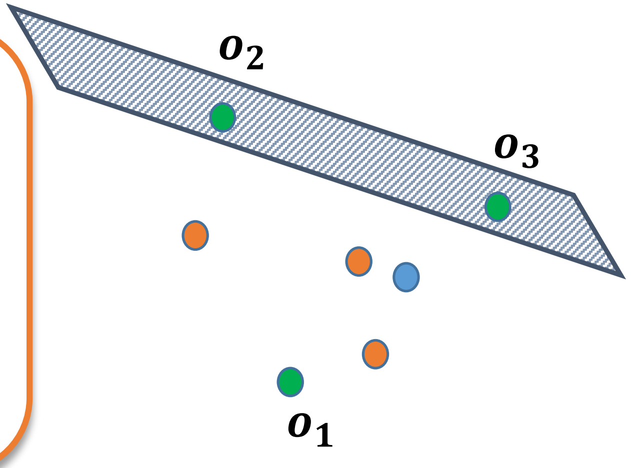
Let $\boldsymbol{S} = \cup_i S_i$ be the union of core-sets

Let $\boldsymbol{Opt} = \{\boldsymbol{o_1}, \dots, \boldsymbol{o_k}\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1\ to\ k$

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any **α core-set** for **k-directional height** gives a α^k composable core-set for **volume maximization**

Let $\boldsymbol{V} = \cup_i V_i$ be the union of the point sets

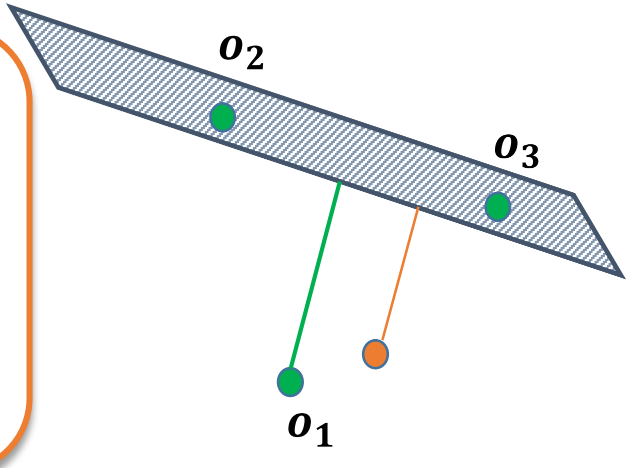
Let $\boldsymbol{S} = \cup_i S_i$ be the union of core-sets

Let $\boldsymbol{Opt} = \{\boldsymbol{o_1}, \dots, \boldsymbol{o_k}\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1$ to k

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any α **core-set** for k**-directional height** gives a α^k composable core-set for **volume maximization**

Let $V = \bigcup_i V_i$ be the union of the point sets

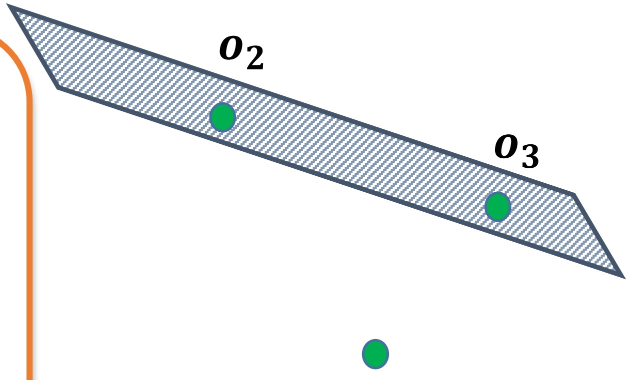
Let $S = \bigcup_i S_i$ be the union of core-sets

Let $Opt = \{o_1, \dots, o_k\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1\ to\ k$

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any α **core-set** for k**-directional height** gives a α^k composable core-set for **volume maximization**

Let $\boldsymbol{V} = \bigcup_i V_i$ be the union of the point sets

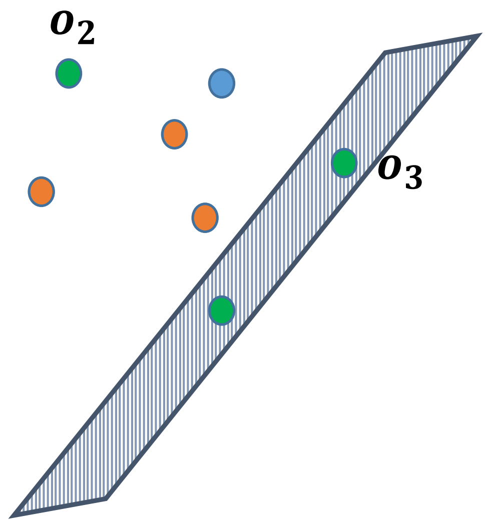
Let $\boldsymbol{S} = \bigcup_i S_i$ be the union of core-sets

Let $\boldsymbol{Opt} = \{\boldsymbol{o_1, \dots, o_k}\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1$ to k

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol\setminus\{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any α **core-set** for k**-directional height** gives a α^k composable core-set for **volume maximization**

Let $V = \bigcup_i V_i$ be the union of the point sets

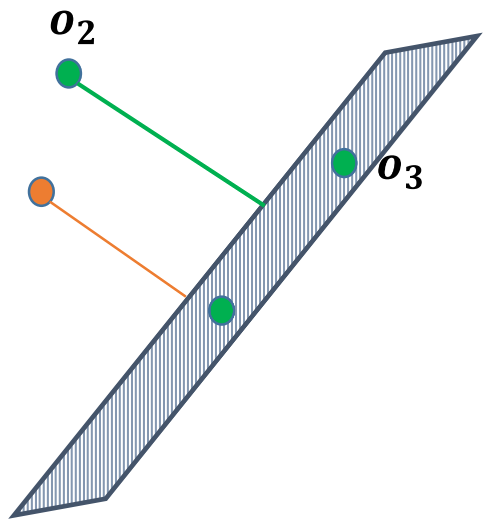
Let $S = \bigcup_i S_i$ be the union of core-sets

Let $Opt = \{o_1, \dots, o_k\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1$ to k

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any α **core-set** for **k-directional height** gives a α^k composable core-set for **volume maximization**

Let $\boldsymbol{V} = \cup_i V_i$ be the union of the point sets

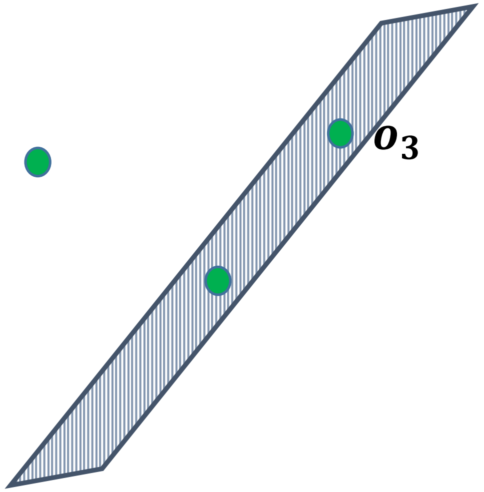
Let $\boldsymbol{S} = \cup_i S_i$ be the union of core-sets

Let $\boldsymbol{Opt} = \{\boldsymbol{o_1}, \ldots, \boldsymbol{o_k}\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1\ to\ k$

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any α **core-set** for k**-directional height** gives a α^k composable core-set for **volume maximization**

Let $\boldsymbol{V} = \bigcup_i V_i$ be the union of the point sets

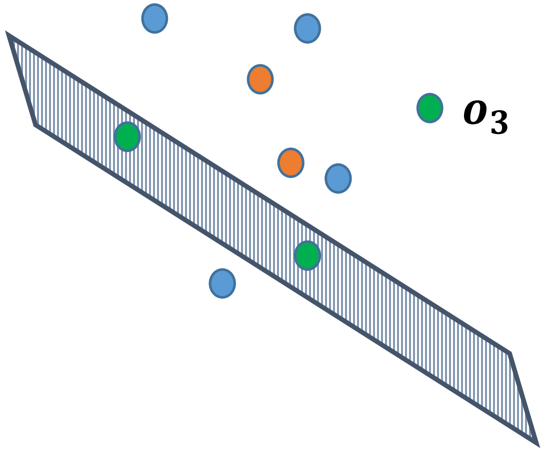
Let $\boldsymbol{S} = \bigcup_i S_i$ be the union of core-sets

Let $\boldsymbol{Opt} = \{\boldsymbol{o_1}, \dots, \boldsymbol{o_k}\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1$ to k

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any α **core-set** for k**-directional height** gives a α^k composable core-set for **volume maximization**

Let $\boldsymbol{V} = \bigcup_i V_i$ be the union of the point sets

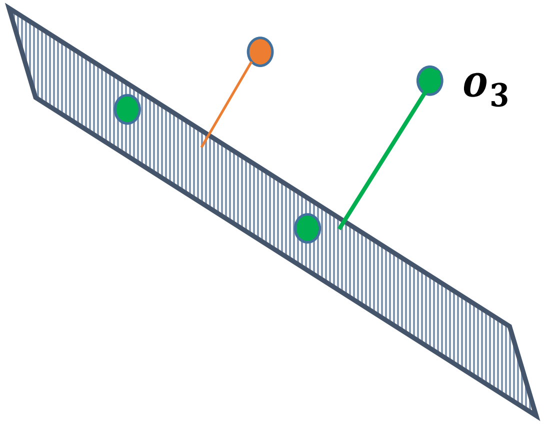
Let $\boldsymbol{S} = \bigcup_i S_i$ be the union of core-sets

Let $\boldsymbol{Opt} = \{\boldsymbol{o_1}, \dots, \boldsymbol{o_k}\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1 \ to \ k$

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma
Any **α core-set** for **k-directional height** gives a α^k composable core-set for **volume maximization**

Let $\boldsymbol{V} = \bigcup_i V_i$ be the union of the point sets

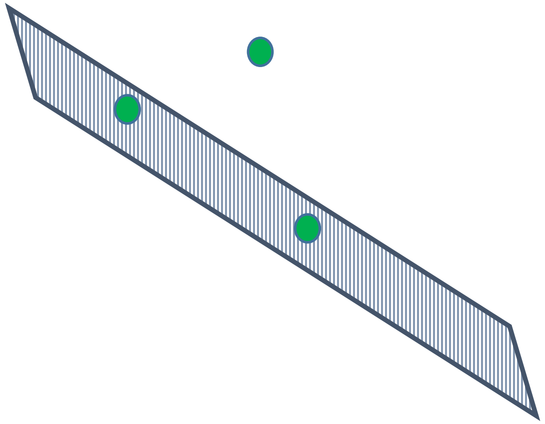
Let $\boldsymbol{S} = \bigcup_i S_i$ be the union of core-sets

Let $\boldsymbol{Opt} = \{\boldsymbol{o_1}, \dots, \boldsymbol{o_k}\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1\ to\ k$

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any α **core-set** for **k-directional height** gives a α^k composable core-set for **volume maximization**

Let $\boldsymbol{V} = \bigcup_i V_i$ be the union of the point sets

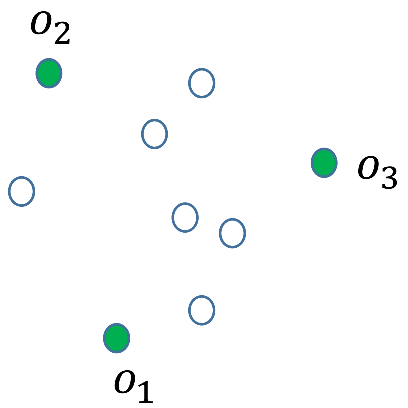
Let $\boldsymbol{S} = \bigcup_i S_i$ be the union of core-sets

Let $\boldsymbol{Opt} = \{\boldsymbol{o_1}, \dots, \boldsymbol{o_k}\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1\ to\ k$

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any α **core-set** for k**-directional height** gives a α^k composable core-set for **volume maximization**

Let $V = \bigcup_i V_i$ be the union of the point sets

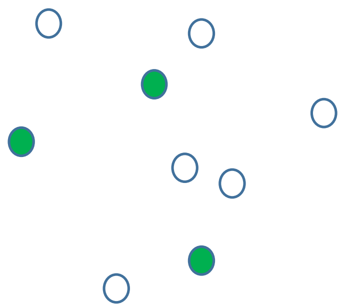
Let $S = \bigcup_i S_i$ be the union of core-sets

Let $Opt = \{o_1, \dots, o_k\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1\ to\ k$

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



Height-Volume Lemma

Any **α core-set** for **k-directional height** gives a α^k composable core-set for **volume maximization**

Let $V = \cup_i V_i$ be the union of the point sets

Let $S = \cup_i S_i$ be the union of core-sets

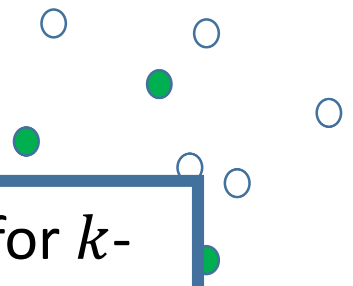
Let $Opt = \{o_1, \dots, o_k\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1 \ to \ k$

- Let $q_i \in S$ be the point that is fa
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$

Since we have a **α core-set** for k-directional height



➢ Lose a factor of at most α at each iteration

Height-Volume Lemma

Any α **core-set** for **k-directional height** gives a α^k composable core-set for **volume maximization**

Let $V = \bigcup_i V_i$ be the union of the point sets

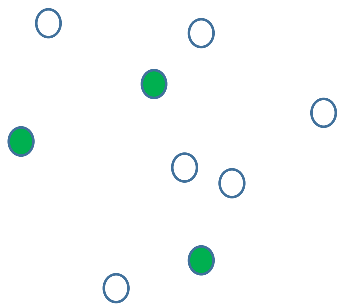
Let $S = \bigcup_i S_i$ be the union of core-sets

Let $Opt = \{o_1, \dots, o_k\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1 \; to \; k$

- Let $q_i \in S$ be the point that is farthest away from $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$



➢ Lose a factor of at most α at each iteration

➢ Total approximation factor α^k

Local Search Lemma

Local Search gives a $2k$ −approximate **core-set** for ***k*-directional height.**

+

Height-Volume Lemma

Any α **core-set** for ***k*-directional height** gives a α^k core-set for **volume maximization**

$$\boldsymbol{\alpha = 2k}$$

Theorem

Local Search produces a $O(k)^k$ core-set for volume maximization.