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: 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.

Solving the problem over core-set $U$

→

Solving the problem over dataset $V$ (approximately)
Core-sets: a small subset $U$ of the data $V$ that represents it well.

- Task specific

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

Solving the problem over core-set $U$

$\approx$

Solving the problem over dataset $V$ (approximately)
Core-sets [Agarwal, Har-Peled, Varadarajan’05]

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

- Task specific

Solving the problem over core-set $U$

$\approx \frac{1}{\alpha}$

Solving the problem over dataset $V$ (approximately)
Core-sets [Agarwal, Har-Peled, Varadarajan’05]

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

Convex Hull is a 1-core-set for Diameter

$$\text{Diameter} \left(\begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array}\right) = \text{Diameter} \left(\begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array}\right)$$
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 $q$,
- $\frac{\text{Far}(q, P)}{\alpha} \leq \text{Far}(q, S) \leq \text{Far}(q, P)$
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 $q$,
- $\frac{\text{Far}(q, P)}{\alpha} \leq \text{Far}(q, S) \leq \text{Far}(q, P)$
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 $q$,
• $\frac{\text{Far}(q, P)}{\alpha} \leq \text{Far}(q, S) \leq \text{Far}(q, P)$
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 $q$,
  • $\text{Far}(q, P)/\alpha \leq \text{Far}(q, S) \leq \text{Far}(q, P)$
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 $q$,
  • $\frac{\text{Far}(q, P)}{\alpha} \leq \text{Far}(q, S) \leq \text{Far}(q, P)$

• 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 $q$, 
  - $\frac{\text{Far}(q, P)}{\alpha} \leq \text{Far}(q, S) \leq \text{Far}(q, P)$

- 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 $q$,
- $\frac{\text{Far}(q, P)}{\alpha} \leq \text{Far}(q, S) \leq \text{Far}(q, P)$

- 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 $q$,
  - $\frac{\text{Far}(q, P)}{\alpha} \leq \text{Far}(q, S) \leq \text{Far}(q, P)$

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

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

• 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 $q$,
  
  $\frac{Far(q, P)}{\alpha} \leq Far(q, S) \leq Far(q, P)$

• General setting?
  
  • Impose a grid of side length $\varepsilon r$
  • 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 $q$,
  
  • $\text{Far}(q, P)/\alpha \leq \text{Far}(q, S) \leq \text{Far}(q, P)$

• General setting?
  • Impose a grid of side length $\epsilon r$
  • 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 $q$,
  
  $\frac{\text{Far}(q, P)}{\alpha} \leq \frac{\text{Far}(q, S)}{\alpha} \leq \text{Far}(q, P)$

• General setting?
  • Impose a grid of side length $\epsilon r$
  • 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 \( q \),
- \( \frac{\text{Far}(q, P)}{\alpha} \leq \text{Far}(q, S) \leq \text{Far}(q, P) \)

- General setting?
  - Cover the unit sphere with vectors \( v_i \) with separation angle at most \( \epsilon \)
  - 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 $q$,
  
  • $\frac{Far(q, P)}{\alpha} \leq Far(q, S) \leq Far(q, P)$

• General setting?
  • Cover the unit sphere with vectors $v_i$ with separation angle at most $\epsilon$
  • 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 $\epsilon$
  • 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 $f$ be an optimization function
- Multiple data sets $V_1, \ldots, V_m$
Composable Core-sets

Core-sets with composability property:

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

- Let $f$ be an optimization function
- Multiple data sets $V_1, \ldots, V_m$ and their core-sets $U_1 \subset V_1, \ldots, U_m \subset V_m$, 
Composable Core-sets

Core-sets with composability property:

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

- Let $f$ be an optimization function
- Multiple data sets $V_1, \ldots, V_m$ and their core-sets $U_1 \subset V_1, \ldots, U_m \subset V_m$,
Composable Core-sets

Core-sets with composability property:

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

- Let $f$ be an optimization function
- Multiple data sets $V_1, \ldots, V_m$ and their core-sets $U_1 \subset V_1, \ldots, U_m \subset V_m$,
  - $f(U_1 \cup \ldots \cup U_m)$ approximates $f(V_1 \cup \ldots \cup V_m)$ by a factor $\alpha$
Core-sets with composability property:

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

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

![Diagram of chunk division](image-url)
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

![Diagram showing Core-sets](image)
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

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)  \[\rightarrow\]  Feature Vectors  \[\rightarrow\]  Points in a high dimensional space
Diversity Maximization: The Model

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

**Goal:** pick $k$ points while maximizing “diversity”.

$k = 3$
**Input:** a set of $n$ vectors $V \subset \mathbb{R}^d$ 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 \( V \subset \mathbb{R}^d \) and a parameter \( k \leq d \),

**Goal:** pick \( k \) points s.t. the minimum pairwise distance of the picked points is maximized.
**Input:** a set of \( n \) vectors \( V \subset \mathbb{R}^d \) 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
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

$k = 3$
Maximizing the minimum pairwise distance

$k = 3$
Maximizing the minimum pairwise distance
Maximizing the minimum pairwise distance

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

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

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

- $\exists q \in S$ such that $\text{dist}(p, q) \leq r$
Maximizing the minimum pairwise distance

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

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

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

- $\text{Opt} \leq 2r$
Maximizing the minimum pairwise distance

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

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

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

$k = 3$ 

$\text{Opt} \leq 3r$
Maximizing the minimum pairwise distance

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

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

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

$k = 3$

$Opt \leq 2r$
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, $V = \bigcup_i V_i$

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

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

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

**Goal:** \[ div_k(S) \geq div_k(V)/c \]

**Goal:** \[ div_k(S) \geq div_k(Opt)/c \]
Let $V_1, \ldots, V_m$ be the set of points, $V = \bigcup_i V_i$

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

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

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

**Goal:** $div_k(S) \geq div_k(Opt)/c$
Let $V_1, \ldots, V_m$ be the set of points, $V = \bigcup_i V_i$

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

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

Let $r$ be the maximum diversity $r = \max_i \text{div}_k(S_i)$

| **Goal:** $\text{div}_k(S) \geq \frac{\text{div}_k(V)}{c}$ |
| **Goal:** $\text{div}_k(S) \geq \frac{\text{div}_k(\text{Opt})}{c}$ |
| **Note:** $\text{div}_k(S) \geq r$ |
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, $S = \bigcup_i S_i$

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

Let $r$ be the maximum diversity $r = \max_i \text{div}_k(S_i)$

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

| Goal: $\text{div}_k(S) \geq \text{div}_k(V)/c$ |
| Goal: $\text{div}_k(S) \geq \text{div}_k(\text{Opt})/c$ |
| Note: $\text{div}_k(S) \geq r$ |
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, $S = \bigcup_i S_i$

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

Let $r$ be the maximum diversity $r = \max_i \text{div}_k(S_i)$

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

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

**Goal:** $\text{div}_k(S) \geq \text{div}_k(V)/c$

**Goal:** $\text{div}_k(S) \geq \text{div}_k(Opt)/c$

**Note:** $\text{div}_k(S) \geq r$
Let $V_1, ..., V_m$ be the set of points, $V = \bigcup_i V_i$

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

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

Let $r$ be the maximum diversity $r = \max_i \text{div}_k(S_i)$

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

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

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

<table>
<thead>
<tr>
<th>Goal: $\text{div}_k(S) \geq \text{div}_k(V)/c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal: $\text{div}_k(S) \geq \text{div}_k(Opt)/c$</td>
</tr>
<tr>
<td>Note: $\text{div}_k(S) \geq r$</td>
</tr>
</tbody>
</table>
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, $S = \bigcup_i S_i$.

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

Let $r$ be the maximum diversity $r = \max_i \text{div}_k(S_i)$.

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

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

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

**Goal:** $\text{div}_k(S) \geq \frac{\text{div}_k(V)}{c}$

**Goal:** $\text{div}_k(S) \geq \frac{\text{div}_k(\text{Opt})}{c}$

**Note:** $\text{div}_k(S) \geq r$
Diversity: Volume
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.
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.
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 $V \subset \mathbb{R}^d$ 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 produces a composable core-set of size $k$ with approximation factor $O(k)^k$ for the volume maximization problem.
Local Search \[ \text{MAX-k-VOL} \] 

\[ \geq \frac{1}{k^k} \cdot \text{MAX-k-VOL} \]
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 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., $S = S \cup \{p\} \setminus \{q\}$

Start with a crude approximation (Greedy algorithm)

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., $S = S \cup \{p\} \setminus \{q\}$

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

$(k - 1)$-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 $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ core-set for volume maximization.
Local Search Lemma
Local Search gives a $2k$-approximate core-set for $k$-directional height.

Height-Volume Lemma
Any $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ core-set for volume maximization.
Local Search Lemma
Local Search gives a $2k$-approximate core-set for $k$-directional height.

Height-Volume Lemma
Any $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ core-set for volume maximization

$\alpha = 2k$

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

Given
• a point set $P$, and
**$k$-Directional Height**

Given

- a point set $P$, and
- a $(k - 1)$-dimensional subspace $G$ (direction),
Given

• a point set $P$, and

• a $(k - 1)$-dimensional subspace $G$ (direction),

The $k$-Directional Height of $P$ in the direction of $G$ is defined as

$$\max_{p \in P} \text{dist}(p, G)$$
A subset of points that preserve the $k$-directional height for all subspaces $G$ of dimension $k - 1$ \textit{at the same time} upto an approximation factor $\alpha$. 
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 $G$

$$\max_{q \in S} \text{dist}(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} \text{dist}(p, G)$$
Local Search Lemma:
For any \((k - 1)\)-dimensional subspace \(G\), the maximum distance of the point set to \(G\) is approximately preserved

\[
\max_{s \in S} \text{dist}(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} \text{dist}(p, G)
\]
Local Search Lemma:
For any \((k - 1)\)-dimensional subspace \(G\), the maximum distance of the point set to \(G\) is approximately preserved

\[
\max_{s \in S} \text{dist}(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} \text{dist}(p, G)
\]
Local Search Lemma:
For any \((k - 1)\)-dimensional subspace \(G\), the maximum distance of the point set to \(G\) is approximately preserved

\[
\max_{s \in S} \text{dist}(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} \text{dist}(p, G)
\]
Local Search Lemma:
For any \((k - 1)\)-dimensional subspace \(G\), the maximum distance of the point set to \(G\) is approximately preserved

\[
\max_{s \in S} \text{dist}(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} \text{dist}(p, G)
\]
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: \(d(p, G) \leq 2kx\)
Local Search Lemma:
For any \((k - 1)\)-dimensional subspace \(G\), the maximum distance of the point set to \(G\) is approximately preserved:

\[
\max_{s \in S} \text{dist}(q, G) \geq \frac{1}{2k} \cdot \max_{p \in V} \text{dist}(p, G)
\]
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**.

$$p_H = \sum_{i=1}^{k} \alpha_i q_i \quad \text{s.t.} \quad \text{all } \left|\alpha_i\right| \leq 1$$
We can write $p_H$ as linear combination of core-set points, with small coefficient:

$$p_H = \sum_{i=1}^{k} \alpha_i q_i \quad \text{s.t.} \quad \text{all } |\alpha_i| \leq 1$$
We can write $p_H$ as linear combination of core-set points, with small coefficient.

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

$$d(p_H, G) \leq kx$$
**Local Search Lemma:**

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

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

\[
d(p, p_H) \leq kx
\]

\[
d(p_H, G) \leq kx
\]
Local Search Lemma:
For any $(k - 1)$-dimensional subspace $G$, the maximum distance of the point set to $G$ is approximately preserved

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

Goal: $d(p, G) \leq 2kx$
Local Search Lemma
Local Search gives a $2k$-approximate core-set for $k$-directional height.

Height-Volume Lemma
Any $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ core-set for volume maximization

$\alpha = 2k$

Theorem:
Local Search produces a $O(k)^k$ core-set for volume maximization.
Height-Volume Lemma
Any \( \alpha \) core-set for \( k \)-directional height gives a \( \alpha^k \) composable core-set for volume maximization

Let \( V = \bigcup_i V_i \) be the union of the point sets
Height-Volume Lemma
Any $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ 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 \( \alpha \) core-set for \( k \)-directional height gives a \( \alpha^k \) 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 \( Opt = \{o_1, \ldots, o_k\} \subset V \) be the optimal subset of points maximizing the volume.
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 $Opt = \{o_1, \ldots, o_k\} \subset V$ be the optimal subset of points maximizing the volume

Let $Sol \leftarrow Opt$

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 $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ 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 $Opt = \{o_1, ..., o_k\} \subseteq V$ 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 $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$
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 $Opt = \{o_1, \ldots, o_k\} \subset V$ 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 $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$
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 $Opt = \{o_1, \ldots, o_k\} \subset V$ be the optimal subset of points maximizing the volume

$Sol \leftarrow Opt$

For $i = 1 \text{ to } k$

- 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 $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ 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 $Opt = \{o_1, \ldots, o_k\} \subset V$ 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 \( \alpha \) core-set for \( k \)-directional height gives a \( \alpha^k \) 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 \( Opt = \{o_1, \ldots, o_k\} \subset V \) be the optimal subset of points maximizing the volume

\[
\text{Sol} \leftarrow Opt \\
\text{For } i = 1 \text{ to } k \\
\quad \text{• Let } q_i \in S \text{ be the point that is farthest away from } H_{Sol \setminus \{o_i\}} \\
\quad \text{• } \text{Sol} \leftarrow \text{Sol} \cup \{q_i\} \setminus \{o_i\}
\]
Let $\mathbf{V} = \bigcup_i V_i$ be the union of the point sets.

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

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

For $i = 1 \text{ to } k$
  - Let $q_i \in \mathbf{S}$ be the point that is farthest away from $H_{\text{Sol}\setminus{o_i}}$.
  - $\text{Sol} \leftarrow \text{Sol} \cup \{q_i\} \setminus \{o_i\}$.
Height-Volume Lemma

Any $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ 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 $Opt = \{o_1, ..., o_k\} \subseteq V$ 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 $H_{Sol\setminus\{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$
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 $Opt = \{o_1, \ldots, o_k\} \subseteq V$ 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 $H_{Sol \setminus \{o_i\}}$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$
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 $\text{Opt} = \{o_1, \ldots, o_k\} \subset V$ be the optimal subset of points maximizing the volume

$\text{Sol} \leftarrow \text{Opt}$

For $i = 1$ to $k$

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

• $\text{Sol} \leftarrow \text{Sol} \cup \{q_i\} \setminus \{o_i\}$
Let $\mathbf{V} = \bigcup_i V_i$ be the union of the point sets

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

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

$\textit{Sol} \leftarrow \textit{Opt}$

For $i = 1$ to $k$

- Let $q_i \in \mathbf{S}$ be the point that is farthest away from $H_{\textit{Sol} \setminus \{o_i\}}$
- $\textit{Sol} \leftarrow \textit{Sol} \cup \{q_i\} \setminus \{o_i\}$
Let $\mathbf{V} = \bigcup_i V_i$ be the union of the point sets

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

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

$\text{Sol} \leftarrow \text{Opt}$

For $i = 1$ to $k$

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

**Height-Volume Lemma**

Any $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ composable core-set for volume maximization
Height-Volume Lemma

Any $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ 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 $Opt = \{o_1, ..., o_k\} \subset V$ 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 $H S S S S S S \setminus S_i$
- $Sol \leftarrow Sol \cup \{q_i\} \setminus \{o_i\}$

Since we have a $\alpha$ core-set for $k$-directional height

- Lose a factor of at most $\alpha$ at each iteration
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 $Opt = \{o_1, ..., o_k\} \subset V$ 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 $\alpha$ at each iteration

➢ Total approximation factor $\alpha^k$
Local Search Lemma
Local Search gives a $2k$-approximate core-set for $k$-directional height.

Height-Volume Lemma
Any $\alpha$ core-set for $k$-directional height gives a $\alpha^k$ core-set for volume maximization.

$$\alpha = 2k$$

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