

Composable Core-sets for Diversity and Coverage Maximization

Piotr Indyk (MIT)

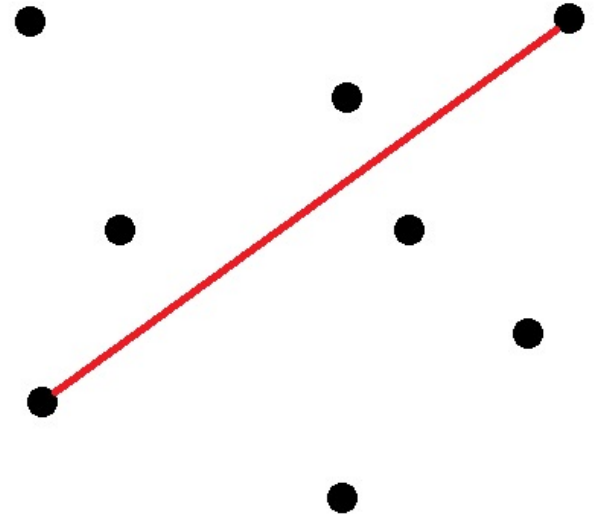
Sepideh Mahabadi (MIT)

Mohammad Mahdian (Google)

Vahab S. Mirrokni (Google)

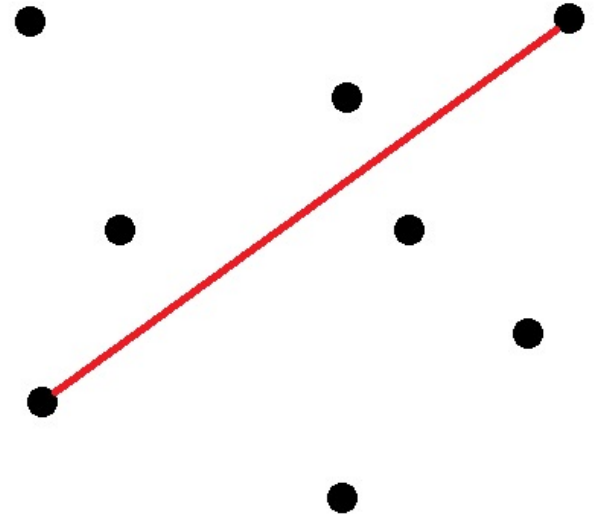
Core-Set Definition

- **Setup**
 - Set of n points \mathbf{P} in d -dimensional space
 - Optimize a function f



Core-Set Definition

- **Setup**
 - Set of n points P in d -dimensional space
 - Optimize a function f
- **c -Core-set:** Small subset of points $S \subset P$ which suffices to c -approximate the optimal solution
- Maximization: $\frac{f_{opt}(P)}{c} \leq f_{opt}(S) \leq f_{opt}(P)$



Core-Set Definition

- **Setup**

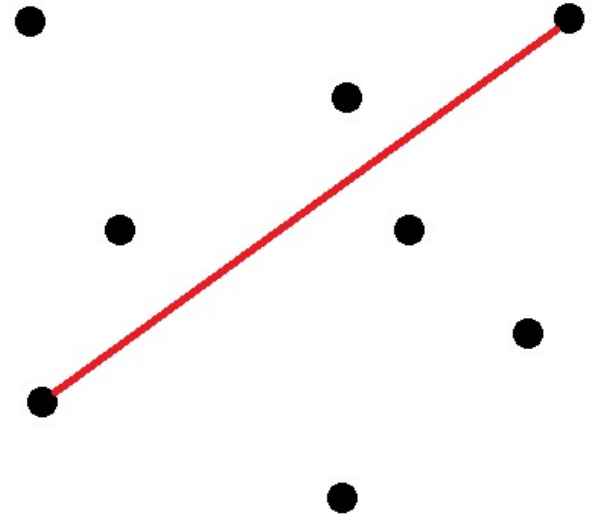
- Set of n points P in d -dimensional space
- Optimize a function f

- **c -Core-set:** Small subset of points $S \subset P$ which suffices to c -approximate the optimal solution

- Maximization: $\frac{f_{opt}(P)}{c} \leq f_{opt}(S) \leq f_{opt}(P)$

- **Example**

- Optimization Function: Distance of the two farthest points



Core-Set Definition

- **Setup**

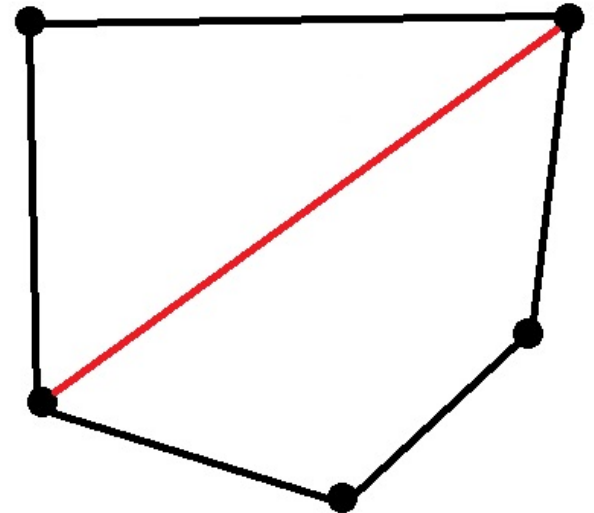
- Set of n points P in d -dimensional space
- Optimize a function f

- **c -Core-set:** Small subset of points $S \subset P$ which suffices to c -approximate the optimal solution

- Maximization: $\frac{f_{opt}(P)}{c} \leq f_{opt}(S) \leq f_{opt}(P)$

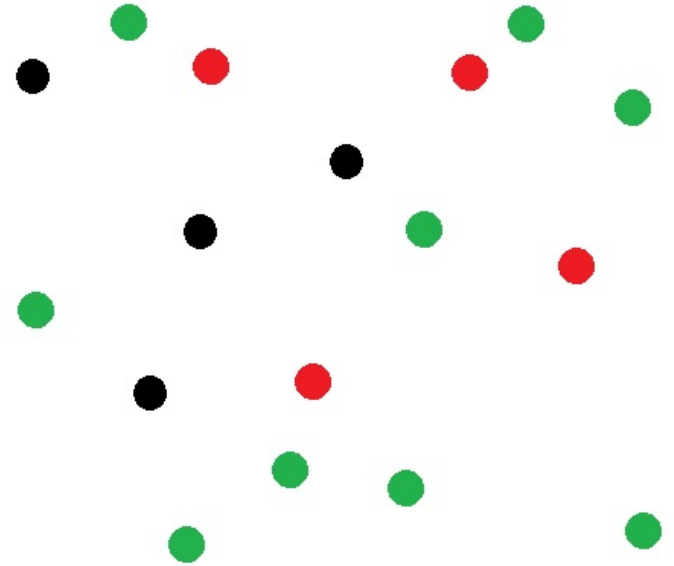
- **Example**

- Optimization Function: Distance of the two farthest points
- 1-Core-set: Points on the convex hull.



Composable Core-sets

- **Setup**
 - P_1, P_2, \dots, P_m are set of points in d -dimensional space
 - Optimize a function f over their union P .



Composable Core-sets

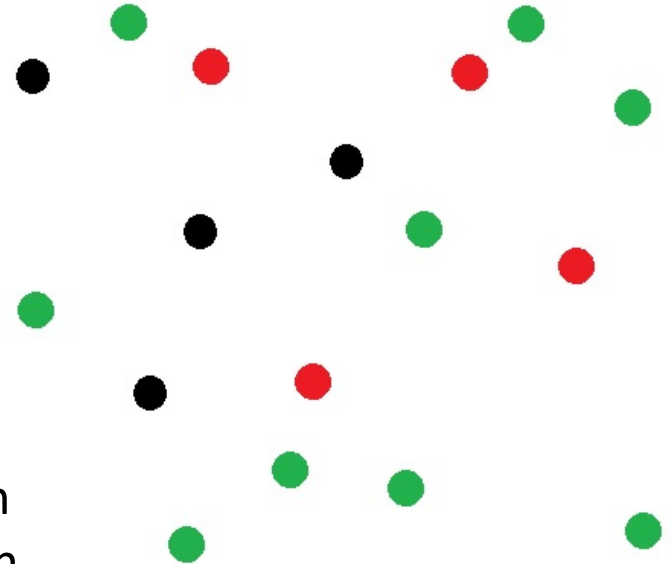
- **Setup**

- P_1, P_2, \dots, P_m are set of points in d -dimensional space
- Optimize a function f over their union P .

- **c -Composable Core-sets:** Subsets of points $S_1 \subset P_1, S_2 \subset P_2, \dots, S_m \subset P_m$ points such that the solution of the union of the core-sets approximates the solution of the point sets.

- Maximization :

$$\frac{1}{c} f_{opt}(P_1 \cup \dots \cup P_m) \leq f_{opt}(S_1 \cup \dots \cup S_m) \leq f_{opt}(P_1 \cup \dots \cup P_m)$$



Composable Core-sets

- **Setup**

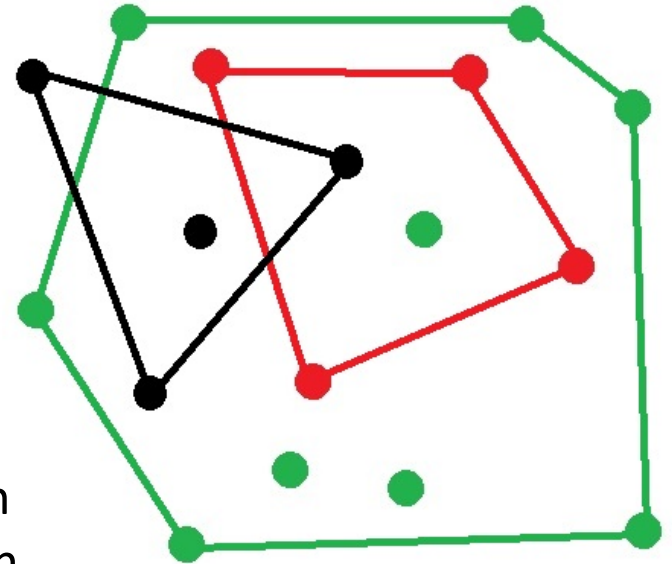
- P_1, P_2, \dots, P_m are set of points in d -dimensional space
- Optimize a function f over their union P .

- **c -Composable Core-sets:** Subsets of points $S_1 \subset P_1, S_2 \subset P_2, \dots, S_m \subset P_m$ points such that the solution of the union of the core-sets approximates the solution of the point sets.

- Maximization :

$$\frac{1}{c} f_{opt}(P_1 \cup \dots \cup P_m) \leq f_{opt}(S_1 \cup \dots \cup S_m) \leq f_{opt}(P_1 \cup \dots \cup P_m)$$

- **Example:** two farthest points



Composable Core-sets

- **Setup**

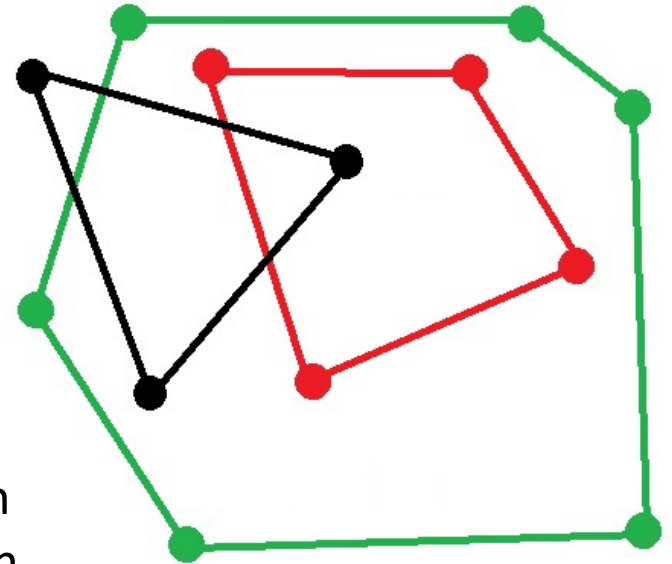
- P_1, P_2, \dots, P_m are set of points in d -dimensional space
- Optimize a function f over their union P .

- **c -Composable Core-sets:** Subsets of points $S_1 \subset P_1, S_2 \subset P_2, \dots, S_m \subset P_m$ points such that the solution of the union of the core-sets approximates the solution of the point sets.

- Maximization :

$$\frac{1}{c} f_{opt}(P_1 \cup \dots \cup P_m) \leq f_{opt}(S_1 \cup \dots \cup S_m) \leq f_{opt}(P_1 \cup \dots \cup P_m)$$

- **Example:** two farthest points



Composable Core-sets

- **Setup**

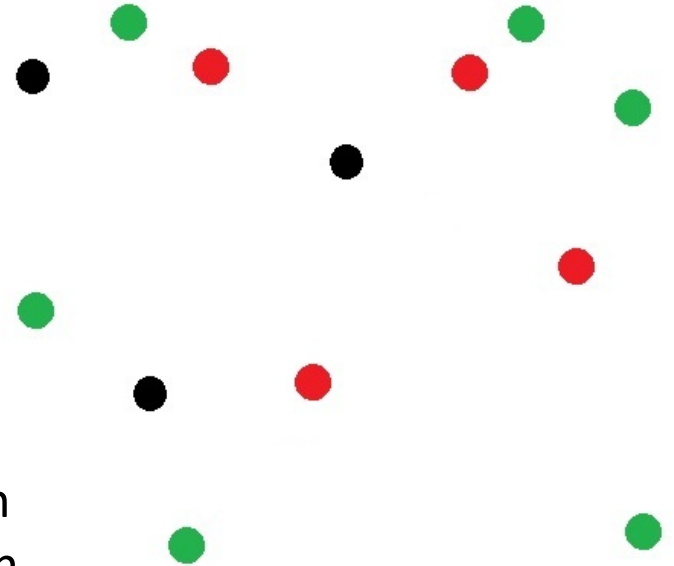
- P_1, P_2, \dots, P_m are set of points in d -dimensional space
- Optimize a function f over their union P .

- **c -Composable Core-sets:** Subsets of points $S_1 \subset P_1, S_2 \subset P_2, \dots, S_m \subset P_m$ points such that the solution of the union of the core-sets approximates the solution of the point sets.

- Maximization :

$$\frac{1}{c} f_{opt}(P_1 \cup \dots \cup P_m) \leq f_{opt}(S_1 \cup \dots \cup S_m) \leq f_{opt}(P_1 \cup \dots \cup P_m)$$

- **Example:** two farthest points



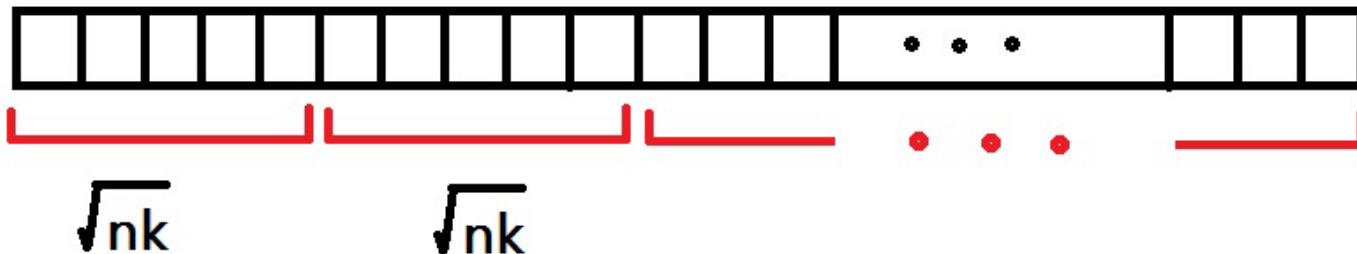
Applications – Streaming Computation

- **Streaming Computation:**
 - Processing sequence of n data elements “on the fly”
 - limited Storage



Applications – Streaming Computation

- **Streaming Computation:**
 - Processing sequence of n data elements “on the fly”
 - limited Storage
- **c -Composable Core-set of size k**
 - Chunks of size \sqrt{nk} , thus number of chunks = $\sqrt{n/k}$



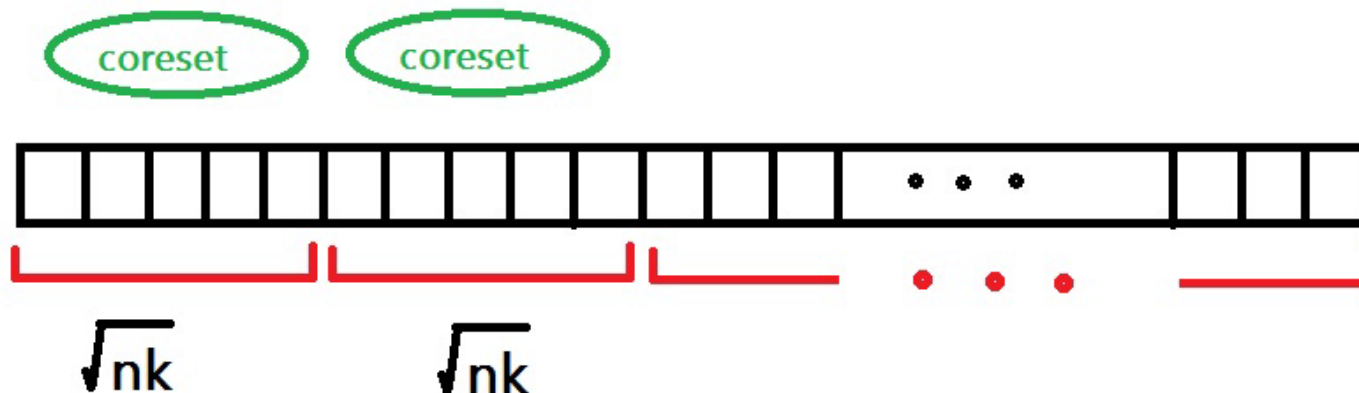
Applications – Streaming Computation

- **Streaming Computation:**

- Processing sequence of n data elements “on the fly”
- limited Storage

- **c -Composable Core-set of size k**

- Chunks of size \sqrt{nk} , thus number of chunks = $\sqrt{n/k}$
- Core-set for each chunk
- Total Space: $k\sqrt{n/k} + \sqrt{nk} = O(\sqrt{nk})$
- Approximation Factor: c

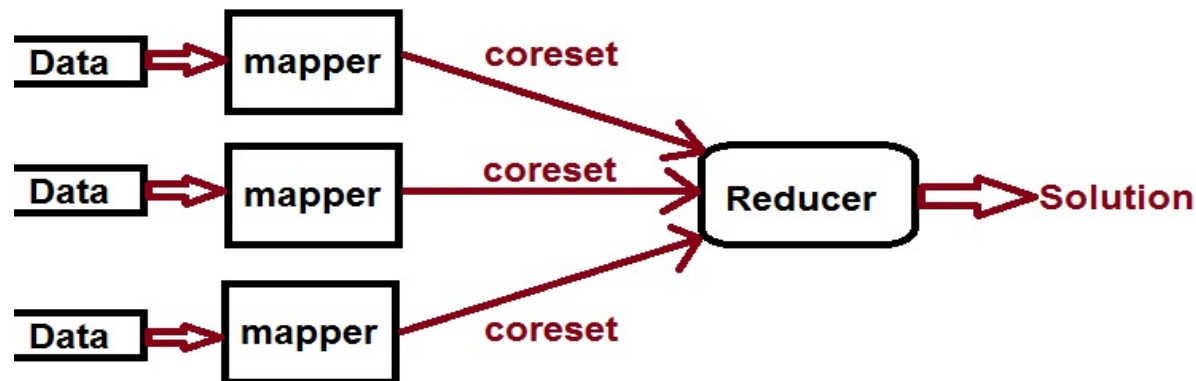


Applications – Distributed Systems

- Streaming Computation
- **Distributed System:**
 - Each machine holds a block of data.
 - A composable core-set is computed and sent to the server

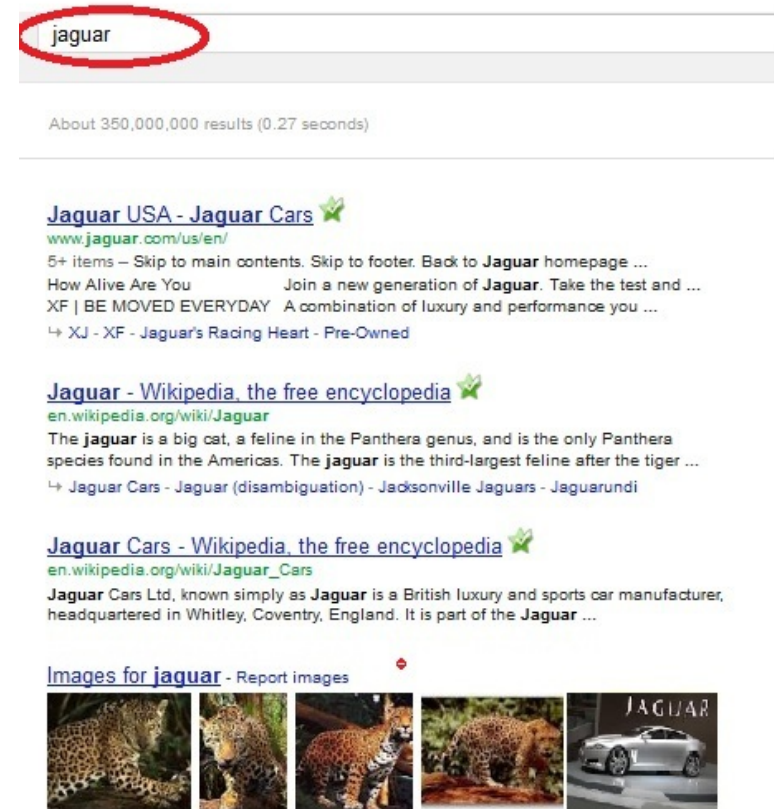
Applications – Distributed Systems

- Streaming Computation
- **Distributed System:**
 - Each machine holds a block of data.
 - A composable core-set is computed and sent to the server
- **Map-Reduce Model:**
 - One round of Map-Reduce
 - $\sqrt{n/k}$ mappers each getting \sqrt{nk} points
 - Mapper computes a composable core-set of size k
 - Will be passed to a single reducer



Applications – Similarity Search

- Streaming Computation
- Distributed System
- **Similarity Search:** Small output size

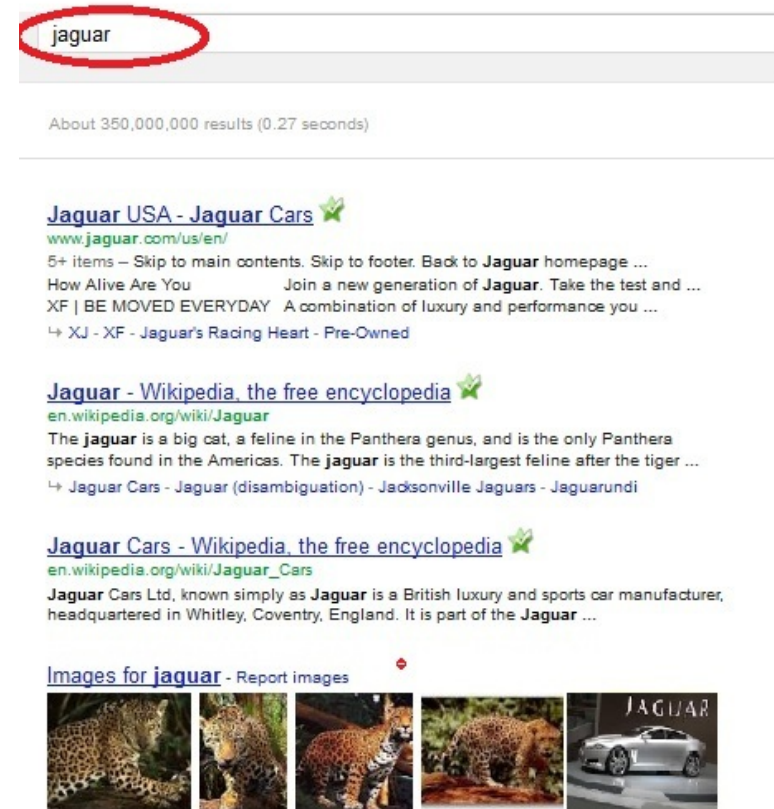


The screenshot shows a search engine interface with the query "jaguar" entered in the search bar. Below the search bar, it indicates "About 350,000,000 results (0.27 seconds)". The results are listed as follows:

- Jaguar USA - Jaguar Cars** (marked with a green star)
www.jaguar.com/us/en/
5+ items – Skip to main contents. Skip to footer. Back to **Jaguar** homepage ...
How Alive Are You Join a new generation of **Jaguar**. Take the test and ...
XF | BE MOVED EVERYDAY A combination of luxury and performance you ...
↳ XJ - XF - Jaguar's Racing Heart - Pre-Owned
- Jaguar - Wikipedia, the free encyclopedia** (marked with a green star)
en.wikipedia.org/wiki/Jaguar
The **jaguar** is a big cat, a feline in the Panthera genus, and is the only Panthera species found in the Americas. The **jaguar** is the third-largest feline after the tiger ...
↳ Jaguar Cars - Jaguar (disambiguation) - Jacksonville Jaguars - Jaguarundi
- Jaguar Cars - Wikipedia, the free encyclopedia** (marked with a green star)
en.wikipedia.org/wiki/Jaguar_Cars
Jaguar Cars Ltd, known simply as **Jaguar** is a British luxury and sports car manufacturer, headquartered in Whitley, Coventry, England. It is part of the **Jaguar** ...
- Images for jaguar** - Report images
A row of five image thumbnails: a leopard, a jaguar, a tiger, another jaguar, and a silver Jaguar car.

Applications – Similarity Search

- Streaming Computation
- Distributed System
- **Similarity Search**: Small output size
- Good to have result from each cluster: **relevant** and **diverse**



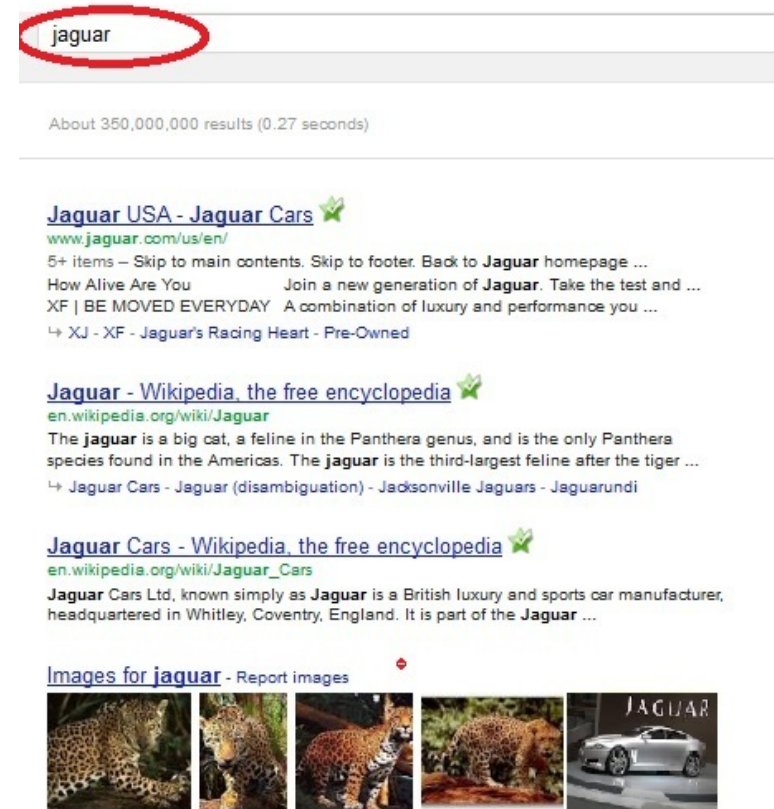
The screenshot shows a search engine interface with the query "jaguar" entered in the search bar. Below the search bar, it indicates "About 350,000,000 results (0.27 seconds)". The results are listed as follows:

- Jaguar USA - Jaguar Cars** with a green star icon and URL www.jaguar.com/us/en/. Below the title, it shows "5+ items" and a list of items: "How Alive Are You", "Join a new generation of Jaguar. Take the test and ...", "XF | BE MOVED EVERYDAY A combination of luxury and performance you ...", and "XJ - XF - Jaguar's Racing Heart - Pre-Owned".
- Jaguar - Wikipedia, the free encyclopedia** with a green star icon and URL en.wikipedia.org/wiki/Jaguar. The snippet reads: "The **jaguar** is a big cat, a feline in the Panthera genus, and is the only Panthera species found in the Americas. The **jaguar** is the third-largest feline after the tiger ...". Below the snippet are links: "Jaguar Cars - Jaguar (disambiguation) - Jacksonville Jaguars - Jaguarundi".
- Jaguar Cars - Wikipedia, the free encyclopedia** with a green star icon and URL en.wikipedia.org/wiki/Jaguar_Cars. The snippet reads: "Jaguar Cars Ltd, known simply as **Jaguar** is a British luxury and sports car manufacturer, headquartered in Whitley, Coventry, England. It is part of the **Jaguar** ...".

At the bottom, there is a section titled "Images for jaguar - Report images" with a red diamond icon. It contains five image thumbnails: four showing jaguars in their natural habitat and one showing a silver Jaguar car.

Applications – Similarity Search

- Streaming Computation
- Distributed System
- **Similarity Search:** Small output size
- Good to have result from each cluster: **relevant** and **diverse**
- Diverse Near Neighbor Problem
[Abbar, Amer-Yahia, Indyk, Mahabadi WWW'13] [Abbar, Amer-Yahia, Indyk, Mahabadi, Varadarajan, SoCG'13]



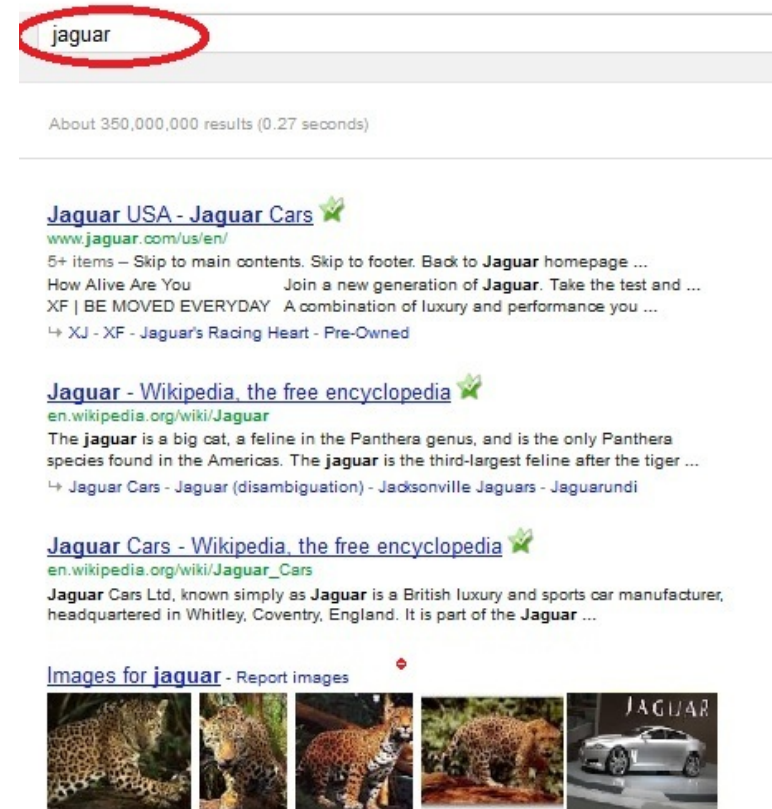
The screenshot shows a search engine interface with the query "jaguar" entered in the search bar. Below the search bar, it indicates "About 350,000,000 results (0.27 seconds)". The results are categorized into three sections:

- Jaguar USA - Jaguar Cars**: Includes the URL www.jaguar.com/us/en/, a snippet about joining a new generation of Jaguar, and a link to "XJ - XF - Jaguar's Racing Heart - Pre-Owned".
- Jaguar - Wikipedia, the free encyclopedia**: Includes the URL en.wikipedia.org/wiki/Jaguar and a snippet describing the jaguar as a big cat in the Panthera genus.
- Jaguar Cars - Wikipedia, the free encyclopedia**: Includes the URL en.wikipedia.org/wiki/Jaguar_Cars and a snippet describing Jaguar Cars Ltd as a British luxury and sports car manufacturer.

At the bottom, there is a section for "Images for jaguar" which displays a row of five images: four images of jaguars in their natural habitat and one image of a silver Jaguar car.

Applications – Similarity Search

- Streaming Computation
- Distributed System
- **Similarity Search**: Small output size
- Good to have result from each cluster: **relevant** and **diverse**
- Diverse Near Neighbor Problem
[Abbar, Amer-Yahia, Indyk, Mahabadi WWW'13] [Abbar, Amer-Yahia, Indyk, Mahabadi, Varadarajan, SoCG'13]
- uses Locality Sensitive Hashing (LSH) and Composable Core-sets techniques.



The screenshot shows a search engine interface with the query "jaguar" entered in the search bar. Below the search bar, it indicates "About 350,000,000 results (0.27 seconds)". The results are categorized into three sections: "Jaguar USA - Jaguar Cars" with a link to www.jaguar.com/us/en/ and a snippet about joining a new generation of Jaguar; "Jaguar - Wikipedia, the free encyclopedia" with a link to en.wikipedia.org/wiki/Jaguar and a snippet describing the jaguar as a big cat; and "Jaguar Cars - Wikipedia, the free encyclopedia" with a link to en.wikipedia.org/wiki/Jaguar_Cars and a snippet about Jaguar Cars Ltd. At the bottom, there is a section for "Images for jaguar" showing five small images: four of jaguars in various poses and one of a silver Jaguar car.

Diversity Maximization Problem

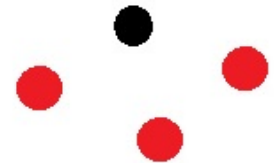
- A set of n points P in metric space $(\Delta, dist)$
- Optimization Problem:
 - Find a subset of k points S which maximizes Diversity



$k=4$
 $n=6$

Diversity Maximization Problem

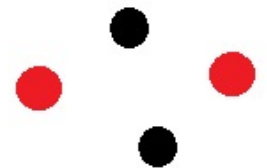
- A set of n points P in metric space $(\Delta, dist)$
- Optimization Problem:
 - Find a subset of k points S which maximizes Diversity
- Diversity:
 - Minimum pairwise distance (Remote Edge)



$k=4$
 $n=6$

Diversity Maximization Problem

- A set of n points P in metric space $(\Delta, dist)$
- Optimization Problem:
 - Find a subset of k points S which maximizes Diversity
- Diversity:
 - Minimum pairwise distance (Remote Edge)
 - Sum of Pairwise distances (Remote Clique)



k=4
n = 6

Diversity Maximization Problem

- A set of n points P in metric space $(\Delta, dist)$
- Optimization Problem:
 - Find a subset of k points S which maximizes Diversity
- Diversity:
 - Minimum pairwise distance (Remote Edge)
 - Sum of Pairwise distances (Remote Clique)
- Long list of variants [Chandra and Halldorsson '01]



$k=4$
 $n=6$

Diversity Functions

Diversity function over a set S of k point	Description
Remote-edge	Minimum Pairwise Distance: $\min_{\{p,q \in S\}} dist(p, q)$
Remote-clique	Sum of Pairwise Distances : $\sum_{\{p,q \in S\}} dist(p, q)$
Remote-tree	Weight of Minimum Spanning Tree (MST) of the set S
Remote-cycle	Weight of minimum Traveling Salesman Tour (TSP) of the set S
Remote-star	Weight of minimum star: $\min_{\{p \in S\}} \sum_{\{q \in S\}} dist(p, q)$
Remote-Pseudoforest	Sum of the distance of each point to its nearest neighbor $\sum_{\{p \in S\}} \min_{\{q \in S\}} dist(p, q)$
Remote-Matching	Weight of minimum perfect Matching of the set S
Max-Coverage	How well the points cover each coordinate $\sum_{i=1}^d \max_{p \in S} p_i$

Our Results

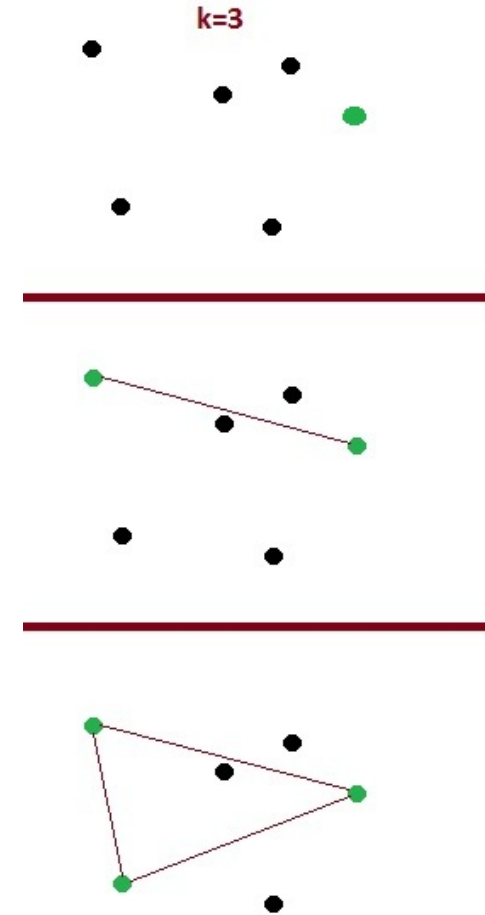
Diversity function		Offline ApproxFactor	Composable Coreset Approx factor [Our Results]
Remote-edge	Minimum Pairwise Distance	$O(1)$ [Tmair 91][White 91] [Ravi et al 94]	$O(1)$
Remote-clique	Sum of Pairwise Distances	$O(1)$ [Hassin et al 97]	$O(1)$
Remote-tree	Weight of MST	$O(1)$ [Halldorsson et al 99]	$O(1)$
Remote-cycle	Weight of minimum TSP	$O(1)$ [Halldorsson et al 99]	$O(1)$
Remote-star	Weight of minimum star	$O(1)$ [Chandra&Halldorsson 01]	$O(1)$
Remote-Pseudoforest	Sum of the distance of each point to its nearest neighbor	$O(\log k)$ [Chandra&Halldorsson 01]	$O(\log k)$
Remote-Matching	Weight of minimum perfect Matching	$O(\log k)$ [Chandra&Halldorsson 01]	$O(\log k)$
Max-Coverage	How well the points cover each coordinate $\sum_{i=1}^d \max_{p \in S} p_i$	$O(1)$ [Feige 98]	No Composable Coreset of Poly size in k with app. factor $\frac{\sqrt{k}}{\log k}$

Review of Offline Algorithms

- We have a set of n point P
- Goal: find a subset S of size k which maximizes the diversity

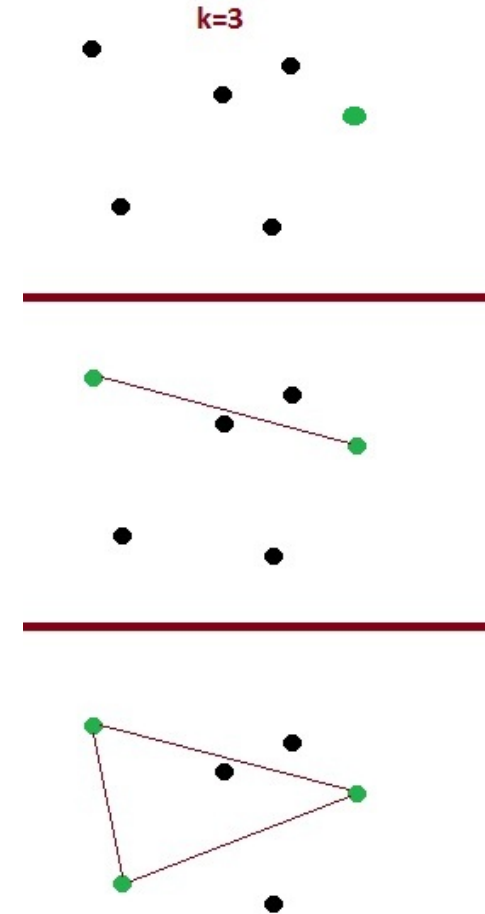
The Greedy Algorithm

- Used for minimum-pairwise distance



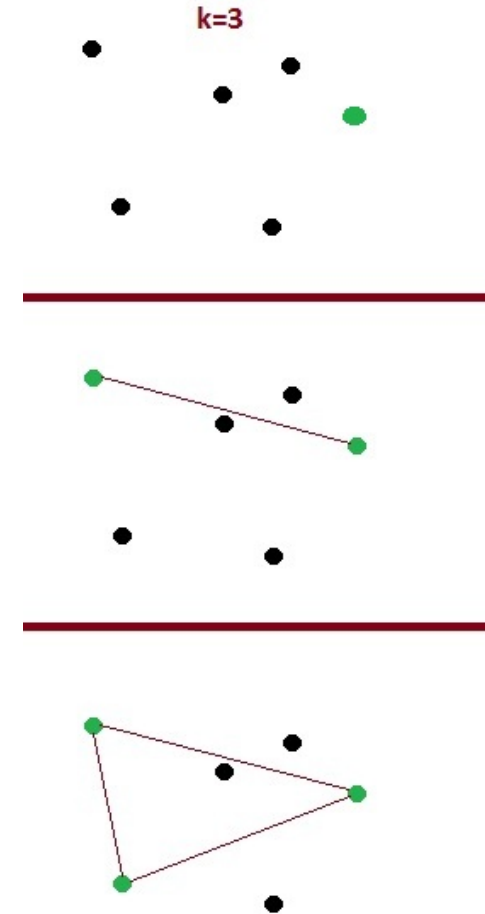
The Greedy Algorithm

- Used for minimum-pairwise distance
- Greedy Algorithm [Ravi, Rosenkrantz, Tayi] [Gonzales]
 - Choose an arbitrary point
 - Repeat $k-1$ times
 - Add the point whose minimum distance to the currently chosen points is maximized



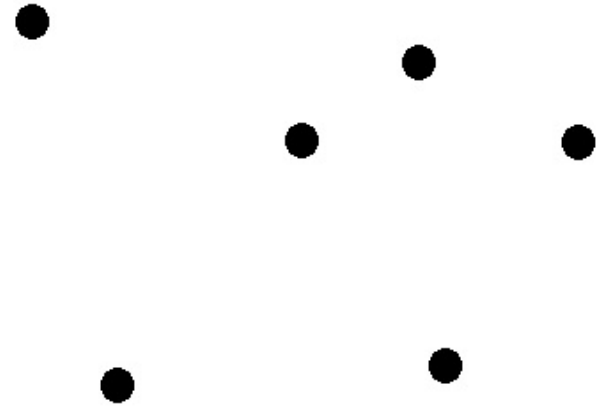
The Greedy Algorithm

- Used for minimum-pairwise distance
- Greedy Algorithm [Ravi, Rosenkrantz, Tayi] [Gonzales]
 - Choose an arbitrary point
 - Repeat $k-1$ times
 - Add the point whose minimum distance to the currently chosen points is maximized
- Remote-edge: computes a 2-approximate set



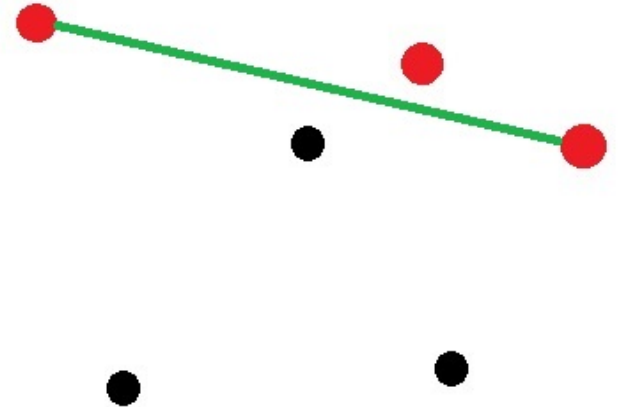
Local Search Algorithm

- Used for sum of pairwise distances



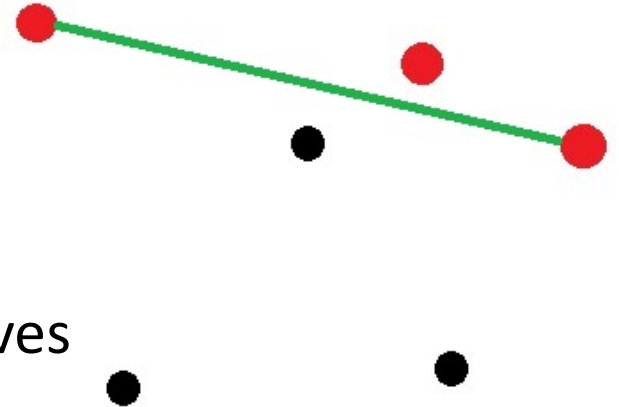
Local Search Algorithm

- Used for sum of pairwise distances
- Algorithm [Abbasi, Mirrokni, Thakur]
 - Initialize S with an arbitrary set of k points which contains the two farthest points



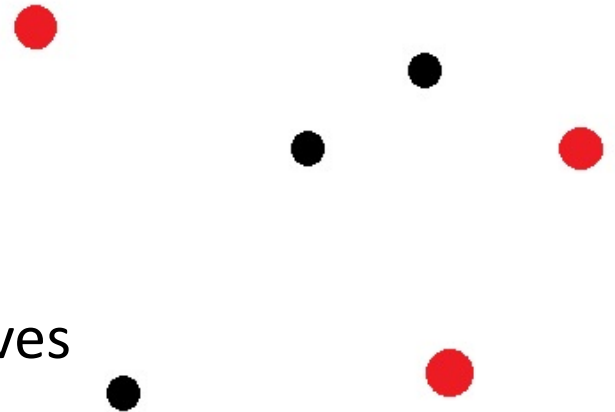
Local Search Algorithm

- Used for sum of pairwise distances
- Algorithm [Abbasi, Mirrokni, Thakur]
 - Initialize S with an arbitrary set of k points which contains the two farthest points
 - While there exists a swap that improves diversity by a factor of $\left(1 + \frac{\epsilon}{n}\right)$



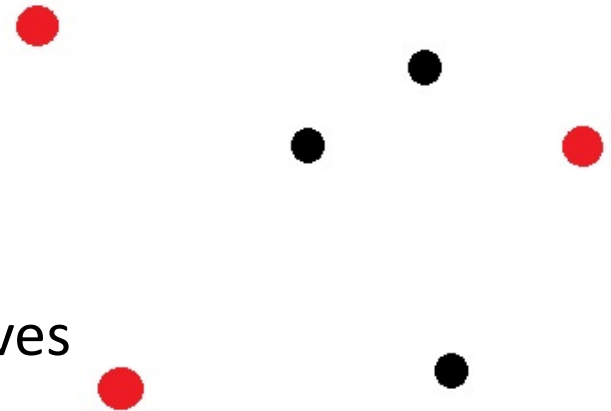
Local Search Algorithm

- Used for sum of pairwise distances
- Algorithm [Abbasi, Mirrokni, Thakur]
 - Initialize S with an arbitrary set of k points which contains the two farthest points
 - While there exists a swap that improves diversity by a factor of $\left(1 + \frac{\epsilon}{n}\right)$
 - » Perform the swap



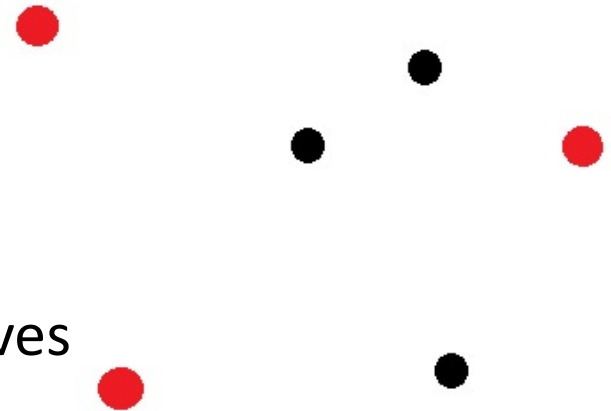
Local Search Algorithm

- Used for sum of pairwise distances
- Algorithm [Abbasi, Mirrokni, Thakur]
 - Initialize S with an arbitrary set of k points which contains the two farthest points
 - While there exists a swap that improves diversity by a factor of $\left(1 + \frac{\epsilon}{n}\right)$
 - » Perform the swap



Local Search Algorithm

- Used for sum of pairwise distances
- Algorithm [Abbasi, Mirrokni, Thakur]
 - Initialize S with an arbitrary set of k points which contains the two farthest points
 - While there exists a swap that improves diversity by a factor of $\left(1 + \frac{\epsilon}{n}\right)$
 - » Perform the swap
- For Remote-Clique
 - Number of rounds: $\log_{\left\{1 + \frac{\epsilon}{n}\right\}} k^2 = O\left(\frac{n}{\epsilon} \log k\right)$
 - Approximation factor is constant.

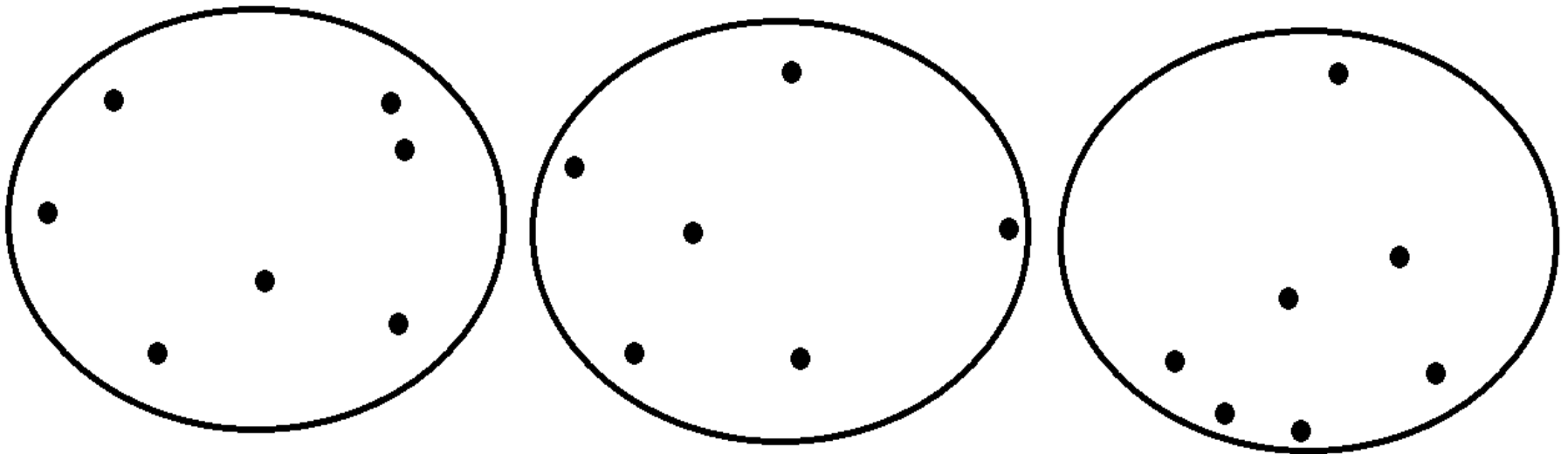


Composable Core-sets

- Greedy Algorithm Computes a 3-composable core-set for minimum pairwise distance
- Local Search Algorithm Computes a constant factor composable core-set for sum of pairwise distances.

Proof Idea

Let P_1, \dots, P_m be the set of points, $P = \cup P_i$

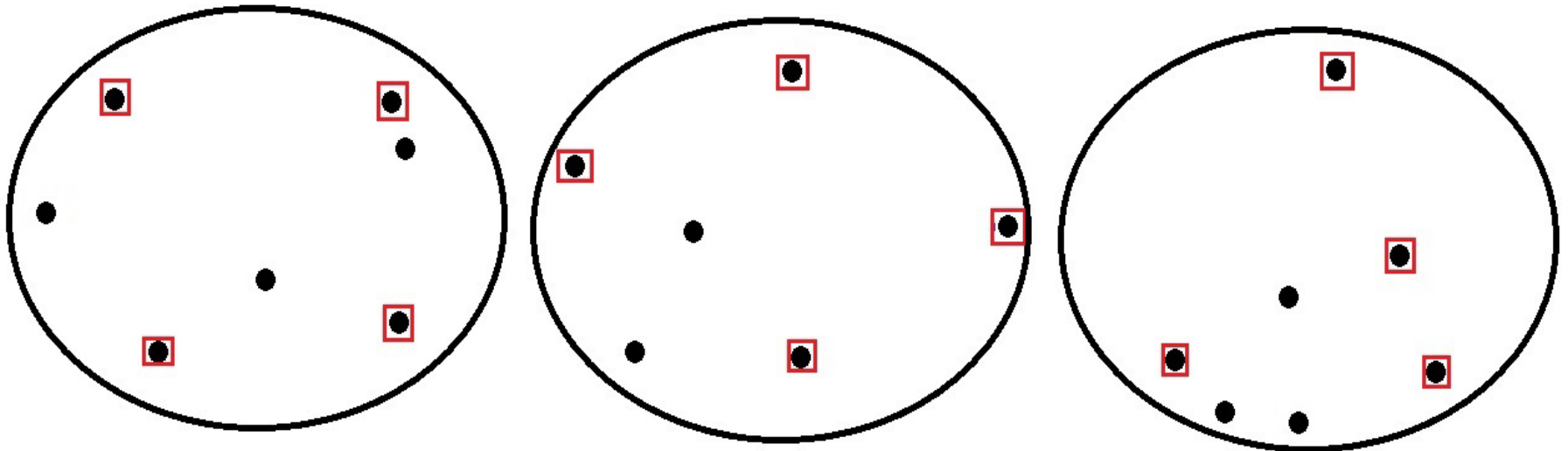


Proof Idea

Let P_1, \dots, P_m be the set of points, $P = \cup P_i$

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

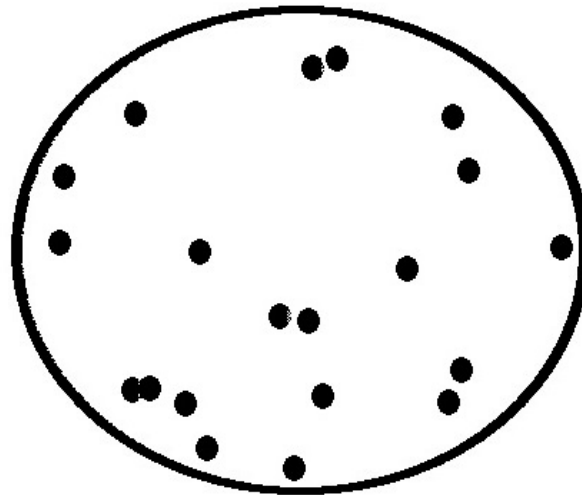
Goal: $div_k(S) \geq div_k(P) / c$



Proof Idea

Let P_1, \dots, P_m be the set of points, $P = \cup P_i$
 S_1, \dots, S_m be their core-sets, $S = \cup S_i$

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



Proof Idea

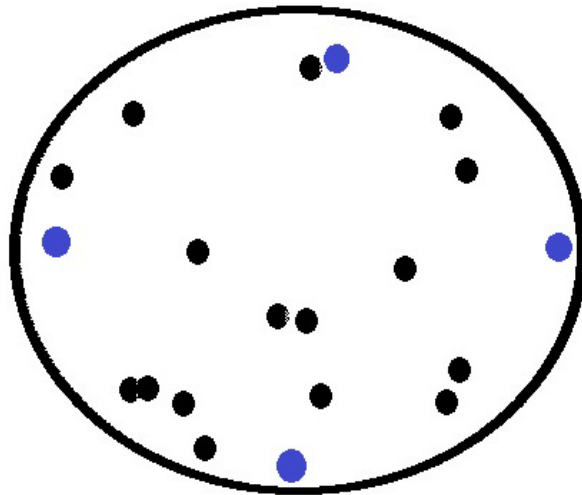
Let P_1, \dots, P_m be the set of points, $P = \cup P_i$

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

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

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

Goal: $div_k(S) \geq div(OPT) / c$



Proof Idea

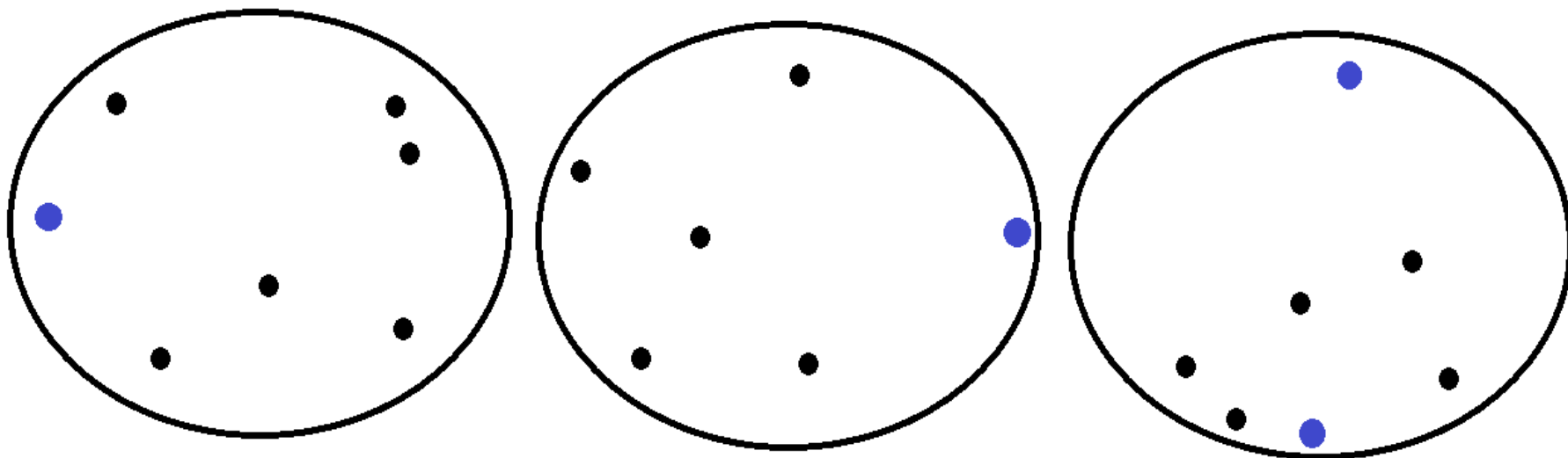
Let P_1, \dots, P_m be the set of points, $P = \cup P_i$

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

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

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

Goal: $div_k(S) \geq div(OPT) / c$



Proof Idea

Let P_1, \dots, P_m be the set of points, $P = \cup P_i$

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

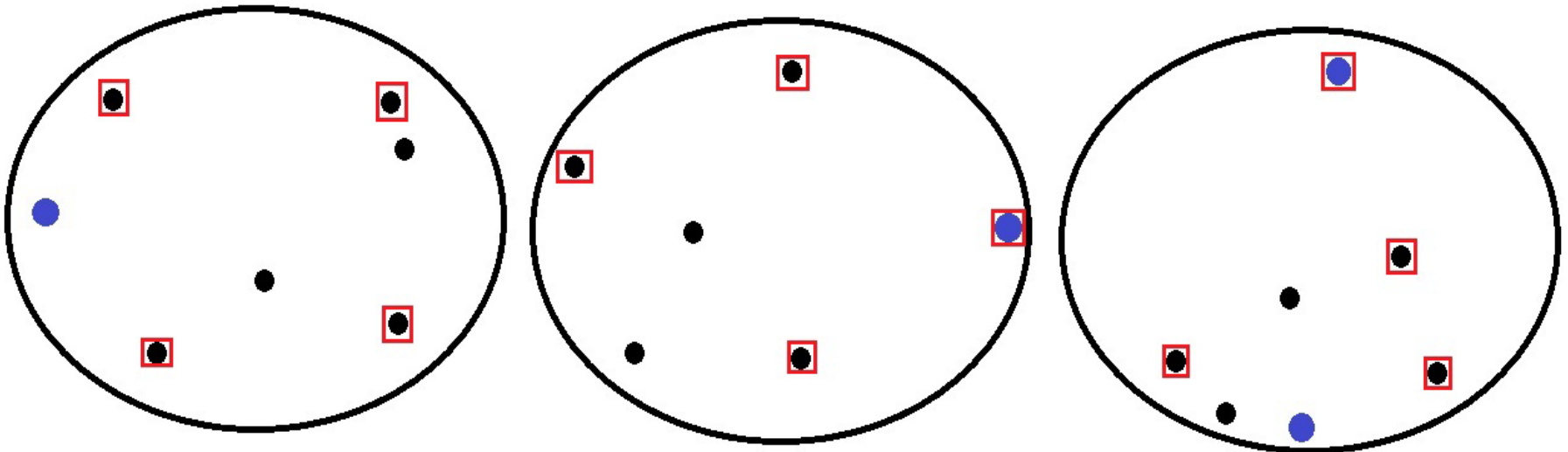
Let $OPT = \{o_1, \dots, o_k\}$ be the optimal solution

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

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

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

Note: $\text{div}_k(S) \geq r$



Proof Idea

Let P_1, \dots, P_m be the set of points, $P = \cup P_i$

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

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

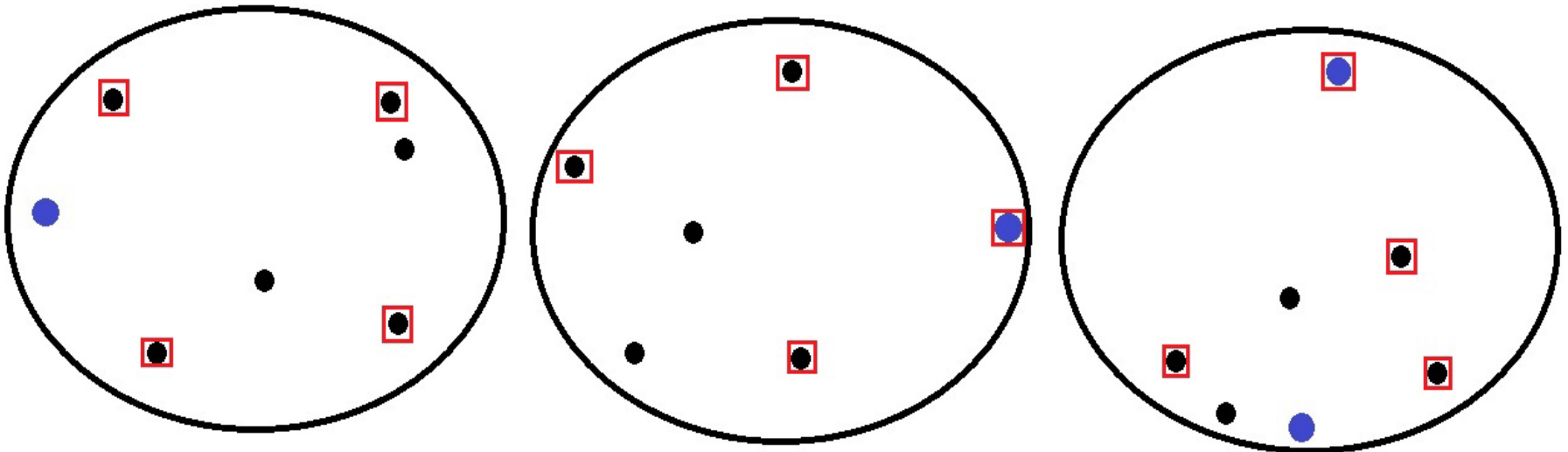
Let r be their maximum diversity, $r = \max_i \text{div}(S_i)$,

Case 1: one of S_i has diversity as good as the optimum: $r \geq \mathbf{O}(\text{div}(OPT))$

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

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

Note: $\text{div}_k(S) \geq r$



Proof Idea

Let P_1, \dots, P_m be the set of points, $P = \cup P_i$

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

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

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

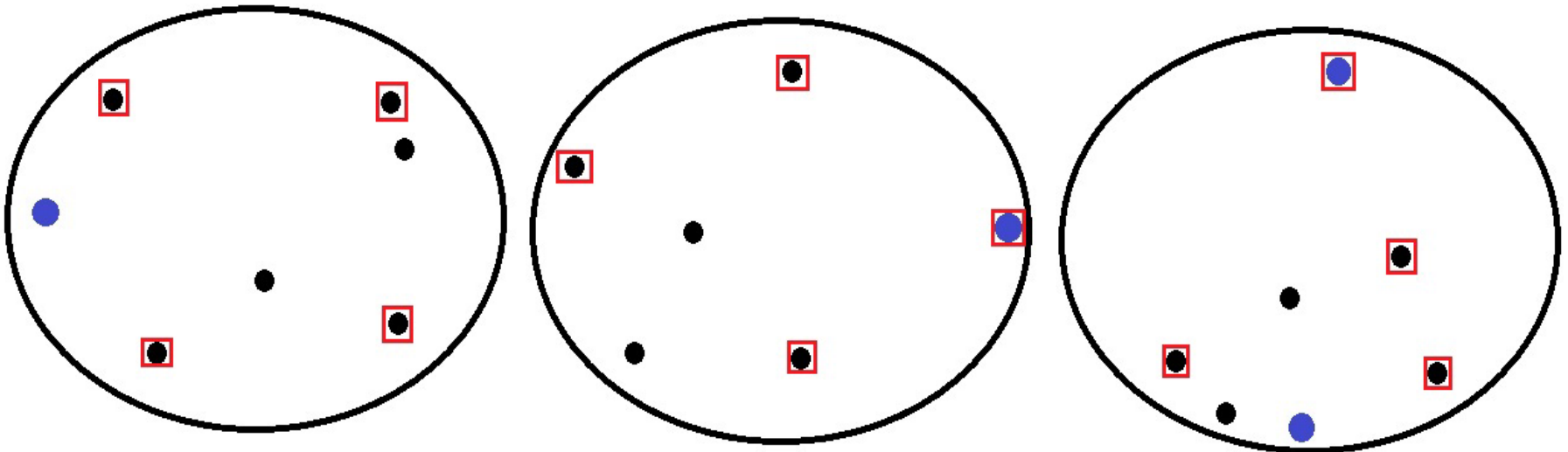
Case 1: one of S_i has diversity as good as the optimum: $r \geq \mathbf{O}(\text{div}(OPT))$

Case 2: $r \leq \mathbf{O}(\text{div}(OPT))$

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

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

Note: $\text{div}_k(S) \geq r$



Proof Idea

Let P_1, \dots, P_m be the set of points, $P = \cup P_i$

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

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

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

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

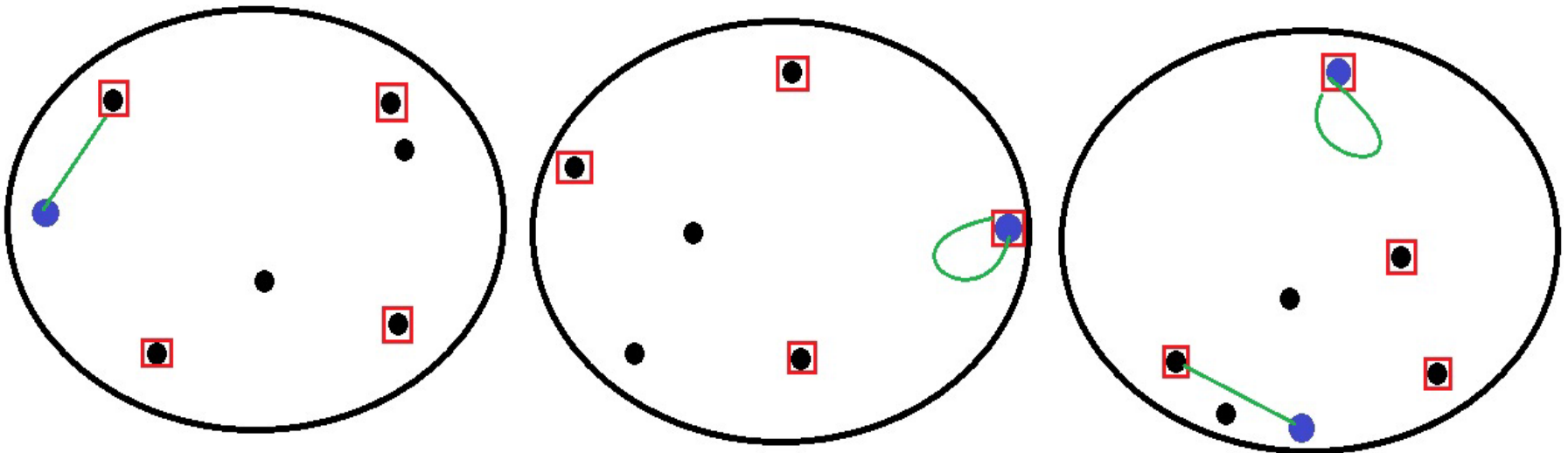
Goal: $\text{div}_k(S) \geq \text{div}(OPT) / c$

Note: $\text{div}_k(S) \geq r$

Case 1: one of S_i has diversity as good as the optimum: $r \geq \mathbf{O}(\text{div}(OPT))$

Case 2: $r \leq \mathbf{O}(\text{div}(OPT))$

- find a **one-to-one** mapping μ from $OPT = \{o_1, \dots, o_k\}$ to $S = S_1 \cup \dots \cup S_m$ s.t.
 $\text{dist}(o_i, \mu(o_i)) \leq \mathbf{O}(r)$



Proof Idea

Let P_1, \dots, P_m be the set of points, $P = \cup P_i$

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

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

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

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

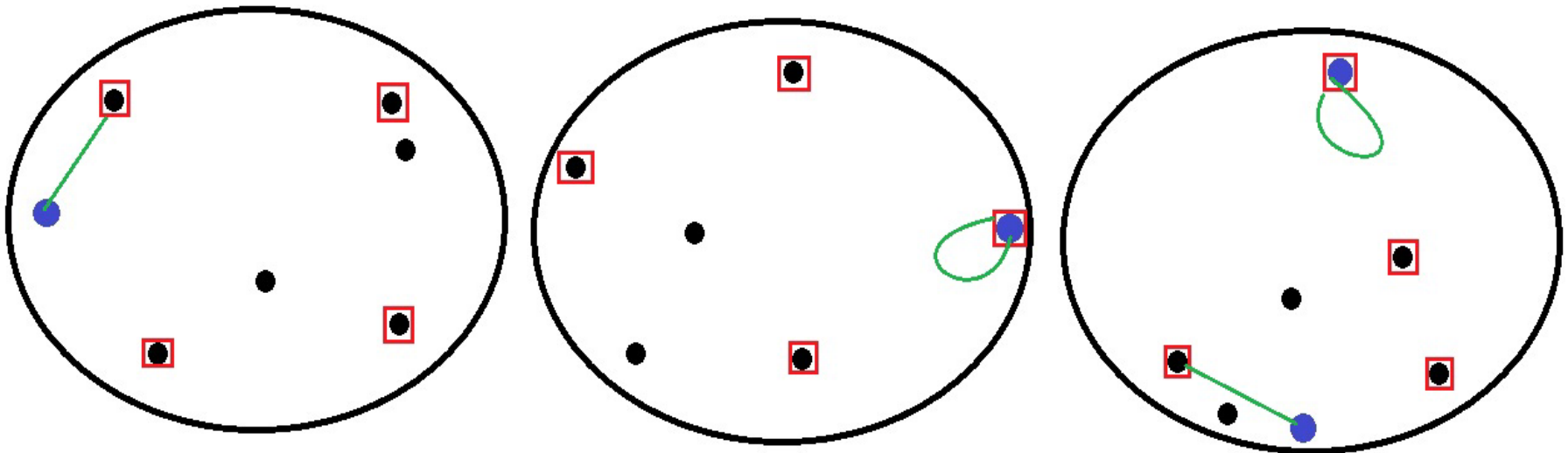
Goal: $\text{div}_k(S) \geq \text{div}(OPT) / c$

Note: $\text{div}_k(S) \geq r$

Case 1: one of S_i has diversity as good as the optimum: $r \geq \mathbf{O}(\text{div}(OPT))$

Case 2: $r \leq \mathbf{O}(\text{div}(OPT))$

- find a **one-to-one** mapping μ from $OPT = \{o_1, \dots, o_k\}$ to $S = S_1 \cup \dots \cup S_m$ s.t.
 $\text{dist}(o_i, \mu(o_i)) \leq \mathbf{O}(r)$
- Replacing o_i with $\mu(o_i)$ has still large diversity
- $\text{div}(\{\mu(o_i)\})$ is approximately as good as $\text{div}(\{o_i\})$



Proof Idea

Let P_1, \dots, P_m be the set of points, $P = \cup P_i$

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

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

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

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

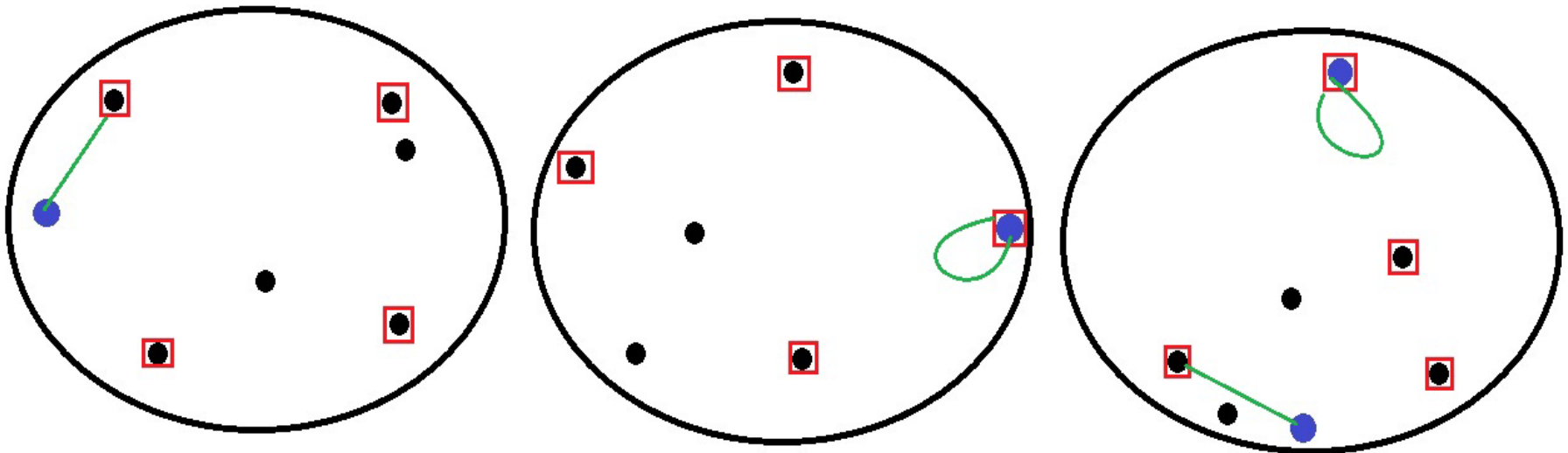
Goal: $\text{div}_k(S) \geq \text{div}(OPT) / c$

Note: $\text{div}_k(S) \geq r$

Case 1: one of S_i has diversity as good as the optimum: $r \geq \mathbf{O}(\text{div}(OPT))$

Case 2: $r \leq \mathbf{O}(\text{div}(OPT))$

- find a **one-to-one** mapping μ from $OPT = \{o_1, \dots, o_k\}$ to $S = S_1 \cup \dots \cup S_m$ s.t.
 $\text{dist}(o_i, \mu(o_i)) \leq \mathbf{O}(r)$
- Replacing o_i with $\mu(o_i)$ has still large diversity
- $\text{div}(\{\mu(o_i)\})$ is approximately as good as $\text{div}(\{o_i\})$
- The actual mapping μ depends on the specific diversity measure we are considering.



Maximum k-Coverage

- A set of n points P in d -dimensional space
- Each dimension corresponds to a feature.
- Goal: choose a set of k points S in P which maximizes the total coverage:

$$- \text{cov}(S) = \sum_{\{i=1\}}^d \max_{\{s \in S\}} s_i$$

Maximum k-Coverage

- A set of n points P in d -dimensional space
- Each dimension corresponds to a feature.
- Goal: choose a set of k points S in P which maximizes the total coverage:

$$- \text{cov}(S) = \sum_{\{i=1\}}^d \max_{\{S \in S\}} s_i$$

- **Special Case hamming space:**
- A collection of n sets P
- Over the universe $U = \{1, \dots, d\}$
- Goal: choose k sets $S = \{S_1, \dots, S_k\}$ in P whose union is maximized.

Maximum k-Coverage

- A set of n points P in d -dimensional space
- Each dimension corresponds to a feature.
- Goal: choose a set of k points S in P which maximizes the total coverage:

$$- \text{cov}(S) = \sum_{i=1}^d \max_{\{S \in S\}} s_i$$

- **Special Case hamming space:**
- A collection of n sets P
- Over the universe $U = \{1, \dots, d\}$
- Goal: choose k sets $S = \{S_1, \dots, S_k\}$ in P whose union is maximized.
- **Theorem:** for any $\alpha < \frac{\sqrt{k}}{\log k}$ and any constant $\beta > 1$, there is no α -composable core-set of size k^β

Proof Idea

Build a set of instances $P_1, \dots, P_{O(k)}$

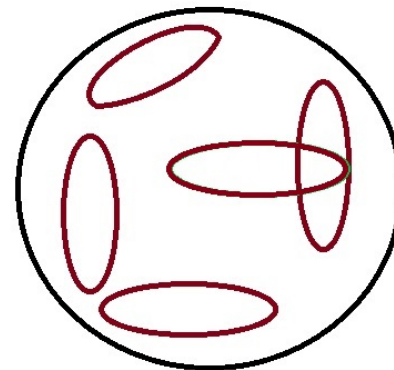
let $U = \{1, \dots, O(k^4)\}$

Proof Idea

Build a set of instances $P_1, \dots, P_{O(k)}$

let $U = \{1, \dots, O(k^4)\}$

- Let V_i be subset of size k of U

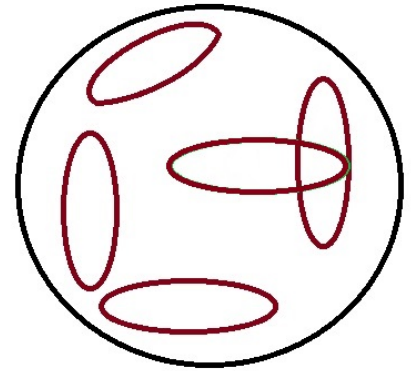


Proof Idea

Build a set of instances $P_1, \dots, P_{O(k)}$

let $U = \{1, \dots, O(k^4)\}$

- Let V_i be subset of size k of U
- P_i is a collection of subsets of size \sqrt{k} from V_i

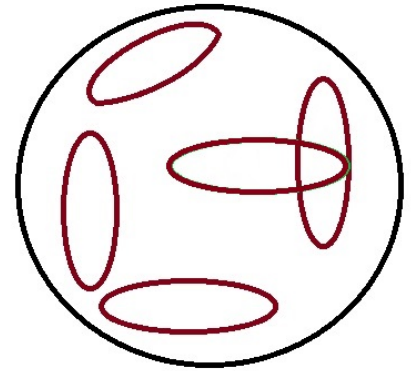


Proof Idea

Build a set of instances $P_1, \dots, P_{O(k)}$

let $U = \{1, \dots, O(k^4)\}$

- Let V_i be subset of size k of U
- P_i is a collection of subsets of size \sqrt{k} from V_i
- P_i has cardinality $\binom{k}{\sqrt{k}}$



Proof Idea

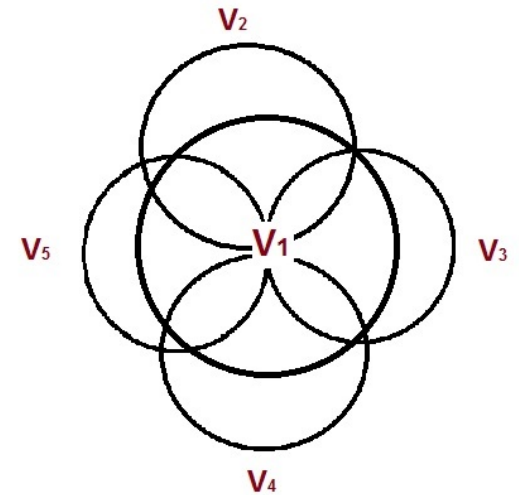
Build a set of instances $P_1, \dots, P_{O(k)}$

let $U = \{1, \dots, O(k^4)\}$

- Let V_i be subset of size k of U
- P_i is a collection of subsets of size \sqrt{k} from V_i
- P_i has cardinality $\binom{k}{\sqrt{k}}$

We show there exists $V_1, \dots, V_{O(k)}$ such that

- $V_i \setminus V_1$ has size \sqrt{k}
- $V_i \setminus V_1$ and $V_j \setminus V_1$ are disjoint for $i \neq j$



Proof Idea

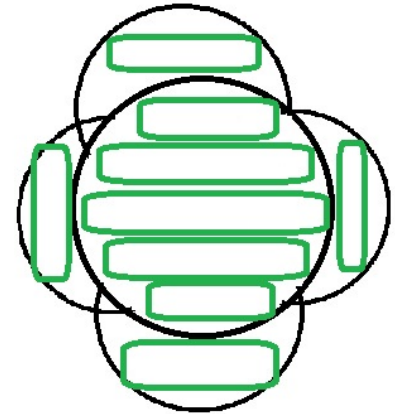
Build a set of instances $P_1, \dots, P_{O(k)}$

let $U = \{1, \dots, O(k^4)\}$

- Let V_i be subset of size k of U
- P_i is a collection of subsets of size \sqrt{k} from V_i
- P_i has cardinality $\binom{k}{\sqrt{k}}$

We show there exists $V_1, \dots, V_{O(k)}$ such that

- $V_i \setminus V_1$ has size \sqrt{k}
- $V_i \setminus V_1$ and $V_j \setminus V_1$ are disjoint for $i \neq j$
- Using k sets everything in $\cup V_i$ can be covered, that is $O(k^{3/2})$ elements.



Proof Idea

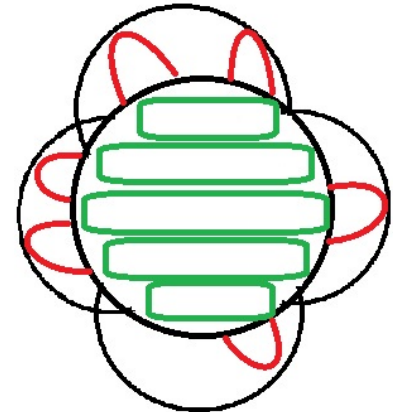
Build a set of instances $P_1, \dots, P_{O(k)}$

let $U = \{1, \dots, O(k^4)\}$

- Let V_i be subset of size k of U
- P_i is a collection of subsets of size \sqrt{k} from V_i
- P_i has cardinality $\binom{k}{\sqrt{k}}$

We show there exists $V_1, \dots, V_{O(k)}$ such that

- $V_i \setminus V_1$ has size \sqrt{k}
- $V_i \setminus V_1$ and $V_j \setminus V_1$ are disjoint for $i \neq j$
- Using k sets everything in $\cup V_i$ can be covered, that is $O(k^{3/2})$ elements.
- Using core-sets only $|V_1| + k \log k = O(k \log k)$ can be covered



Conclusion

- Applications of composable core-sets

Conclusion

- Applications of composable core-sets
- We showed construction of composable core-sets for a wide range of diversity measures

Conclusion

- Applications of composable core-sets
- We showed construction of composable core-sets for a wide range of diversity measures
- We showed non existence of core-sets of polynomial size in k for maximum coverage

Conclusion

- Applications of composable core-sets
- We showed construction of composable core-sets for a wide range of diversity measures
- We showed non existence of core-sets of polynomial size in k for maximum coverage
- **Open Problems**
 - Are there any other applications of composable core-sets?

Conclusion

- Applications of composable core-sets
- We showed construction of composable core-sets for a wide range of diversity measures
- We showed non existence of core-sets of polynomial size in k for maximum coverage
- **Open Problems**
 - Are there any other applications of composable core-sets?
 - Is there a general characterization of measures for which composable core-sets exist?

Conclusion

- Applications of composable core-sets
- We showed construction of composable core-sets for a wide range of diversity measures
- We showed non existence of core-sets of polynomial size in k for maximum coverage
- **Open Problems**
 - Are there any other applications of composable core-sets?
 - Is there a general characterization of measures for which composable core-sets exist?
 - Better approximation factors?

Thank You!

Questions?