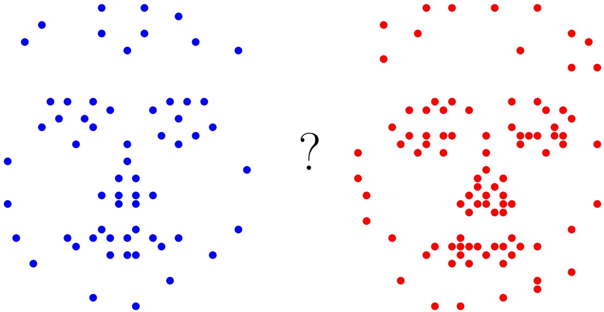
Bipartite Diameter and Other Measures Under Translation

Boris Aronov, **Omrit Filtser**, Matthew J. Katz, and Khadijeh Sheikhan

March 14, 2019

Similarity between two sets of points

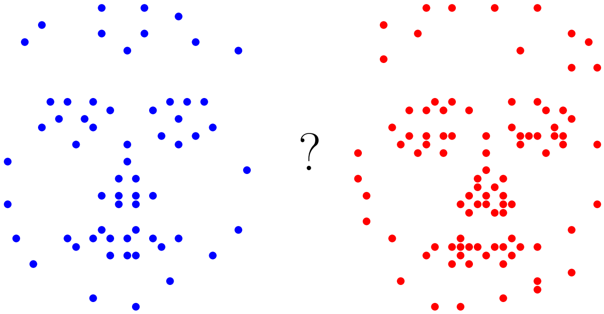
Goal: Determining the similarity between two sets of points.



Similarity between two sets of points

Goal: Determining the similarity between two sets of points.

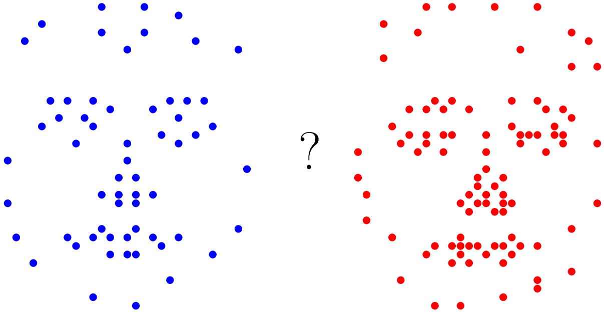
- A well investigated problem in computational geometry.



Similarity between two sets of points

Goal: Determining the similarity between two sets of points.

- A well investigated problem in computational geometry.
- **Problem**: Sometimes, a bipartite measure is meaningless, unless one of the sets undergoes some transformation.



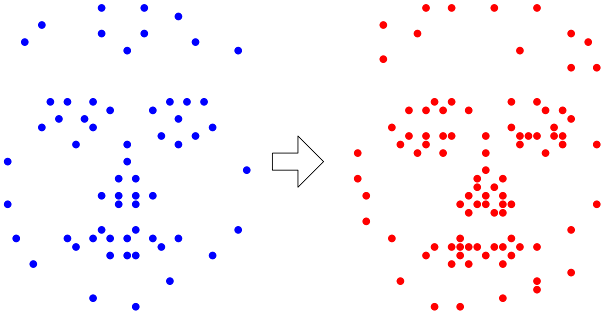
Similarity between two sets of points

Goal: Determining the similarity between two sets of points.

- A well investigated problem in computational geometry.
- **Problem**: Sometimes, a bipartite measure is meaningless, unless one of the sets undergoes some transformation.

This paper:

Find a translation which minimizes some bipartite measure.

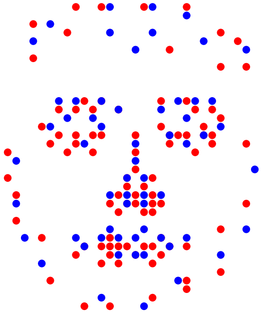


Similarity between two sets of points

Goal: Determining the similarity between two sets of points.

- A well investigated problem in computational geometry.
- **Problem**: Sometimes, a bipartite measure is meaningless, unless one of the sets undergoes some transformation.

This paper:
Find a translation which minimizes some bipartite measure.



Bipartite measures under translation

$A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ – two sets of points in $\mathbb{R}^d$.

Problem

Find a translation t^* that minimizes **some bipartite measure** *of A and $B + t$ over all translations t.*

Bipartite measures under translation

$A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ – two sets of points in $\mathbb{R}^d$.

Problem

Find a translation t^* that minimizes **some bipartite measure** *of A and $B + t$ over all translations t.*

Remarks

Bipartite measures under translation

$A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ – two sets of points in $\mathbb{R}^d$.

Problem

Find a translation t^* that minimizes **some bipartite measure** *of A and $B + t$ over all translations t.*

Remarks

- For the sake of simplicity, we assume that $m = n$.

Bipartite measures under translation

$A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ – two sets of points in $\mathbb{R}^d$.

Problem

Find a translation t^* that minimizes **some bipartite measure** *of A and $B + t$ over all translations t.*

Remarks

- For the sake of simplicity, we assume that $m = n$.
- This class of problems naturally extends to other types of transformations, such as rotations, rigid motions, etc.

Some bipartite measure?

Some bipartite measure?

When comparing two sets of points A and B of the same size:

Some bipartite measure?

When comparing two sets of points A and B of the same size:

- **Congruence testing**: decide if there exists a transformation that maps A exactly or approximately into B.

Some bipartite measure?

When comparing two sets of points A and B of the same size:

- **Congruence testing**.
- **RMS distance**: minimize the sum of squares of distances in a perfect matching between A and B.

Some bipartite measure?

When comparing two sets of points A and B of the same size:

- **Congruence testing**.
- **RMS distance**.

When comparing two sets of points A and B of different sizes:

Some bipartite measure?

When comparing two sets of points A and B of the same size:

- **Congruence testing**.
- **RMS distance**.

When comparing two sets of points A and B of different sizes:

- **Hausdorff distance**: the maximum of the distances from a point in each of the sets to the nearest point in the other set.
  Huttenlocher,Kedem, Sharir: $\tilde{O}(n^3)$ in 2D.

Some bipartite measure?

When comparing two sets of points A and B of the same size:

- **Congruence testing**.
- **RMS distance**.

When comparing two sets of points A and B of different sizes:

- **Hausdorff distance**: $\tilde{O}(n^3)$ in 2D.
- Maximum **overlap between the convex hulls** of the sets A and B. de Berg et al.:$O(n \log n)$ in 2D, Ahn et al.: $\tilde{O}(n^3)$ in 3D.

Some bipartite measure?

When comparing two sets of points A and B of the same size:

- **Congruence testing**.
- **RMS distance**.

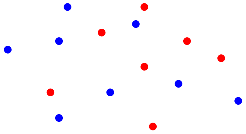
When comparing two sets of points A and B of different sizes:

- **Hausdorff distance**: $\tilde{O}(n^3)$ in 2D.
- Maximum **overlap between the convex hulls** of the sets A and B. de Berg et al.:$O(n \log n)$ in 2D, Ahn et al.: $\tilde{O}(n^3)$ in 3D.

All the above measures (under various geometric transformations) were widely investigated in the literature.

Our results

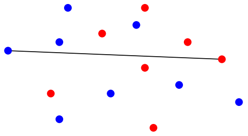
The main bipartite measures that we consider are:



Our results

The main bipartite measures that we consider are:

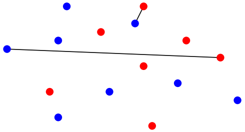
- **diameter** – the distance between the farthest bichromatic pair, i.e. $\max\{\|a-b\| \mid (a,b) \in A \times B\}$.



Our results

The main bipartite measures that we consider are:

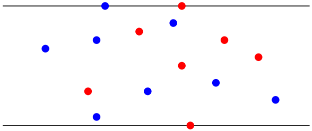
- **diameter** – $\max\{\|a - b\| \mid (a, b) \in A \times B\}$.
- **uniformity** – the difference between the bipartite diameter and the distance between the closest bichromatic pair, i.e. $\mathrm{diam}(A, B) - \min\{\|a - b\| \mid (a, b) \in A \times B\}$.



Our results

The main bipartite measures that we consider are:

- **diameter** – $\max\{\|a-b\| \mid (a,b) \in A \times B\}$.
- **uniformity** – $\mathrm{diam}(A,B) - \min\{\|a-b\| \mid (a,b) \in A \times B\}$.
- **union width** – the width of $A \cup B$, where the *width* of a set of points in the plane is the smallest distance between a pair of parallel lines, such that the closed strip between the lines contains the entire set.



Our results

The main bipartite measures that we consider are:

- **diameter** – $\max\{\|a-b\| \mid (a,b) \in A \times B\}$.
- **uniformity** – $\mathrm{diam}(A,B) - \min\{\|a-b\| \mid (a,b) \in A \times B\}$.
- **union width** – the width of $A \cup B$.
- **red-blue width** – ...

Our results

The main bipartite measures that we consider are:

- **diameter** – $\max\{\|a-b\| \mid (a,b) \in A \times B\}$.
- **uniformity** – $\mathrm{diam}(A,B) - \min\{\|a-b\| \mid (a,b) \in A \times B\}$.
- **union width** – the width of $A \cup B$.
- **red-blue width** – ...

Surprisingly, all of these measures (under translation) were not investigated previously in the literature.

Our results

measure	dimension	running time
diameter	$d = 2$	$O(n \log n)$
	$d = 3$	$O(n \log^2 n)$
	$d > 3$ (fixed)	$O(n^2)$
uniformity	$d = 2$	$O(n^{9/4+\varepsilon})$
union width	$d = 2$	$O(n \log n)$
	$d = 3$	$O(n^2)$

Our results

measure	dimension	running time
diameter	$d = 2$	$O(n \log n)$
	$d = 3$	$O(n \log^2 n)$
	$d > 3$ (fixed)	$O(n^2)$
uniformity	$d = 2$	$O(n^{9/4+\varepsilon})$
union width	$d = 2$	$O(n \log n)$
	$d = 3$	$O(n^2)$

Diameter

$A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ – two sets of points in $\mathbb{R}^d$.

$\text{diam}(A, B) = \max\{\|a - b\| \mid (a, b) \in A \times B\}$

Problem (Bipartite Diameter under Translation)

Find a translation t^* such that for any translation t,
$\text{diam}(A, B + t^*) \leq \text{diam}(A, B + t)$.

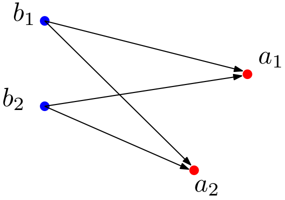
Diameter

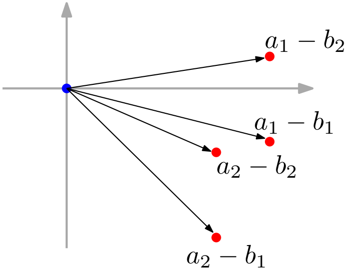
$$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$$

Diameter

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$

- The set of all possible translations taking a point of B to a point of A.

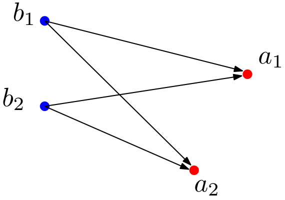


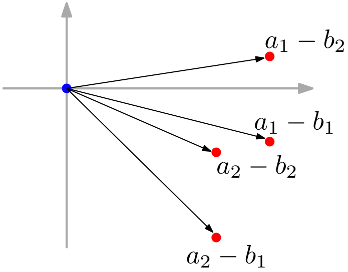


Diameter

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$

- The set of all possible translations taking a point of B to a point of A.
- Clearly, $|\mathcal{P}| = O(n^2)$.



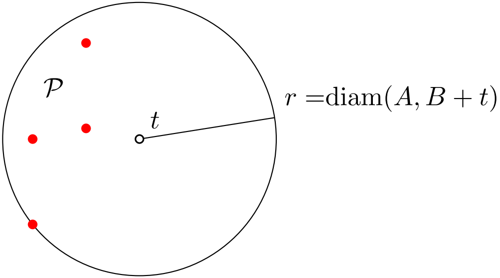


Diameter

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$

Claim

Given a point t, the radius of the **minimum enclosing ball** *of $\mathcal{P}$ centered at t is equal to* $\mathrm{diam}(A, B + t)$.



Diameter

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$

Claim

Given a point t, the radius of the **minimum enclosing ball** *of $\mathcal{P}$ centered at t is equal to* $\mathrm{diam}(A, B + t)$.

Proof.

This radius is at most

$$\max_{(a-b)\in\mathcal{P}} \|(a - b) - t\| = \max_{(a,b)\in A\times B} \|a - (b + t)\| = \mathrm{diam}(A, B + t).$$

□

Diameter

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$

Claim

Given a point t, the radius of the **minimum enclosing ball** *of $\mathcal{P}$ centered at t is equal to* $\mathrm{diam}(A, B + t)$.

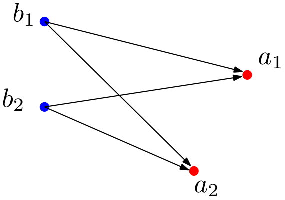
Corollary

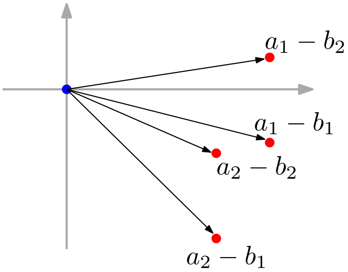
The optimal translation t^* minimizing the bipartite diameter coincides with the center of the minimum enclosing ball of $\mathcal{P}$.

Diameter: Algorithm (naive implementation)

Diameter: Algorithm (naive implementation)

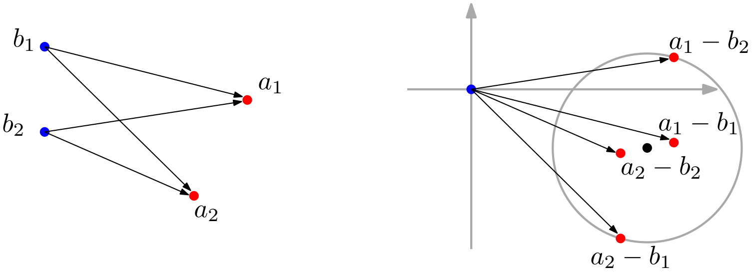
- Compute the set of translations $\mathcal{P}$.





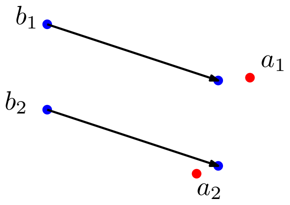
Diameter: Algorithm (naive implementation)

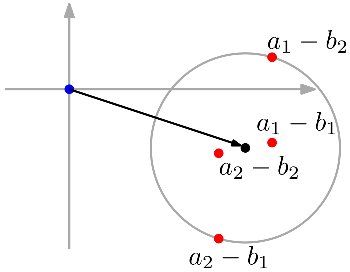
- Compute the set of translations $\mathcal{P}$.
- Find the center c of the minimum enclosing ball of $\mathcal{P}$.



Diameter: Algorithm (naive implementation)

- Compute the set of translations $\mathcal{P}$.
- Find the center c of the minimum enclosing ball of $\mathcal{P}$.
- Translating B by c minimizes the diameter.





Diameter: Running time

The minimum enclosing ball can be computed in:

- linear time using Megiddo's ('83) algorithm, or
- expected linear time using Welzl's ('91) simpler randomized algorithm.

Diameter: Running time

The minimum enclosing ball can be computed in:

- linear time using Megiddo's ('83) algorithm, or
- expected linear time using Welzl's ('91) simpler randomized algorithm.

$\Rightarrow O(n^2)$-time solution.

Diameter: Running time

The minimum enclosing ball can be computed in:

- linear time using Megiddo's ('83) algorithm, or
- expected linear time using Welzl's ('91) simpler randomized algorithm.

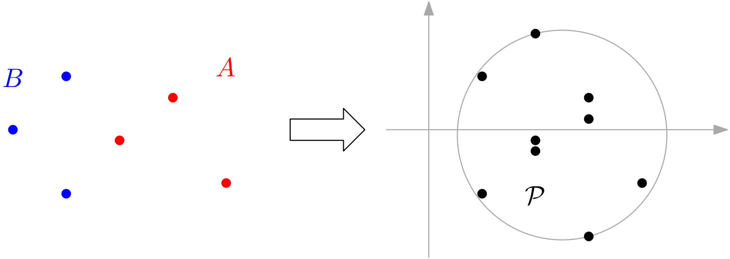
$\Rightarrow O(n^2)$-time solution.

BUT, in 2D and 3D, we can do better!

In fact, computing the minimum enclosing ball of $\mathcal{P}$ in 2D and 3D (*without computing $\mathcal{P}$ explicitly*) can be done in **near-linear time**...

Diameter

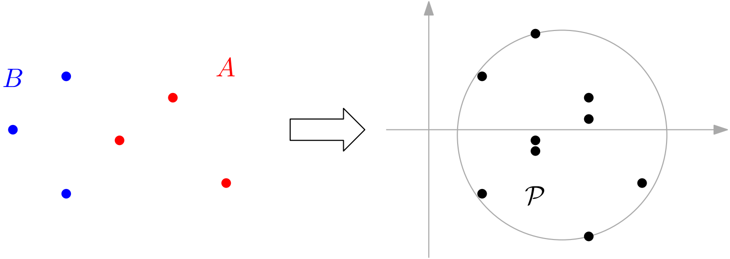
$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$
Goal: compute the minimum enclosing ball of $\mathcal{P}$ implicitly.



Diameter

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$
Goal: compute the minimum enclosing ball of $\mathcal{P}$ implicitly.

Observations



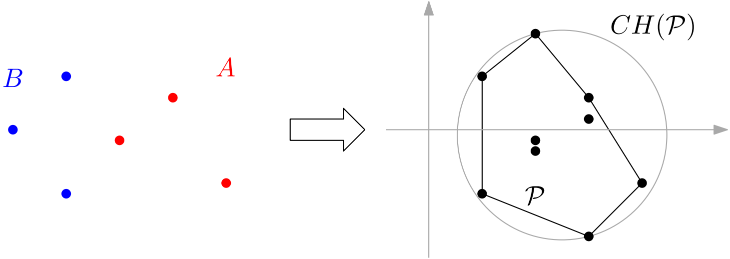
Diameter

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$

Goal: compute the minimum enclosing ball of $\mathcal{P}$ implicitly.

Observations

1. We only need to look at the **convex hull** (CH) of $\mathcal{P}$.

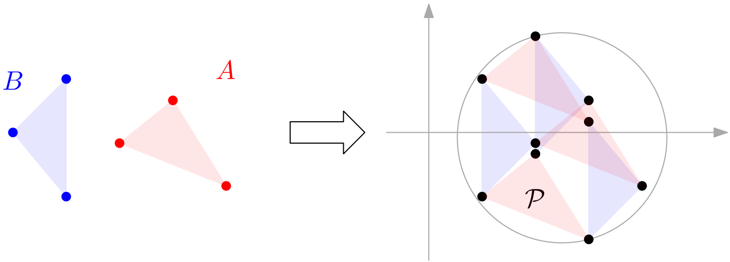


Diameter

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$
Goal: compute the minimum enclosing ball of $\mathcal{P}$ implicitly.

Observations

1. We only need to look at the **convex hull** (CH) of $\mathcal{P}$.
2. $\mathcal{P}$ is the Minkowski sum of A and $-B$, i.e. $\mathcal{P} = A \oplus -B$.



Diameter in 2D

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$
Goal: compute the minimum enclosing ball of $\mathcal{P}$ in 2D.

Diameter in 2D

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$
Goal: compute the minimum enclosing ball of $\mathcal{P}$ in 2D.

A fact from the textbook

For points in 2D, the size of $\mathrm{CH}(A \oplus -B)$ is $O(n)$,
and it can be constructed in $O(n)$ time from $\mathrm{CH}(A)$ and $\mathrm{CH}(B)$
using the well-known rotating calipers method...

Diameter in 2D

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$
Goal: compute the minimum enclosing ball of $\mathcal{P}$ in 2D.

A fact from the textbook

For points in 2D, the size of $\mathrm{CH}(A \oplus -B)$ is $O(n)$,
and it can be constructed in $O(n)$ time from $\mathrm{CH}(A)$ and $\mathrm{CH}(B)$
using the well-known rotating calipers method...

$\Rightarrow O(n \log n)$-time solution for points in 2D!

Diameter in 3D

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$
Goal: compute the minimum enclosing ball of $\mathcal{P}$ in 3D.

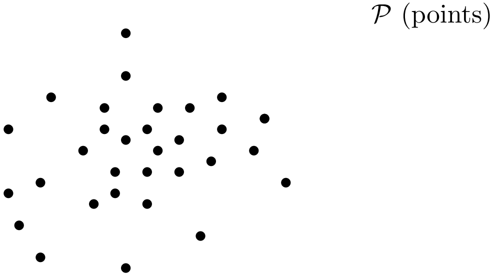
Diameter in 3D

$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$
Goal: compute the minimum enclosing ball of $\mathcal{P}$ in 3D.

Idea: The minimum enclosing ball is an LP-type problem $\Rightarrow$
adapt Clarkson's ('95) scheme for solving LP-type problems.

Diameter in 3D: Algorithm

X – an empty set of points.
Repeat until the minimum enclosing ball is found:

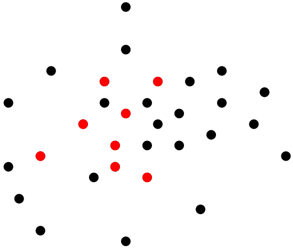


Diameter in 3D: Algorithm

X – an empty set of points.

Repeat until the minimum enclosing ball is found:

1: Pick a random sample $\mathcal{R}$ of $\mathcal{P}$ of size $4n$.



$\mathcal{P}$ (points)

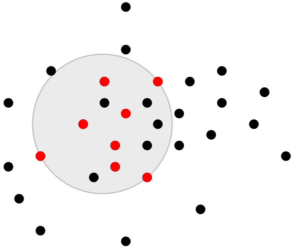
$\mathcal{R}$ (random sample)

Diameter in 3D: Algorithm

X – an empty set of points.

Repeat until the minimum enclosing ball is found:

2: Compute the minimum enclosing ball S of $\mathcal{R} \cup X$.



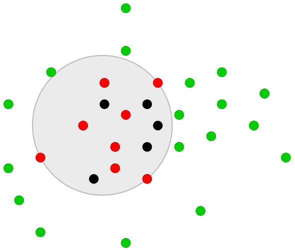
$\mathcal{P}$ (points)

$\mathcal{R}$ (random sample)

Diameter in 3D: Algorithm

X – an empty set of points.
Repeat until the minimum enclosing ball is found:

3: Find the set of *violators* V. If $|V| \geq 2n$, go to 1.



$\mathcal{P}$ (points)

$\mathcal{R}$ (random sample)

V (violators)

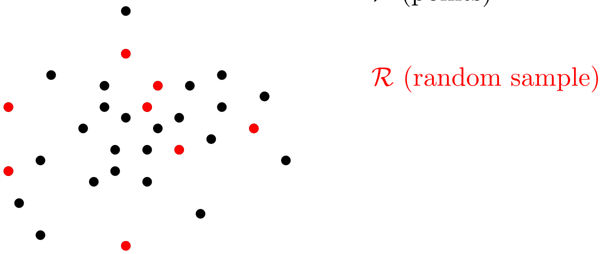
$|V| \geq 2n$, **“bad”** iteration :(

Diameter in 3D: Algorithm

X – an empty set of points.

Repeat until the minimum enclosing ball is found:

1: Pick a random sample $\mathcal{R}$ of $\mathcal{P}$ of size $4n$.

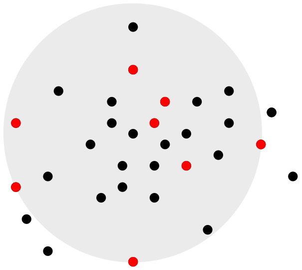


Diameter in 3D: Algorithm

X – an empty set of points.

Repeat until the minimum enclosing ball is found:

2: Compute the minimum enclosing ball S of $\mathcal{R} \cup X$.



$\mathcal{P}$ (points)

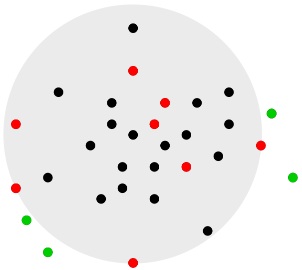
$\mathcal{R}$ (random sample)

Diameter in 3D: Algorithm

X – an empty set of points.

Repeat until the minimum enclosing ball is found:

3: Find the set of *violators* V. If $|V| \geq 2n$, go to 1.



$\mathcal{P}$ (points)

$\mathcal{R}$ (random sample)

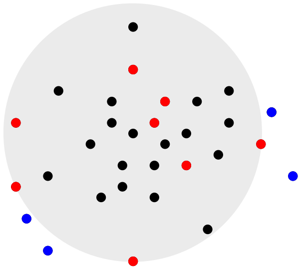
V (violators)

$|V| < 2n$, **"good"** iteration :)

Diameter in 3D: Algorithm

X – an empty set of points.
Repeat until the minimum enclosing ball is found:

4: If $V \neq \emptyset$, then $X \leftarrow X \cup V$ and go to 1. Else, return S.



$\mathcal{P}$ (points)

$\mathcal{R}$ (random sample)

V (violators)

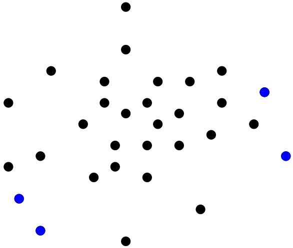
$X \leftarrow X \cup V$

Diameter in 3D: Algorithm

X – an empty set of points.

Repeat until the minimum enclosing ball is found:

1: Pick a random sample $\mathcal{R}$ of $\mathcal{P}$ of size $4n$.



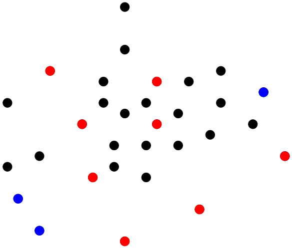
$\mathcal{P}$ (points)

$X \leftarrow X \cup V$

Diameter in 3D: Algorithm

X – an empty set of points.
Repeat until the minimum enclosing ball is found:

1: Pick a random sample $\mathcal{R}$ of $\mathcal{P}$ of size $4n$.



$\mathcal{P}$ (points)

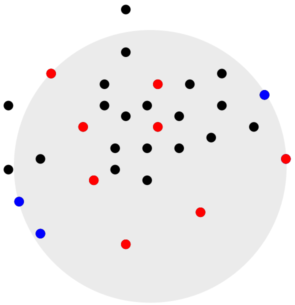
$\mathcal{R}$ (random sample)

$X \leftarrow X \cup V$

Diameter in 3D: Algorithm

X – an empty set of points.
Repeat until the minimum enclosing ball is found:

2: Compute the minimum enclosing ball S of $\mathcal{R} \cup X$.



$\mathcal{P}$ (points)

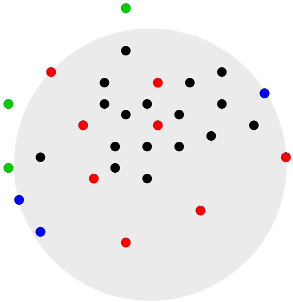
$\mathcal{R}$ (random sample)

$X \leftarrow X \cup V$

Diameter in 3D: Algorithm

X – an empty set of points.
Repeat until the minimum enclosing ball is found:

3: Find the set of *violators* V. If $|V| \geq 2n$, go to 1.



$\mathcal{P}$ (points)

$\mathcal{R}$ (random sample)

V (violators)

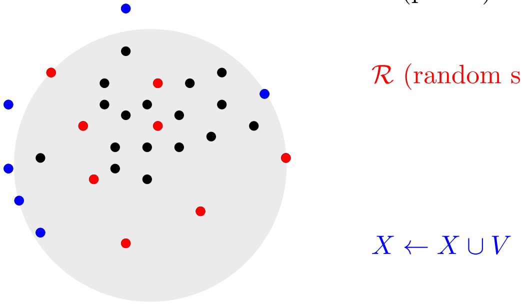
$X \leftarrow X \cup V$

Diameter in 3D: Algorithm

X – an empty set of points.

Repeat until the minimum enclosing ball is found:

4: If $V \neq \emptyset$, then $X \leftarrow X \cup V$ and go to 1. Else, return S.

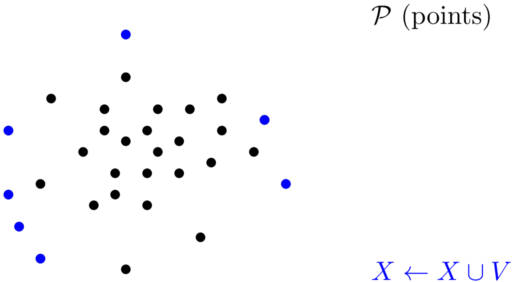


Diameter in 3D: Algorithm

X – an empty set of points.

Repeat until the minimum enclosing ball is found:

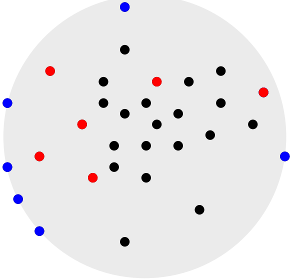
4: If $V \neq \emptyset$, then $X \leftarrow X \cup V$ and go to 1. Else, return S.



Diameter in 3D: Algorithm

X – an empty set of points.
Repeat until the minimum enclosing ball is found:

4: If $V \neq \emptyset$, then $X \leftarrow X \cup V$ and go to 1. Else, return S.



$\mathcal{P}$ (points)

$X \leftarrow X \cup V$

Diameter in 3D: number of iterations

How many iterations?

Diameter in 3D: number of iterations

How many iterations?

- The number of **good** iterations cannot exceed five! (why?)

Diameter in 3D: number of iterations

How many iterations?

- The number of **good** iterations cannot exceed five! (why?)

$\Rightarrow |X| = O(n)$.

Diameter in 3D: number of iterations

How many iterations?

- The number of **good** iterations cannot exceed five! (why?)

$\Rightarrow |X| = O(n)$.

- By Clarkson's analysis:
  In each iteration the expected size of V is n.

Diameter in 3D: number of iterations

How many iterations?

- The number of **good** iterations cannot exceed five! (why?)

$\Rightarrow |X| = O(n)$.

- By Clarkson's analysis:
  In each iteration the expected size of V is n.

$\Rightarrow$ By Markov's inequality, $Pr(|V| \geq 2n) \leq \frac{1}{2}$

Diameter in 3D: number of iterations

How many iterations?

- The number of **good** iterations cannot exceed five! (why?)

$\Rightarrow |X| = O(n)$.

- By Clarkson's analysis:
  In each iteration the expected size of V is n.

$\Rightarrow$ By Markov's inequality, $Pr(|V| \geq 2n) \leq \frac{1}{2}$

$\Rightarrow$ Expected number of **bad** iterations before a good one is $O(1)$!

Diameter in 3D: number of iterations

How many iterations?

- The number of **good** iterations cannot exceed five! (why?)

$\Rightarrow |X| = O(n)$.

- By Clarkson's analysis:
  In each iteration the expected size of V is n.

$\Rightarrow$ By Markov's inequality, $Pr(|V| \geq 2n) \leq \frac{1}{2}$

$\Rightarrow$ Expected number of **bad** iterations before a good one is $O(1)$!

$\Rightarrow$ Expected number of iterations is constant!

Diameter in 3D: implementing an iteration

What is the running time for one iteration?

Diameter in 3D: implementing an iteration

What is the running time for one iteration?

1: Pick a random sample $\mathcal{R}$ of $\mathcal{P}$ of size $4n$.

Diameter in 3D: implementing an iteration

What is the running time for one iteration?

1: Pick a random sample $\mathcal{R}$ of $\mathcal{P}$ of size $4n$.

- Repeatedly pick random points $a \in A$ and $b \in B$ and return $a - b$.

Diameter in 3D: implementing an iteration

What is the running time for one iteration?

2: Compute the minimum enclosing ball S of $\mathcal{R} \cup X$.

Diameter in 3D: implementing an iteration

What is the running time for one iteration?

2: Compute the minimum enclosing ball S of $\mathcal{R} \cup X$.

- Invoke a standard minimum-ball algorithm on $O(n)$ points, requiring $O(n)$ expected time.

Diameter in 3D: implementing an iteration

What is the running time for one iteration?

3: Find the set of violators V.

Diameter in 3D: implementing an iteration

What is the running time for one iteration?

3: Find the set of violators V.

- ?

Diameter in 3D: implementing an iteration

What is the running time for one iteration?

3: Find the set of violators V.

Idea: We solve the following problem.

Problem

Given two sets A and B, each of n points in $\mathbb{R}^3$, a distance r and a parameter k, report all the pairs of points $a \in A$, $b \in B$ with $\|a - b\| > r$, if there are at most k such pairs.
Otherwise, return "TOO MANY".

Diameter in 3D: implementing an iteration

What is the running time for one iteration?

3: Find the set of violators V.

Idea: We solve the following problem.

Problem

Given two sets A and B, each of n points in $\mathbb{R}^3$, a distance r and a parameter k, report all the pairs of points $a \in A$, $b \in B$ with $\|a - b\| > r$, if there are at most k such pairs. Otherwise, return "TOO MANY".

- Expected running time $O((n + k)\log^2 n)$.

Diameter in 3D: implementing an iteration

What is the running time for one iteration?

3: Find the set of violators V.

Idea: We solve the following problem.

Problem

Given two sets A and B, each of n points in $\mathbb{R}^3$, a distance r and a parameter k, report all the pairs of points $a \in A$, $b \in B$ with $\|a - b\| > r$, if there are at most k such pairs. Otherwise, return "TOO MANY".

- Expected running time $O((n + k) \log^2 n)$.

$\Rightarrow O(n \log^2 n)$-time solution for points in 3D!

Uniformity

$A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ – two sets of points in $\mathbb{R}^d$.

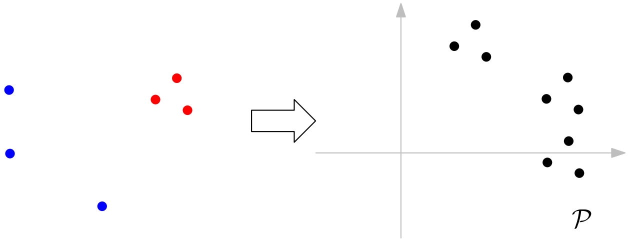
$\text{uni}(A, B) = \text{diam}(A, B) - \min\{\|a - b\| \mid (a, b) \in A \times B\}$

Problem (Uniformity under Translation)

Find a translation t^* such that for any translation t,
$\text{uni}(A, B + t^*) \leq \text{uni}(A, B + t)$.

Uniformity

$$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$$

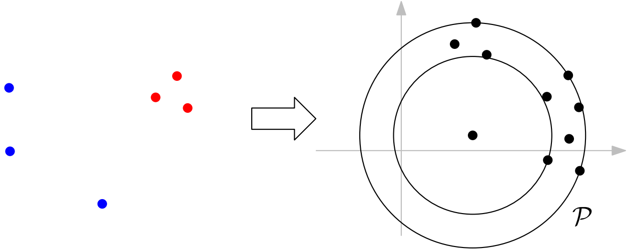


Uniformity

$$\mathcal{P} = \{a - b \mid (a, b) \in A \times B\}$$

Claim

The optimal translation t^* minimizing the **uniformity** *coincides with the center of the* **minimum-width annulus** *containing $\mathcal{P}$.*



Uniformity in 2D

- Agarwal and Sharir ('96): The minimum enclosing annulus of n points in 2D can be computed in $O(n^{3/2+\varepsilon})$ expected time...

Uniformity in 2D

- Agarwal and Sharir ('96): The minimum enclosing annulus of n points in 2D can be computed in $O(n^{3/2+\varepsilon})$ expected time...

$\Rightarrow O(n^{3+\varepsilon})$-time solution.

Uniformity in 2D

- Agarwal and Sharir ('96): The minimum enclosing annulus of n points in 2D can be computed in $O(n^{3/2+\varepsilon})$ expected time...

$\Rightarrow O(n^{3+\varepsilon})$-time solution.

Claim

The minimum enclosing annulus of n points in 2D
— **with only** $O(\sqrt{n})$ **extreme points** —
can be computed in $O(n^{9/8+\varepsilon})$ *expected time.*

Uniformity in 2D

- Agarwal and Sharir ('96): The minimum enclosing annulus of n points in 2D can be computed in $O(n^{3/2+\varepsilon})$ expected time...

$\Rightarrow O(n^{3+\varepsilon})$-time solution.

Claim

The minimum enclosing annulus of n points in 2D
— **with only** $O(\sqrt{n})$ **extreme points** —
can be computed in $O(n^{9/8+\varepsilon})$ *expected time.*

$\Rightarrow O(n^{9/4+\varepsilon})$-time solution!

Thank You!

Open questions

- Consider other types of transformations?

Open questions

- Consider other types of transformations?
- **Diameter**:

Open questions

- Consider other types of transformations?
- **Diameter**:
  - We showed near linear time algorithms in 2D and 3D.

Open questions

- Consider other types of transformations?
- **Diameter**:
  - We showed near linear time algorithms in 2D and 3D.
  - Can we also obtain a near linear algorithm in higher dimensions ($O(n \log^{O(d)} n)$)?

Open questions

- Consider other types of transformations?
- **Diameter**:
  - We showed near linear time algorithms in 2D and 3D.
  - Can we also obtain a near linear algorithm in higher dimensions ($O(n \log^{O(d)} n)$)?
- **Uniformity**:

Open questions

- Consider other types of transformations?
- **Diameter**:
  - We showed near linear time algorithms in 2D and 3D.
  - Can we also obtain a near linear algorithm in higher dimensions ($O(n \log^{O(d)} n)$)?
- **Uniformity**:
  - Consider minimum ratio instead of minimum difference?

Open questions

- Consider other types of transformations?
- **Diameter**:
  - We showed near linear time algorithms in 2D and 3D.
  - Can we also obtain a near linear algorithm in higher dimensions ($O(n \log^{O(d)} n)$)?
- **Uniformity**:
  - Consider minimum ratio instead of minimum difference?
  - Higher dimensions?

Open questions

- Consider other types of transformations?
- **Diameter**:
  - We showed near linear time algorithms in 2D and 3D.
  - Can we also obtain a near linear algorithm in higher dimensions ($O(n\log^{O(d)} n)$)?
- **Uniformity**:
  - Consider minimum ratio instead of minimum difference?
  - Higher dimensions?
- **Width**:

Open questions

- Consider other types of transformations?
- **Diameter**:
  - We showed near linear time algorithms in 2D and 3D.
  - Can we also obtain a near linear algorithm in higher dimensions ($O(n \log^{O(d)} n)$)?
- **Uniformity**:
  - Consider minimum ratio instead of minimum difference?
  - Higher dimensions?
- **Width**:
  - For width in 3D (without translation) there is an $O(n^{3/2+\epsilon})$-time algorithm (Agarwal and Sharir).

Open questions

- Consider other types of transformations?
- **Diameter**:
  - We showed near linear time algorithms in 2D and 3D.
  - Can we also obtain a near linear algorithm in higher dimensions ($O(n \log^{O(d)} n)$)?
- **Uniformity**:
  - Consider minimum ratio instead of minimum difference?
  - Higher dimensions?
- **Width**:
  - For width in 3D (without translation) there is an $O(n^{3/2+\epsilon})$-time algorithm (Agarwal and Sharir).
  - Our algorithm (for width in 3D under translation) runs in $O(n^2)$ time. Can we do better?