Approximate Convex Intersection Detection with Applications to Width and Minkowski Sums

Sunil Arya
Hong Kong University of Science and Technology

Guilherme D. da Fonseca
INRIA Sophia-Antipolis, Université Clermont Auvergne, and LIMOS

David M. Mount
University of Maryland, College Park

ESA, August 2018
Approximate polytope intersection in $O(\text{polylog}\frac{1}{\varepsilon})$ time
- Given two preprocessed polytopes
- Storage: $O(1/\varepsilon^{(d-1)/2})$

Approximation to Minkowski sum in $O(n \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^{(d-1)/2}+\alpha})$ time
- Any $\alpha > 0$
- Previously $O(n + 1/\varepsilon^{d-1})$

Width approximation in $O(n \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^{(d-1)/2}+\alpha})$ time
- Any $\alpha > 0$
- Previously $O(n + 1/\varepsilon^{d-1})$
Approximate polytope intersection in $O(\text{polylog}\frac{1}{\varepsilon})$ time

- Given two preprocessed polytopes
- Storage: $O(1/\varepsilon^{(d-1)/2})$

Approximation to Minkowski sum in $O(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{(d-1)/2+\alpha})$ time

- Any $\alpha > 0$
- Previously $O(n + 1/\varepsilon^{d-1})$

Width approximation in $O(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{(d-1)/2+\alpha})$ time

- Any $\alpha > 0$
- Previously $O(n + 1/\varepsilon^{d-1})$
Results

1. Approximate polytope intersection in $O(\text{polylog}\frac{1}{\epsilon})$ time
   - Given two preprocessed polytopes
   - Storage: $O\left(\frac{1}{\epsilon^{(d-1)/2}}\right)$

2. Approximation to Minkowski sum in $O(n \log \frac{1}{\epsilon} + \frac{1}{\epsilon^{(d-1)/2+\alpha}})$ time
   - Any $\alpha > 0$
   - Previously $O(n + \frac{1}{\epsilon^{d-1}})$

3. Width approximation in $O(n \log \frac{1}{\epsilon} + \frac{1}{\epsilon^{(d-1)/2+\alpha}})$ time
   - Any $\alpha > 0$
   - Previously $O(n + \frac{1}{\epsilon^{d-1}})$
**Directional Width**

**Exact directional width**

Given:
- \( S \): set of \( n \) points in \( \mathbb{R}^d \)
- \( v \): unit vector

Define \( \text{width}_v(S) \):
- Smallest distance between two hyperplanes orthogonal to \( v \) enclosing \( S \)

**Approximate directional width:**
- Given \( \varepsilon > 0 \)
- Find points \( p, q \in S \) with
  \[ \text{width}_v(\{p, q\}) \geq (1 - \varepsilon) \text{width}_v(S) \]
**Directional Width**

### Exact directional width

**Given:**
- $S$: set of $n$ points in $\mathbb{R}^d$
- $v$: unit vector

**Define $\text{width}_v(S)$:**
- Smallest distance between two hypeplanes orthogonal to $v$ enclosing $S$

### Approximate directional width:

- Given $\varepsilon > 0$
- Find points $p, q \in S$ with $\text{width}_v(\{p, q\}) \geq (1 - \varepsilon) \cdot \text{width}_v(S)$
Preprocess into a data structure: \([AFM17a, AFM17b]\)

- \(S\): set of \(n\) points in \(\mathbb{R}^d\)
- \(\varepsilon\): small positive parameter

Given query vector \(v\):

- Answer approximate directional width

Complexity of directional width

- Query time: \(O\left(\log^2 \frac{1}{\varepsilon}\right)\)
- Storage: \(O\left(\frac{1}{\varepsilon^{d-1/2}}\right)\)
- Preprocessing time: \(O\left(n \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^{d-1/2}} + \alpha\right)\)
  for any \(\alpha > 0\)
Preprocess into a data structure: [AFM17a,AFM17b]

- \( S \): set of \( n \) points in \( \mathbb{R}^d \)
- \( \varepsilon \): small positive parameter

Given query vector \( v \):

- Answer approximate directional width

Complexity of directional width

- Query time: \( O(\log^2 \frac{1}{\varepsilon}) \)
- Storage: \( O \left( \frac{1}{\varepsilon^{\frac{d-1}{2}}} \right) \)
- Preprocessing time: \( O \left( n \log \frac{1}{\varepsilon} + \frac{d-1}{\varepsilon^{\frac{d-1}{2}}} + \alpha \right) \)
  for any \( \alpha > 0 \)
Diameter vs Width

- **Diameter**: \( \max_v \text{width}_v(S) \)
- **Width**: \( \min_v \text{width}_v(S) \)

- **Diameter**: Approximated using \( O(1/\varepsilon^{d-1}) \) directional width queries [Cha02]
  
  **Time**: \( O\left(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{\frac{d-1}{2} + \alpha}\right) \) [AFM17b,Cha17]

- **Width**: Known algorithms take \( O(n + 1/\varepsilon^{d-1}) \) time [Cha02,Cha06]

- Can we approximate the width faster?
Diameter vs Width

- **Diameter:** \( \max_v \ \text{width}_v(S) \)
- **Width:** \( \min_v \ \text{width}_v(S) \)

- **Diameter:** Approximated using \( O(1/\varepsilon^{d-1}) \) directional width queries [Cha02]
  
  Time: \( O \left( n \log \frac{1}{\varepsilon} + 1/\varepsilon^{d-1/2} + \alpha \right) \) [AFM17b, Cha17]

- **Width:** Known algorithms take \( O(n + 1/\varepsilon^{d-1}) \) time [Cha02, Cha06]

- Can we approximate the width faster?
Diameter vs Width

- **Diameter:** $\max_v \text{width}_v(S)$
- **Width:** $\min_v \text{width}_v(S)$

- **Diameter:** Approximated using $O(1/\varepsilon \frac{d-1}{2})$ directional width queries [Cha02]
  
  - **Time:** $O\left(n \log \frac{1}{\varepsilon} + 1/\varepsilon \frac{d-1}{2} + \alpha\right)$ [AFM17b,Cha17]

- **Width:** Known algorithms take $O(n + 1/\varepsilon^{d-1})$ time [Cha02,Cha06]

- Can we approximate the width faster?
Minkowski Sum

**Minkowski sum**

- **$A, B$:** Sets of points
- **$A \oplus B = \{p + q : p \in A, q \in B\}$**

- **Applications:** motion planning, CAD, biology, engineering...
- **Slow** to compute: $O(n^2)$

- **What if we approximate?**

---

Minkowski sum is a fundamental concept in geometry that combines two sets of points into a new set. Given two sets $A$ and $B$, the Minkowski sum $A \oplus B$ is defined as the set of all possible sums of a point from $A$ and a point from $B$, i.e., $A \oplus B = \{p + q : p \in A, q \in B\}$. This concept has a wide range of applications, including motion planning, CAD, biology, and engineering, and it is computationally slow, with a complexity of $O(n^2)$.
Minkowski Sum

A, B: Sets of points

\[ A \oplus B = \{ p + q : p \in A, q \in B \} \]

Applications: motion planning, CAD, biology, engineering...

Slow to compute: \( O(n^2) \)

What if we approximate?
Important Properties

1. \( \text{width}_v(A \oplus B) = \text{width}_v(A) + \text{width}_v(B) \)
   - We can query \( A \oplus B \) using data structures for \( A \) and \( B \)

2. Width of \( A \): \( \text{Min} \|v\| \) for \( v \in \partial(A \oplus (-A)) \)
   - Easy if \( A \oplus (-A) \) is represented by hyperplanes

3. \( A \cap B \neq \emptyset \Leftrightarrow O \in A \oplus -B \)
   - We’ll use in the next slide

Strategy to approximate width

Build hyperplane representation of \( A \oplus -A \)
using only directional width queries
### Important Properties

1. \[ \text{width}_v(A \oplus B) = \text{width}_v(A) + \text{width}_v(B) \]
   - We can query \( A \oplus B \) using data structures for \( A \) and \( B \)

2. **Width of** \( A \): \[ \text{Min} \|v\| \text{ for } v \in \partial(A \oplus (-A)) \]
   - Easy if \( A \oplus (-A) \) is represented by hyperplanes

3. \( A \cap B \neq \emptyset \iff O \in A \oplus (-B) \)
   - We’ll use in the next slide

---

**Strategy to approximate width**

Build **hyperplane** representation of \( A \oplus -A \) using only **directional width** queries
Important Properties

1. \[ \text{width}_v(A \oplus B) = \text{width}_v(A) + \text{width}_v(B) \]
   - We can query \( A \oplus B \) using data structures for \( A \) and \( B \)

2. Width of \( A \): \[ \text{Min} \|v\| \text{ for } v \in \partial(A \oplus (-A)) \]
   - Easy if \( A \oplus (-A) \) is represented by hyperplanes

3. \[ A \cap B \neq \emptyset \iff O \in A \oplus -B \]
   - We’ll use in the next slide

Strategy to approximate width

Build hyperplane representation of \( A \oplus -A \)
using only directional width queries
Important Properties

1. \( \text{width}_v(A \oplus B) = \text{width}_v(A) + \text{width}_v(B) \)
   - We can query \( A \oplus B \) using data structures for \( A \) and \( B \)

2. Width of \( A \): \( \text{Min } \|v\| \) for \( v \in \partial(A \oplus (-A)) \)
   - Easy if \( A \oplus (-A) \) is represented by hyperplanes

3. \( A \cap B \neq \emptyset \iff O \in A \oplus -B \)
   - We’ll use in the next slide

Strategy to approximate width

Build hyperplane representation of \( A \oplus -A \) using only directional width queries
### Polytope Intersection

**Property 3**

\[ A \cap B \neq \emptyset \iff O \in A \oplus -B \]

- \( S \): set of points
- Question: Is \( O \in \text{conv}(S) \)?
- Classic linear programming problem
  - Faster approximate solution after preprocessing?
  - Look at the dual

**Intersection, Minkowski Sum, and Width**

- Results
- Dir. Width
- Black Box
- Diam vs Width
- Minkowski Properties
- Origin
- Duality
- Minimization
- d-Dimensional
- Intersection
- Dudley
- Fatness
- Fattening
- Closest
- Minkowski Apx
- Width
- Conclusion
- Bibliography
- Thanks
Property 3

\[ A \cap B \neq \emptyset \iff O \in A \oplus -B \]

- \( S \): set of points
- Question: Is \( O \in \text{conv}(S) \)?
- Classic linear programming problem
- Faster approximate solution after preprocessing?
- Look at the dual
**Point-Hyperplane Duality**

### Duality

Point \((p_1, \ldots, p_d)\) maps to hyperplane

\[ x_d = p_1 x_1 + \cdots + p_{d-1} x_{d-1} - p_d \]

We want to solve:

- Primal: \(O \in \text{conv}(S)\)
- Dual: hyperplane \(O^* : x_d = 0\) between upper and lower envelopes

We have access to:

- Primal: directional width
- Dual: vertical ray shooting
Point-Hyperplane Duality

**Duality**

Point \((p_1, \ldots, p_d)\) maps to hyperplane

\[ x_d = p_1 x_1 + \cdots + p_{d-1} x_{d-1} - p_d \]

We want to solve:
- **Primal:** \(O \in \text{conv}(S)\)
- **Dual:** hyperplane \(O^* : x_d = 0\) between upper and lower envelopes

We have access to:
- **Primal:** directional width
- **Dual:** vertical ray shooting
Point-Hyperplane Duality

Duality

Point \((p_1, \ldots, p_d)\) maps to hyperplane

\[ x_d = p_1 x_1 + \cdots + p_{d-1} x_{d-1} - p_d \]

We want to solve:
- **Primal:** \(O \in \text{conv}(S)\)
- **Dual:** hyperplane \(O^*: x_d = 0\)
  between upper and lower envelopes

We have access to:
- **Primal:** directional width
- **Dual:** vertical ray shooting
One-Dimensional Convex Minimization

- Upper envelope is convex
- Minimize convex function using evaluations
- Slope at most $c$
- Binary search:
  - Sample 4 points
  - Recurse $2/3$ (or $1/3$) interval containing smallest sample
  - Stop with interval size $\varepsilon/c$
- $O(\log \frac{1}{\varepsilon})$ time for $f : [0, 1] \rightarrow \mathbb{R}$
One-Dimensional Convex Minimization

- Upper envelope is convex
- Minimize convex function using evaluations
- Slope at most $c$
- Binary search:
  - Sample 4 points
  - Recurse $2/3$ (or $1/3$) interval containing smallest sample
  - Stop with interval size $\varepsilon/c$
- $O(\log \frac{1}{\varepsilon})$ time for $f : [0, 1] \to \mathbb{R}$
One-Dimensional Convex Minimization

- Upper envelope is convex
- Minimize convex function using evaluations
- Slope at most $c$
- Binary search:
  - Sample 4 points
  - Recurse $2/3$ (or $1/3$) interval containing smallest sample
  - Stop with interval size $\varepsilon/c$
- $O(\log \frac{1}{\varepsilon})$ time for $f : [0, 1] \to \mathbb{R}$
One-Dimensional Convex Minimization

- Upper envelope is convex
- Minimize convex function using evaluations
- Slope at most $c$
- Binary search:
  - Sample 4 points
  - Recurse $2/3$ (or $1/3$) interval containing smallest sample
  - Stop with interval size $\varepsilon/c$
- $O(\log \frac{1}{\varepsilon})$ time for $f : [0, 1] \rightarrow \mathbb{R}$
One-Dimensional Convex Minimization

- Upper envelope is convex
- **Minimize** convex function using evaluations
- Slope at most $c$
- Binary search:
  - Sample 4 points
  - Recurse $2/3$ (or $1/3$) interval containing smallest sample
  - Stop with interval size $\varepsilon/c$
- $O(\log \frac{1}{\varepsilon})$ time for $f : [0, 1] \to \mathbb{R}$
One-Dimensional Convex Minimization

- Upper envelope is convex
- **Minimize** convex function using evaluations
- Slope at most $c$
- **Binary search**:
  - Sample 4 points
  - Recurse $2/3$ (or $1/3$) interval containing smallest sample
  - Stop with interval size $\varepsilon/c$
- $O(\log \frac{1}{\varepsilon})$ time for $f : [0, 1] \rightarrow \mathbb{R}$
$g(x_1) = \min_{x_2,\ldots,x_d \in [0,1]^{d-1}} f(x_1,\ldots,x_d)$

- $g : [0,1] \to \mathbb{R}$ is convex
- Minimize $g(\cdot)$
- Solve $(d-1)$-dimensional minimization to evaluate $g(\cdot)$
  - $t(1) = O(\log \frac{1}{\varepsilon})$
  - $t(d) = t(d-1) \cdot t(1)$
- $t(d) = O(\log^d \frac{1}{\varepsilon})$ time for $f : [0,1]^d \to \mathbb{R}$
\( d \)-Dimensional Convex Minimization

\[
g(x_1) = \min_{x_2, \ldots, x_d \in [0,1]^{d-1}} f(x_1, \ldots, x_d)
\]

- \( g : [0, 1] \rightarrow \mathbb{R} \) is convex
- Minimize \( g(\cdot) \)
- Solve \((d-1)\)-dimensional minimization to evaluate \( g(\cdot) \)
  - \( t(1) = O(\log \frac{1}{\varepsilon}) \)
  - \( t(d) = t(d-1) \cdot t(1) \)
  - \( t(d) = O(\log^d \frac{1}{\varepsilon}) \) time for \( f : [0,1]^d \rightarrow \mathbb{R} \)
Approximate Polytope Intersection

- If $A$ intersects $B$: answer **yes**
- If the distance between $A$ and $B$ is more than $\varepsilon \cdot (\text{diam}(A) + \text{diam}(B))$: answer **no**
- Otherwise either answer is ok

(1) Approximate polytope intersection

- Query time: $O(\text{polylog} \frac{1}{\varepsilon})$
- Storage: $O(\frac{1}{\varepsilon^{(d-1)/2}})$
- Preprocessing time: $O(n \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^{(d-1)/2+\alpha}})$, for any $\alpha > 0$
Dudley’s result: [Dud74]

A convex body $K$ of diameter 1 can be $\varepsilon$-approximated by a polytope $P$ with $O\left(\frac{1}{\varepsilon^{\frac{d-1}{2}}}\right)$ facets.

- Fatten $K$ into $K'$
- Ball $B$ of radius $2 \cdot \text{diam}(K')$
- $\sqrt{\varepsilon}$-net $N$ on $B$
- Closest point on $K'$ for each point in $N$
- $P'$ bounded by tangent hyperplanes
- Unfatten $P'$ into $P$
Dudley’s result: [Dud74]

A convex body $K$ of diameter 1 can be $\varepsilon$-approximated by a polytope $P$ with $O\left(\frac{1}{\varepsilon^{\frac{d-1}{2}}}\right)$ facets.

- Fatten $K$ into $K'$
- Ball $B$ of radius $2 \cdot \text{diam}(K')$
- $\sqrt{\varepsilon}$-net $N$ on $B$
- Closest point on $K'$ for each point in $N$
- $P'$ bounded by tangent hyperplanes
- Unfatten $P'$ into $P$
Dudley’s result: [Dud74]

A convex body $K$ of diameter 1 can be $\varepsilon$-approximated by a polytope $P$ with $O(1/\varepsilon^{\frac{d-1}{2}})$ facets.

- Fatten $K$ into $K'$
- Ball $B$ of radius $2 \cdot \text{diam}(K')$
- $\sqrt{\varepsilon}$-net $N$ on $B$
- Closest point on $K'$ for each point in $N$
- $P'$ bounded by tangent hyperplanes
- Unfatten $P'$ into $P$
Dudley's result: [Dud74]

A convex body $K$ of diameter 1 can be $\varepsilon$-approximated by a polytope $P$ with $O(1/\varepsilon^{d-1/2})$ facets.

- Fatten $K$ into $K'$
- Ball $B$ of radius $2 \cdot \text{diam}(K')$
- $\sqrt{\varepsilon}$-net $N$ on $B$
- Closest point on $K'$ for each point in $N$
- $P'$ bounded by tangent hyperplanes
- Unfatten $P'$ into $P$
Dudley’s result: [Dud74]

A convex body \( K \) of diameter 1 can be \( \varepsilon \)-approximated by a polytope \( P \) with \( O\left(\frac{1}{\varepsilon^{\frac{d-1}{2}}}\right) \) facets.

- Fatten \( K \) into \( K' \)
- Ball \( B \) of radius \( 2 \cdot \text{diam}(K') \)
- \( \sqrt{\varepsilon} \)-net \( N \) on \( B \)
- Closest point on \( K' \) for each point in \( N \)
- \( P' \) bounded by tangent hyperplanes
- Unfatten \( P' \) into \( P \)
Dudley’s result: [Dud74]

A convex body $K$ of diameter 1 can be $\varepsilon$-approximated by a polytope $P$ with $O(1/\varepsilon^{d-1/2})$ facets.

- Fatten $K$ into $K'$
- Ball $B$ of radius $2 \cdot \text{diam}(K')$
- $\sqrt{\varepsilon}$-net $N$ on $B$
- Closest point on $K'$ for each point in $N$
- $P'$ bounded by tangent hyperplanes
- Unfatten $P'$ into $P$
Dudley’s result: [Dud74]

A convex body $K$ of diameter 1 can be $\varepsilon$-approximated by a polytope $P$ with $O(1/\varepsilon^{d-1/2})$ facets.

- Fatten $K$ into $K'$
- Ball $B$ of radius $2 \cdot \text{diam}(K')$
- $\sqrt{\varepsilon}$-net $N$ on $B$
- Closest point on $K'$ for each point in $N$
- $P'$ bounded by tangent hyperplanes
- Unfatten $P'$ into $P$
A convex body $K$ is fat if it is sandwiched between balls of radii $r$ and $c \cdot r$ for some constant $c$ that does not depend on $K$.

Fatten by scaling John Ellipsoid to a ball:

John Ellipsoid [Joh48]

For every convex body $K$ in $\mathbb{R}^d$, there exist ellipsoids $E_1, E_2$ such that $E_1 \subseteq K \subseteq E_2$ and $E_2$ is a $d$-scaling of $E_1$. 

Fatness

A convex body $K$ is fat if it is sandwiched between balls of radii $r$ and $c \cdot r$ for some constant $c$ that does not depend on $K$. 

Fatten by scaling John Ellipsoid to a ball:
Fatness and John Ellipsoid

Fatness

A convex body $K$ is fat if it is sandwiched between balls of radii $r$ and $c \cdot r$ for some constant $c$ that does not depend on $K$.

Fatten by scaling John Ellipsoid to a ball:

John Ellipsoid [Joh48]

For every convex body $K$ in $\mathbb{R}^d$, there exist ellipsoids $E_1, E_2$ such that $E_1 \subseteq K \subseteq E_2$ and $E_2$ is a $d$-scaling of $E_1$. 
Minkowski sum of ellipsoids is not an ellipsoid

It follows from John that:

For every convex body $K$ in $\mathbb{R}^d$, there exist rectangles $R_1, R_2$ such that $R_1 \subseteq K \subseteq R_2$ and $R_2$ is a $(3d/2)$-scaling of $R_1$

- Store $R_1(A)$ with $A$
- For $A \oplus B$ use $R_1(A) \oplus R_1(B)$
- $R_1(A) \oplus R_1(B)$ has $O(1)$ vertices
- Fatten $A \oplus B$ scaling $R_1(A) \oplus R_1(B)$ into a fat polytope
Minkowski sum of ellipsoids is not an ellipsoid
It follows from John that:

For every convex body $K$ in $\mathbb{R}^d$, there exist rectangles $R_1, R_2$ such that $R_1 \subseteq K \subseteq R_2$ and $R_2$ is a $(3d/2)$-scaling of $R_1$

- Store $R_1(A)$ with $A$
  - For $A \oplus B$ use $R_1(A) \oplus R_1(B)$
  - $R_1(A) \oplus R_1(B)$ has $O(1)$ vertices
  - Fatten $A \oplus B$ scaling $R_1(A) \oplus R_1(B)$ into a fat polytope
Fattening Minkowski Sums

- Minkowski sum of ellipsoids is not an ellipsoid
- It follows from John that:

\[
\text{For every convex body } K \text{ in } \mathbb{R}^d, \text{ there exist rectangles } R_1, R_2 \text{ such that } R_1 \subseteq K \subseteq R_2 \text{ and } R_2 \text{ is a } (3d/2)\text{-scaling of } R_1
\]

- Store \( R_1(A) \) with \( A \)
- For \( A \oplus B \) use \( R_1(A) \oplus R_1(B) \)
- \( R_1(A) \oplus R_1(B) \) has \( O(1) \) vertices
- Fatten \( A \oplus B \) scaling \( R_1(A) \oplus R_1(B) \) into a fat polytope
Fattening Minkowski Sums

- Minkowski sum of ellipsoids is **not** an ellipsoid
- It follows from John that:

  For every convex body $K$ in $\mathbb{R}^d$, there exist rectangles $R_1, R_2$ such that $R_1 \subseteq K \subseteq R_2$ and $R_2$ is a $(3d/2)$-scaling of $R_1$

- Store $R_1(A)$ with $A$
- For $A \oplus B$ use $R_1(A) \oplus R_1(B)$
- $R_1(A) \oplus R_1(B)$ has $O(1)$ vertices
- Fatten $A \oplus B$ scaling $R_1(A) \oplus R_1(B)$ into a fat polytope
Closest Point

Approximate closest point

Given:
- $K$: preprocessed polytope
- $q$: query point with $\text{dist}(q, K) = \Theta(1)$

Find:
- $p \in K$ with $\|pq\| \leq \text{dist}(q, K) + \varepsilon$

- Binary search
- $O(\log \frac{1}{\varepsilon})$ intersection queries
  $\rightarrow$ approximate closest point
Approximate closest point

Given:
- $K$: preprocessed polytope
- $q$: query point with $\text{dist}(q, K) = \Theta(1)$

Find:
- $p \in K$ with $\|pq\| \leq \text{dist}(q, K) + \varepsilon$

- Binary search
- $O(\log \frac{1}{\varepsilon})$ intersection queries
  $\rightarrow$ approximate closest point
Closest Point

Approximate closest point

Given:
- $K$: preprocessed polytope
- $q$: query point with $\text{dist}(q, K) = \Theta(1)$

Find:
- $p \in K$ with $\|pq\| \leq \text{dist}(q, K) + \varepsilon$

- Binary search
- $O(\log \frac{1}{\varepsilon})$ intersection queries
  $\rightarrow$ approximate closest point
Approximate closest point

Given:

- $K$: preprocessed polytope
- $q$: query point with $\text{dist}(q, K) = \Theta(1)$

Find:

- $p \in K$ with $\|pq\| \leq \text{dist}(q, K) + \varepsilon$

- Binary search
- $O(\log \frac{1}{\varepsilon})$ intersection queries
  $\rightarrow$ approximate closest point
Approximate closest point

Given:
- \( K \): preprocessed polytope
- \( q \): query point with \( \text{dist}(q, K) = \Theta(1) \)

Find:
- \( p \in K \) with \( \|pq\| \leq \text{dist}(q, K) + \varepsilon \)

- Binary search
- \( O(\log \frac{1}{\varepsilon}) \) intersection queries
  \( \rightarrow \) approximate closest point
Minkowski Sum Approximation

- Build **directional width** data structures for \( A \) and \( B \)
- Let \( K = A \oplus B \)
- Run Dudley’s algorithm
  - Fatten using rectangles
  - Answer closest point queries using polytope intersection

(2) Minkowski sum approximation

Time: \( O(n \log \frac{1}{\epsilon} + \frac{1}{\epsilon^{(d-1)/2+\alpha}}) \)

for any \( \alpha > 0 \)
Minkowski Sum Approximation

- Build directional width data structures for $A$ and $B$
- Let $K = A \oplus B$
- Run Dudley’s algorithm
  - Fatten using rectangles
  - Answer closest point queries using polytope intersection

(2) Minkowski sum approximation
Time: $O(n \log \frac{1}{\epsilon} + \frac{1}{\epsilon^{(d-1)/2 + \alpha}})$, for any $\alpha > 0$
Build directional width data structures for $A$ and $B$

Let $K = A \oplus B$

Run Dudley’s algorithm
  - Fatten using rectangles
  - Answer closest point queries using polytope intersection

(2) Minkowski sum approximation

Time: $O(n \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^{(d-1)/2+\alpha}})$, for any $\alpha > 0$
Minkowski Sum Approximation

- Build **directional width** data structures for $A$ and $B$
- Let $K = A \oplus B$
- Run Dudley’s algorithm
  - Fatten using rectangles
  - Answer closest point queries using polytope intersection

### (2) Minkowski sum approximation

Time: $O(n \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^{(d-1)/2+\alpha}})$, for any $\alpha > 0$
Width Approximation

- Compute Dudley of $A \oplus -A$
- Dudley has $O(1/\varepsilon^{(d-1)/2})$ bounding hyperplanes
- Find closest boundary point to the origin naively

(3) Approximate width
Time: $O(n \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^{(d-1)/2+\alpha}})$, for any $\alpha > 0$
Width Approximation

- Compute Dudley of $A \oplus -A$
- Dudley has $O(1/\varepsilon^{(d-1)/2})$ bounding hyperplanes
- Find closest boundary point to the origin naively

(3) Approximate width

Time: $O(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{(d-1)/2 + \alpha})$, for any $\alpha > 0$
Conclusion

Using approximate directional width we solved:

1. Approximate polytope intersection queries in $O(\text{polylog}\frac{1}{\varepsilon})$ time with $O(1/\varepsilon^{(d-1)/2})$ storage
2. Approximation to Minkowski sum in $O(n \log\frac{1}{\varepsilon} + 1/\varepsilon^{(d-1)/2+\alpha})$ time
3. Width approximation in $O(n \log\frac{1}{\varepsilon} + 1/\varepsilon^{(d-1)/2+\alpha})$ time

Open problems:

- Remove the $1/\varepsilon^\alpha$ factor
- Lower bounds (or improved upper bounds): Is $1/\varepsilon^{(d-1)/2}$ necessary?
- Diameter for non-Euclidean metrics
- Approximate the separation depth
Intersection, Minkowski Sum, and Width

Results
Dir. Width
Black Box
Diam vs Width
Minkowski Properties
Origin
Duality
Minimization
d-Dimensional Intersection
Dudley
Fatness
Fattening
Closest Minkowski Apx
Width
Conclusion
Bibliography
Thanks

Bibliography


Thank you!