Economical Delone Sets for Approximating Convex Bodies

Ahmed Abdelkader and David Mount

Department of Computer Science
University of Maryland, College Park

Scandinavian Symposium and Workshops on Algorithm Theory
Malmö, Sweden

June 18th, 2018

Curse of dimensionality

Upper-Bound Theorem [McMullen, 1970], also [Seidel, 1995]

Let $P \subset \mathbb{R}^d$ be a convex polytope.

- If P has n vertices, then the number of faces is $O(n^{\lfloor d/2 \rfloor})$.
- If P has n facets, then the number of vertices is $O(n^{\lfloor d/2 \rfloor})$.

./figs/images/mom1.png

./figs/images/mom2.png

Moment curve: $t \mapsto (t, t^2, \ldots, t^d)$

Figure from [George Papazafeiropoulos]

Curse of dimensionality

Upper-Bound Theorem [McMullen, 1970], also [Seidel, 1995]

Let $P \subset \mathbb{R}^d$ be a convex polytope.

- If P has n vertices, then the number of faces is $O(n^{\lfloor d/2 \rfloor})$.
- If P has n facets, then the number of vertices is $O(n^{\lfloor d/2 \rfloor})$.

Workarounds

Settle for an approximation

- Explicit: find an approximating polytope
- Implicit: approximate membership testing

Polytope Membership Queries

.. implementing a membership oracle

Polytope membership queries

Given a polytope K in $\mathbb{R}^d$, preprocess K to answer membership queries:

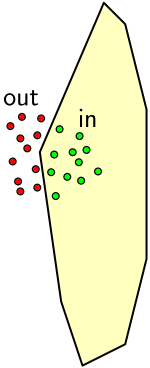
Given a query point $q \in \mathbb{R}^d$, is $q \in K$?

Assumptions

- Dimension d is a constant
- K is given as the intersection of n halfspaces

Dual: Halfspace emptiness queries [Matoušek'92, Erickson'98]

- $d \leq 3 \;\Rightarrow$ Query time: $O(\log n)$ with $O(n)$ storage
- $d \geq 4 \;\Rightarrow$ (roughly) $O(n^{1-1/\lfloor d/2 \rfloor})$ time with $O(n)$ storage
- Restriction $\Rightarrow \tilde{\Omega}(n^{1-1/d})$ time if storage is $\tilde{O}(n)$

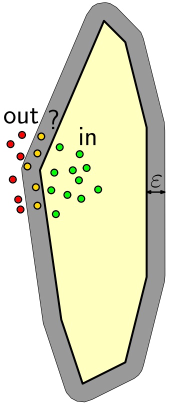


Approximate Polytope Membership Queries

How about an *approximate* membership oracle?

ε-APM queries

- Specify $\varepsilon > 0$ at preprocessing time
- Assume the polytope scaled to unit diameter
- If the query point's distance from K:
  - $0 \Rightarrow$ Inside
  - $> \varepsilon \Rightarrow$ Outside
  - Otherwise: Either answer is acceptable



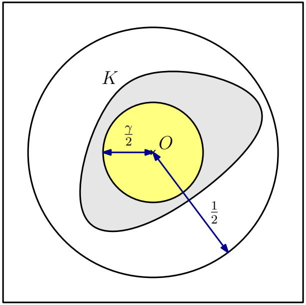
Goal

Query time: $O(\log \frac{1}{\varepsilon})$ ← Logarithmic
Storage: $O(1/\varepsilon^{O(d)})$ ← Efficient?

Canonical Form

Easier to work with fat bodies

In $O(n)$, find an invertible affine transform T s.t. $T(K)$ is in canonical form.



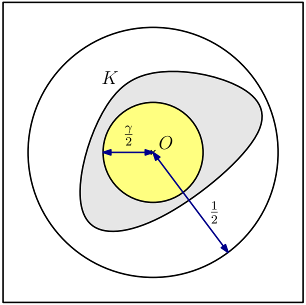
.. and absolute rather than relative errors

If q is a point at distance greater than $\varepsilon \cdot diam(K)$ from K, then $T(q)$ is at distance greater than ε/d from $T(K)$.

Canonical Form

Easier to work with fat bodies

In $O(n)$, find an invertible affine transform T s.t. $T(K)$ is in canonical form.

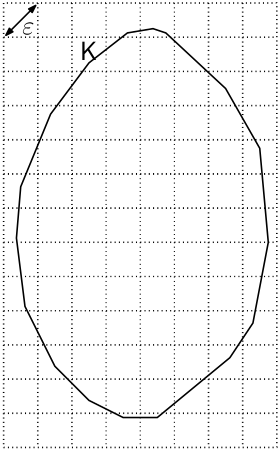


.. and absolute rather than relative errors

If q is a point at distance greater than $\varepsilon \cdot diam(K)$ from K, then $T(q)$ is at distance greater than ε/d from $T(K)$.

Time-efficient Solution [Bentley et al. (1982)]

- Create a grid with cells of diameter ε
- For each column, store the topmost and bottommost cells intersecting K
- Query processing:
    - Locate the column that contains q
    - Compare q with the two extreme values

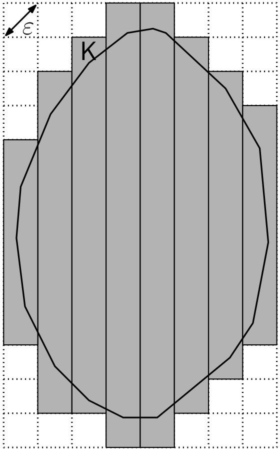


Time-efficient:

- $O(1/\varepsilon^{d-1})$ columns
- Query time: $O(\log \frac{1}{\varepsilon})$ ← optimal
- Storage: $O(1/\varepsilon^{d-1})$ ← high!

Time-efficient Solution [Bentley et al. (1982)]

- Create a grid with cells of diameter ε
- For each column, store the topmost and bottommost cells intersecting K
- Query processing:
  - Locate the column that contains q
  - Compare q with the two extreme values

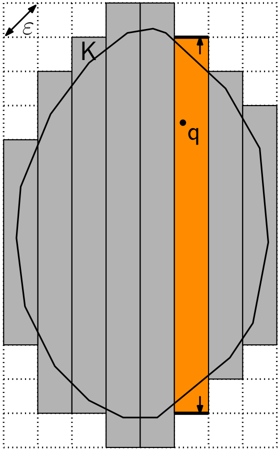


Time-efficient:

- $O(1/\varepsilon^{d-1})$ columns
- Query time: $O(\log \frac{1}{\varepsilon})$ ← optimal
- Storage: $O(1/\varepsilon^{d-1})$ ← high!

Time-efficient Solution [Bentley et al. (1982)]

- Create a grid with cells of diameter ε
- For each column, store the topmost and bottommost cells intersecting K
- Query processing:
  - Locate the column that contains q
  - Compare q with the two extreme values

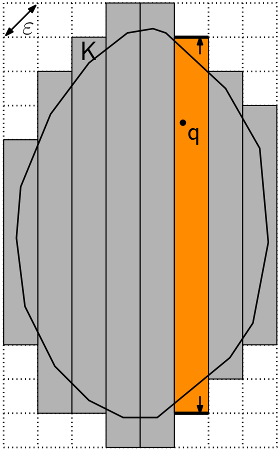


Time-efficient:

- $O(1/\varepsilon^{d-1})$ columns
- Query time: $O(\log \frac{1}{\varepsilon})$ $\leftarrow$ optimal
- Storage: $O(1/\varepsilon^{d-1})$ $\leftarrow$ high!

Time-efficient Solution [Bentley et al. (1982)]

- Create a grid with cells of diameter ε
- For each column, store the topmost and bottommost cells intersecting K
- Query processing:
  - Locate the column that contains q
  - Compare q with the two extreme values

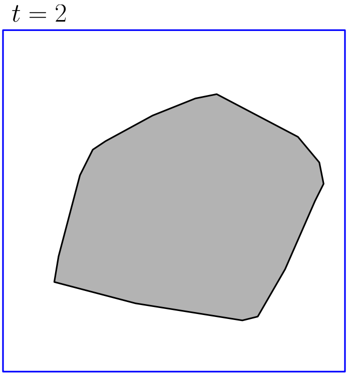


Time-efficient:

- $O(1/\varepsilon^{d-1})$ columns
- Query time: $O(\log \frac{1}{\varepsilon})$ $\leftarrow$ optimal
- Storage: $O(1/\varepsilon^{d-1})$ $\leftarrow$ high!

Space-efficient Solution [Arya et al. (2011, 2012)]

- $Q \leftarrow$ unit hypercube; $t \leftarrow \widetilde{O}(1/\varepsilon^{(d-1)/8})$
- Find an ε-approximation of $Q \cap P$
- If $\leq t$ facets suffice, Q stores them
- Otherwise, subdivide Q and recurse
- Query processing:
  - Locate the cell that contains q
  - Test q against up to t facets



Storage-efficient:

- Query time: $\widetilde{O}(1/\varepsilon^{(d-1)/8})$ $\leftarrow$ high!
- Storage: $O(1/\varepsilon^{(d-1)/2})$ $\leftarrow$ optimal

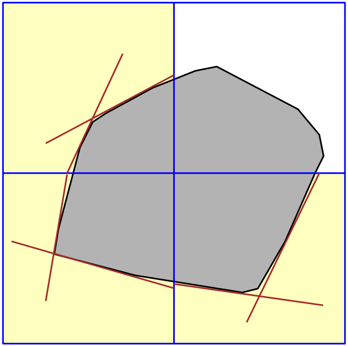
Space-efficient Solution [Arya et al. (2011, 2012)]

- $Q \leftarrow$ unit hypercube; $t \leftarrow \widetilde{O}(1/\varepsilon^{(d-1)/8})$
- Find an ε-approximation of $Q \cap P$
- If $\leq t$ facets suffice, Q stores them
- Otherwise, subdivide Q and recurse
- Query processing:
  - Locate the cell that contains q
  - Test q against up to t facets

Storage-efficient:

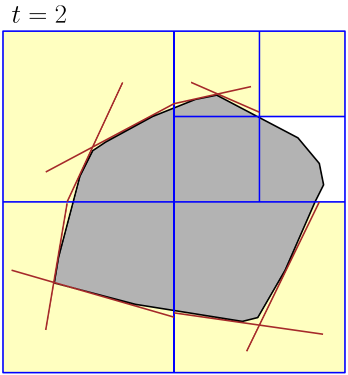
- Query time: $\widetilde{O}(1/\varepsilon^{(d-1)/8})$ $\leftarrow$ high!
- Storage: $O(1/\varepsilon^{(d-1)/2})$ $\leftarrow$ optimal

$t = 2$



Space-efficient Solution [Arya et al. (2011, 2012)]

- $Q \leftarrow$ unit hypercube; $t \leftarrow \widetilde{O}(1/\varepsilon^{(d-1)/8})$
- Find an ε-approximation of $Q \cap P$
- If $\leq t$ facets suffice, Q stores them
- Otherwise, subdivide Q and recurse
- Query processing:
  - Locate the cell that contains q
  - Test q against up to t facets

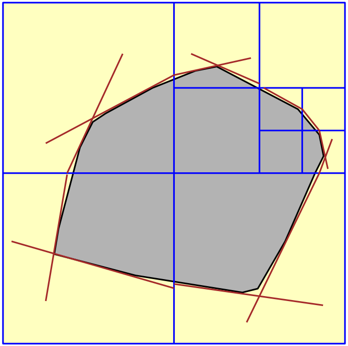


Storage-efficient:

- Query time: $\widetilde{O}(1/\varepsilon^{(d-1)/8})$ $\leftarrow$ high!
- Storage: $O(1/\varepsilon^{(d-1)/2})$ $\leftarrow$ optimal

Space-efficient Solution [Arya et al. (2011, 2012)]

- $Q \leftarrow$ unit hypercube; $t \leftarrow \widetilde{O}(1/\varepsilon^{(d-1)/8})$
- Find an ε-approximation of $Q \cap P$
- If $\leq t$ facets suffice, Q stores them
- Otherwise, subdivide Q and recurse
- Query processing:
    - Locate the cell that contains q
    - Test q against up to t facets



Storage-efficient:

- Query time: $\widetilde{O}(1/\varepsilon^{(d-1)/8})$ $\leftarrow$ high!
- Storage: $O(1/\varepsilon^{(d-1)/2})$ $\leftarrow$ optimal

Digression?

Polytope approximation

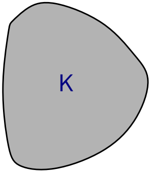
Let $K \subset \mathbb{R}^d$ be a convex body (polytope) of unit diameter. Find P s.t.

- The Hausdorff distance between K and P is at most $\varepsilon > 0$,
- P is a convex polytope having few faces.

Combinatorial Complexity of Approximating Polytopes

Dudley (1974) or Bronshteyn-Ivanov (1976)

Easy construction of P with $O(1/\varepsilon^{(d-1)/2})$ vertices (or facets).



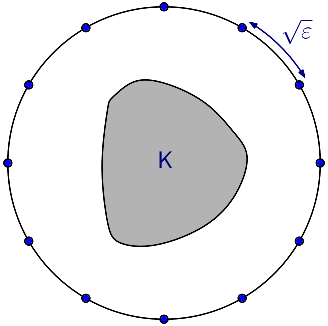
State-of-the-art [Arya et al. 2016]

Constructon of P with $O(1/\hat{\varepsilon}^{(d-1)/2})$ faces, where $\hat{\varepsilon} = \varepsilon / \log(1/\varepsilon)$.

Combinatorial Complexity of Approximating Polytopes

Dudley (1974) or Bronshteyn-Ivanov (1976)

Easy construction of P with $O(1/\varepsilon^{(d-1)/2})$ vertices (or facets).



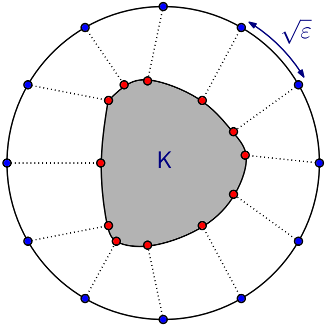
State-of-the-art [Arya et al. 2016]

Constructon of P with $O(1/\hat{\varepsilon}^{(d-1)/2})$ faces, where $\hat{\varepsilon} = \varepsilon / \log(1/\varepsilon)$.

Combinatorial Complexity of Approximating Polytopes

Dudley (1974) or Bronshteyn-Ivanov (1976)

Easy construction of P with $O(1/\varepsilon^{(d-1)/2})$ vertices (or facets).



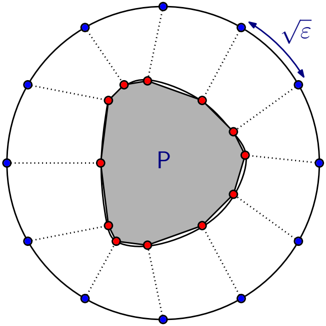
State-of-the-art [Arya et al. 2016]

Constructon of P with $O(1/\hat{\varepsilon}^{(d-1)/2})$ faces, where $\hat{\varepsilon} = \varepsilon / \log(1/\varepsilon)$.

Combinatorial Complexity of Approximating Polytopes

Dudley (1974) or Bronshteyn-Ivanov (1976)

Easy construction of P with $O(1/\varepsilon^{(d-1)/2})$ vertices (or facets).



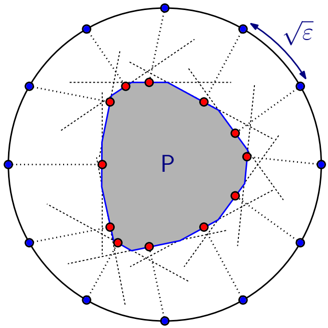
State-of-the-art [Arya et al. 2016]

Constructon of P with $O(1/\hat{\varepsilon}^{(d-1)/2})$ faces, where $\hat{\varepsilon} = \varepsilon / \log(1/\varepsilon)$.

Combinatorial Complexity of Approximating Polytopes

Dudley (1974) or Bronshteyn-Ivanov (1976)

Easy construction of P with $O(1/\varepsilon^{(d-1)/2})$ vertices (or facets).



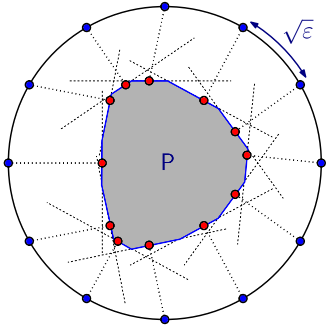
State-of-the-art [Arya et al. 2016]

Constructon of P with $O(1/\hat{\varepsilon}^{(d-1)/2})$ faces, where $\hat{\varepsilon} = \varepsilon / \log(1/\varepsilon)$.

Combinatorial Complexity of Approximating Polytopes

Dudley (1974) or Bronshteyn-Ivanov (1976)

Easy construction of P with $O(1/\varepsilon^{(d-1)/2})$ vertices (or facets).

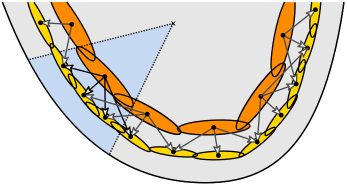


State-of-the-art [Arya et al. 2016]

Constructon of P with $O(1/\hat{\varepsilon}^{(d-1)/2})$ faces, where $\hat{\varepsilon} = \varepsilon / \log(1/\varepsilon)$.

State-of-the-art in ε-APM [Arya et al. (2017)]

- Create a hierarchy of eroded bodies
- For each erosion, cover the boundary with a ring of ellipsoids
- Query processing:
  - Shoot a ray from a central point to q
  - Trace the ray through the rings
  - Follow parent-child links



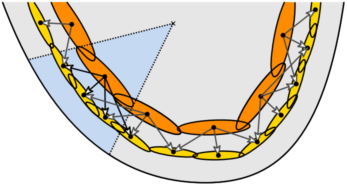
Asymptotically optimal

Query time: $O(\log \frac{1}{\varepsilon})$ ← optimal
Storage: $O(1/\varepsilon^{(d-1)/2})$ ← optimal

- Lower-bound realized by $\mathbb{S}^{d-1}$
- Ray shooting vs. point location

State-of-the-art in ε-APM [Arya et al. (2017)]

- Create a hierarchy of eroded bodies
- For each erosion, cover the boundary with a ring of ellipsoids
- Query processing:
  - Shoot a ray from a central point to q
  - Trace the ray through the rings
  - Follow parent-child links



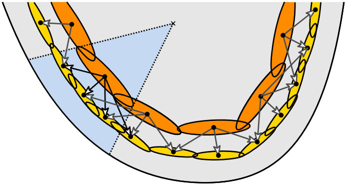
Asymptotically optimal

Query time: $O(\log \frac{1}{\varepsilon})$ $\leftarrow$ optimal
Storage: $O(1/\varepsilon^{(d-1)/2})$ $\leftarrow$ optimal

- Lower-bound realized by $\mathbb{S}^{d-1}$
- Ray shooting vs. point location

State-of-the-art in ε-APM [Arya et al. (2017)]

- Create a hierarchy of eroded bodies
- For each erosion, cover the boundary with a ring of ellipsoids
- Query processing:
    - Shoot a ray from a central point to q
    - Trace the ray through the rings
    - Follow parent-child links



Asymptotically optimal

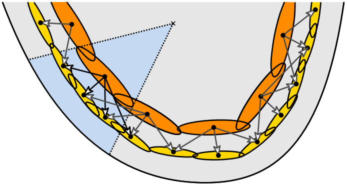
Query time: $O(\log \frac{1}{\varepsilon})$ ← optimal

Storage: $O(1/\varepsilon^{(d-1)/2})$ ← optimal

- Lower-bound realized by $\mathbb{S}^{d-1}$
- Ray shooting vs. point location

State-of-the-art in ε-APM [Arya et al. (2017)]

- Create a hierarchy of eroded bodies
- For each erosion, cover the boundary with a ring of ellipsoids
- Query processing:
  - Shoot a ray from a central point to q
  - Trace the ray through the rings
  - Follow parent-child links



Asymptotically optimal

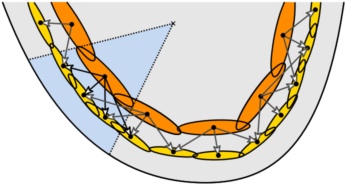
Query time: $O(\log \frac{1}{\varepsilon})$ ← optimal

Storage: $O(1/\varepsilon^{(d-1)/2})$ ← optimal

- Lower-bound realized by $\mathbb{S}^{d-1}$
- Ray shooting vs. point location

State-of-the-art in ε-APM [Arya et al. (2017)]

- Create a hierarchy of eroded bodies
- For each erosion, cover the boundary with a ring of ellipsoids
- Query processing:
  - Shoot a ray from a central point to q
  - Trace the ray through the rings
  - Follow parent-child links



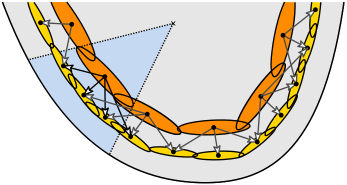
Asymptotically optimal

Query time: $O(\log \frac{1}{\varepsilon})$ $\leftarrow$ optimal
Storage: $O(1/\varepsilon^{(d-1)/2})$ $\leftarrow$ optimal

- Lower-bound realized by $\mathbb{S}^{d-1}$
- Ray shooting vs. point location

State-of-the-art in ε-APM [Arya et al. (2017)]

- Create a hierarchy of eroded bodies
- For each erosion, cover the boundary with a ring of ellipsoids
- Query processing:
    - Shoot a ray from a central point to q
    - Trace the ray through the rings
    - Follow parent-child links



Asymptotically optimal

Query time: $O(\log \frac{1}{\varepsilon})$ ← optimal
Storage: $O(1/\varepsilon^{(d-1)/2})$ ← optimal

- Lower-bound realized by $\mathbb{S}^{d-1}$
- Ray shooting vs. point location

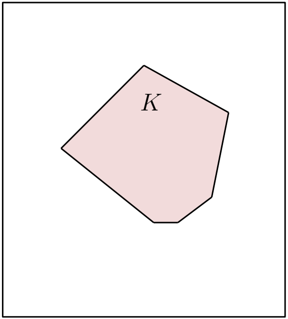
Representational Technicality

A convenient alternative to $K \oplus B(\delta)$

Assuming K is represented as $K = \cap_i H_i$ where each H_i is of the form

$$H_i = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i\}, \text{ where } c_1\delta \leq \|v_i\| \leq c_2,$$

we work with $K_\delta = \cap_i H_{i,\delta}$, where $H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$.



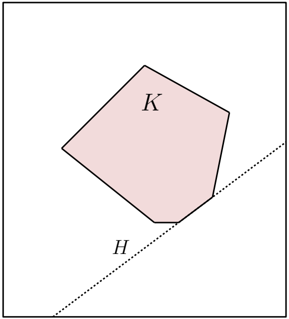
Representational Technicality

A convenient alternative to $K \oplus B(\delta)$

Assuming K is represented as $K = \cap_i H_i$ where each H_i is of the form

$$H_i = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i\}, \text{ where } c_1\delta \leq \|v_i\| \leq c_2,$$

we work with $K_\delta = \cap_i H_{i,\delta}$, where $H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$.



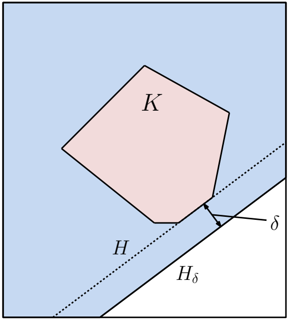
Representational Technicality

A convenient alternative to $K \oplus B(\delta)$

Assuming K is represented as $K = \cap_i H_i$ where each H_i is of the form

$$H_i = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i\}, \text{ where } c_1\delta \leq \|v_i\| \leq c_2,$$

we work with $K_\delta = \cap_i H_{i,\delta}$, where $H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$.



APM by Quadtree Search

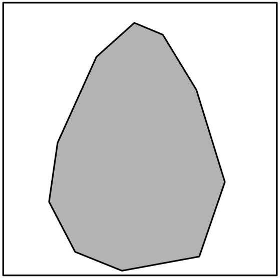
Quadtree-based approach

- Preprocessing: Build a quadtree, subdividing each node that cannot be resolved as being inside or outside
- Stop at diameter ε
- Query: Find the leaf node containing q and return its label

Analysis

Query time: $O(\log \frac{1}{\varepsilon})$ (Quadtree descent)
Storage: $O(1/\varepsilon^{d-1})$ (Number of leaves)

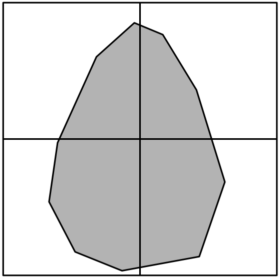
What does it take to beat $O(1/\varepsilon^{d-1})$?



APM by Quadtree Search

Quadtree-based approach

- Preprocessing: Build a quadtree, subdividing each node that cannot be resolved as being inside or outside
- Stop at diameter ε
- Query: Find the leaf node containing q and return its label



Analysis

Query time: $O(\log \frac{1}{\varepsilon})$ (Quadtree descent)
Storage: $O(1/\varepsilon^{d-1})$ (Number of leaves)

What does it take to beat $O(1/\varepsilon^{d-1})$?

APM by Quadtree Search

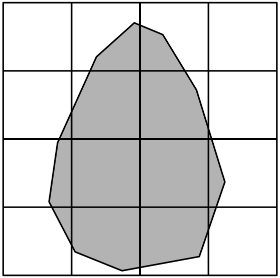
Quadtree-based approach

- Preprocessing: Build a quadtree, subdividing each node that cannot be resolved as being inside or outside
- Stop at diameter ε
- Query: Find the leaf node containing q and return its label

Analysis

Query time: $O(\log \frac{1}{\varepsilon})$ (Quadtree descent)
Storage: $O(1/\varepsilon^{d-1})$ (Number of leaves)

What does it take to beat $O(1/\varepsilon^{d-1})$?



APM by Quadtree Search

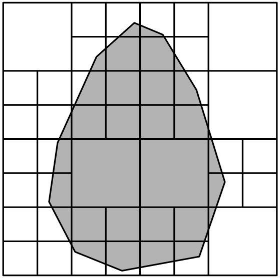
Quadtree-based approach

- Preprocessing: Build a quadtree, subdividing each node that cannot be resolved as being inside or outside
- Stop at diameter ε
- Query: Find the leaf node containing q and return its label

Analysis

Query time: $O(\log \frac{1}{\varepsilon})$ (Quadtree descent)
Storage: $O(1/\varepsilon^{d-1})$ (Number of leaves)

What does it take to beat $O(1/\varepsilon^{d-1})$?



APM by Quadtree Search

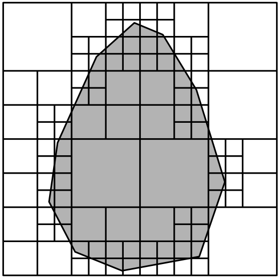
Quadtree-based approach

- Preprocessing: Build a quadtree, subdividing each node that cannot be resolved as being inside or outside
- Stop at diameter ε
- Query: Find the leaf node containing q and return its label

Analysis

Query time: $O(\log \frac{1}{\varepsilon})$ (Quadtree descent)
Storage: $O(1/\varepsilon^{d-1})$ (Number of leaves)

What does it take to beat $O(1/\varepsilon^{d-1})$?



APM by Quadtree Search

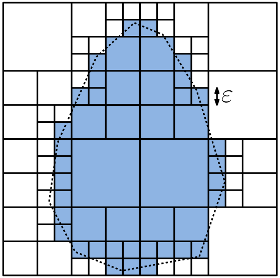
Quadtree-based approach

- Preprocessing: Build a quadtree, subdividing each node that cannot be resolved as being inside or outside
- Stop at diameter ε
- Query: Find the leaf node containing q and return its label

Analysis

Query time: $O(\log \frac{1}{\varepsilon})$ (Quadtree descent)
Storage: $O(1/\varepsilon^{d-1})$ (Number of leaves)

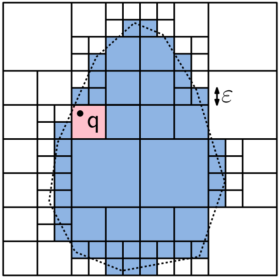
What does it take to beat $O(1/\varepsilon^{d-1})$?



APM by Quadtree Search

Quadtree-based approach

- Preprocessing: Build a quadtree, subdividing each node that cannot be resolved as being inside or outside
- Stop at diameter ε
- Query: Find the leaf node containing q and return its label



Analysis

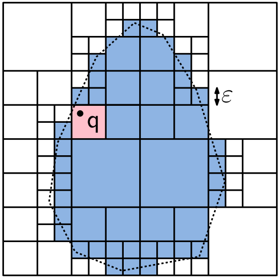
Query time: $O(\log \frac{1}{\varepsilon})$ (Quadtree descent)
Storage: $O(1/\varepsilon^{d-1})$ (Number of leaves)

What does it take to beat $O(1/\varepsilon^{d-1})$?

APM by Quadtree Search

Quadtree-based approach

- Preprocessing: Build a quadtree, subdividing each node that cannot be resolved as being inside or outside
- Stop at diameter ε
- Query: Find the leaf node containing q and return its label



Analysis

Query time: $O(\log \frac{1}{\varepsilon})$ (Quadtree descent)
Storage: $O(1/\varepsilon^{d-1})$ (Number of leaves)

What does it take to beat $O(1/\varepsilon^{d-1})$?

APM by Quadtree Search

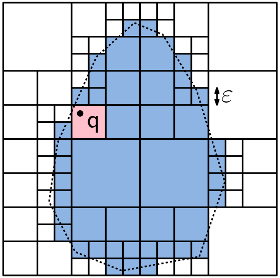
Quadtree-based approach

- Preprocessing: Build a quadtree, subdividing each node that cannot be resolved as being inside or outside
- Stop at diameter ε
- Query: Find the leaf node containing q and return its label

Analysis

Query time: $O(\log \frac{1}{\varepsilon})$ (Quadtree descent)
Storage: $O(1/\varepsilon^{d-1})$ (Number of leaves)

What does it take to beat $O(1/\varepsilon^{d-1})$?



Macbeath Regions

Want cells that conform to K's shape

Macbeath Region [Macbeath (1952)]

Given convex body K, $x \in K$, and $\lambda > 0$:

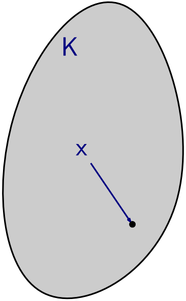
- $M_K(x) = x + (K - x) \cap (x - K)$
  - Intersection of K and its reflection about x
- $M_K^\lambda(x)$: Scaling of $M_K(x)$ by factor λ

Will omit K when clear

John ellipsoid [John (1948)]

May use ellipsoids instead

$$E^\lambda(x) \subseteq M^\lambda(x) \subseteq E^{\lambda\sqrt{d}}(x)$$



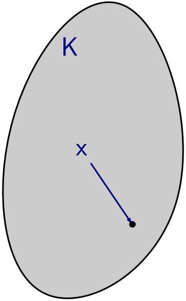
Macbeath Regions

Want cells that conform to K's shape

Macbeath Region [Macbeath (1952)]

Given convex body K, $x \in K$, and $\lambda > 0$:

- $M_K(x) = x + (K - x) \cap (x - K)$
    - Intersection of K and its reflection about x
- $M_K^\lambda(x)$: Scaling of $M_K(x)$ by factor λ



Will omit K when clear

John ellipsoid [John (1948)]

May use ellipsoids instead

$$E^\lambda(x) \subseteq M^\lambda(x) \subseteq E^{\lambda\sqrt{d}}(x)$$

Macbeath Regions

Want cells that conform to K's shape

Macbeath Region [Macbeath (1952)]

Given convex body K, $x \in K$, and $\lambda > 0$:

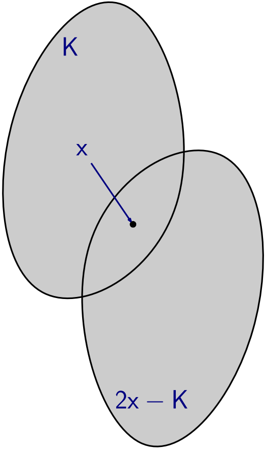
- $M_K(x) = x + (K - x) \cap (x - K)$
  - Intersection of K and its reflection about x
- $M_K^\lambda(x)$: Scaling of $M_K(x)$ by factor λ

Will omit K when clear

John ellipsoid [John (1948)]

May use ellipsoids instead

$$E^\lambda(x) \subseteq M^\lambda(x) \subseteq E^{\lambda\sqrt{d}}(x)$$



Macbeath Regions

Want cells that conform to K's shape

Macbeath Region [Macbeath (1952)]

Given convex body K, $x \in K$, and $\lambda > 0$:

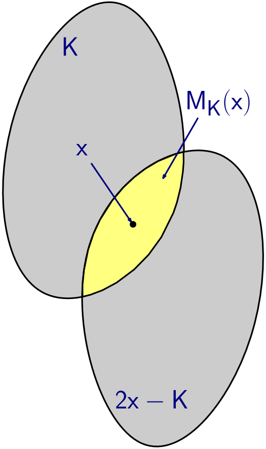
- $M_K(x) = x + (K - x) \cap (x - K)$
  - Intersection of K and its reflection about x
- $M_K^\lambda(x)$: Scaling of $M_K(x)$ by factor λ

Will omit K when clear

John ellipsoid [John (1948)]

May use ellipsoids instead

$$E^\lambda(x) \subseteq M^\lambda(x) \subseteq E^{\lambda\sqrt{d}}(x)$$



Macbeath Regions

Want cells that conform to K's shape

Macbeath Region [Macbeath (1952)]

Given convex body K, $x \in K$, and $\lambda > 0$:

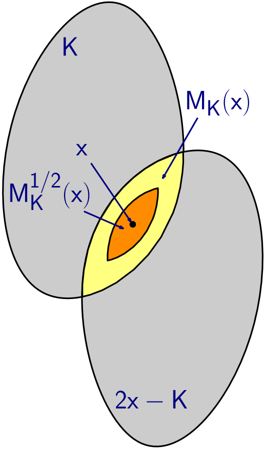
- $M_K(x) = x + (K - x) \cap (x - K)$
    - Intersection of K and its reflection about x
- $M_K^\lambda(x)$: Scaling of $M_K(x)$ by factor λ

Will omit K when clear

John ellipsoid [John (1948)]

May use ellipsoids instead

$$E^\lambda(x) \subseteq M^\lambda(x) \subseteq E^{\lambda\sqrt{d}}(x)$$



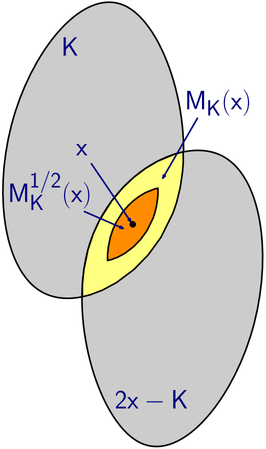
Macbeath Regions

Want cells that conform to K's shape

Macbeath Region [Macbeath (1952)]

Given convex body K, $x \in K$, and $\lambda > 0$:

- $M_K(x) = x + (K - x) \cap (x - K)$
  - Intersection of K and its reflection about x
- $M_K^\lambda(x)$: Scaling of $M_K(x)$ by factor λ



Will omit K when clear

John ellipsoid [John (1948)]

May use ellipsoids instead

$$E^\lambda(x) \subseteq M^\lambda(x) \subseteq E^{\lambda\sqrt{d}}(x)$$

Macbeath Regions

Want cells that conform to K's shape

Macbeath Region [Macbeath (1952)]

Given convex body K, $x \in K$, and $\lambda > 0$:

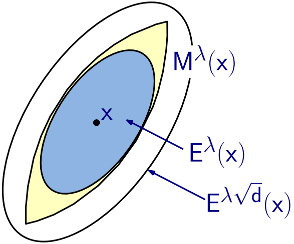
- $M_K(x) = x + (K - x) \cap (x - K)$
  - Intersection of K and its reflection about x
- $M_K^\lambda(x)$: Scaling of $M_K(x)$ by factor λ

Will omit K when clear

John ellipsoid [John (1948)]

May use ellipsoids instead

$$E^\lambda(x) \subseteq M^\lambda(x) \subseteq E^{\lambda\sqrt{d}}(x)$$



Properties of Macbeath Regions

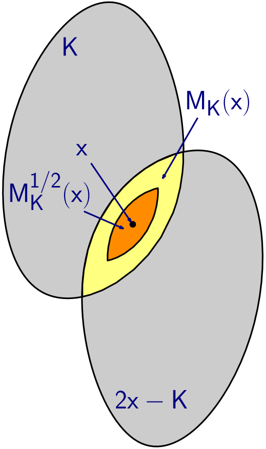
Properties

- $M^{\lambda}(x)$ is convex and symmetric about x
- Expansion-Containment: [Ewald et al (1970)]
  If $M^{\lambda}(x)$ intersects $M^{\lambda}(y)$, with $\lambda < 1$, then

$$M^{\lambda}(y) \subseteq M^{c\lambda}(x), \text{ where } c = \frac{3+\lambda}{1-\lambda}.$$

Good approximation primitives

- Macbeath regions act like metric balls
- But, with respect to which metric?

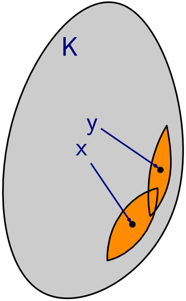


Properties of Macbeath Regions

Properties

- $M^{\lambda}(x)$ is convex and symmetric about x
- Expansion-Containment: [Ewald et al (1970)]
  If $M^{\lambda}(x)$ intersects $M^{\lambda}(y)$, with $\lambda < 1$, then

  $$M^{\lambda}(y) \subseteq M^{c\lambda}(x), \text{ where } c = \frac{3+\lambda}{1-\lambda}.$$



Good approximation primitives

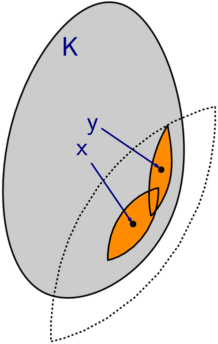
- Macbeath regions act like metric balls
- But, with respect to which metric?

Properties of Macbeath Regions

Properties

- $M^{\lambda}(x)$ is convex and symmetric about x
- Expansion-Containment: [Ewald et al (1970)]
  If $M^{\lambda}(x)$ intersects $M^{\lambda}(y)$, with $\lambda < 1$, then

$$M^{\lambda}(y) \subseteq M^{c\lambda}(x), \text{ where } c = \frac{3+\lambda}{1-\lambda}.$$



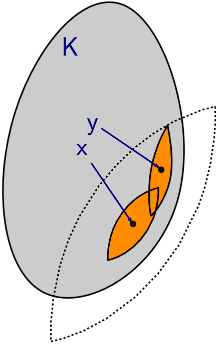
Good approximation primitives

- Macbeath regions act like metric balls
- But, with respect to which metric?

Properties of Macbeath Regions

Properties

- $M^\lambda(x)$ is convex and symmetric about x
- Expansion-Containment: [Ewald et al (1970)]
  If $M^\lambda(x)$ intersects $M^\lambda(y)$, with $\lambda < 1$, then
  $$M^\lambda(y) \subseteq M^{c\lambda}(x), \text{ where } c = \frac{3+\lambda}{1-\lambda}.$$



Good approximation primitives

- Macbeath regions act like metric balls
- But, with respect to which metric?

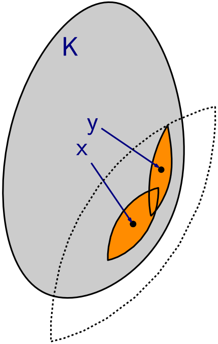
Properties of Macbeath Regions

Properties

- $M^{\lambda}(x)$ is convex and symmetric about x
- Expansion-Containment: [Ewald et al (1970)]
  If $M^{\lambda}(x)$ intersects $M^{\lambda}(y)$, with $\lambda < 1$, then
  $$M^{\lambda}(y) \subseteq M^{c\lambda}(x), \text{ where } c = \frac{3+\lambda}{1-\lambda}.$$

Good approximation primitives

- Macbeath regions act like metric balls
- But, with respect to which metric?



Macbeath Regions and the Hilbert Geometry

From Hilbert geometry [Hilbert (1895)]

- Hilbert Metric: Given $x, y \in K$, define

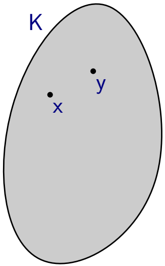
$$f_K(x,y) = \frac{1}{2} \ln \left(\frac{\|x' - y\|}{\|x' - x\|} \frac{\|x - y'\|}{\|y - y'\|} \right)$$

- Hilbert Ball: $B_H(x,\delta) = \{y \in K : f_K(x,y) \leq \delta\}$
- Locally-sensitive to the shape of the convex body

.. to Macbeath regions [Vernicos and Walsh (2016)]

For all $x \in K$ and $0 \leq \lambda < 1$:

$$B_H(x, r_1(\lambda)) \subseteq M^\lambda(x) \subseteq B_H(x, r_2(\lambda))$$



Macbeath Regions and the Hilbert Geometry

From Hilbert geometry [Hilbert (1895)]

- Hilbert Metric: Given $x, y \in K$, define

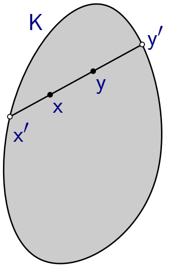
$$f_K(x,y) = \frac{1}{2} \ln \left(\frac{\|x' - y\|}{\|x' - x\|} \frac{\|x - y'\|}{\|y - y'\|} \right)$$

- Hilbert Ball: $B_H(x, \delta) = \{y \in K : f_K(x,y) \leq \delta\}$
- Locally-sensitive to the shape of the convex body

.. to Macbeath regions [Vernicos and Walsh (2016)]

For all $x \in K$ and $0 \leq \lambda < 1$:

$$B_H(x, r_1(\lambda)) \subseteq M^\lambda(x) \subseteq B_H(x, r_2(\lambda))$$



Macbeath Regions and the Hilbert Geometry

From Hilbert geometry [Hilbert (1895)]

- Hilbert Metric: Given $x, y \in K$, define

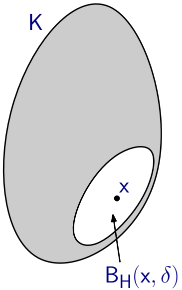
$$f_K(x, y) = \frac{1}{2} \ln \left(\frac{\|x' - y\|}{\|x' - x\|} \frac{\|x - y'\|}{\|y - y'\|} \right)$$

- Hilbert Ball: $B_H(x, \delta) = \{y \in K : f_K(x, y) \leq \delta\}$
- Locally-sensitive to the shape of the convex body

.. to Macbeath regions [Vernicos and Walsh (2016)]

For all $x \in K$ and $0 \leq \lambda < 1$:

$$B_H(x, r_1(\lambda)) \subseteq M^\lambda(x) \subseteq B_H(x, r_2(\lambda))$$



Macbeath Regions and the Hilbert Geometry

From Hilbert geometry [Hilbert (1895)]

- Hilbert Metric: Given $x, y \in K$, define

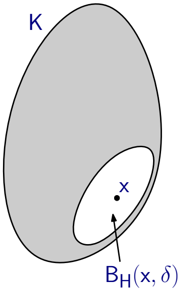
$$f_K(x,y) = \frac{1}{2} \ln \left(\frac{\|x' - y\|}{\|x' - x\|} \frac{\|x - y'\|}{\|y - y'\|} \right)$$

- Hilbert Ball: $B_H(x,\delta) = \{y \in K : f_K(x,y) \leq \delta\}$
- Locally-sensitive to the shape of the convex body

.. to Macbeath regions [Vernicos and Walsh (2016)]

For all $x \in K$ and $0 \leq \lambda < 1$:

$$B_H(x, r_1(\lambda)) \subseteq M^\lambda(x) \subseteq B_H(x, r_2(\lambda))$$



Macbeath Regions and the Hilbert Geometry

From Hilbert geometry [Hilbert (1895)]

- Hilbert Metric: Given $x, y \in K$, define

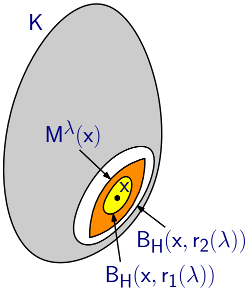
$$f_K(x, y) = \frac{1}{2} \ln \left(\frac{\|x' - y\|}{\|x' - x\|} \frac{\|x - y'\|}{\|y - y'\|} \right)$$

- Hilbert Ball: $B_H(x, \delta) = \{y \in K : f_K(x, y) \leq \delta\}$
- Locally-sensitive to the shape of the convex body

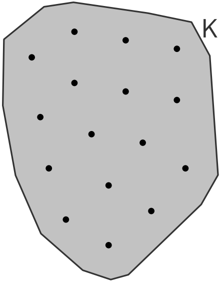
.. to Macbeath regions [Vernicos and Walsh (2016)]

For all $x \in K$ and $0 \leq \lambda < 1$:

$$B_H(x, r_1(\lambda)) \subseteq M^\lambda(x) \subseteq B_H(x, r_2(\lambda))$$



More from Metric Spaces



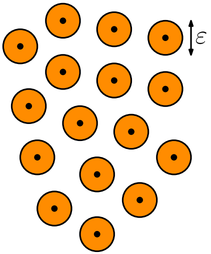
Where to put the Macbeath regions?

Point set layouts in a metric space $\mathbb{X}$

- ε-packing: If the balls of radius $\varepsilon/2$ centered at every point of X are disjoint
- ε-covering: If every point of $\mathbb{X}$ is within distance ε of some point of X
- $(\varepsilon_p, \varepsilon_c)$-Delone Set: If X is an ε_p-packing and an ε_c-covering
- ε-Net: If X is an $(\varepsilon, \varepsilon)$-Delone set

We seek economical Delone sets for K, that fit within an expansion of K by $\delta \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots, \varepsilon\}$

More from Metric Spaces



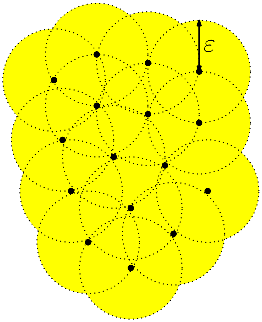
Where to put the Macbeath regions?

Point set layouts in a metric space $\mathbb{X}$

- ε-packing: If the balls of radius $\varepsilon/2$ centered at every point of X are disjoint
- ε-covering: If every point of $\mathbb{X}$ is within distance ε of some point of X
- $(\varepsilon_p, \varepsilon_c)$-Delone Set: If X is an ε_p-packing and an ε_c-covering
- ε-Net: If X is an $(\varepsilon, \varepsilon)$-Delone set

We seek economical Delone sets for K, that fit within an expansion of K by $\delta \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots, \varepsilon\}$

More from Metric Spaces



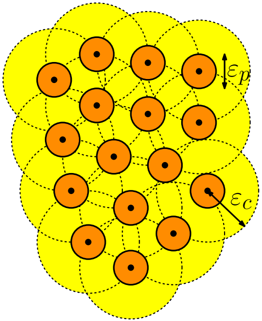
Where to put the Macbeath regions?

Point set layouts in a metric space $\mathbb{X}$

- ε-packing: If the balls of radius $\varepsilon/2$ centered at every point of X are disjoint
- ε-covering: If every point of $\mathbb{X}$ is within distance ε of some point of X
- $(\varepsilon_p, \varepsilon_c)$-Delone Set: If X is an ε_p-packing and an ε_c-covering
- ε-Net: If X is an $(\varepsilon, \varepsilon)$-Delone set

We seek economical Delone sets for K, that fit within an expansion of K by $\delta \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots, \varepsilon\}$

More from Metric Spaces



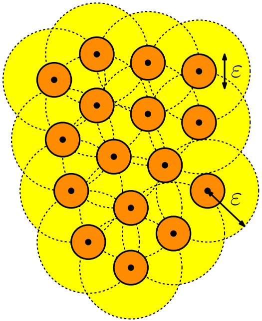
Where to put the Macbeath regions?

Point set layouts in a metric space $\mathbb{X}$

- ε-packing: If the balls of radius $\varepsilon/2$ centered at every point of X are disjoint
- ε-covering: If every point of $\mathbb{X}$ is within distance ε of some point of X
- $(\varepsilon_p, \varepsilon_c)$-Delone Set: If X is an ε_p-packing and an ε_c-covering
- ε-Net: If X is an $(\varepsilon, \varepsilon)$-Delone set

We seek economical Delone sets for K, that fit within an expansion of K by $\delta \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots, \varepsilon\}$

More from Metric Spaces



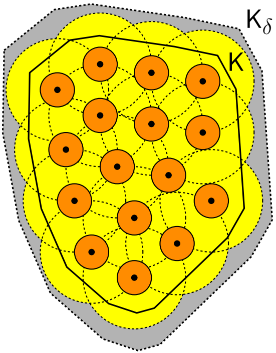
Where to put the Macbeath regions?

Point set layouts in a metric space $\mathbb{X}$

- ε-packing: If the balls of radius $\varepsilon/2$ centered at every point of X are disjoint
- ε-covering: If every point of $\mathbb{X}$ is within distance ε of some point of X
- $(\varepsilon_p, \varepsilon_c)$-Delone Set: If X is an ε_p-packing and an ε_c-covering
- ε-Net: If X is an $(\varepsilon, \varepsilon)$-Delone set

We seek economical Delone sets for K, that fit within an expansion of K by $\delta \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots, \varepsilon\}$

More from Metric Spaces



Where to put the Macbeath regions?

Point set layouts in a metric space $\mathbb{X}$

- ε-packing: If the balls of radius $\varepsilon/2$ centered at every point of X are disjoint
- ε-covering: If every point of $\mathbb{X}$ is within distance ε of some point of X
- $(\varepsilon_p, \varepsilon_c)$-Delone Set: If X is an ε_p-packing and an ε_c-covering
- ε-Net: If X is an $(\varepsilon, \varepsilon)$-Delone set

We seek economical Delone sets for K, that fit within an expansion of K by $\delta \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots, \varepsilon\}$

Macbeath Ellipsoids and Delone Sets

Delone sets from Macbeath ellipsoids

- Let K_δ be an expansion of K by $\delta > 0$
- Let $X_\delta \subset K$ be a maximal set of points such that $E^{\lambda_0}(x)$ are disjoint for all $x \in X_\delta$

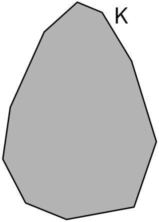
Lemma

$$K \subseteq \bigcup_{x \in X_\delta} E^{\frac{1}{2}}(x) \subseteq K_\delta$$

Macbeath-Based Delone Set

X_δ is essentially a $(\frac{1}{2}, 2\lambda_0)$-Delone set for K

λ_0 is a small constant $1/(4\sqrt{d}+1)$



Macbeath Ellipsoids and Delone Sets

Delone sets from Macbeath ellipsoids

- Let K_δ be an expansion of K by $\delta > 0$
- Let $X_\delta \subset K$ be a maximal set of points such that $E^{\lambda_0}(x)$ are disjoint for all $x \in X_\delta$

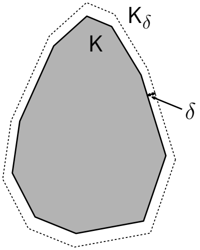
Lemma

$$K \subseteq \bigcup_{x \in X_\delta} E^{\frac{1}{2}}(x) \subseteq K_\delta$$

Macbeath-Based Delone Set

X_δ is essentially a $(\frac{1}{2}, 2\lambda_0)$-Delone set for K

λ_0 is a small constant $1/(4\sqrt{d}+1)$



Macbeath Ellipsoids and Delone Sets

Delone sets from Macbeath ellipsoids

- Let K_δ be an expansion of K by $\delta > 0$
- Let $X_\delta \subset K$ be a maximal set of points such that $E^{\lambda_0}(x)$ are disjoint for all $x \in X_\delta$

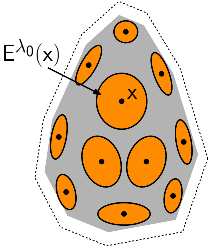
Lemma

$$K \subseteq \bigcup_{x \in X_\delta} E^{\frac{1}{2}}(x) \subseteq K_\delta$$

Macbeath-Based Delone Set

X_δ is essentially a $(\frac{1}{2}, 2\lambda_0)$-Delone set for K

λ_0 is a small constant $1/(4\sqrt{d}+1)$



(Ellipses not drawn to scale)

Macbeath Ellipsoids and Delone Sets

Delone sets from Macbeath ellipsoids

- Let K_δ be an expansion of K by $\delta > 0$
- Let $X_\delta \subset K$ be a maximal set of points such that $E^{\lambda_0}(x)$ are disjoint for all $x \in X_\delta$

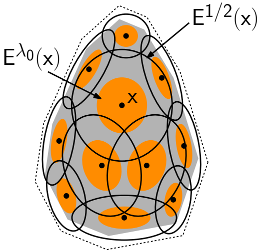
Lemma

$$K \subseteq \bigcup_{x \in X_\delta} E^{\frac{1}{2}}(x) \subseteq K_\delta$$

Macbeath-Based Delone Set

X_δ is essentially a $(\frac{1}{2}, 2\lambda_0)$-Delone set for K

λ_0 is a small constant $1/(4\sqrt{d}+1)$



(Ellipses not drawn to scale)

Macbeath Ellipsoids and Delone Sets

Delone sets from Macbeath ellipsoids

- Let K_δ be an expansion of K by $\delta > 0$
- Let $X_\delta \subset K$ be a maximal set of points such that $E^{\lambda_0}(x)$ are disjoint for all $x \in X_\delta$

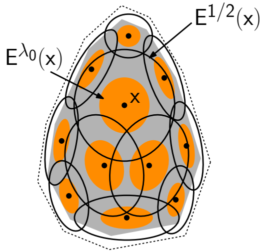
Lemma

$$K \subseteq \bigcup_{x \in X_\delta} E^{\frac{1}{2}}(x) \subseteq K_\delta$$

Macbeath-Based Delone Set

X_δ is essentially a $(\frac{1}{2}, 2\lambda_0)$-Delone set for K

λ_0 is a small constant $1/(4\sqrt{d}+1)$



(Ellipses not drawn to scale)

Macbeath Ellipsoids and Delone Sets

Delone sets from Macbeath ellipsoids

- Let K_δ be an expansion of K by $\delta > 0$
- Let $X_\delta \subset K$ be a maximal set of points such that $E^{\lambda_0}(x)$ are disjoint for all $x \in X_\delta$

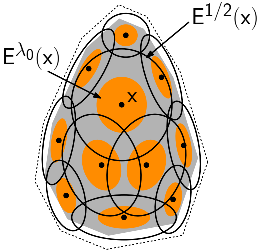
Lemma

$$K \subseteq \bigcup_{x \in X_\delta} E^{\frac{1}{2}}(x) \subseteq K_\delta$$

Macbeath-Based Delone Set

X_δ is essentially a $(\frac{1}{2}, 2\lambda_0)$-Delone set for K

λ_0 is a small constant $1/(4\sqrt{d}+1)$



(Ellipses not drawn to scale)

Expansion-Containment over δ

Lemma

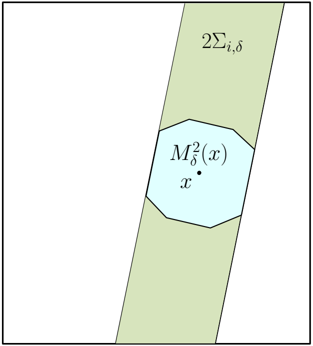
For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M_\delta^2(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where

$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$

- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M_\delta^2(x)$
- Hence, $M_{2\delta}(x) \subseteq M_\delta^2(x)$ $\square$



Expansion-Containment over δ

Lemma

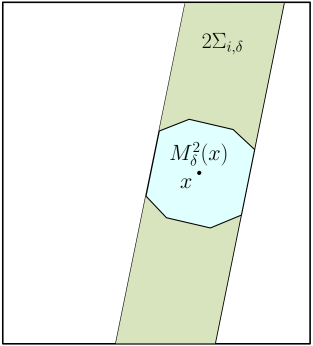
For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M_\delta^2(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where

$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$

- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M_\delta^2(x)$
- Hence, $M_{2\delta}(x) \subseteq M_\delta^2(x)$ □



Expansion-Containment over δ

Lemma

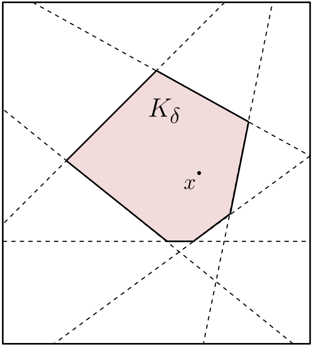
For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M_\delta^2(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where

$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$

- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M_\delta^2(x)$
- Hence, $M_{2\delta}(x) \subseteq M_\delta^2(x)$ $\square$



Expansion-Containment over δ

Lemma

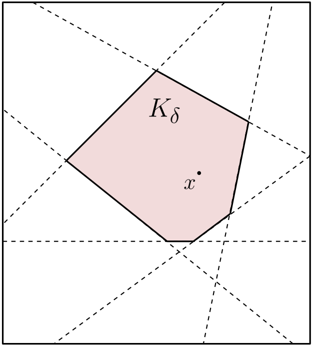
For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M_\delta^2(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where

$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$

- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M_\delta^2(x)$
- Hence, $M_{2\delta}(x) \subseteq M_\delta^2(x)$ □



Expansion-Containment over δ

Lemma

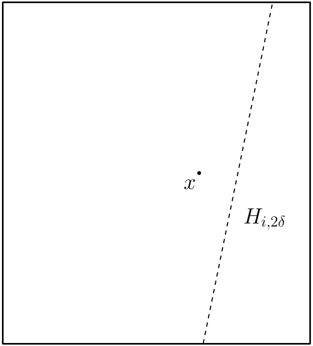
For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M_\delta^2(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where

$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$

- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M_\delta^2(x)$
- Hence, $M_{2\delta}(x) \subseteq M_\delta^2(x)$ $\square$



Expansion-Containment over δ

Lemma

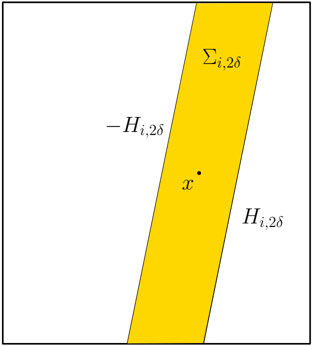
For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M^2_\delta(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where

$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$

- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M^2_\delta(x)$
- Hence, $M_{2\delta}(x) \subseteq M^2_\delta(x)$ □



Expansion-Containment over δ

Lemma

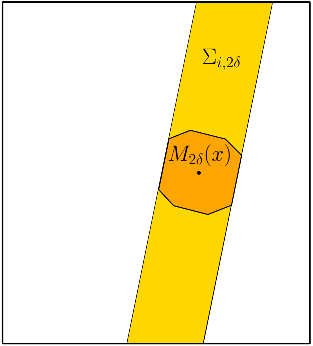
For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M_\delta^2(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where

$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$

- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M_\delta^2(x)$
- Hence, $M_{2\delta}(x) \subseteq M_\delta^2(x)$ $\square$



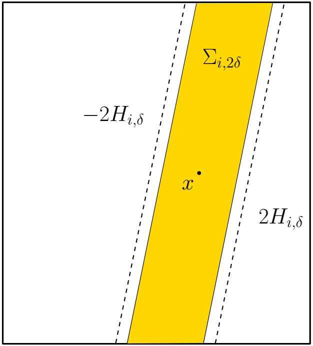
Expansion-Containment over δ

Lemma

For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M_\delta^2(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where
$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$
- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M_\delta^2(x)$
- Hence, $M_{2\delta}(x) \subseteq M_\delta^2(x)$ □



Expansion-Containment over δ

Lemma

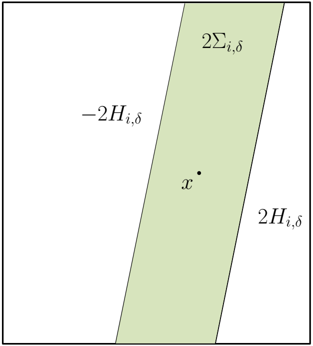
For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M_\delta^2(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where

$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$

- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M_\delta^2(x)$
- Hence, $M_{2\delta}(x) \subseteq M_\delta^2(x)$ $\square$



Expansion-Containment over δ

Lemma

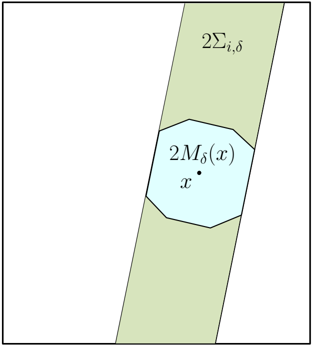
For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M_\delta^2(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where

$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$

- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M_\delta^2(x)$
- Hence, $M_{2\delta}(x) \subseteq M_\delta^2(x)$ $\square$



Expansion-Containment over δ

Lemma

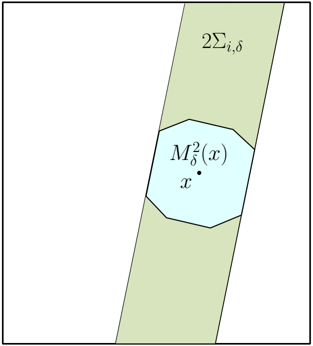
For a convex body $K \subset \mathbb{R}^d$ and $\delta \geq 0$, for all $x \in K$

$$M_\delta(x) \subseteq M_{2\delta}(x) \subseteq M_\delta^2(x).$$

- First inclusion follows from $K_\delta \subset K_{2\delta}$
- Recall $K_\delta = \cap_i H_{i,\delta}$ where

$$H_{i,\delta} = \{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq a_i + \delta\}$$

- Put the origin at x, so $M_\delta(x) = K_\delta \cap -K_\delta$
- Define the slab $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$
- We may write $M_\delta(x) = \cap_i \Sigma_{i,\delta}$
- Define $2H_{i,\delta}$ as $\{x \in \mathbb{R}^d \mid \langle x, v_i \rangle \leq 2(a_i + \delta)\}$
- Observe $\Sigma_{i,2\delta} \subseteq 2\Sigma_{i,\delta} = 2H_{i,\delta} \cap -2H_{i,\delta}$
- But, $\cap_i 2\Sigma_{i,\delta} = 2M_\delta(x)$ is the same as $M_\delta^2(x)$
- Hence, $M_{2\delta}(x) \subseteq M_\delta^2(x)$ $\square$



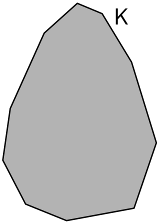
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



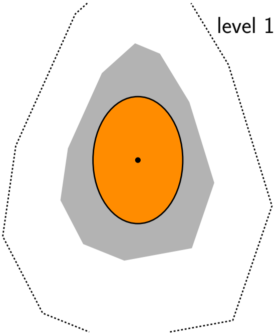
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



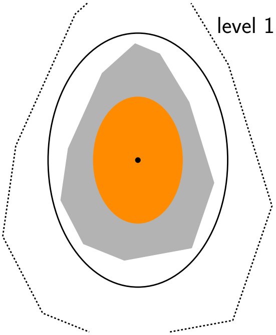
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



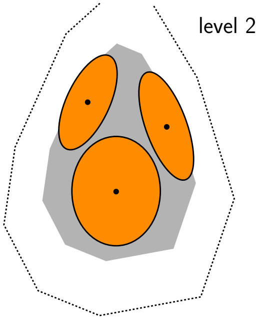
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i-1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



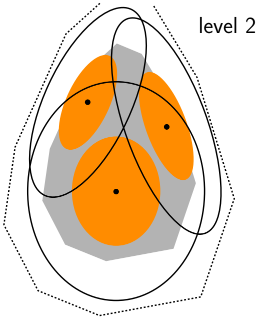
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i-1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



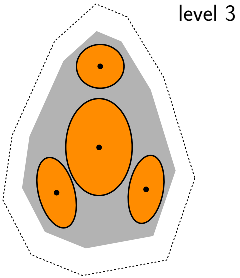
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



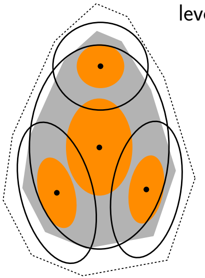
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



APM Data Structure

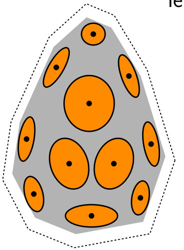
Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"

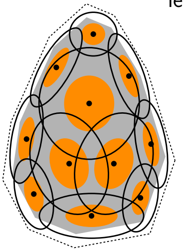
level 4



APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)



Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"

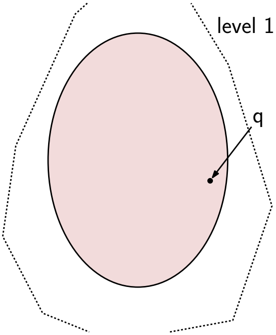
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



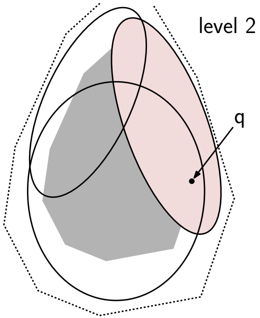
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



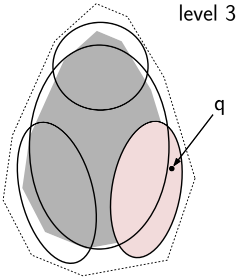
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



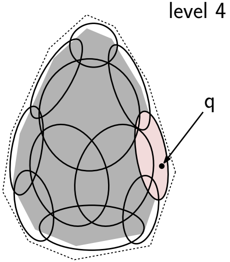
APM Data Structure

Preprocessing

- Input: K and $\varepsilon > 0$
- For $i = 0, 1, \ldots$
  - $\delta_i \leftarrow 2^i \varepsilon$
  - $X_i \leftarrow$ Macbeath Delone set for K_{δ_i}
  - Create a node at level i for each $x \in X_i$
  - Create child links to nodes at level $i - 1$ whose $\frac{1}{2}$-scaled Macbeath ellipsoids overlap
- Stop when $|E_\ell| = 1$ (at $\delta_\ell = O(1)$)

Query Processing

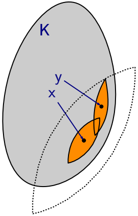
- Descend the DAG from root (level ℓ) until:
  - $q \notin \frac{1}{2}$-scaled child ellipsoids $\Rightarrow$ "outside"
  - Reach leaf $u \Rightarrow$ "inside"



Analysis

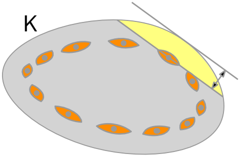
Query time

- Out-degree: $O(1)$ (exp-containment + packing)
- Query time per level: $O(1)$
- Number of levels: $O(\log \frac{1}{\varepsilon})$ (ε to $O(1)$)
- Total query time: $O(\log 1/\varepsilon)$



Storage

- #M-regions to cover K_{δ_i}: $O(1/\delta_i^{(d-1)/2})$ (ECC)
- Storage for bottom level: $O(1/\varepsilon^{(d-1)/2})$
- Leaf level dominates (geometric progression)
- Total storage: $O(1/\varepsilon^{(d-1)/2})$



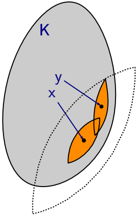
Economical Cap Cover

[Barany and Larman (1989), Arya, de Fonseca, Mount (2016)]

Analysis

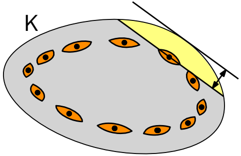
Query time

- Out-degree: $O(1)$ (exp-containment + packing)
- Query time per level: $O(1)$
- Number of levels: $O(\log \frac{1}{\varepsilon})$ (ε to $O(1)$)
- Total query time: $O(\log 1/\varepsilon)$



Storage

- #M-regions to cover K_{δ_i}: $O(1/\delta_i^{(d-1)/2})$ (ECC)
- Storage for bottom level: $O(1/\varepsilon^{(d-1)/2})$
- Leaf level dominates (geometric progression)
- Total storage: $O(1/\varepsilon^{(d-1)/2})$



Economical Cap Cover

[Barany and Larman (1989), Arya, de Fonseca, Mount (2016)]

Summary

Delone sets for convex bodies

- Intuitive and space-efficient cover
- $O(1/\delta^{(d-1)/2})$ ellipsoids for δ-approximation
- Simple DAG hierarchy for time-efficient look-ups
    - Asymptotically optimal

Thanks for listening

Questions?
akader@cs.umd.edu

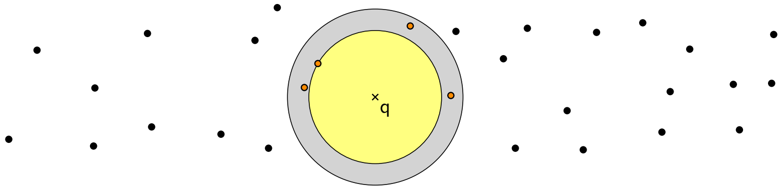
Proximity Search

Approximate Nearest-Neighbor Searching (ANN)

Given a set of n points $P \subset \mathbb{R}^d$, preprocess P into a data structure to such that, given a query point $q \in \mathbb{R}^d$ one can efficiently find a $p \in P$ where

$$\|p - q\| \leq \|nn(q) - q\|,$$

where $nn(q)$ is the *nearest-neighbor* of q in P.



ANN Searching and Polytope Approximation

Lifting and Voronoi Diagrams

Lift the points of P to Ψ, take the upper envelope of the tangent hyperplanes, and project the skeleton back onto the plane. The result is the Voronoi diagram of P.

Intuition: Improved representations of polytopes lead to improvements for ANN