Approximate Nearest Neighbor Searching and Polytope Approximation

David M. Mount

Department of Computer Science
University of Maryland, College Park

(Joint with Sunil Arya and Guilherme da Fonseca)

Presented at the 2017 SCGP Spring School on Discrete and Computational Geometry
Proximity Searching

Proximity searching:
A set of related geometric retrieval problems that involve finding the objects close to a given query object.

Given an \( n \)-element set \( P \) of points in a metric space:

- **Nearest neighbor searching**: Find the closest point of \( P \) to query point \( q \)
- **Range searching**: Count/report points of \( P \) lying in a bounded range \( Q \)

We assume a vector space of low-dimension with the Euclidean metric.
Proximity Searching: Applications

- Pattern recognition and classification
- Object recognition in images
- Content-based retrieval:
  - Shape matching
  - Image retrieval
  - Document retrieval
  - Biometric identification (face/fingerprint/voice recognition)
- Clustering and phylogeny
- Data compression (vector quantization)
- Physical simulation (collision detection and response)
- Computer graphics: photon mapping and point-based modeling

...and many more
Overview

- Nearest Neighbor Searching
- Approximate Nearest Neighbor Searching
- Polytope Approximation
  - Simple Trade-off
  - Split-Reduce
  - Back to ANN
- Area-Sensitive Approximation
- Conclusions
Nearest Neighbor Searching

Space
- Real $d$-dimensional space, $\mathbb{R}^d$
- Each point given by its coordinate vector $p = (p_1, \ldots, p_d)$

Nearest Neighbor Problem
Given a set $P \subset \mathbb{R}^d$, preprocess $P$, so that given any query point $q \in \mathbb{R}^d$, can efficiently compute $p \in P$ that minimizes the distance $\text{dist}(p, q)$.

Assumptions
- Dimension is low (constant, independent of $n$)
- Assume Euclidean distance: $\text{dist}(p, q) = \|pq\| = \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}$
Ideal: $O(n)$ space and $O(\log n)$ query time

Voronoi Diagrams
- Subdivide space into regions according to which point is closest
- Apply point location to answer queries

In the plane, $O(n)$ space and $O(\log n)$ query time

...but in higher dimensions
- In dimension $d$, Voronoi diagram has worst-case complexity $O(n^{\lceil d/2 \rceil})$
- Point location is not well solved if $d \geq 3$
Approximate Nearest Neighbor

This has motivated approximate solutions:

**Approximate Nearest Neighbor**

Given a query point $q$, whose true nearest neighbor is $p^*$, return any point $p \in P$, such that

$$\|pq\| \leq (1 + \varepsilon)\|p^*q\|$$
Overview

- Nearest Neighbor Searching
- **Approximate Nearest Neighbor Searching**
- Polytope Approximation
  - Simple Trade-off
  - Split-Reduce
  - Back to ANN
- Area-Sensitive Approximation
- Conclusions
Balanced Box-Decomposition (BBD) Tree

- **Cell**: Difference of two quadtree boxes (inner and outer)
- **Centroid Decomposition**: Used to guarantee $O(\log n)$ depth
Nearest Neighbor Searching with BBD trees

**ε-NN Searching with BBD trees**

- **Preprocessing:** $O(n)$ space
- **Query Processing:**
  - Locate the cell containing $q$ ($O(\log n)$ time)
  - Establish initial search radius
  - Recursively visit nodes only if they are close enough to offer a closer point
- **Query time:** $O(\log n + (1/\varepsilon)^d)$
Approximate Voronoi Diagrams

**Trade-offs:** More space but lower query times?

**Approximate Voronoi Diagram (AVD)**
- Quadtree subdivision into cells
- Each cell stores a representative, \( r \in P \), such that \( r \) is an \( \varepsilon \)-ANN of any point \( q \) in the cell

**Har-Peled (2001)**
Given a set of \( n \) points in \( \mathbb{R}^d \), \( \varepsilon \)-approximate nearest neighbor queries can be answered in space \( \tilde{O}(n/\varepsilon^d) \) and in time \( O(\log(n/\varepsilon)) \)
Approximate Nearest Neighbor Searching

Approximate Voronoi Diagrams

**Trade-offs:** If we allow multiple reps, can we decrease space?

**Multi-Rep AVDs [Arya, Malamatos (2002)]**

- Quadtree subdivision into cells
- Each cell stores up to $t$ representatives, $\{r_1, \ldots, r_t\} \in P$
- Given any point $q$ in the cell, at least one rep is an $\varepsilon$-ANN of $q$

By adjusting $t$, it is possible to trade off space and query time
Separation Properties [AMM (2009)]

Given \( P \subset \mathbb{R}^d \) and \( \gamma \geq 2 \), can partition space into \( \tilde{O}(n\gamma^d) \) cells, such that for each cell \( Q \), all the points of \( P \) are \( \gamma \cdot \text{diam}(Q) \) far from \( Q \) except either:

- A single point in \( Q \)
- A \((1/\varepsilon)\)-separated cluster of points

Can select \( O((1/\varepsilon\gamma)^{(d-1)/2}) \) representatives, one of which is an \( \varepsilon \)-ANN to any point of \( Q \).

By adjusting \( \gamma \), can achieve a trade-off between space and query time.
Approximate Nearest Neighbor Searching

Approximate Voronoi Diagrams

Space-Time Trade-offs [AMM (2009)]

For any $\gamma \geq 2$, $\varepsilon$-NN queries can be answered in query time roughly $O(1/\varepsilon^{d/\gamma})$ with storage roughly $O(n/\varepsilon^{d(1-2/\gamma)})$

Lower bounds were also established in the AVD model

- This is optimal in the extremes
- Can we close the gap?
- Which of two bounds (upper or lower) is likely to be the “truth”?
Overview

- Nearest Neighbor Searching
- Approximate Nearest Neighbor Searching
- Polytope Approximation
  - Simple Trade-off
  - Split-Reduce
  - Back to ANN
- Area-Sensitive Approximation
- Conclusions
ANN Searching and Polytope Approximation

Lifting and Distances
- Project a point \( p \) vertically to \( p^\uparrow \) on a paraboloid \( \Psi \)
- Let \( h \) be the tangent hyperplane at \( p^\uparrow \)
- For any point \( q \) at distance \( \delta \) from \( p \), the vertical distance between \( \Psi \) and \( h \) is \( \delta^2 \)

Lifting and Voronoi Diagrams
- Lift the points of \( P \) vertically to \( \Psi \)
- Intersect their tangent upper halfspaces
- The projected skeleton of the resulting polytope is the Voronoi diagram of \( P \)
Lifting and Voronoi Diagrams

Lift the points of $P$ to $\Psi$, take the upper envelope of the tangent hyperplanes, and project the skeleton back onto the plane. The result is the Voronoi diagram of $P$. 
Polytope Membership Queries

Given a polytope $P$ in $d$-dimensional space, preprocess $P$ to answer membership queries:

Given a point $q$, is $q \in P$?

- Assume that dimension $d$ is a constant and $P$ is given as intersection of $n$ halfspaces.
Approximate Version

- An approximation parameter $\varepsilon$ is given (at preprocessing time)
- Assume the polytope has diameter 1
- If the query point's distance from $P$'s boundary:
  - $> \varepsilon$: answer must be correct
  - $\leq \varepsilon$: either answer is acceptable
Approximate nearest neighbor searching can be reduced to approximate polytope membership:

- Recall that ($\gamma = 2$) AVDs partition space into cells, each associated with representatives, such that:
  - Total number of representatives over all cells is roughly $O(n)$
  - All but one representative is inside a constant-radius annulus
- Through lifting, we can reduce the nearest neighbor search to a small number of approximate polytope membership queries
Dudley’s (Outer) Approximation

Every unit-diameter polytope can be $\varepsilon$-approximated as the intersection of $O(1/\varepsilon^{(d-1)/2})$ halfspaces [Dudley (1974)]

Space-Efficient Solution

Check whether $q$ lies within each halfspace:

- Storage: $O(1/\varepsilon^{(d-1)/2})$
- Query time: $O(1/\varepsilon^{(d-1)/2})$
- Note: Each halfspace is used to cover a surface patch of size $\sqrt{\varepsilon}$
A Simple Trade-off

- Generate a grid of diameter $r \in [\varepsilon, 1]$

- **Preprocessing:** For each cell $Q$ intersecting $P$’s boundary:
  - Apply Dudley to $P \cap Q$
  - $O((r/\varepsilon)^{(d-1)/2})$ halfspaces per cell

- **Query Processing:**
  - Find the cell containing $q$
  - Check whether $q$ lies within every halfspace for this cell

---

**Trade-off (Rephrased: $r = \varepsilon^{1-2\alpha}$)**

- **Storage:** $O(1/\varepsilon^{(d-1)(1-1/\alpha)})$
- **Query time:** $O(1/\varepsilon^{(d-1)/\alpha})$
Can we do better? Need a little sensitivity

- Dudley tends to **oversample** regions of very low and very high curvature
- Finding the smallest number of halfspaces reduces to **set cover**
- A $\log(1/\varepsilon)$-approximation can be found efficiently [Mitchell and Suri (2009) and Clarkson (1993)]

**Simple Idea:** Recursively subdivide space (quadtree) until the number of approximating halfspaces is **small enough**
Preprocess:
- Input $P$, $\varepsilon$, and desired query time $t$
- $Q \leftarrow$ unit hypercube
- Split-Reduce($Q$)

Split-Reduce($Q$)
- Find an $\varepsilon$-approximation of $Q \cap P$
- If at most $t$ facets, then $Q$ stores them
- Otherwise, subdivide $Q$ and recurse

Query time: $O(\log(1/\varepsilon) + t)$
Storage: ???
**Split-Reduce**

\[ t = 2 \]

Preprocess:
- Input \( P, \varepsilon \), and desired query time \( t \)
- \( Q \leftarrow \) unit hypercube
- Split-Reduce(\( Q \))

Split-Reduce(\( Q \))
- Find an \( \varepsilon \)-approximation of \( Q \cap P \)
- If at most \( t \) facets, then \( Q \) stores them
- Otherwise, subdivide \( Q \) and recurse

Query time: \( O(\log(1/\varepsilon) + t) \)
Storage: ???
Split-Reduce

**Preprocess:**
- Input $P$, $\varepsilon$, and desired query time $t$
- $Q \leftarrow$ unit hypercube
- Split-Reduce($Q$)

**Split-Reduce($Q$)**
- Find an $\varepsilon$-approximation of $Q \cap P$
- If at most $t$ facets, then $Q$ stores them
- Otherwise, subdivide $Q$ and recurse

- Query time: $O(\log(1/\varepsilon) + t)$
- Storage: ???
Preprocess:
- Input $P$, $\varepsilon$, and desired query time $t$
- $Q \leftarrow$ unit hypercube
- Split-Reduce($Q$)

Split-Reduce($Q$)
- Find an $\varepsilon$-approximation of $Q \cap P$
- If at most $t$ facets, then $Q$ stores them
- Otherwise, subdivide $Q$ and recurse

Query time: $O(\log(1/\varepsilon) + t)$
Storage: ???
**General Trade-off**

**Space-Time Trade-off [AFM (2011)]**

Using Split-Reduce we can answer $\epsilon$-approximate polytope membership queries with

**Storage:** $O\left(\frac{1}{\epsilon^{(d-1)/(1-k/2^k)}}\right)$

**Query time:** $O\left(\frac{1}{\epsilon^{(d-1)/2^k}}\right)$

![Graph showing trade-offs for polytope membership](image)
Approximate Nearest Neighbor (ANN) Searching

Recall the AVD-based space-time trade-off [AMM (2009)]

Through the use of the split-reduce data structure, the trade-off improves throughout the spectrum
Approximate Nearest Neighbor (ANN) Searching

Recall the AVD-based space-time trade-off [AMM (2009)]

Through the use of the split-reduce data structure, the trade-off improves throughout the spectrum
Overview

- Nearest Neighbor Searching
- Approximate Nearest Neighbor Searching
- Polytope Approximation
  - Simple Trade-off
  - Split-Reduce
  - Back to ANN
- Area-Sensitive Approximation
- Conclusions
Better polytope approximation can lead to faster nearest-neighbor searching

**Better Approximations for Skinny Bodies [AFM (2012)]**

A convex body $K$ can be $\varepsilon$-approximated by a polytope $P$ with

$$\tilde{O}(\sqrt{\text{area}(K)}/\varepsilon^{(d-1)/2})$$

facets (alternatively, vertices).

- Uses area instead of diameter
- Matches Dudley’s bound up to a log factor when the body is fat
- Significant improvement for skinny bodies
- Analysis uses several new techniques for the problem (polarity, Mahler volume, $\varepsilon$-nets...
The Mahler Volume

- \( K \): convex body
- Polar body of \( K \): set of points \( p \) such that \( p \cdot q \leq 1 \) for \( q \in K \)
- Mahler volume of \( K \): product of the volume of \( K \) and the volume of \( \text{polar}(K) \)

Important for us:
The Mahler volume of \( K \) is bounded below by a constant [Kuperberg (2008)]
The Mahler Volume

We show that:

An $\epsilon$-dual cap $D$ and its Voronoi patch are related in a manner that is similar to the polar transform (up to an $\epsilon$-scaling).

Using the fact that the Mahler volume is at least a constant:

**Key lemma:**

For any $\epsilon$-dual cap $D$, the product of $\text{area}(D)$ and $\text{area}(\text{Vor}(D) \cap S)$ is $\Omega(\epsilon^{d-1})$.

Less formally: If $D$ has small area, then its Voronoi patch is large.
Improved trade-off for approximate polytope membership queries

**New bounds:**

For integer $k \geq 2$, we can answer $\varepsilon$-approximate polytope membership queries with:

- **Storage:** $O\left(\frac{1}{\varepsilon^{(d-1)/(1-k/2^k)}}\right)$
- **Query time:** $O\left(\frac{1}{\varepsilon^{(d-1)/2^{k+1} \log(1/\varepsilon)}}\right)$

- For the same storage, the query time is reduced to roughly the square root
- Leads to improved approximate nearest neighbor data structures

**Storage:** $O\left(\frac{1}{\varepsilon^{(d-1)/2}}\right)$

**Query time:** $\tilde{O}\left(\frac{1}{\varepsilon^{(d-1)/8}}\right)$
Approximate Nearest Neighbor (ANN) Searching

This results in improved query time for ANN searching.

\[ y: \text{Query time is } O(\log n) + \frac{1}{\varepsilon y(d-\Theta(1))} \]

\[ x: \text{Storage is } \frac{n}{\varepsilon^x(d-\Theta(1))} \]
Concluding Remarks

- Improved upper bounds for approximate polytope membership queries
- Space-time trade-offs
- Simple algorithm – Split-Reduce
- Area-sensitive polytope approximation
- Significant improvements to ANN searching

Open Problems:

- Generalizations to other Minkowski metrics (lifting fails)
- Better approximations to polytope covering (eliminating log factors)
Bibliography