Approximate Nearest Neighbor Searching

Sunil Arya
Department of Computer Science and Engineering
HKUST

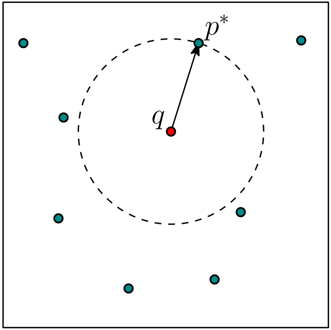
Nearest Neighbor Searching

Nearest Neighbor Searching

Given a set S of n data points, preprocess S into a data structure so that given a query point q, the data point p^* closest to q can be found quickly.

Assumptions and Goals

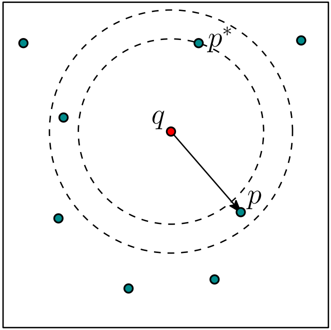
- Dimension d is constant.
- Euclidean metric.
- Desire $O(n)$ space and $O(\log n)$ query time.



Approximate Nearest Neighbor Searching

Approximate Nearest Neighbor (ANN) Searching

Given query point q and $\varepsilon > 0$, return a data point p whose distance from q is no more than $(1 + \varepsilon)$ times the distance from q to its nearest neighbor p^*. We call p an ε-approximate nearest neighbor of q.



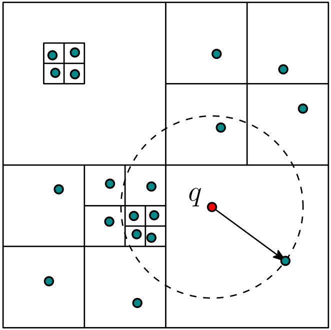
Balanced Box-Decomposition (BBD) Tree

Data Structure

The tree combines the balance of k-d trees with the fatness properties of quadtrees.

Theoretical Result (Arya et al. '98)

- Space: $O(n)$
- Query Time: $O(\log n + (1/\varepsilon)^d)$



- Can the ε-dependencies be reduced, albeit at the expense of using more space?
- Can we achieve tradeoffs between space and query time?

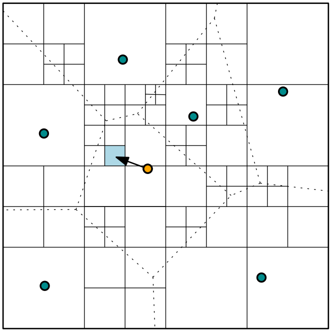
Approximate Voronoi Diagrams

Data Structure

- Quadtree-like subdivision of space.
- Each cell stores a representative $r \in S$ such that r is an ε-ANN of any point q in the cell.

Theoretical Result (Har-Peled '01)

- Space: $O(n \cdot (1/\varepsilon)^d)$
- Query Time: $O(\log n + \log(1/\varepsilon))$



Space-Time Tradeoffs

Results

Method	Space	Time	Space x Time
BBD-Trees (Arya et al. '98)	n	$\left(\frac{1}{\varepsilon}\right)^{d}$	$n \cdot \left(\frac{1}{\varepsilon}\right)^{d}$
BBD-Trees + Cones (Clarkson '94, Chan '98)	$n \cdot \left(\frac{1}{\varepsilon}\right)^{d/2}$	$\left(\frac{1}{\varepsilon}\right)^{d/2}$	$n \cdot \left(\frac{1}{\varepsilon}\right)^{d}$
AVDs (Har-Peled '01)	$n \cdot \left(\frac{1}{\varepsilon}\right)^{d}$	1	$n \cdot \left(\frac{1}{\varepsilon}\right)^{d}$

(Ignoring small factors)

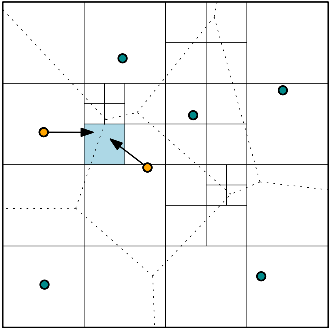
Approximate Voronoi Diagrams: Multiple Representatives

Data Structure

- Each cell is allowed up to $t \geq 1$ representatives.
- Given any point q in the cell, at least one of the representatives is an ε-ANN of q.
- We can achieve space-time tradeoffs by adjusting t.

Theoretical Result (Arya et al. '09)

- Space: $O(n)$
- Query Time: $O((1/\varepsilon)^{d/2})$



Conclusions

- We achieve continuous space-time tradeoffs.
- We break the $n \cdot (1/\varepsilon)^d$ space-time product barrier.
- Finding the best space-time tradeoffs is still an open problem.