Approximate Nearest Neighbors via Point Location in Balls

- **Nearest Neighbor Problem**: Given a set of n points, P in a metric space M; preprocess it to be able to determine the closest point in P, to a query point $q \in M$.

Requires prohibitive pre-processing time. Need to compute Voronoi diagrams and process for point location queries. $O(n^{\lceil d/2 \rceil})$ time.

- So we ..... Approximate !

Let $nn(q, P)$ be said nearest neighbor, and $d(q, P)$ be the distance between q and $nn(q, P)$.

A point $s \in M$ is a $(1+\varepsilon)$-approximate nearest neighbor if $d_M(q, s) \leq (1+\varepsilon) \cdot d(q, P)$.

- Ball is defined by its center p & radius r.

$$b(p, r) = \{ q \in M \mid d_M(p, q) \leq r \}$$

- Target Ball of q in a given set B, $\odot_B(q)$, is the smallest ball in B that contains the query q.

- Objective: Show that $(1+\varepsilon)$-ANN queries can be reduced to target ball queries on a "small" # of balls.

- Notation: Union of balls of radius r, centered at points of P $\quad \mathcal{U}_{balls}(P, r) := \bigcup_{p \in P} b(p, r)$

⇒ Quadratic # of balls

- Lemma 841: Given P, q in M. Let $B = \bigcup_{i=-\infty}^{\infty} \mathcal{U}_{balls}(P, (1+\varepsilon)^i)$.
  Let $b = \mathcal{O}_B(q)$ be the target ball and $p \in P$ its center.
  Then p is $(1+\varepsilon)$-ANN in P to q.

Pf: Let $s = nn(q, P)$ be the nearest neighbor, and i be the index such that

$$r_1 := (1+\varepsilon)^i < d_M(s,q) \le (1+\varepsilon)^{i+1} =: r_2$$

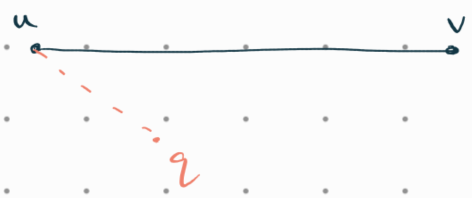
No ball of radius r_1 can contain the query, but $\exists$ a ball $b(s, (1+\varepsilon)^{i+1})$ that does contain it. So, the target ball must have radius $\le (1+\varepsilon)^{i+1}$.

$$\alpha = \frac{r(\text{target ball})}{r(\text{OPT ball})} \le \frac{(1+\varepsilon)^{i+1}}{(1+\varepsilon)^i} \le 1+\varepsilon \qquad \blacksquare$$

So, is the objective achieved? Nope ∞ balls is not "small number" by any measure.

Consider two points u, v.

Case 1 : $d_M(q,u) \ge \frac{2 d_M(u,v)}{\varepsilon}$



Case Close(d) : $d_M(q,u) \le d_M(u,v)/4$

Intuitively, only $d(q, \{u,v\}) \in \overbrace{[1/4, 2/\varepsilon] \cdot d_M(u,v)}^{\mathcal{I}(u,u)}$ hard to decide.

→ Construction.

$B(u,v) = \{ b(u,r), b(v,r) \mid r = (1+\varepsilon)^i \in Z(u,v) \text{ and } i \in \mathbb{Z} \}$.
For every pair $u,v \in P$, add balls $B(u,v)$ to the set B being constructed.

→ Size $\in O(n^2 \varepsilon^{-1} \log \varepsilon^{-1})$.

Ratio between high and low values

$$R = \frac{2d_M(u,v)}{\varepsilon} \Big/ \frac{d_M(u,v)}{4} \in O(1/\varepsilon)$$

Total no. of balls is,

$$O(n^2 \log_{1+\varepsilon} R) = O(n^2 \varepsilon^{-1} \log R) = O(n^2 \varepsilon^{-1} \log \varepsilon^{-1}).$$

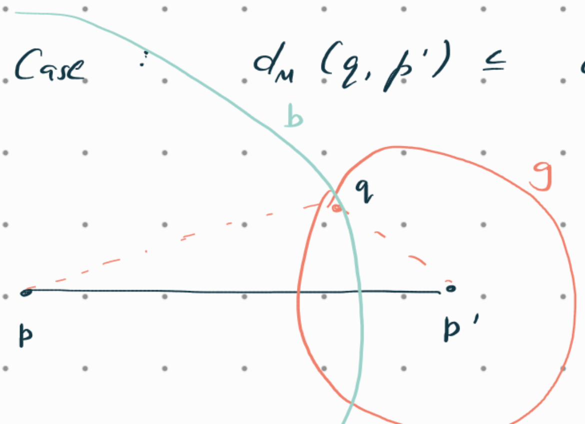
→ Correctness.

Let $b = O_B(q)$ be the target ball, and p its center.
Let $p' = nn(q, P)$ be the nearest neighbor.

- Far Case : $d_M(q,p') \geq 2\, d_M(p,p')/\varepsilon$

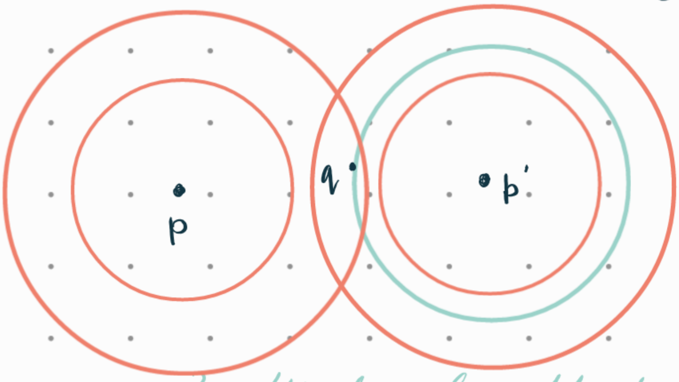
$$\frac{d(q,p)}{d(q,p')} \leq \frac{d(q,p') + d(p',p)}{d(q,p')} \leq 1 + \varepsilon/2$$

- Close Case : $d_M(q,p') \leq d_M(p,p')/4$



$rad(g) = d_M(p',q) \leq d_M(p,p')/4$.
but $rad(b) \geq 3/4\, d_M(p,p')$.
b can never be returned $\Rightarrow \Leftarrow$.

- Bad Case: $d_M(q,p') \in [1/4, 2/\varepsilon]\, d_M(p,p')$
  then by construction, there is a good ball.



Are we done now? Kind of. Maybe we can still improve on the quadratic size?

⇒ Auxilliaries

→ Proximity Queries (a.k.a. Near Neighbor Data Structure)

- Given point set P, query pt q, and $r > 0 \in \mathbb{R}$.
  $D_{near}(P, r)$ is a data structure, which decides whether
  $d(q,P) \leq r$ or $d(q,P) > r$.
  In the former case, it returns a witness point.

- Trivial realization: balls of radius r, with centers at P.
  Target ball query for checking.

→ Interval Query (a.k.a. Interval Near Neighbor)
  to check whether query lies in dist. $[a,b]$ from P.

$\hat{I}(P, a, b, \varepsilon) := \{N_0, N_1, \dots, N_M\}$
where $N_i = D_{near}(P, r_i = \min((1+\varepsilon)^i a, b))$, & $M = \lceil \log_{1+\varepsilon}(b/a) \rceil$.

- Constructed with $O(\varepsilon^{-1} \log(b/a))$ PQ data structures.

- Contains $O(\varepsilon^{-1} n \log(b/a))$ balls overall.

- Requires 2 queries to report not in range, and requires $O(\log(\varepsilon^{-1}\log(b/a)))$ PQ to report $(1+\varepsilon)$-ANN in range.

→ Far Case.

Let Q be a set of m balls. And $r > 0 \in \mathbb{R}$ such that $\mathcal{U}_{balls}(Q, r)$ is a connected set. Then,

i) any points $p, q \in Q$ are within $2r(m-1)$ distance.

ii) for query $q \in M$ s.t. $d(q, Q) \geq \text{diam}(Q)/\delta$, we have that any point of Q is a $(1+\delta)$-ANN to q in Q.

→ Binary Heirarchically well Separated Tree (BHST)

- Given a set of points P. Consider a binary tree H, having elements of P as leaves. H defines an HST over points of P if
  - $\forall u \in H$, we associate a label $\Delta(u) \geq 0$.
  - $\Delta(u) = 0 \iff u$ is a leaf.
  - u is child of $v \implies \Delta(u) \leq \Delta(v)$
  - distance between leaves $u, v \in H$ defined as $\Delta(\text{lca}(u,v))$.

- distances satisfy triangle inequality, hence HST defines metric. useful, since can be manipulated algorithmically.

- Quadtree with 'diameter of point within cell' as label defines an HST.

- In linear time HST can be converted to BHST, while retaining distances.

- Metric N t-approximates a metric M, if
$$d_M(u,v) \le d_N(u,v) \le t \cdot d_M(u,v).$$

- Any n point metric can be $(n-1)$ approximated by some HST.

$\Rightarrow$ Linear # of balls.

- We use intuition of close / far / bad cases from earlier to refine our construction. We construct a tree to search for the ANN, such that traversal is based on that case work.

- Given set of points P, and a t-approximate BHST H of P for a leaf u of H, rep_u is the point stored at u, and for an internal node, rep_u is the representative of one of its children.

For a subtree $H' \subseteq H$, $P(H') := \{ rep_u \mid u \in H' \}$

not just rooted subtrees?

- A separator is a node u of tree H, such that its removal breaks H into connected components, each of size at most $|V(H)|/2$.

It can be found in linear time.

$\longrightarrow$ Construction

- Initialize subtree $S = H$.
- Set root of tree T as node $v = v(S)$. Notation: n_v is # of nodes of S, and $P^v = P(S) = \bigcup_{u \in S} rep_u$.

- Find $u^v \in V(S)$ as a separator of S. Let S_L and S_R be the subtrees rooted in the left and right child of u^v resp. And let $S_{out} := S \setminus \{S_L \cup S_R\}$.
  define $P_L^v = P(S_L)$, P_R^v, P_{out}^v accordingly.
- Build Interval Query data structure $\hat{I}(P^v, r_v, R_v, \varepsilon/4)$ and store it in node v. Here

$$r_v = \Delta(u^v)/4t \qquad R_v = \mu\,\Delta(u^v)$$

  global parameter $\mu \in O(\varepsilon^{-1}\log n)$ to be set later.
  $\Delta(u^v)$ is (upper bound) t-approx diameter of subtree at u^v.
- Recursively build tree on S_L, S_R, S_{out} and attach as children v_L, v_R, v_{out} of v.

→ Query

Perform query at $\hat{I}_{root}$:

i) $d(q, P^v) \leq r_v$: Then $q \in \mathcal{U}_{balls}(P^v, r_v)$, and DS returns a point $p \in P^v$ s.t. $d_M(q,p) \leq r_v$. If $p \in P_L^v$ or P_R^v recurse there, otherwise recurse in v_{out}.

ii) $d(q, P^v) \in (r_v, R_v]$: DS already returns an ANN.

iii) $d(q, P^v) > R_v$: recurse in v_{out}.

→ Correctness

- Observations:
  - $P_R^v \cap P_L^v = \emptyset$
  - $P_R^v \cap (P_{out}^v \setminus \{rep_u\}) = \emptyset$
  - $P_L^v \cap (P_{out}^v \setminus \{rep_u\}) = \emptyset$
  - $d_M(P_L^v \cup P_R^v, P_{out}^v \setminus \{rep_u\}) \geq \Delta(u^v)/t$
  - $d_M(P_L^v, P_R^v) \geq \Delta(u^v)/t$
  - $\operatorname{diam}(P_L^v \cup P_R^v) \leq \Delta(u^v)$

For some node v on the search path.

- Close Case $d(q, P^v) \leq r_v$: we continue search in the subtree that contains req. ANN, & hence incur no loss.

Say we recurse on v_L, then $d(q, P_L^v) \leq r_v$.
but $d_M(P_L^v, P \setminus P_L^v) \geq \Delta(u^v)/t$

Let $q_L = nn(q, P_L^v)$. So

$$d(q, P^v \setminus P_L^v) \geq d(q_L, P^v \setminus P_L^v) - d_M(q, q_L)$$
$$\geq 4r_v - r_v > r_v.$$

- Far Case $d(q, P^v) > R_v$: If $s = nn(q, P^v)$ is in r_{out}, we are fine (no loss). If not, if $s \in P_L^v \cup P_R^v$?

we know $\quad \text{diam}(P_L^v \cup P_R^v) \leq \Delta(u^v) = R_v/\mu$

$$\Rightarrow \frac{\text{diam}(P_L^v \cup P_R^v)}{1/\mu} \leq R_v$$

$\Rightarrow$ Any point of $P_L^v \cup P_R^v$ is $(1 + 1/\mu)$-ANN.
and rep_{u^v} (still in search space) is in $P_L^v \cup P_R^v$.
So we introduce $(1 + 1/\mu)$ error.

- Bad Case: actually pretty good. We output an answer introducing $(1 + \varepsilon/4)$ error.

Now, depth of tree $N = 2\ln 2n \in O(\log n)$. Let $\mu = 4N/\varepsilon$.
Quality of ANN,

$$\left(1 + 1/\mu\right)^N \left(1 + \varepsilon/4\right) \leq \exp\left(\frac{N}{\mu} + \frac{\varepsilon}{4}\right) = \exp\left(\frac{N}{4N/\varepsilon} + \varepsilon/4\right) = e^{\varepsilon/2} \leq 1 + \varepsilon.$$

→ # of balls. $O(n/\varepsilon \cdot \log^2 n)$.

per structure.

$U(n_v) = n/\varepsilon \cdot \log(b/a) = n_v/\varepsilon \log\left(\frac{R_v}{r_v}\right) = n_v/\varepsilon \log\left(\frac{t \log n}{\varepsilon}\right)$

recurrence

$$B(n) = U(n) + \underbrace{B(n_L) + B(n_R) + B(n_{out})}_{\text{each } n_i \le n/2 + 1}.$$

set $t = n^{O(1)}$ and $1/\varepsilon \in O(n)$, to get

$$B(2n) = O(n/\varepsilon \cdot \log^2 n)$$

→ # of Proximity Queries. $O(\log(n/\varepsilon))$.

every internal node on the search path takes at most 2 queries. The final query takes,

$$O(\log(\varepsilon^{-1} \log(b/a)))$$

$$= \log\left(\frac{\log(R_v/r_v)}{\varepsilon}\right) = \log 1/\varepsilon + \log\log\left(\frac{t \log n}{\varepsilon}\right)$$

$$= \log 1/\varepsilon + \log\log n.$$

at most $O(\log n)$ internal nodes in the query, so overall

$$O(\log(n/\varepsilon)).$$