October 25, 2023 10:51 AM

Approx Nearest Neighbor Search in High Dims

Problem Given n pts P in $\mathbb{R}^d$,
build data structure s.t.
given query pt $q \in \mathbb{R}^d$, can find $p \in P$
minimizing $d(p,q)$.

Euclidean (L_2): $\sqrt{\sum_{i=1}^{d} (p_i - q_i)^2}$

L_1: $\sum_{i=1}^{d} |p_i - q_i|$

L_∞: $\max_{i=1}^{d} |p_i - q_i|$

Known: $d=2$: $O(n)$ space, $O(\log n)$ time by PL. & Voronoi diag.

larger const d:

L_1 / L_∞: $O(n \log^d n)$ space, $O(\log^d n)$ query time

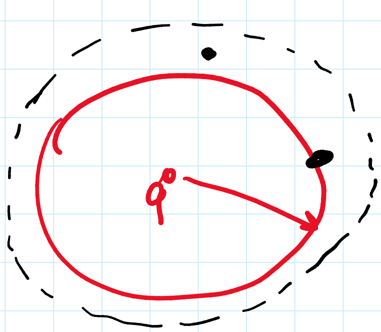
L_2: $O(n^{\lceil d/2 \rceil})$ space, $O(\log n)$ time
or $O(n)$ space, $\tilde{O}(n^{1-\frac{1}{\lceil d/2 \rceil}})$ time

terrible for large d!

Relax problem: given $q \in \mathbb{R}^d$, find $p \in P$ s.t.

$$d(p,q) \leq c \cdot \min_{p' \in P} d(p',q)$$

c: approximation factor



Suffice to solve approx <u>decision</u> problem, <u>for fixed radius r</u>:

given $q \in \mathbb{R}^d$,

return some pt $p \in P$ with $d(p,q) \le cr$

or declare all pts $p \in P$ with $d(p,q) > r$.

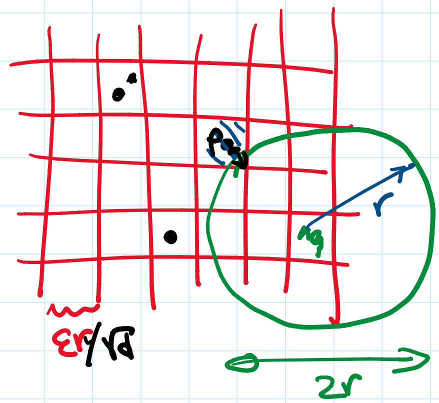
(original prob can be solved by approx binary search extra factor $\log U$)

try all $r = (1+\varepsilon)^i$, $(i = 0, \ldots, \underline{\log_{1+\varepsilon} U})$

<u>Method 0: Grid</u>

form uniform grid of side length $\varepsilon r / \sqrt{d}$

dictionary → store S = all nonempty grid cells



query(q):

check if any grid cell intersecting ball(q,r) is in S

by hashing in $O(1)$ time per grid cell →

$\le r + \varepsilon r = (1+\varepsilon) r$

$\Rightarrow$ $1+\varepsilon$ approx factor

query time $O(\#\text{grid cells intersecting ball}(q,r))$

$$\le O\left(\left(\frac{2r}{\varepsilon r/\sqrt{d}}\right)^d\right)$$

$$= \boxed{O\left(\left(\frac{2\sqrt{d}}{\varepsilon}\right)^d\right)}$$

← improves to $(O(1))^d$

$= \boxed{O\left(\left(\frac{O(1)}{\varepsilon}\right)^d\right)}$ $\left(\frac{O(1)}{\varepsilon}\right)^d$

Space $\boxed{O(dn)}$

Alternative: to reduce query time

store $S =$ all grid cells intersecting $\bigcup_{p \in P} \text{ball}(p, r)$

space $\boxed{\left(\frac{O(1)}{\varepsilon}\right)^d n}$

query time $\boxed{O(d)}$

Q: how to avoid exponential dependence on d?

Method 1: Dimension Reduction (Indyk-Motwani '98)

Johnson-Lindenstrauss Lemma ('84)

For n pts P in $\mathbb{R}^d$, for Euclidean, (or L_1)

$\exists$ mapping $f: P \to \mathbb{R}^k$, s.t. $\forall p, q \in P$,

$$d(p,q) \leq d(f(p), f(q)) \leq (1+\varepsilon)\, d(p,q)$$

with $k = \boxed{O\left(\frac{1}{\varepsilon^2} \log n\right)}$

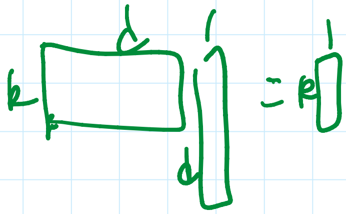
(Pf: take a random projection f)
Chernoff bd ...

take a random projection ?))

Chernoff bd . . .

$(a^{b\log n} = n^{b\log a})$

Combine with grid $\Longrightarrow$



space $\left(\frac{O(1)}{\varepsilon}\right)^{O(\frac{1}{\varepsilon^2}\log n)}$

$= \boxed{n^{O(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon})}}$

query $O(dk) = \boxed{O(\frac{1}{\varepsilon^2} d \log n)}$

no exponential in d !

Method 2: Locality-Sensitive Hashing (LSH)

(Indyk, Motwani '98)

idea - if p,q are close, more likely that $h(p)=h(q)$

Consider special case of Hamming space $\{0,1\}^d$

given $p, q \in \{0,1\}^d$,

define $d(p,q) =$ # bit positions that are diff.

e.g. $p = 10\,0\,1\,0\,1$
$q = 10\,1\,1\,1\,1$ $\quad d(p,q) = 2.$

Will solve approx decis problem for fixed r, approx factor c.

Preproc Alg/m:

let k be param.

define hash fn $h: \{0,1\}^d \to \{0,1\}^k$

by projecting to k rand dims

eg. $p = 10111110011$

$h(p) = 010$

for each $p \in P$

put p in bucket for $h(p)$.

Query(q): if each $p \in P$ in bucket for $h(q)$

if $d(p,q) \leq cr$ return p

return no

Prop $\forall p, q \in \{0,1\}^d$,

$$\Pr[h(p) = h(q)] = \left(1 - \frac{d(p,q)}{d}\right)^k$$

eg. $p = 1011100001$ $h(p) = 010$

$q = 1000111001$ $h(q) = 010$

$d(p,q) = 2.$