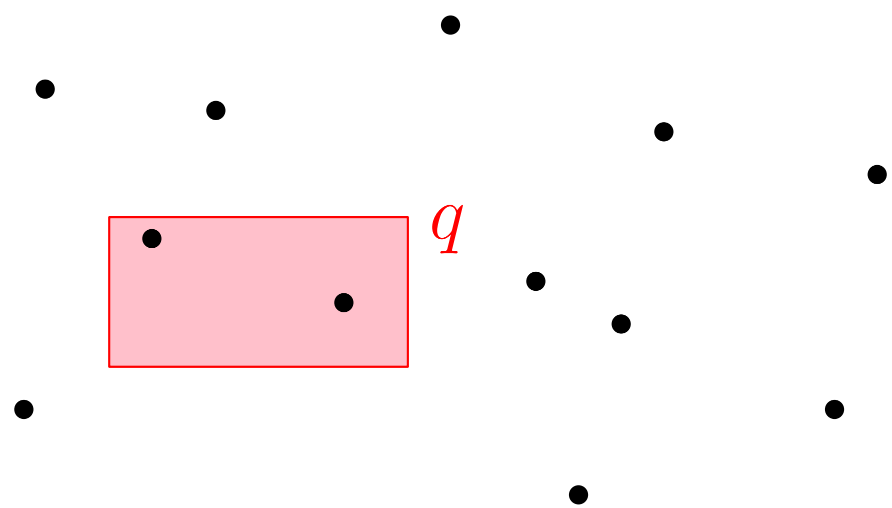
Geometric Problems in Moderate Dimensions

Timothy Chan

UIUC

Basic Problems in Comp. Geometry

- Orthogonal range search
  - preprocess n points in $\mathbb{R}^d$ s.t. we can detect, or count, or report points inside a query axis-aligned box

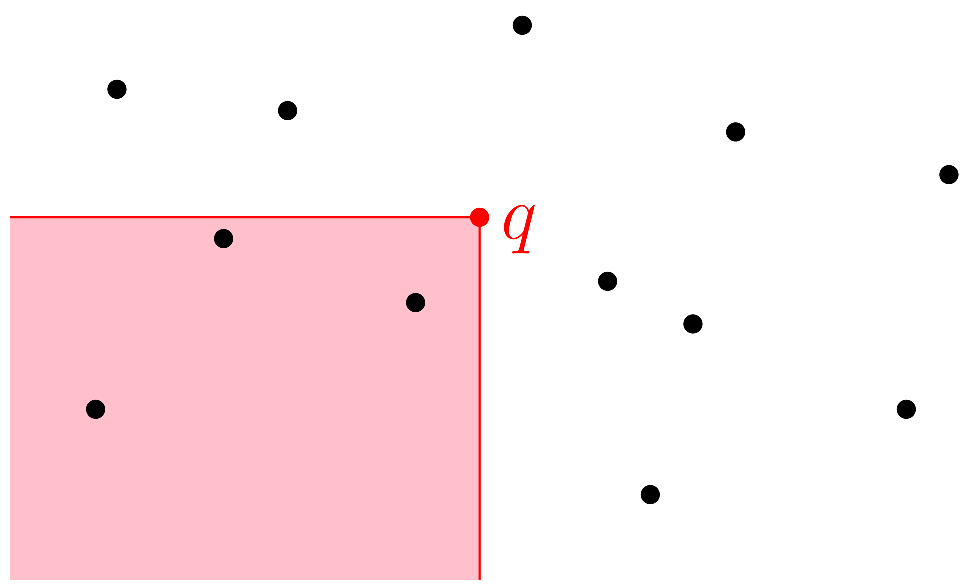


Basic Problems in Comp. Geometry

- Orthogonal range search
- Dominance range search
  - preprocess n points in $\mathbb{R}^d$ s.t. we can detect, or count, or report points dominated by q, i.e., inside $(-\infty, q_1] \times \cdots \times (-\infty, q_d]$

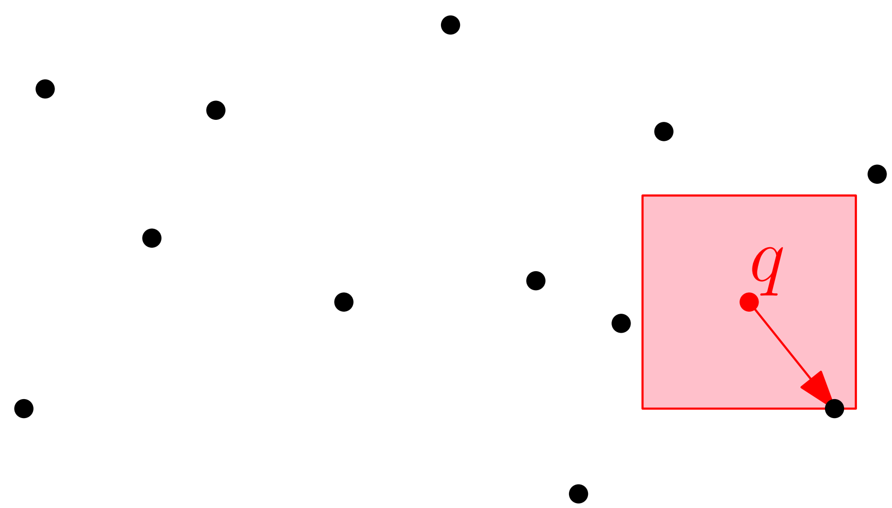
    (many geom. appl'ns: computing skylines, . . .)

    (orthogonal range search in $\mathbb{R}^d$ reduces to dominance in $\mathbb{R}^{2d}$)



Basic Problems in Comp. Geometry

- Orthogonal range search
- Dominance range search
- ℓ_∞ nearest neighbor search
  - preprocess n points in $\mathbb{R}^d$ s.t. we can find ℓ_∞-closest point to a query point

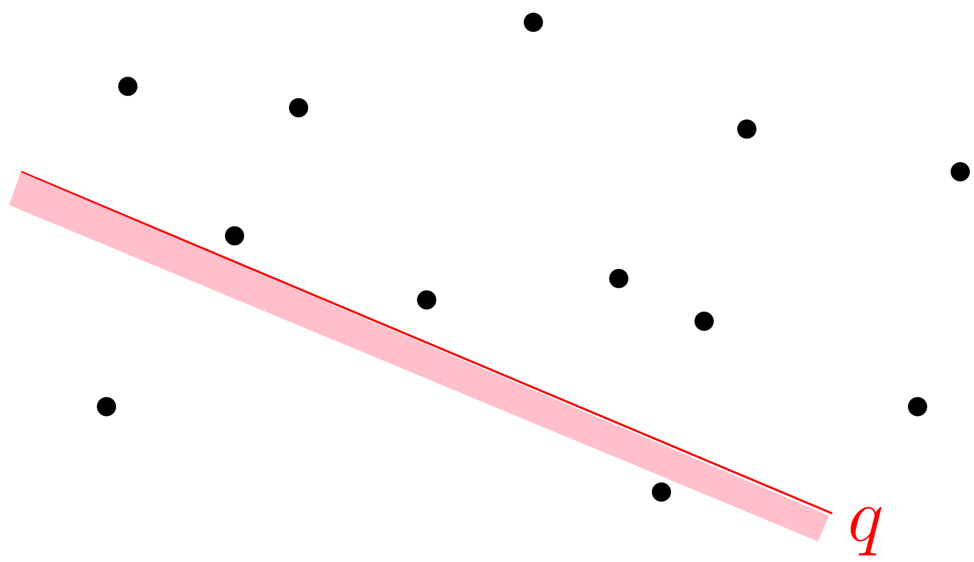


Basic Problems in Comp. Geometry

- Orthogonal range search
- Dominance range search
- ℓ_∞ nearest neighbor search
- ℓ_1 nearest neighbor search

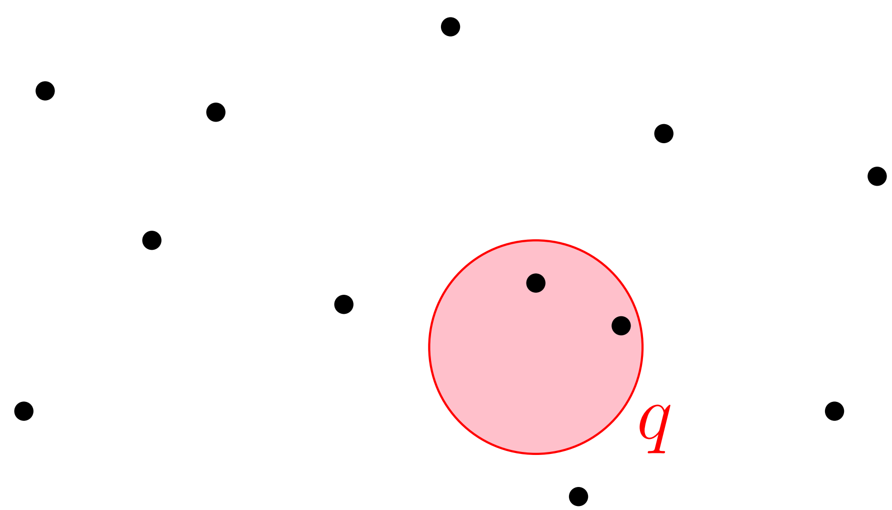
Basic Problems in Comp. Geometry

- Non-orthogonal range search
- Halfspace range search



Basic Problems in Comp. Geometry

- Non-orthogonal range search
- Halfspace range search
- Ball range search



Basic Problems in Comp. Geometry

- Non-orthogonal range search
- Halfspace range search
- Ball range search
- ℓ_2 nearest neighbor search

(many geom. appl'ns: bichromatic closest/farthest pair, min spanning tree, convex hull, ...)

Setting

- focus on orthogonal problems
- focus on exact, not approximate
- focus on upper bounds
- n online vs. offline queries

Low Dimensions: Classical Results from Comp. Geometry

- Orthogonal range search:

$$d^{O(d)} \cdot n \log^{O(d)} n \text{ time}$$

(but meaningful only when $d \leq \delta_0 \log n \ldots$)

- Non-orthogonal range search:

$$d^{O(d)} \cdot n^{2-1/O(d)} \text{ time}$$

Connection to Non-Geom. Problems

- Boolean orthogonal vector problem (OV)
  - given sets A, B of n vectors in $\{0, 1\}^d$, decide $\exists a \in A, b \in B$ s.t. $a \cdot b = 0$

  (appl'ns: subset queries, partial match queries for strings, . . .)

  (equiv. to Boolean version of offline dominance)

  (OV Conjecture: no $O(n^{2-\delta})$ alg'm for $d = \omega(\log n)$)

- Boolean matrix multiplication
  - given matrices $A \in \{0, 1\}^{n \times n}$, $B \in \{0, 1\}^{n \times n}$, compute $c_{ij} = \vee_k (a_{ik} \wedge b_{kj})$ for each i, j

Connection to Non-Geom. Problems

- Boolean orthogonal vector problem (OV)
  - given sets A, B of n vectors in $\{0,1\}^d$, decide $\exists a \in A, b \in B$ s.t. $a \cdot b = 0$

  (appl'ns: subset queries, partial match queries for strings, ...)

  (equiv. to Boolean version of offline dominance)

  (OV Conjecture: no $O(n^{2-\delta})$ alg'm for $d = \omega(\log n)$)

- Boolean matrix multiplication
  - given matrices $A \in \{0,1\}^{n \times d}$, $B \in \{0,1\}^{d \times n}$, compute $c_{ij} = \vee_k (a_{ik} \wedge b_{kj})$ for each i, j

Connection to Non-Geom. Problems

- Boolean orthogonal vector problem (OV)
  - given sets A, B of n vectors in $\{0, 1\}^d$, decide $\exists a \in A, b \in B$ s.t. $a \cdot b = 0$

  (appl'ns: subset queries, partial match queries for strings, ...)

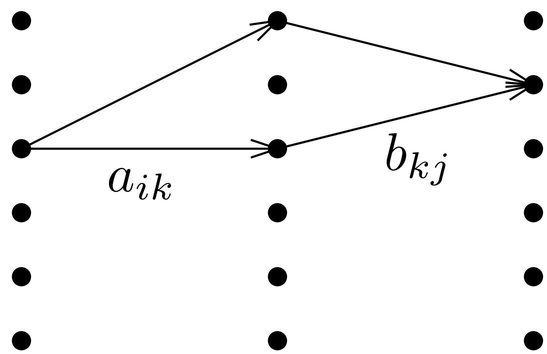
  (equiv. to Boolean version of offline dominance)

  (OV Conjecture: no $O(n^{2-\delta})$ alg'm for $d = \omega(\log n)$)

- Boolean matrix multiplication
  - i.e., given sets A, B of n vectors in $\{0, 1\}^d$, decide whether $a \cdot b = 0$ for each $a \in A, b \in B$

Connection to Non-Geom. Problems

- All-pairs shortest paths (APSP)



Connection to Non-Geom. Problems

- All-pairs shortest paths (APSP) or (min,+)-matrix multiplication
  - given matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times n}$, compute $c_{ij} = \min_k(a_{ik} + b_{kj})$ for each i, j

(appl'ns: graph diameter/radius/etc., max 2D subarray, language edit distance, min-weight triangulation of polygons, . . .)

Connection to Non-Geom. Problems

- All-pairs shortest paths (APSP)
  or (min,+)-matrix multiplication
  - given matrices $A \in \mathbb{R}^{n\times d}$, $B \in \mathbb{R}^{d\times n}$, compute
    $c_{ij} = \min_k(a_{ik} + b_{kj})$ for each i, j

Connection to Non-Geom. Problems

- All-pairs shortest paths (APSP) or (min,+)-matrix multiplication
  - i.e., given sets A, B of n vectors in $\mathbb{R}^d$, compute $\min_k(a_k + b_k)$ for each $a \in A, b \in B$

(reduces to d instances of offline dominance by "Fredman's trick":

$$a_{k_0} + b_{k_0} \le a_k + b_k \iff a_{k_0} - a_k \le b_k - b_{k_0})$$

Connection to Non-Geom. Problems

- (min,+)-convolution
  - given vectors $a, b \in \mathbb{R}^n$, compute $c_i = \min_k(a_k + b_{i-k})$ for each i

  (appl'ns: jumbled string matching, knapsack, min k-enclosing rectangles, . . .)

  (reduces to $O(\sqrt{n})$ (min,+)-matrix multiplication of $\sqrt{n} \times \sqrt{n}$ matrices)

- 3SUM
  - given vectors $a, b, c \in \mathbb{R}^n$, decide $\exists i, j, k$ s.t. $a_i + b_j = c_k$

Connection to Non-Geom. Problems

- SAT for sparse instances
  - given CNF formula in n Boolean vars & cn clauses, decide $\exists$ satisfying assignment

(reduces to OV with $2^{n/2}$ Boolean vectors in cn dimensions)

  - for each partial assignment of first $n/2$ vars, define vector a with $a_i = 0$ iff i-th clause is already satisfied;
  - for each partial assignment of last $n/2$ vars, define vector b with $b_i = 0$ iff i-th clause is already satisfied)

Connection to Non-Geom. Problems

- SAT for sparse instances
  - given CNF formula in n Boolean vars & cn clauses, decide $\exists$ satisfying assignment

  (reduces to OV with $2^{n/2}$ Boolean vectors in cn dimensions)

- kSAT

  (reduces to sparse case)

  (SETH Conjecture: no $(2-\delta)^n$ alg'm for $k = \omega(1)$)

- MAX-SAT for sparse instances
- MAX-kSAT

Connection to Non-Geom. Problems

- Integer 0-1 linear programming for sparse instances
  - given cn linear inequalities with real coeffs. over n 0-1 vars, decide $\exists$ satisfying assignment

(reduces to dominance with $2^{n/2}$ real vectors in cn dimensions)

High Dimensions: By Fast Matrix Multiplication

- offline dominance in Boolean case (i.e., OV) can be trivially solved by Boolean rect. matrix multiplication in $M(n,d,n)$ time
  - $M(n,n,n) = O(n^{2.373})$
  - $M(n,d,n) = \widetilde{O}(n^2)$ for $d \approx n^{0.1}$
- offline dominance for real case can be reduced to Boolean case, in $O(\min_s(M(n,ds,n) + dn^2/s))$ time [Matoušek'91]
  - by dividing into s buckets of size n/s

(but not clear how to beat n^2 time...)

Moderate Dimensions: This Talk

- subquadratic time for d beyond logarithmic?

- let $d = c \log n$ (c not necessarily constant)

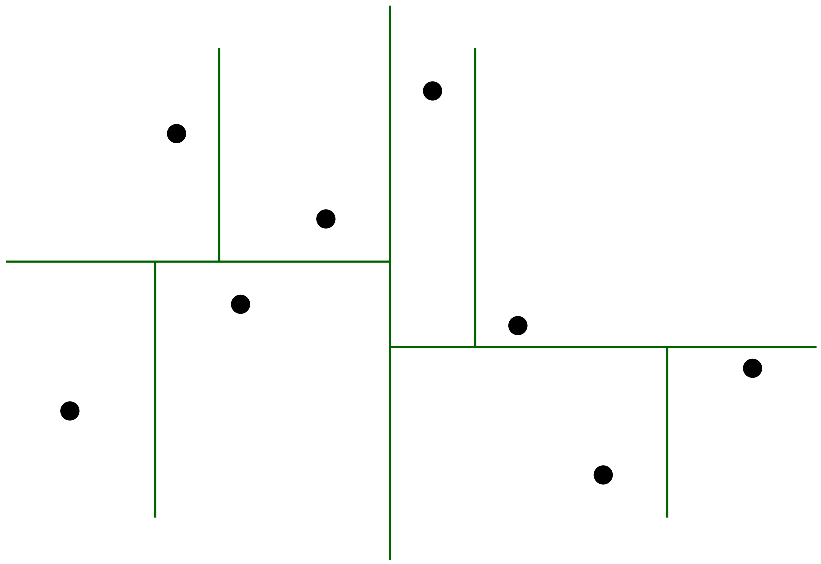
Two Approaches

Part I. Comp. Geometry Techniques
(k-d trees, range trees)

Part II. Polynomial Method

k-d Trees ['75]

1. pick next axis $i \in \{1, \ldots, d\}$
2. divide by median i-th coord.
3. recurse on both sides

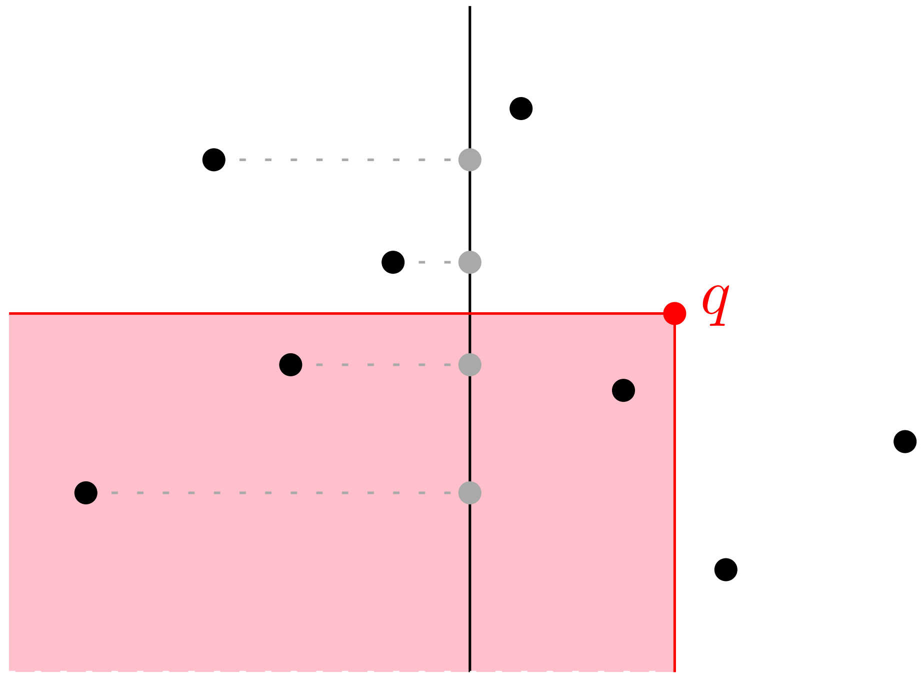


preproc. time $O(dn \log n)$

dominance query time $O(n^{1-1/d})$

Range Trees ['79]

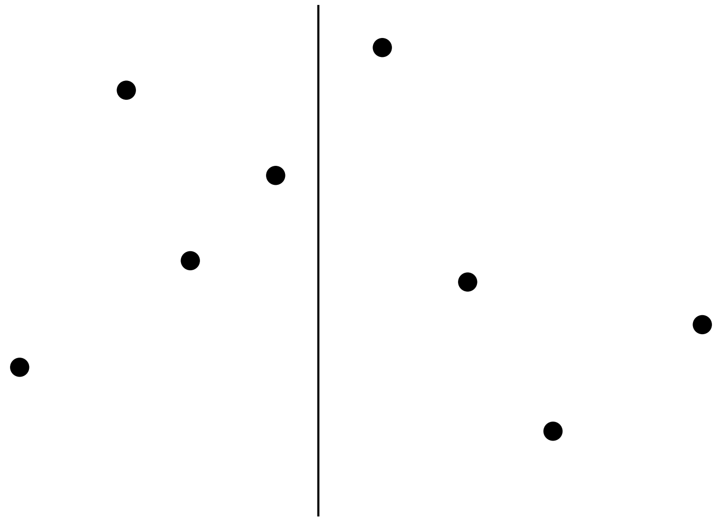
0. recurse on projection along 1st coord.
1. divide by median 1st coord.
2. recurse on both sides



$$P_d(n) \le 2P_d(n/2) + P_{d-1}(n/2) \;\Rightarrow\; O(n \log^d n)$$
$$Q_d(n) \le Q_d(n/2) + Q_{d-1}(n/2) \;\Rightarrow\; O(\log^d n)$$

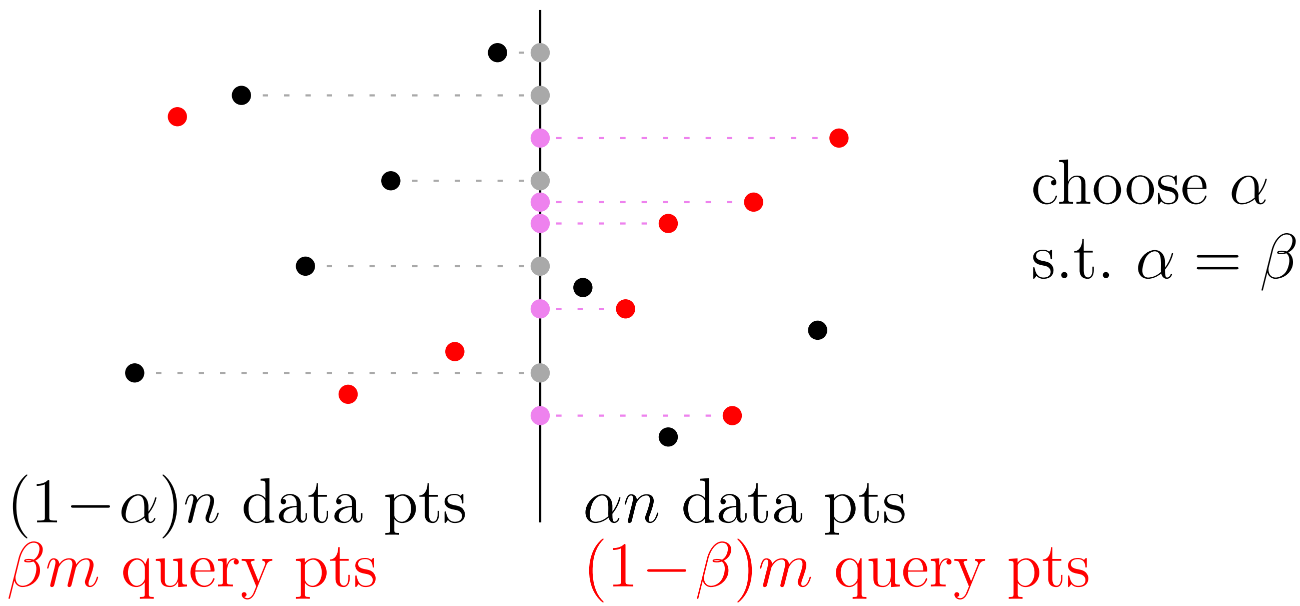
“Lopsided” Range Tree for Offline Dominance [Impagliazzo–Lovett–Paturi–Schneider’14]

0. recurse on projection along 1st coord.
1. divide by median 1st coord.
2. recurse on both sides



“Lopsided” Range Tree for Offline Dominance [Impagliazzo–Lovett–Paturi–Schneider’14]

0. recurse on projection along 1st coord.
1. divide by (αn)-th largest 1st coord. for some α
2. recurse on both sides



$$T_d(n,m) \le \max_{\alpha} [T_d((1-\alpha)n, \alpha m) + T_d(\alpha n, (1-\alpha)m) + T_{d-1}((1-\alpha)n, (1-\alpha)m)]$$

“Lopsided” Range Tree for Offline Dominance [Impagliazzo–Lovett–Paturi–Schneider’14]

$$T_d(n,m) \;\le\; \max_\alpha\, [T_d((1-\alpha)n, \alpha m) + T_d(\alpha n, (1-\alpha)m) + T_{d-1}((1-\alpha)n, (1-\alpha)m)]$$

- Impagliazzo et al.: $\boxed{n^{2-1/\widetilde{O}(c^{15})}}$ time for n offline queries (for $d = c \log n$)
- C. [SODA’15]: $\boxed{n^{2-1/\widetilde{O}(c)}}$ time

  (subquadratic for $c \ll \log n$, i.e., $d \ll \log^2 n$)

  (appl’n: integer linear programming with cn constraints in $(2-1/\widetilde{O}(c))^n$ time)

  (but only works for offline. . .)

New “Lopsided” k-d Tree for Online Dominance [C., SoCG’17]

1. pick next axis $i \in \{1, \ldots, d\}$
2. divide by median i-th coord.
3. recurse on both sides

New "Lopsided" k-d Tree for Online Dominance [C., SoCG'17]

0. directly build data structure for each possible (d/b)-dimensional projection, with $b \approx (1/\varepsilon)c \log c$
1. pick random axis $i \in \{1, \ldots, d\}$
2. divide by (αn)-th largest i-th coord. with $\alpha \approx 1/c^4$
3. recurse on both sides

projections $\binom{d}{d/b} = b^{O(d/b)} = b^{O((c \log n)/b)} = n^{O(\varepsilon)}$

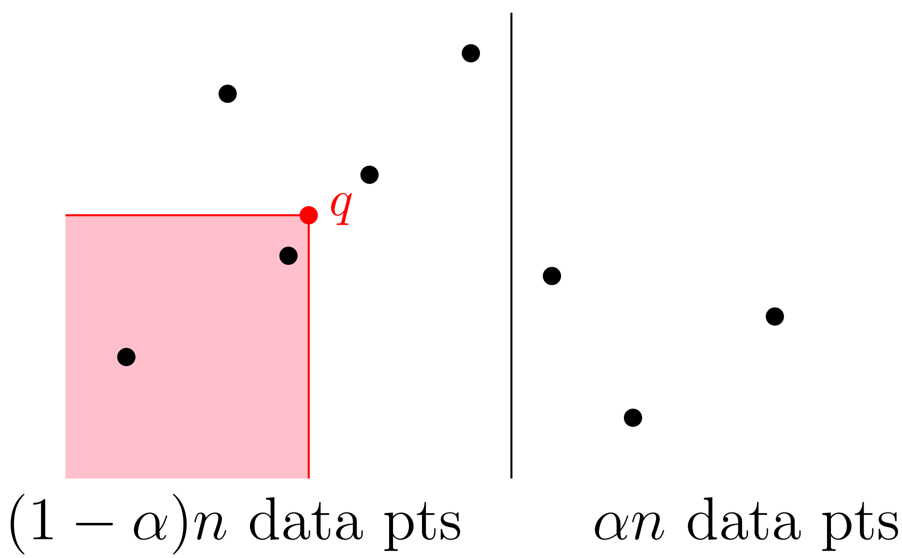
$\Rightarrow$ preproc. time $\boxed{n^{1+O(\varepsilon)}}$

New “Lopsided” k-d Tree for Online Dominance [C., SoCG’17]

Let $Q_j(n)$ = time for query pt with j bounded coords

If $j \le d/b$ then base case else

$$Q_j(n) \le \max_{\ell} \begin{cases} Q_j((1-\alpha)n) \\ \\ \\ \end{cases}$$

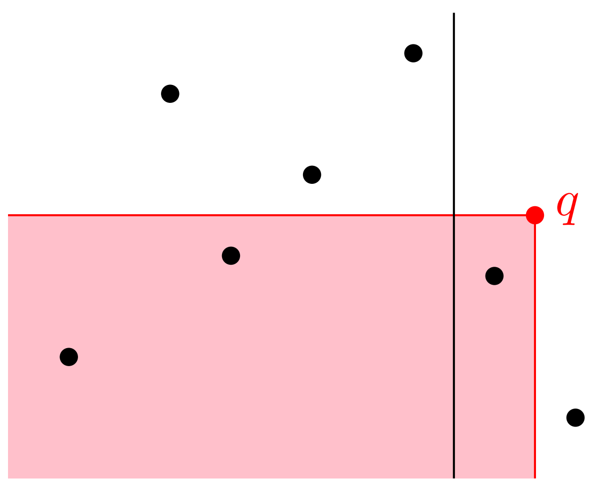


New "Lopsided" k-d Tree for Online Dominance [C., SoCG'17]

Let $Q_j(n) =$ time for query pt with j bounded coords

If $j \le d/b$ then base case else

$$Q_j(n) \le \max_{\ell} \begin{cases} Q_j((1-\alpha)n) \\ (Q_j(\alpha n) + Q_{j-1}((1-\alpha)n)) \\ \\ \end{cases}$$



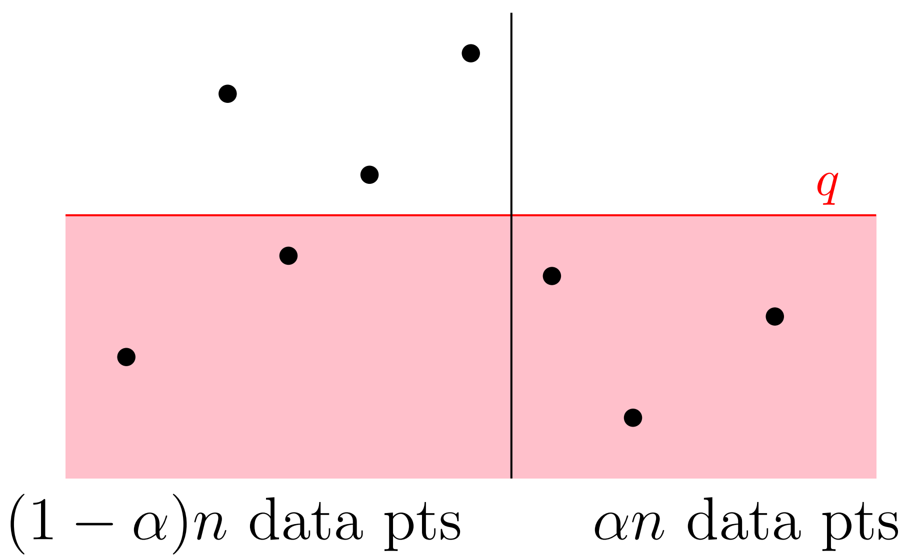
$(1-\alpha)n$ data pts $\qquad$ αn data pts

New "Lopsided" k-d Tree for Online Dominance [C., SoCG'17]

Let $Q_j(n)$ = time for query pt with j bounded coords

If $j \le d/b$ then base case else

$$Q_j(n) \le \max_{\ell} \begin{cases} Q_j((1-\alpha)n) \\ (Q_j(\alpha n) + Q_{j-1}((1-\alpha)n)) \\ (Q_j(\alpha n) + Q_j((1-\alpha)n)) \end{cases}$$



New “Lopsided” k-d Tree for Online Dominance [C., SoCG’17]

Let $Q_j(n)$ = time for query pt with j bounded coords

If $j \le d/b$ then base case else

$$Q_j(n) \le \max_{\ell} \begin{cases} \frac{\ell}{d} \cdot & Q_j((1-\alpha)n) + \\ \frac{j-\ell}{d} \cdot & (Q_j(\alpha n) + Q_{j-1}((1-\alpha)n)) + \\ \frac{d-j}{d} \cdot & (Q_j(\alpha n) + Q_j((1-\alpha)n)) \end{cases}$$

$\Rightarrow$ expected online query time $\boxed{n^{1-1/\widetilde{O}(c)}}$

(appl’n to APSP: $\widetilde{O}(n^3/\log^3 n)$ combinatorial alg’m [C., SoCG’17])

(specialization to Boolean case: k-d tree $\rightarrow$ trie)

Open Problems

- deterministic online?
- lower bounds for geometric tree-based methods?

s-Way Range Tree: Reducing Offline Real Dominance to Boolean [C., SoCG'17]

Assume that offline Boolean dominance (i.e., OV) can be solved in $n^{2-f(c)}$ time

Let $T_j(n)$ be time for n data & query pts in $\mathbb{R}^j \times [s]^{d-j}$

If $j = 0$ then $T_0(n) \le n^{2-f(cs)}$ else

$$T_j(n) \le sT_j(n/s) + T_{j-1}(n)$$

Set $s = c^4$

$\Rightarrow$ $\boxed{n^{2-f(c^5)+O(1/c)}}$ time for offline real dominance

Two Approaches

Part I. Comp. Geometry Techniques
(k-d trees, range trees)

Part II. Polynomial Method

Boolean OV [Abboud–Williams–Yu, SODA'15]

- First reduce # of input vectors from n to n/s, by treating each group of s vectors as one

$\Rightarrow$ given sets A, B of n/s vectors in $\{0, 1\}^{ds}$, evaluate $f(a, b)$ for each $a \in A, b \in B$, for this "funny" function

$$f(a,b) = \bigwedge_{i,j \in [s]} \bigvee_{k \in [d]} (a_{ik} \wedge b_{jk})$$

$\Rightarrow$ "funny" rect. matrix multiplication problem

Boolean OV [Abboud–Williams–Yu, SODA'15]

- If we can express f as polynomial, "funny" rect. matrix multiplication reduces to standard rect. matrix multiplication

- Example:

$$\begin{aligned} f(a,b) &= a_1b_2 + 4a_2b_1b_2 + 3a_1a_3b_1 - 5a_2b_1b_3 \\ &= (a_1,\, 4a_2,\, 3a_1a_3,\, -5a_2) \cdot (b_2,\, b_1b_2,\, b_1,\, b_1b_3) \end{aligned}$$

- new dim. = # of monomials
- $\widetilde{O}((n/s)^2)$ time if # of monomials $\ll (n/s)^{0.1}$...

Boolean OV [Abboud–Williams–Yu, SODA'15]

- New Problem: express

$$f(a,b) = \bigwedge_{i,j \in [s]} \bigvee_{k \in [d]} (a_{ik} \wedge b_{jk})$$

as a polynomial with small ~~# of monomials~~ degree

Boolean OV [Abboud–Williams–Yu, SODA'15]

- New Problem: express

$$\text{AND-of-OR}(x) = \bigwedge_{\ell \in [s^2]} \bigvee_{k \in [d]} x_{\ell k}$$

as a polynomial with small degree

- Naive Sol'n:

$$\sum_{\ell} \left(1 - \prod_{k \in [d]} (1 - x_{\ell k}) \right)$$

$\Rightarrow$ degree d

Boolean OV [Abboud–Williams–Yu, SODA'15]

- New Problem: express

$$\text{AND-of-OR}(x) = \bigwedge_{\ell \in [s^2]} \bigvee_{k \in [d]} x_{\ell k}$$

as a polynomial with small degree

- Rand. Sol'n: by Razborov–Smolensky's trick ('87)
  - replace each OR with random linear combination in $\mathbb{F}_2$
  - repeat $\log(100s^2)$ times to lower error prob. to $1/(100s^2)$
  - replace AND with another random linear combination in $\mathbb{F}_2$

$\Rightarrow$ degree $\boxed{O(\log s)}$

Boolean OV [Abboud–Williams–Yu, SODA'15]

- degree $\boxed{O(\log s)}$
- # monomials $\approx s^2 \cdot \binom{d}{O(\log s)}$

$$
\begin{aligned}
&= \left(\tfrac{d}{\log s}\right)^{O(\log s)} \\
&= (c/\alpha)^{O(\alpha \log n)} && \text{for } d = c\log n,\ \ s = n^\alpha \\
&= n^{O(\alpha \log(c/\alpha))} \\
&\ll (n/s)^{0.1} && \text{for } \alpha \approx 1/O(\log c)
\end{aligned}
$$

$\Rightarrow \widetilde{O}((n/s)^2) = \boxed{n^{2-1/O(\log c)}}$ rand. time

(better than $n^2/\text{poly}(d)$ when $d \ll 2^{\sqrt{\log n}}$)

(similar ideas used in Williams's APSP alg'm [STOC'14] in $n^3/2^{\Omega(\sqrt{\log n})}$ time)

Boolean OV [Abboud–Williams–Yu, SODA'15]

- Derandomization [C.–Williams, SODA'16]
  - use ε-biased space for the random linear combinations in $\mathbb{F}_2$
  - sum over entire sample space
  - use modulus-amplifying polynomials before summing

- Extends to counting problem #OV (via SUM-of-OR)

  (appl'ns: SAT & #SAT with cn clauses in $(2-1/O(\log c))^n$ time, kSAT & #kSAT in $(2-1/O(k))^n$ time)

Offline Hamming Nearest Neighbor

[Alman–Williams, FOCS'15; Alman–C.–Williams, FOCS'16]

- New Problem: express

$$f(a,b) = \bigwedge_{i,j\in[s]} \left[\sum_{k\in[d]} (a_{ik} - b_{jk})^2 \geq t \right]$$

as a polynomial with small degree

Offline Hamming Nearest Neighbor

[Alman–Williams, FOCS'15; Alman–C.–Williams, FOCS'16]

- New Problem: express

$$\text{AND-of-THR}(x) = \bigwedge_{\ell \in [s^2]} \left[\sum_{k \in [d]} x_{\ell k} \leq t \right]$$

  as a polynomial with small degree

- Rand. Sol'n 1: [Alman–Williams]
  - replace AND with sum
  - for each THR, take random sample of size $d/2$ & recurse
  - if count $\in t \pm O(\sqrt{d \log s})$, use interpolating polynomial

$\Rightarrow$ degree $\boxed{O(\sqrt{d \log s})}$

Offline Hamming Nearest Neighbor

[Alman–Williams, FOCS'15; Alman–C.–Williams, FOCS'16]

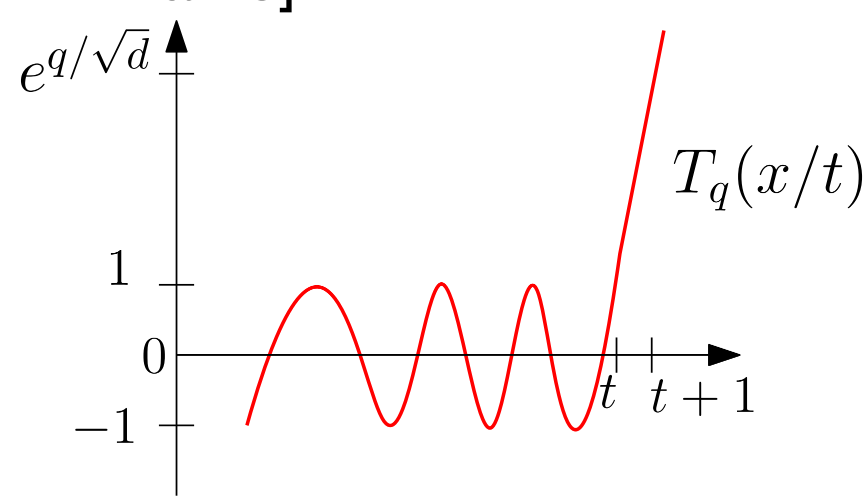
- New Problem: express

$$\text{AND-of-THR}(x) = \bigwedge_{\ell \in [s^2]} \left[\sum_{k \in [d]} x_{\ell k} \le t \right]$$

as a polynomial with small degree

- Simple Det. Sol'n 2: [Alman–C.–Williams]
  - sum of Chebyshev polynomials

$\Rightarrow$ degree $\boxed{O(\sqrt{d}\,\log s)}$



Offline Hamming Nearest Neighbor

[Alman–Williams, FOCS'15; Alman–C.–Williams, FOCS'16]

- New Problem: express

$$\text{AND-of-THR}(x) = \bigwedge_{\ell \in [s^2]} \left[\sum_{k \in [d]} x_{\ell k} \le t \right]$$

as a polynomial with small degree

- Combined Sol'n 3: [Alman–C.–Williams]
  - for each THR, take random sample of size $r = d^{2/3} \log^{1/3} s$
  - use Sol'n 1 on sample
  - if count $\in t \pm O((d/\sqrt{r})\sqrt{\log s})$, use Chebyshev polynomial

$\Rightarrow$ degree $\boxed{O(d^{1/3} \log^{2/3} s)}$

Offline Hamming Nearest Neighbor

[Alman–Williams, FOCS'15; Alman–C.–Williams, FOCS'16]

- degree $\boxed{O(d^{1/3}\log^{2/3} s)}$
- # monomials $\approx\ s^2 \cdot \binom{d}{O(d^{1/3}\log^{2/3} s)}$

$$
\begin{aligned}
&= \left(\tfrac{d}{\log s}\right)^{O(d^{1/3}\log^{2/3} s)} \\
&\le (c/\alpha)^{O(c^{1/3}\alpha^{2/3}\log n)} && \text{for } d = c\log n,\ \ s = n^{\alpha} \\
&= n^{\widetilde{O}(c^{1/3}\alpha^{2/3})} \\
&\ll (n/s)^{0.1} && \text{for } \alpha \approx 1/\widetilde{O}(\sqrt{c})
\end{aligned}
$$

$\Rightarrow \widetilde{O}((n/s)^2) = \boxed{n^{2-1/\widetilde{O}(\sqrt{c})}}$ rand. time

(subquadratic when $c \ll \log^2 n$, i.e., $d \ll \log^3 n$)

Offline Hamming Nearest Neighbor

[Alman–Williams, FOCS'15; Alman–C.–Williams, FOCS'16]

- extends to offline ℓ_1 nearest neighbor in $[U]^d$ in $n^{2-1/\widetilde{O}(\sqrt{cU})}$ rand. time

- offline $(1+\varepsilon)$-approximate (ℓ_1 or ℓ_2) nearest neighbor in $n^{2-\widetilde{\Omega}(\varepsilon^{-1/3})}$ rand. time (via AND-of-Approx-THR)

  (improving over Valiant [FOCS'12] & LSH for small ε)

  (appl'n: MAX-SAT with cn clauses in $(2-1/\widetilde{O}(c^{1/3}))^n$ time)

  (appl'n: MAX-3-SAT with cn clauses in $(2-1/\text{polylog}\, c)^n$ time)

Open Problems

- derandomize?
- improve $d^{1/3}$ degree for AND-of-THR?
- ℓ_1 nearest neighbor search for larger universe U?
- beat LSH for offline 2-approximate nearest neighbor?

- online? (Larsen–Williams [SODA'17] solved online Boolean OV)

- better offline dominance: disprove OV/SETH??

- non-orthogonal problems are harder

  (Williams [SODA'18]: offline ℓ_2 nearest neighbor search for $d = \omega(\log\log n)^2$ can't be solved in $O(n^{2-\delta})$ time, assuming OV conjecture)