Computational Geometry, from Low to High Dimensions

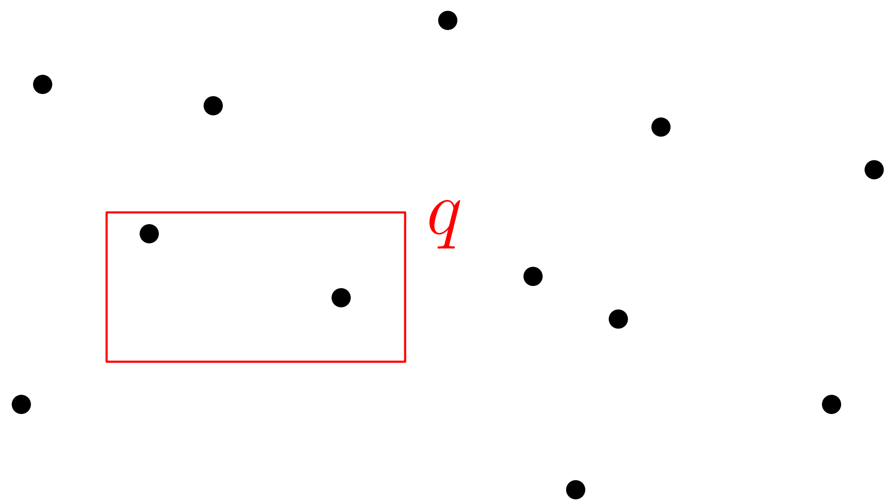
Timothy Chan

School of CS

U of Waterloo

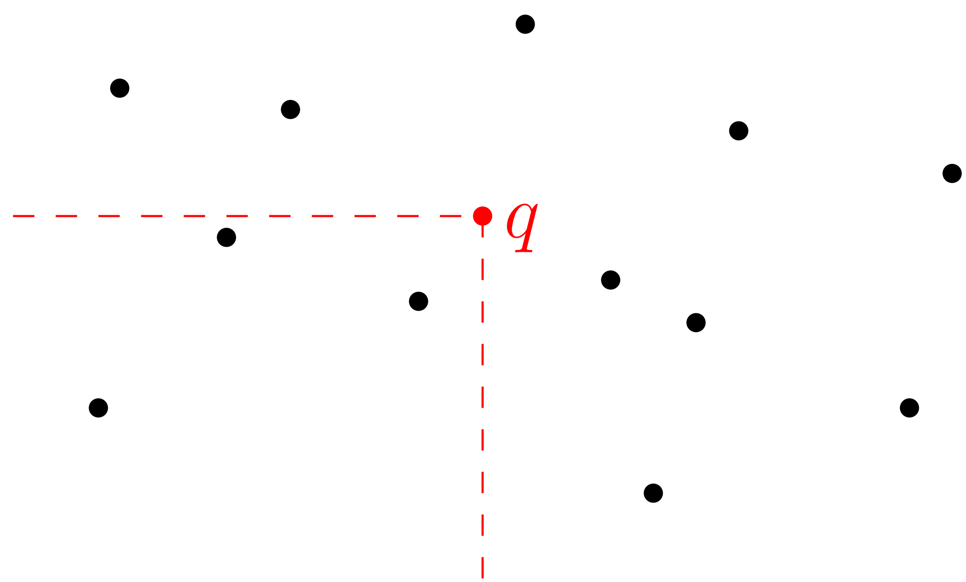
Classic Problems in CG: Orthogonal Range Search

- preprocess n points in $\mathbb{R}^d$, to detect/report/count points inside any query rectangle q



Classic Problems in CG: Dominance Range Search

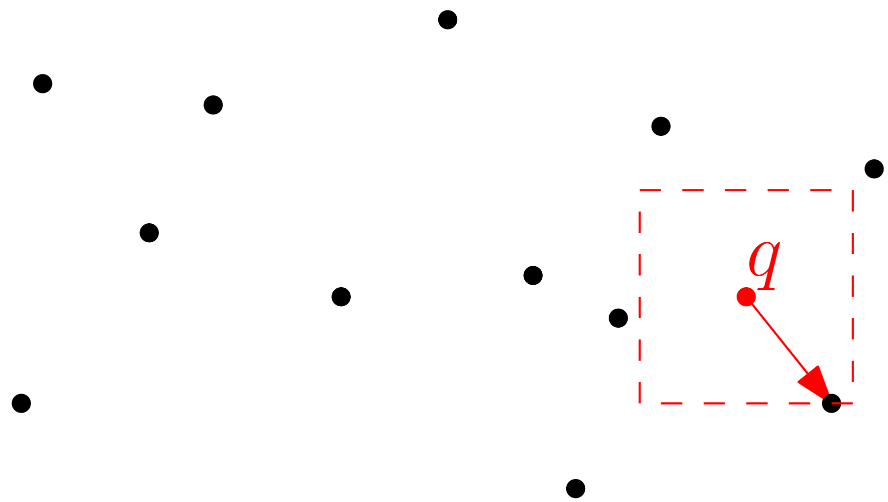
- preprocess n points in $\mathbb{R}^d$, to detect/report/count points dominated by any query point q



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

Classic Problems in CG: ℓ_∞ Nearest Neighbor Search

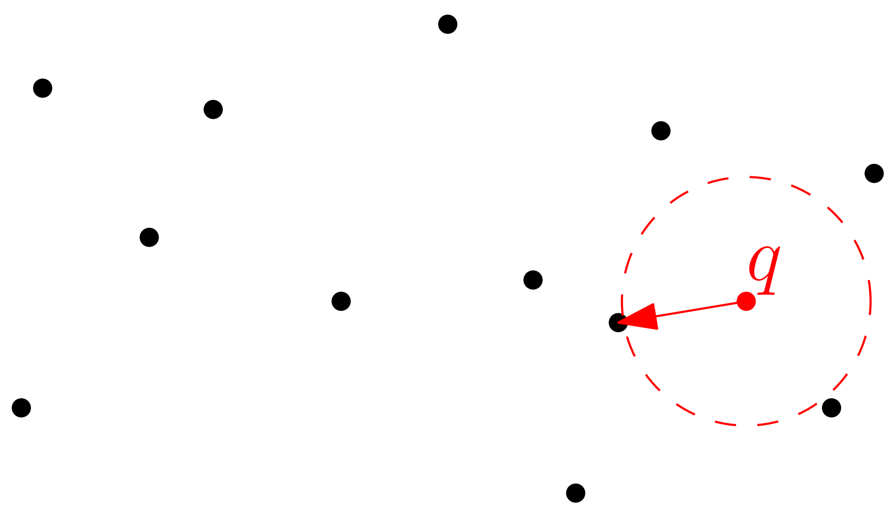
- preprocess n points in $\mathbb{R}^d$, to find nearest point to any query point q



(ℓ_∞ nearest neighbor search reduces to orthogonal range search)

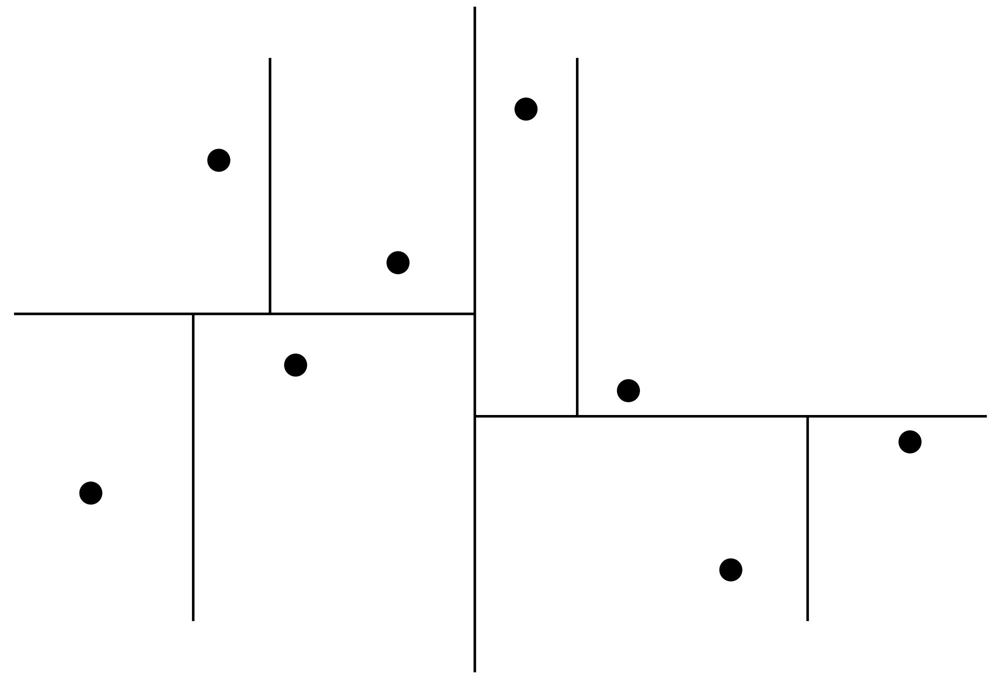
Classic Problems in CG: ℓ_2 Nearest Neighbor Search

- preprocess n points in $\mathbb{R}^d$, to find ℓ_2 nearest point to any query point q



Standard Alg'ms in Low Dimensions: k-d Trees [Bentley'75]

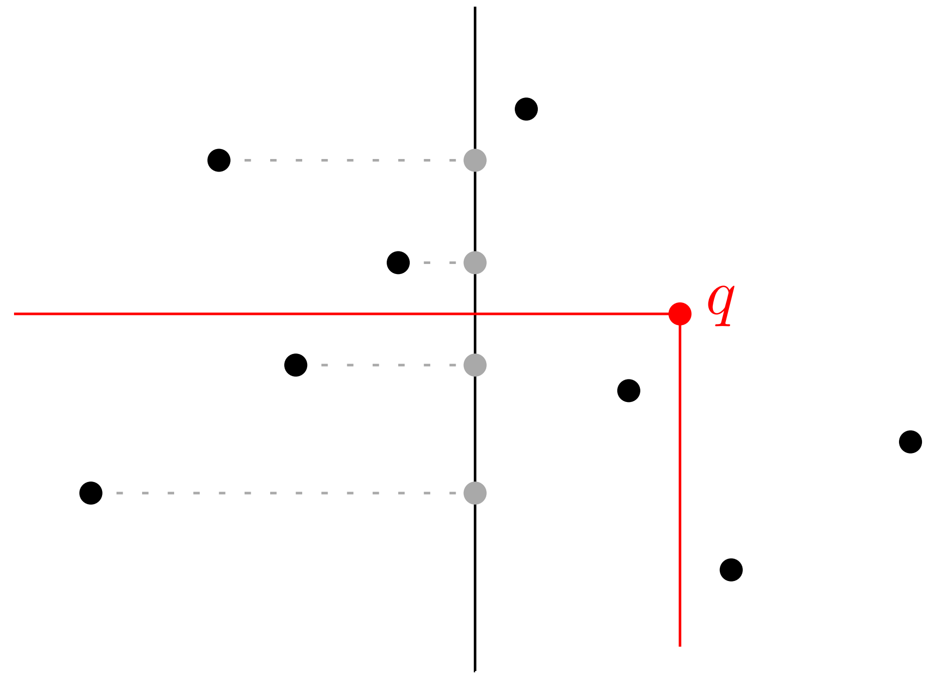
- divide by median x-coord.
- then by median y-coord.
- etc.



$\Rightarrow$ $\boxed{O(n)}$ space, $\boxed{O(n^{1-1/d})}$ query time

Standard Alg'ms in Low Dimensions: Range Trees [Lueker'78, Willard'79, Bentley'79, Lee–Wong'80]

- divide by median x-coord.
- recurse on left & right
- recurse on projection



$$\Rightarrow S_d(n) \le 2S_d(n/2) + S_{d-1}(n)$$
$$Q_d(n) \le Q_d(n/2) + Q_{d-1}(n)$$

$\Rightarrow$ $\boxed{O(n \log^d n)}$ space, $\boxed{O(\log^d n)}$ query time

Improvements
for Orthogonal Range Reporting

	space	query time
Willard'85	$n \log^{d-1} n$	$\log^{d-1} n$
Chazelle [FOCS'83]	$n \log^{d-1} n / \log\log n$	$\log^{d-1} n$
Chazelle [FOCS'85]	$n \log^{d-2+\varepsilon} n$	$\log^{d-1} n$
Willard [SODA'92]	$n \log^{d-1} n / \log\log n$	$\log^{d-1} n / \log\log n$
Subramanian–Ramaswamy [SODA'95]	$n \log^{d-1} n$	$\log^{d-2} n \log^{**} n$
Alstrup–Brodal–Rauhe [FOCS'00]	$n \log^{d-2+\varepsilon} n$	$\log^{d-2} n / \log^{d-3} \log n$
Nekrich [SoCG'07]	$n \log^{d+1+\varepsilon} n$	$\log^{d-3} n / \log^{d-5} \log n$
Afshani'08	$n \log^{d+\varepsilon} n$	$\log^{d-3} n / \log^{d-5} \log n$
Karpinski–Nekrich'09	$n \log^{d-2+\varepsilon} n$	$\log^{d-3} n / \log^{d-6} \log n$
C. [SODA'11]	$n \log^{d-2+\varepsilon} n$	$\log^{d-3} n / \log^{d-5} \log n$
C.–Larsen–Pătraşcu [SoCG'11]	$n \log^{d-2+\varepsilon} n$	$\log^{d-3} n / \log^{d-4} \log n$

Improvements
for Offline Dominance Range Detection

	total time for n queries
Kung–Luccio–Preparata'75	$n \log^{d-2} n$
Gabow–Bentley–Tarjan [STOC'84]	$n \log^{d-3} n \log\log n$
C.–Larsen–Pătraşcu [SoCG'11]	$n \log^{d-3} n$

Improvements
for Offline Dominance Range Counting

	total time for n queries
Bentley'80	$n \log^{d-1} n$
Willard [SODA'02]	$n \log^{d-1} n / \log \log n$
C.–Pătraşcu [SODA'10]	$n \log^{d-2+1/d} n$

Known Alg'ms in Low Dimensions
for ℓ_2 Nearest Neighbor Search

	space	query time
Dobkin–Lipton [STOC'74]	$n^{2^{d+1}}$	$\log n$
Clarkson [STOC'85]	$n^{\lceil d/2 \rceil + \varepsilon}$	$\log n$
Haussler–Welzl [SoCG'86]	n	$n^{1-1/(d(d+1)+1)+\varepsilon}$
Chazelle–Sharir–Welzl [SoCG'90]	n	$n^{1-1/(d+1)+\varepsilon}$
Matoušek [SoCG'91]	n	$n^{1-1/(d+1)} \operatorname{polylog} n$
Matoušek [FOCS'91]	n	$n^{1-1/\lceil d/2 \rceil} \operatorname{polylog} n$

"Curse of Dimensionality"

- all these alg'ms have exponential dependencies in d

  e.g., range tree's $O(\log^d n)$ time is sublinear only for $d \ll \log n / \log\log n$

  [C.'05: range tree still OK for $d \ll 0.29 \log n$]

  [Short Proof:
  $$Q_d(n) \le Q_d(n/2) + Q_{d-1}(n) \;\; \Rightarrow \;\; O(\tbinom{\log n + d}{d})$$]

In Very High Dimensions...

- Let $M(n,d,n)$ be time to multiply $n \times d$ & $d \times n$ matrix
- Folklore: offline ℓ_2 nearest neighbor search in $O(M(n,d,n))$ time

  e.g., $O^*(n^2)$ for $d \ll n^{0.30}$ [Coppersmith'82 ... Le Gall'12]
- Matoušek'91: offline dominance range search in $O(M(n,ds,n) + dn^2/s)$ time

  e.g., $O^*(n^2)$ for $d \ll n^{0.15}$ (by picking $s = d$)

Focus of This Talk

For what d can we obtain subquadratic exact alg'ms?

Part I: Range Trees Strike Back

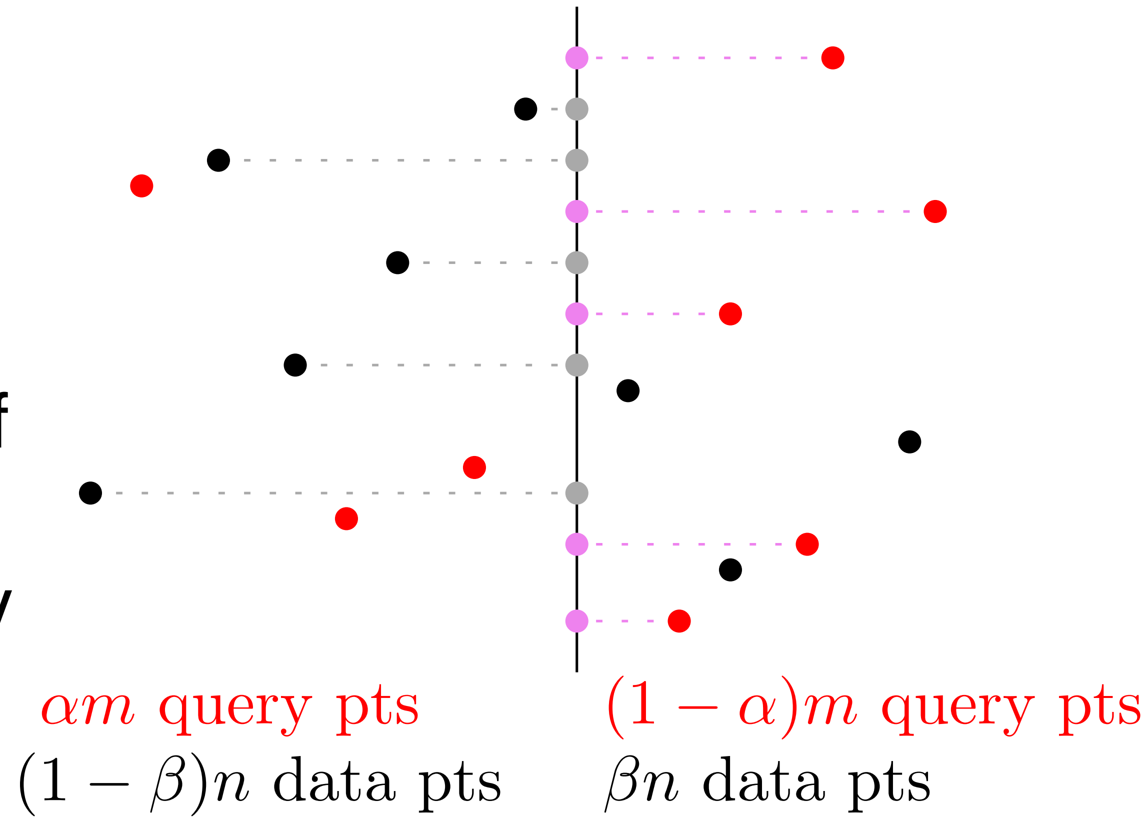
(or, How CG can help nongeometric problems. . .)

Surprisingly. . .

- range-tree-like divide&conquer can still work well beyond log dimensions!
- Impagliazzo–Lovett–Paturi–Schneider'14: offline dominance range search in $\boxed{n^{2-1/\widetilde{O}(c^{15})}}$ time for $d = c \log n$
- C. [SODA'15]: improves to $\boxed{n^{2-1/\widetilde{O}(c)}}$ time
  (e.g., subquadratic for $d \ll \log^2 n$)

Range-Tree-Like Method for Offline Dominance [Impagliazzo–Lovett–Paturi–Schneider'14]

- divide by some vertical hyperplane
- recurse on left & right
- recurse on projection of left data pts with projection of right query pts



$$\Rightarrow T_d(m,n) = T_d(\alpha m, (1-\beta)n) + T_d((1-\alpha)m, \beta n) + T_{d-1}((1-\alpha)m, (1-\beta)n)$$

- pick dividing hyperplane s.t. $\alpha = \beta$

Analysis

[C. (SODA'15)]

$$T_d(m,n) \;=\; T_d(\alpha m, (1-\alpha)n) + T_d((1-\alpha)m, \alpha n) + T_{d-1}((1-\alpha)m, (1-\alpha)n)$$

- How to solve recurrence: guess!
- suppose $T_d(m,n) \le (1+\delta)^d (nm^{1-\varepsilon} + mn^{1-\varepsilon})$
- induction goes through by picking $\delta \approx 1/c^2$, $\varepsilon \approx 1/c$

  $\Rightarrow$ $\boxed{n^{2-1/\widetilde{O}(c)}}$ time

Remark: Online Dominance?

- C.'17(?): $O(n^{1+\varepsilon})$ space/preproc. time, $n^{1-1/\widetilde{O}(c)}$ expected query time

  by k-d-tree-like divide&conquer (lopsided, randomized, with secondary structures for lower-dimensional projections. . .)

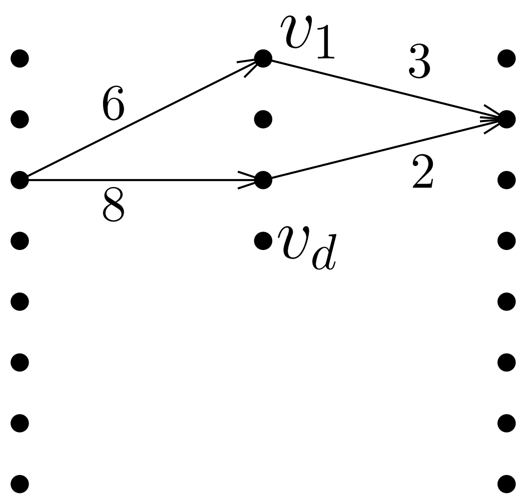
Application 1: All-Pairs Shortest Paths (APSP)

- given a real-weighted dense graph with n vertices, compute shortest path from s to t for every pair (s, t)

- Textbook [Floyd–Warshall]: $O(n^3)$ time
- Subcubic?

From APSP to Dominance [C.'05]

- Main Case: $n \times d \times n$ tripartite graph



Subproblem: which shortest paths go through v_1?

- same as reporting all pairs (s, t) s.t. $\forall k = 1, \ldots, d$,

$$w(s, v_1) + w(v_1, t) \ \le \ w(s, v_k) + w(v_k, t), \quad \text{i.e.,}$$
$$w(s, v_1) - w(s, v_k) \ \le \ w(v_k, t) - w(v_1, t)$$

$\Rightarrow$ offline dominance for n pts in d dimensions!

- implies APSP alg'm in $\boxed{\widetilde{O}(n^3/\log^2 n)}$ time
- C.'17(?): combine with bit packing tricks
  $\Rightarrow$ APSP alg'm in $\boxed{\widetilde{O}(n^3/\log^3 n)}$ time (combinatorial)
- extends earlier $\widetilde{O}(n^3/\log^3 n)$ combinatorial alg'm for Boolean matrix multiplication [C. (SODA'15)]

Application 2: 0-1 Integer Linear Programming (ILP)

- Find an assignment of n variables $x_1, \ldots, x_n \in \{0, 1\}$ to satisfy cn given constraints

- Beating brute-force 2^n time?

From 0-1 ILP to Dominance

[Impagliazzo–Lovett–Paturi–Schneider'14]

- same as finding $(x_1, \ldots, x_{n/2}), (x_{n/2+1}, \ldots, x_n) \in \{0,1\}^{n/2}$ s.t. $\forall i = 1, \ldots, cn$,

$$a_{i,1}x_1 + \cdots + a_{i,n}x_n \;\le\; b_i, \quad \text{i.e.,}$$

$$a_{i,1}x_1 + \cdots + a_{i,n/2}x_{n/2} \;\le\; b_i - a_{i,n/2+1}x_{n/2+1} - \cdots - a_{i,n}x_n$$

$\Rightarrow$ offline dominance for $2^{n/2}$ pts in cn dimensions!

- implies 0-1 ILP alg'm in $(2^{n/2})^{2-1/\widetilde{O}(c)} =$ $\boxed{2^{(1-1/\widetilde{O}(c))n}}$ time

Part II: The Polynomial Method

(or, How nongeometric techniques can help CG. . .)

Recent Breakthrough

- Williams [STOC'14]:
  APSP in $n^3/2^{\Omega(\sqrt{\log n})}$ rand. time

- Abboud–Williams–Yu [SODA'15]:
  offline dominance range detection in Boolean case in $n^{2-1/O(\log c)}$ rand. time for $d = c \log n$

  e.g., $n^2/2^{\Omega(\sqrt{\log n})}$ for $d \ll 2^{\sqrt{\log n}}$

  (derandomized by C.–Williams [SODA'16])

- C.'17(?): same for offline dominance in general case

Polynomial Method for Offline Boolean Dominance [Abboud–Williams–Yu (SODA'15)]

- Given group of s data pts $\mathbf{x} \in (\{0,1\}^d)^s$ & a query pt $\mathbf{y} \in \{0,1\}^d$, define predicate

  $F(\mathbf{x}, \mathbf{y})$ = [all s pts in $\mathbf{x}$ are not dominated by $\mathbf{y}$]

- Goal: evaluate F over all n/s groups $\mathbf{x}$ & all n query pts $\mathbf{y}$

- Approach: express F as a multivariate polynomial!

Polynomial Method for Offline Boolean Dominance [Abboud–Williams–Yu (SODA'15)]

- e.g., $F(\mathbf{x},\mathbf{y}) = x_1y_2 + 5x_2y_1y_2 + 4x_1x_2y_1$
  $= (x_1, 5x_2, 4x_1x_2) \cdot (y_2, y_1y_2, y_1)$

- goal reduces to computing dot products between n/s vectors & n vectors with
  dimension $d' = \#$ monomials in F

  i.e., multiply an $n/s \times d'$ with $d' \times n$ matrix
  $\Rightarrow$ $\boxed{\widetilde{O}(n^2/s)}$ time for $d' \ll (n/s)^{0.1}$ [Coppersmith'82]

Polynomial Method for Offline Boolean Dominance [Abboud–Williams–Yu (SODA'15)]

- New Problem: express

$$F(\mathrm{x}, \mathrm{y}) = \text{[all } s \text{ pts in } \mathrm{x} \text{ are not dominated by } \mathrm{y}\text{]}$$
$$= \bigwedge_{i=1}^{s} \bigvee_{j=1}^{d} (x_{ij} \wedge \overline{y_j})$$

  as a polynomial with small # monomials

- aim for small degree. . .

OR Polynomial [Razborov–Smolensky'87]

- Subproblem: express $\bigvee_{j=1}^{d} z_j$ as a polynomial
- Easy Sol'n: $1 - \prod_{j=1}^{d} (1 - z_j)$ (but degree $= d$, too big!)

OR Polynomial [Razborov–Smolensky'87]

- Subproblem: express $\bigvee_{j=1}^{d} z_j$ as a polynomial
- Rand. Sol'n: take random vector $\mathbf{r} \in \{0,1\}^d$
  return $\sum_{j=1}^{d} r_j z_j \pmod 2$
- degree $= 1$!
- 1-sided error prob. $= 1/2$
- can lower error prob. to $1/s$ by repeating $\log s$ times & taking product $\Rightarrow$ degree $\approx \log s$

$$F(\mathbf{x}, \mathbf{y}) = \bigwedge_{i=1}^{s} \bigvee_{j=1}^{d} (x_{ij} \wedge \overline{y_j})$$

- apply Razborov–Smolensky twice

  (for top AND, use de Morgan & const error prob.)

- degree $\approx \log s$

- # monomials $\approx s \cdot \binom{d}{\log s}$

$$= s \cdot \binom{c \log n}{\alpha \log n} \qquad \text{for } d = c \log n,\ s = n^{\alpha}$$

$$\leq (c/\alpha)^{O(\alpha \log n)}$$

$$= n^{O(\alpha \log(c/\alpha))}$$

$$\ll (n/s)^{0.1} \qquad \text{for } \alpha \approx \boxed{1/O(\log c)}$$

Offline Dominance: From Boolean to General [C.'17(?)]

- Idea: combine with range-tree divide&conquer!
- assume all but the first j coordinates are in $\{1, \ldots, b\}$

$$T_{d,j}(n) = \begin{cases} n^{2-1/O(\log(bc))} & \text{if } j = 0 \\ bT_{d,j}(n/b) + T_{d,j-1}(n) & \text{else} \end{cases}$$

$\Rightarrow$ $\boxed{n^{2-1/O(\log c)}}$ (by guessing, picking $b = c^{O(1)} \ldots$)

More Developments

- Alman–Williams [FOCS'15]: offline nearest/farthest neighbor search in Hamming case in $n^{2-1/\widetilde{O}(c)}$ rand. time for $d = c \log n$

- Alman–C.–Williams'16(?): improves to $n^{2-1/\widetilde{O}(\sqrt{c})}$

  e.g., subquadratic for $d \ll \log^3 n$

Polynomial Method for Offline Hamming Farthest Neighbor [Alman–Williams (FOCS'15)]

- again consider group of s data pts $\mathbf{x} \in (\{0,1\}^d)^s$ & a query pt $\mathbf{y} \in \{0,1\}^d$

- New Problem: express

$$
\begin{aligned}
F(\mathbf{x},\mathbf{y}) &= [\text{one of the } s \text{ pts in } \mathbf{x} \text{ has dist.} > t \text{ from } \mathbf{y}] \\
&= \bigvee_{i=1}^{s} \left[\sum_{j=1}^{d} (x_{ij} - y_j)^2 > t^2 \right]
\end{aligned}
$$

as a polynomial with small # monomials/degree

Threshold Polynomial: Method 1

[Alman–Williams (FOCS'15)]

- Subproblem: express $\left[\sum_{j=1}^{d} z_j > t\right]$ as a polynomial
- Idea 1: take rand. sample R of size $d/2$
- degree $D(d) \approx D(d/2) + \sqrt{d \log s}$

$\uparrow$

interpolating polynomial

$\approx \sqrt{d \log s}$ (optimal)

$$F(\mathbf{x},\mathbf{y}) \;=\; \sum_{i=1}^{s} \left[\sum_{j=1}^{d} (x_{ij} - y_j)^2 > t^2 \right]$$

- degree $\approx \sqrt{d \log s}$
- # monomials $\approx \; s \cdot \binom{d}{\sqrt{d \log s}}$

$$\approx \; s \cdot \binom{c \log n}{\sqrt{c\alpha} \log n} \qquad \text{for } d = c \log n, \;\; s = n^{\alpha}$$

$$\leq \; (c/\alpha)^{O(\sqrt{c\alpha} \log n)}$$

$$= \; n^{O(\sqrt{c\alpha} \log(c/\alpha))}$$

$$\ll \; (n/s)^{0.1} \qquad\qquad \text{for } \alpha \approx \boxed{1/\widetilde{O}(c)}$$

Threshold Polynomial: Method 2

[Alman–C.–Williams'16(?)]

- Subproblem: express $\left[\sum_{j=1}^{d} z_j > t\right]$ as a polynomial

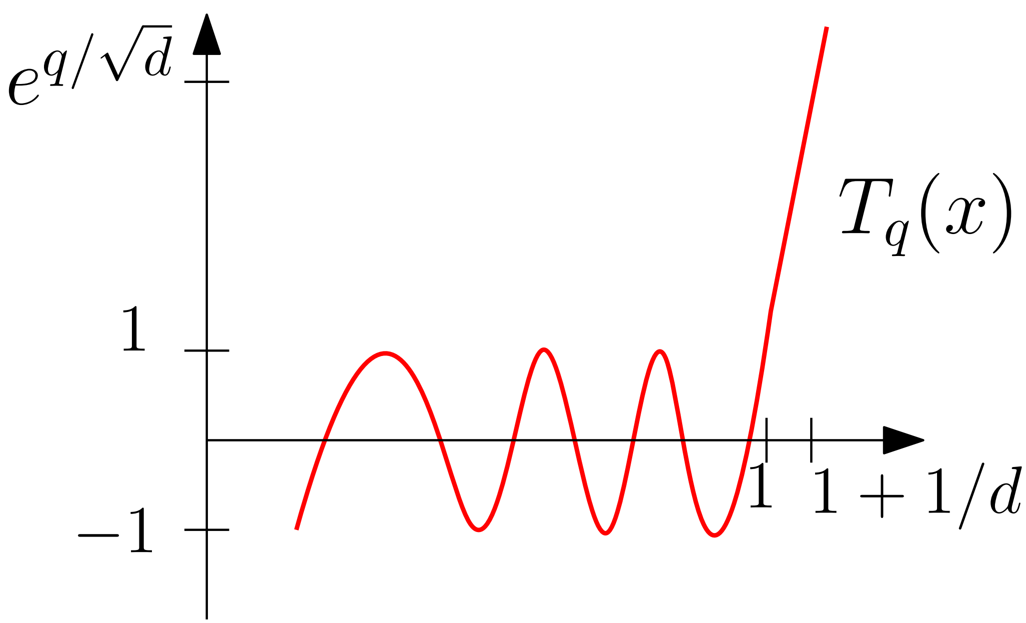
- Idea 2: just take $T_q\left(\sum_{j=1}^{d} z_j / t\right)$, where T_q is the degree-q Chebyshev polynomial (1854)!

$$
\begin{aligned}
T_q(x) &= \cos(q \arccos(x)) \\
&= \cosh(q \operatorname{arcosh}(x)) \\
&= \sum_{i=0}^{\lfloor q/2 \rfloor} \binom{q}{2i} (x^2 - 1)^i x^{q-2i}
\end{aligned}
$$

Threshold Polynomial: Method 2

[Alman–C.–Williams'16(?)]

- Subproblem: express $\left[\sum_{j=1}^{d} z_j > t\right]$ as a polynomial
- Idea 2: just take $T_q\left(\sum_{j=1}^{d} z_j/t\right)$, where T_q is the degree-q Chebyshev polynomial (1854)!



false $\Rightarrow$ output $\in [-1, 1]$

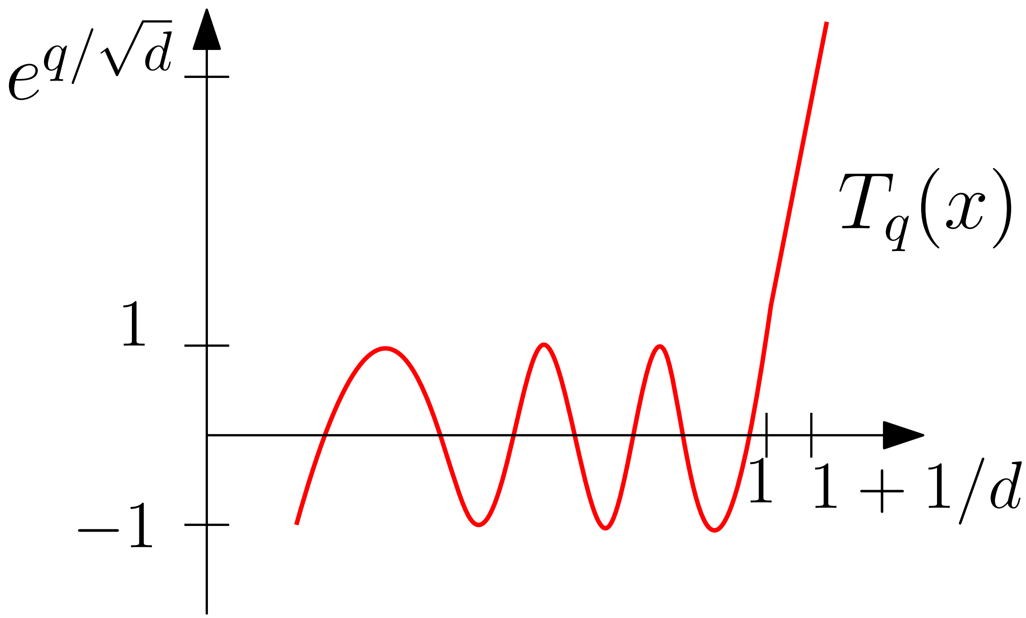
true $\Rightarrow$ output $> 100s$

by picking $q \approx \sqrt{d}\log s$

Threshold Polynomial: Method 2

[Alman–C.–Williams'16(?)]

- Subproblem: express $\left[\sum_{j=1}^{d} z_j > t\right]$ as a polynomial
- Idea 2: just take $T_q\left(\sum_{j=1}^{d} z_j/t\right)$, where T_q is the degree-q "discrete" Chebyshev polynomial (1864)!



false $\Rightarrow$ output $\in [-1, 1]$

true $\Rightarrow$ output $> 100s$

by picking $q \approx \sqrt{d \log s}$

(no improvement, sadly. . .)

Threshold Polynomial: Final Method

[Alman–C.–Williams'16(?)]

- Subproblem: express $\left[\sum_{j=1}^{d} z_j > t\right]$ as a polynomial
- Final Idea: combine!
- take rand. sample R of size r
- degree $\approx \sqrt{r \log s} + \sqrt{(d/\sqrt{r})\sqrt{\log s}\log s}$

$\uparrow$ Method 1 $\uparrow$ Method 2

$$\approx d^{1/3} \log^{2/3} s$$

$$F(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{s} \left[\sum_{j=1}^{d} (x_{ij} - y_j)^2 > t^2 \right]$$

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

$$= s \cdot \binom{c \log n}{c^{1/3} \alpha^{2/3} \log n} \quad \text{for } d = c \log n,\; s = n^{\alpha}$$

$$\leq (c/\alpha)^{O(c^{1/3}\alpha^{2/3} \log n)}$$

$$= n^{O(c^{1/3}\alpha^{2/3} \log(c/\alpha))}$$

$$\ll (n/s)^{0.1} \quad \text{for } \alpha \approx \boxed{1/\widetilde{O}(\sqrt{c})}$$

Remark: Offline Approximate Nearest Neighbor Search

- LSH [Indyk–Motwani'98, Andoni–Indyk'06]: $\widetilde{O}(n^{1+1/(1+\varepsilon)^2})$ rand. time
- Data-dependent LSH [Andoni–Indyk–Nguyen–Razenshteyn'14, Andoni–Razenshteyn'15]: $\widetilde{O}(n^{1+1/(2(1+\varepsilon)^2-1)}) = \boxed{n^{2-\Omega(\varepsilon)}}$
- G. Valiant [FOCS'12]: $\boxed{n^{2-\Omega(\sqrt{\varepsilon})}}$
- Alman–C.–Williams'16(?): $\boxed{n^{2-\widetilde{\Omega}(\varepsilon^{1/3})}}$

Final Remarks

- further consequences [Alman–C.–Williams'16(?)]:
  - faster exponential alg'ms for MAX-SAT with cn constraints
  - circuit lower bounds (for depth-2 threshold circuits. . .)
- many open problems:
  - online nearest neighbor search, for general ℓ_2?
  - on threshold polynomials: $o(d^{1/3})$ degree?
  - on range-tree-like methods: lower bounds?

THE END