Instance-Optimal Geometric Algorithms

Timothy Chan
School of CS
U. Waterloo

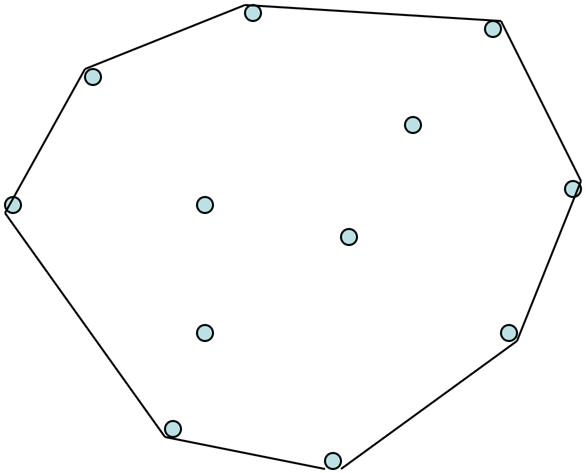
joint work with
Peyman Afshani (MADALGO, Aarhus U)
Jérémy Barbay (U. Chile)

Theme

- Beyond worst-case analysis
- "Adaptive" algorithms

[a theory w. connections to output-sensitive alg'ms, average-case alg'ms, decision-tree lower bds, partition trees, adversary arguments, entropy, distribution-sensitive data structures…]

Example: 2D Convex Hull



Example: 2D Convex Hull

- Background:
  - O(n log n) time alg'ms
    - Graham's scan'72
    - Divide&conquer [Preparata,Hong'77]
    - Randomized incremental [Clarkson,Shor'88]
  - Ω(n log n) lower bd [Ben-Or'83]

2D Convex Hull (Cont'd)

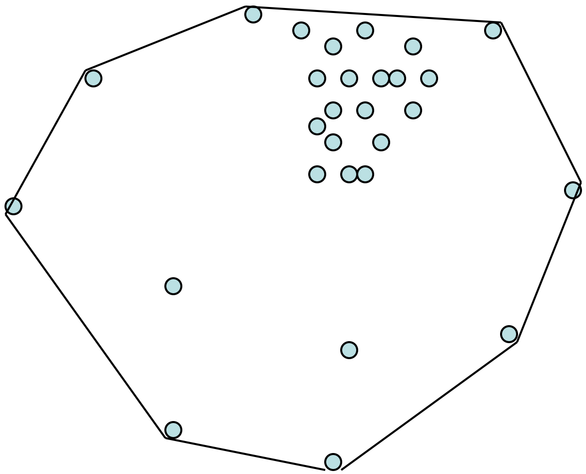
- "Output-Sensitive" Alg'ms
  - O(nh) time
    - Jarvis' march'73
  - O(n log h) time
    - Kirkpatrick,Seidel'86 ["ultimate… ?"]
    - Clarkson,Shor'88 [random sampling]
    - Chan,Snoeyink,Yap'95 [prune,divide&conquer]
    - Chan'95 [grouping+Jarvis]
  - Ω(n log h) lower bd

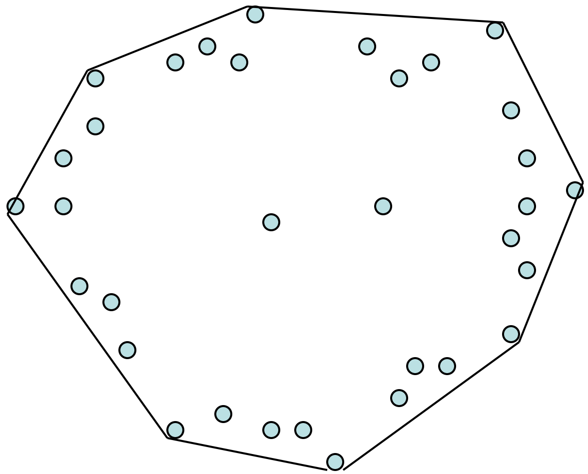
2D Convex Hull (Cont'd)

- "Average-Case" Alg'ms
  - O(n) expected time for
    - uniformly distributed pts inside square/disk/…
    - normally distributed pts [Bentley,Shamos'78]

2D Convex Hull (Cont'd)

- Easy vs. Hard Point Sets





New Result for 2D Convex Hull

- An adaptive alg'm that is optimal
  in terms of every parameter imaginable !

New Result for 2D Convex Hull

- An adaptive alg'm that is optimal for every point set !!

New Result for 2D Convex Hull

- An adaptive alg'm that is optimal for every instance !!

Def'n of "Instance Optimality" (First Attempt)

- Let $T_A(S)$ = runtime of alg'm A on input sequence S
- Let $OPT(S)$ = min $T_A(S)$ over all alg'ms A
- A is instance-optimal if $T_A(S) \leq O(1) \cdot OPT(S) \;\; \forall S$

Def'n of "Instance Optimality"
(First Attempt)

- Let $T_A(S)$ = runtime of alg'm A on input sequence S
- Let $OPT(S)$ = min $T_A(S)$ over all alg'ms A
- A is instance-optimal if $T_A(S) \le O(1) \cdot OPT(S) \quad \forall S$

... but not possible for 2D convex hull !!

[for every input sequence S, there is an alg'm with runtime O(n) on S]

Our Def'n of "Instance Optimality"

- Let $T_A(S)$ = max runtime of alg'm A over all permutations of input set S

- Let $OPT(S)$ = min $T_A(S)$ over all alg'ms A

- A is instance-optimal in the order-oblivious setting if $T_A(S) \leq O(1) \cdot OPT(S) \quad \forall S$

Our Def'n of "Instance Optimality"

- Let $T_A(S)$ = max runtime of alg'm A over all permutations of input set S

- Let $OPT(S)$ = min $T_A(S)$ over all alg'ms A

- A is instance-optimal in the order-oblivious setting if $T_A(S) \leq O(1) \cdot OPT(S) \;\; \forall S$

[subsumes output-sensitive alg'ms, & any alg'm that does not exploit input order, etc.]

Our Def'n of "Instance Optimality"
(Slightly Stronger Version)

- Let $t_A(S)$ = average runtime of alg'm A over all permutations of input set S

- Let $opt(S)$ = $\min t_A(S)$ over all alg'ms A

- A is instance-optimal in the random-order setting if $T_A(S) \le O(1) \cdot opt(S) \;\; \forall S$

Our Def'n of "Instance Optimality" (Slightly Stronger Version)

- Let $t_A(S)$ = average runtime of alg'm A over all permutations of input set S
- Let $opt(S)$ = $\min t_A(S)$ over all alg'ms A
- A is instance-optimal in the random-order setting if $T_A(S) \leq O(1) \cdot opt(S) \;\; \forall S$

[subsumes average-case alg'ms for any distribution, & randomized incremental alg'ms, etc.]

Related Work on Instance Optimality

- Fagin,Lotem,Naor'03 [in database]
- Competitive binary search trees [Sleator,Tarjan'85's "dynamic optimality conjecture"]
- Competitive analysis of on-line alg'ms
- Various adaptive alg'ms, e.g.,
  [Demaine,Lopez-Ortiz,Munro'00: set union/intersection;
  Baran,Demaine'04: approx problems about "black-box" curves; etc.]

New Result for 2D Convex Hull

- An alg'm that is instance-optimal in the order-oblivious (& random-order) setting

Outline

1. What is OPT(S)?
2. Upper Bound
3. Lower Bound
4. Applications to Other Problems

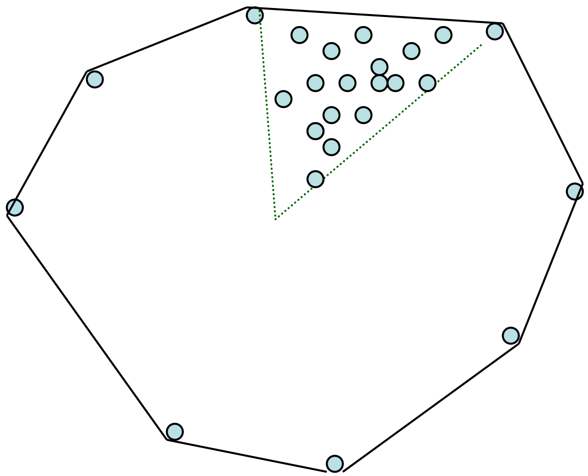
1. What is OPT(S)?

A Measure of Difficulty

- Given point set S of size n
- Consider a partition P of S into subsets S_i s.t.

  each subset can be enclosed in a triangle inside (*) convex hull(S)

- Let $H(P) := \sum_i |S_i| \log (n/|S_i|)$



A Measure of Difficulty

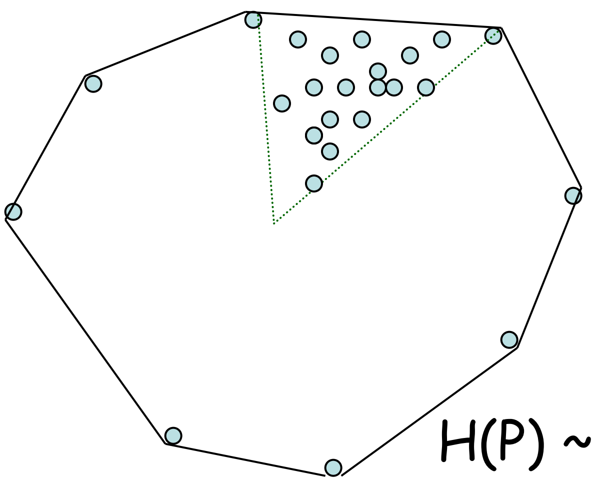
- Given point set S of size n
- Consider a partition P of S into subsets S_i s.t.
  each subset can be enclosed in a triangle inside convex hull(S) (*)
- Let $H(P) := \sum_i |S_i| \log (n/|S_i|)$

A Measure of Difficulty

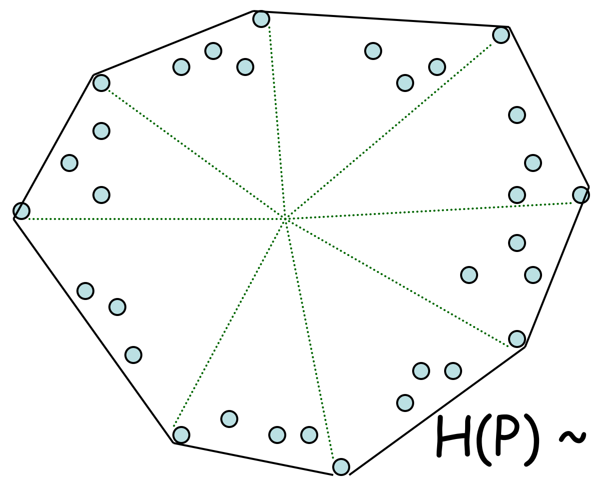
- Given point set S of size n
- Consider a partition P of S into subsets S_i s.t.

  each subset can be enclosed in a triangle inside (*) convex hull(S)

- Let $H(P) := \sum_i |S_i| \log (n/|S_i|)$



H(P) ~ n + h log n



H(P) ~ n log h

A Measure of Difficulty

- Given point set S of size n
- Consider a partition P of S into subsets S_i s.t.

  each subset can be enclosed in a triangle inside (*)
  convex hull(S)

- Let $H(P) := \sum_i |S_i| \log (n/|S_i|)$

- Define the difficulty of S to be

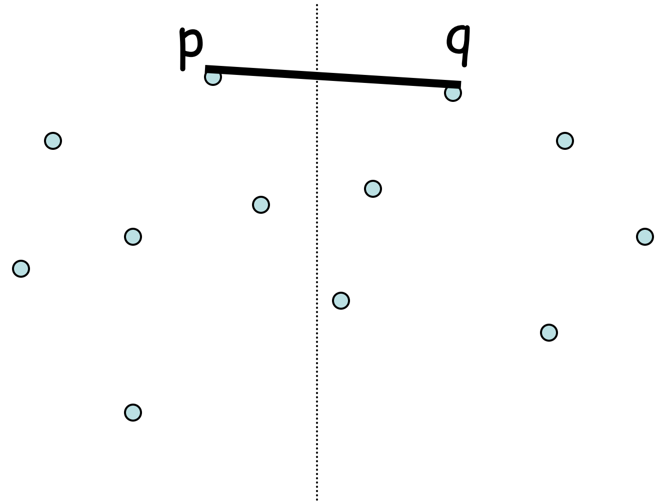
  H(S) := min H(P) over all valid partitions P
  satisfying (*)

Connections

- Multiset sorting requires time $O(\sum_i n_i \log (n/n_i))$ for multiplicities n_i
- Biased search trees require average query time $O(\sum_i p_i \log (1/p_i))$ (the entropy) for probabilities p_i
- $H(P) = \sum_i |S_i| \log (n/|S_i|)$ corresponds to the "entropy" of the partition P [after dividing by n]
- Sen,Gupta'99…

2. Upper Bound

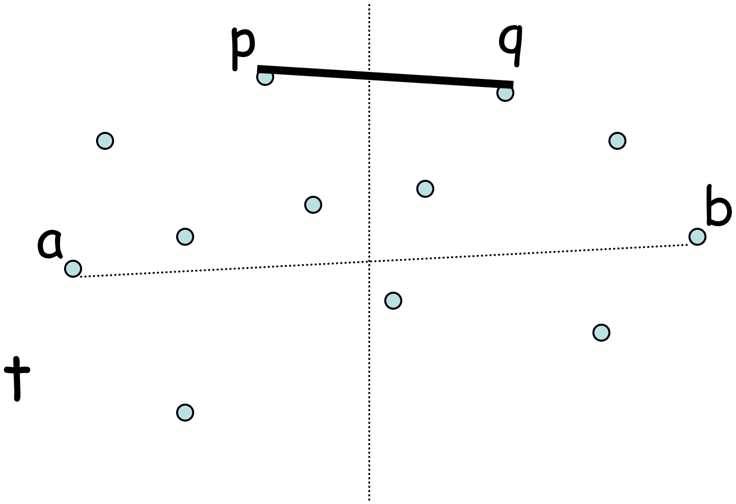
Kirkpatrick,Seidel's Alg'm



UpperHull(S):

1. if $|S| \leq 2$ return …
2. x_m = median x in S
3. find hull edge pq ("bridge") at $x = x_m$ [by 2D LP]
4. prune all pts below pq
5. UpperHull({all pts left of $x = x_m$})
6. UpperHull({all pts right of $x = x_m$})

Kirkpatrick,Seidel's Alg'm (Slightly Modified)

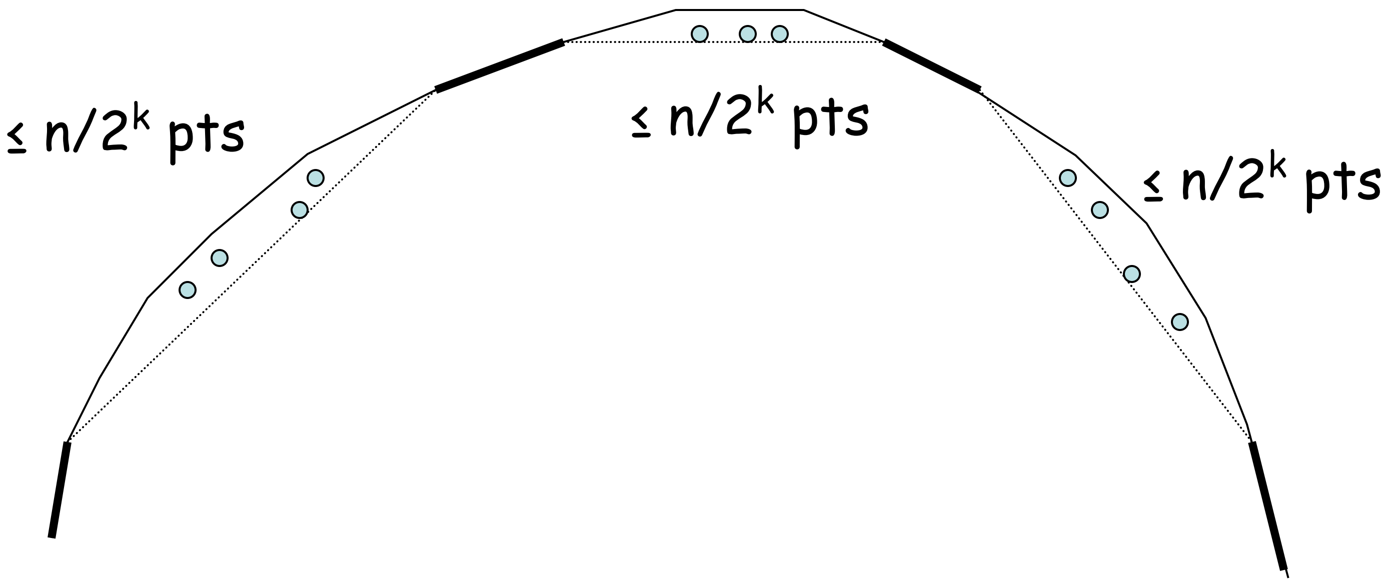


UpperHull(S):

0. if $|S| \le 2$ return ...
1. prune all pts below ab, where
   a = leftmost pt, b = rightmost pt
2. x_m = median x in S
3. find hull edge pq ("bridge") at $x = x_m$ [by 2D LP]
4. prune all pts below pq
5. UpperHull({all pts left of $x = x_m$})
6. UpperHull({all pts right of $x = x_m$})

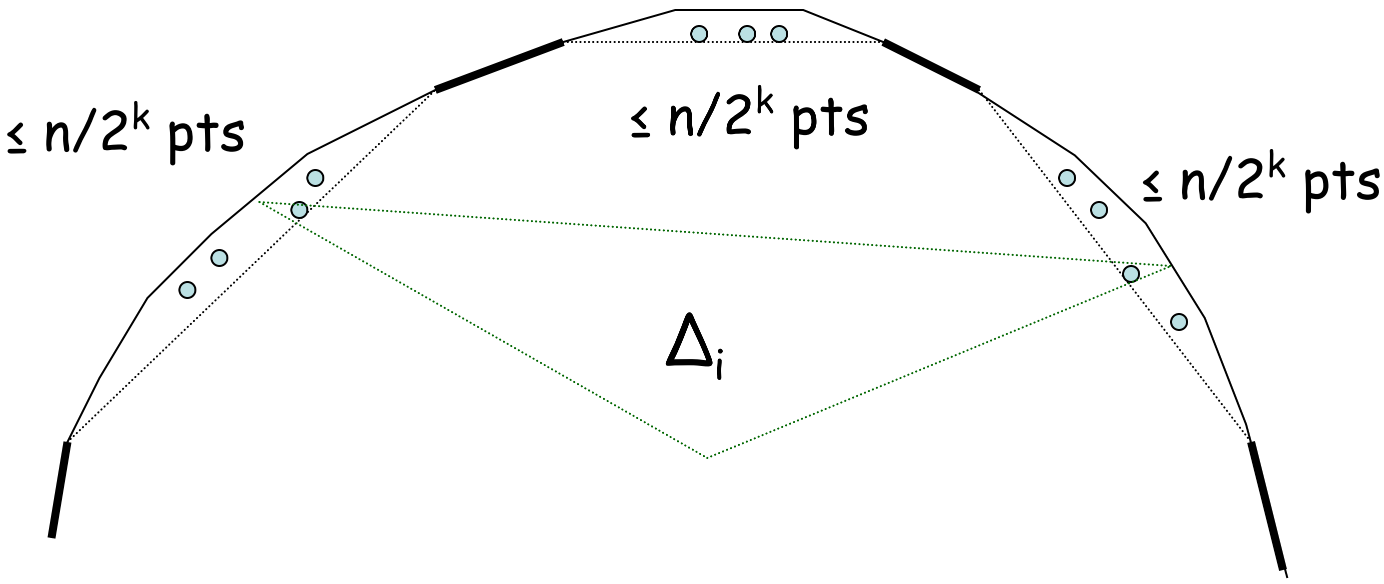
Analysis

- At level k of recursion:



Analysis

- At level k of recursion:



- Let P be any valid partition
- Let S_i be any subset of P, enclosed in triangle Δ_i

$\Rightarrow$ # pts in S_i that survive level k $\leq$ min{ $|S_i|$, $3n/2^k$ }

$\Rightarrow$ total # pts that survive level k $\leq$ $O(\sum_i \min\{|S_i|, n/2^k\})$

Analysis (Cont'd)

- Runtime

  $\leq O(\sum_k \sum_i \min\{ |S_i|, n/2^k\})$

  $= O(\sum_i \sum_k \min\{ |S_i|, n/2^k \})$

  $= O(\sum_i (|S_i| + \ldots + |S_i| + |S_i|/2 + |S_i|/4 + \ldots))$

  log $(n/|S_i|)$ times

  $= O(\sum_i |S_i| \log (n/|S_i|)) = O(H(P))$

$\Rightarrow$ Runtime $\leq O(\min_P H(P)) = O(H(S))$

3. Lower Bound

Traditional $\Omega(n \log n)$ Pfs (via Topology)

- Van Emde Boas'80: linear decision trees
  [but convex hull not solvable in this model !]
- Yao'82: quadratic decision tree
- Steeles,Yao'82: const-deg algebraic decision trees
  [but not quite successful for convex hull…]
- Ben-Or'83: const-deg algebraic decision trees
  & algebraic computation trees

… but none of these gives instance-specific lower bds !

A Different, Simple $\Omega(n \log n)$ Pf
(No Topology Required !)

- Toy Problem: given n pts $x_1, \ldots, x_n$ in R^1, are they distinct?
- Pf by adversary argument
- Simulate alg'm on unknown input
- Maintain an interval I_i for each x_i (initially $I_i = [0,1]$)

Simple $\Omega(n \log n)$ Pf (Cont'd)

- When alg'm compares x_i ? x_j :
  - if midpoint(I_i) < midpoint(I_j) then
    set $I_i \leftarrow$ left half of I_i
    $I_j \leftarrow$ right half of I_j & declare "<"
  - else similar

- Let depth(x_i) := log (1/length(I_i))
- After T comparisons, total depth ≤ O(T)

- At the end, can't have 2 pts whose intervals coincide/overlap [otherwise, answer could change]

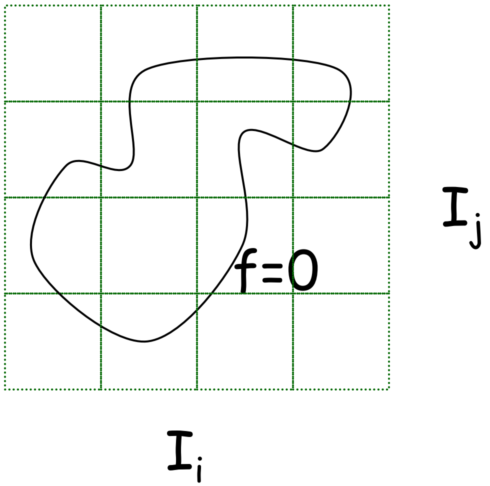
$\Rightarrow$ can't have > n/2 pts with depth ≤ log(n/2)

$\Rightarrow$ T ≥ Ω(total depth) ≥ $\Omega(n \log n)$ Q.E.D.

Simple $\Omega(n \log n)$ Pf (Generalized Version)

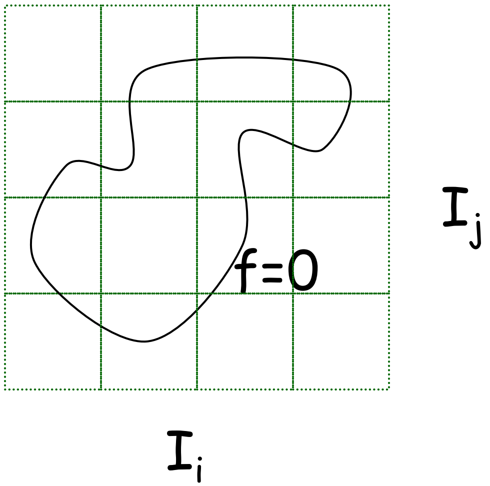
- When alg'm tests $f(x_i, x_j)$? 0 for const-deg alg. fn f:
  - take r grid subintervals I_i' of I_i
    r grid subintervals I_j' of I_j
  - among the r^2 grid cells $I_i' \times I_j'$,
    f = 0 intersects O(r) cells

  $\Rightarrow$ for suff. large const r,
  can set $I_i \leftarrow$ one of I_i'
  $I_j \leftarrow$ one of I_j' s.t. sign(f) is determined



Simple Ω(n log n) Pf (Generalized Version)

- When alg'm tests $f(x_i, x_j)$? 0 for const-deg alg. fn f:
  - take r grid subintervals I_i' of I_i
    r grid subintervals I_j' of I_j
  - among the r^2 grid cells $I_i' \times I_j'$,
    f = 0 intersects O(r) cells



  $\Rightarrow$ for suff. large const r,
    can set $I_i \leftarrow$ one of I_i'
    $I_j \leftarrow$ one of I_j' s.t. sign(f) is determined

- Note: extends to decision trees w. const-deg algebraic test fns w. any const # of args [Moran,Snir,Manber'85 had diff. pf for arbitrary test fns w. const # of args, via Ramsey theory, but it's not instance-specific…]

An Instance-Specific Lower Bd Pf for Convex Hull

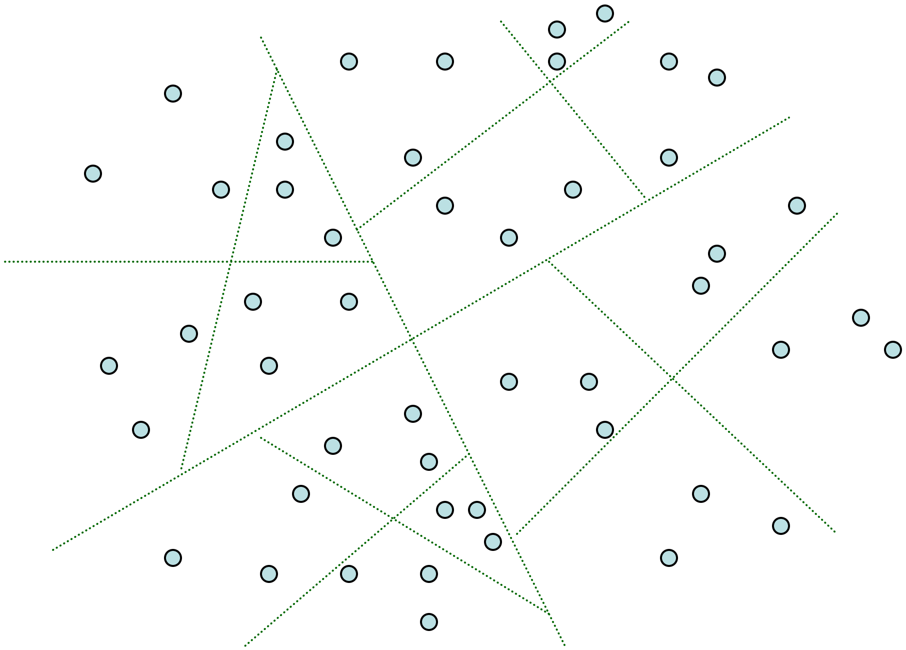
- Note: holds for decision trees w. multilinear test fns w. const # of args

- Ex: $f((x_1,y_1),(x_2,y_2)) = x_1y_2 + x_2y_1$ is multilinear
  $f((x_1,y_1),(x_2,y_2)) = x_1y_1 + x_2y_2$ is not
  the determinant is…

Instance-Specific Lower Bd Pf (Cont'd)

- Partition Thm: [Willard'82,…,Matoušek'91]

  Any point set S can be partitioned into ~ r subsets S_i of size n/r, each enclosed in (disjoint) cell Δ_i of size $\tilde{O}(1)$ s.t. each line crosses $O(r^{1-\varepsilon})$ cells



Instance-Specific Lower Bd Pf (Cont'd)

- Recurse $\Rightarrow$ partition tree
  where each cell at depth k contains n/r^k pts
  i.e., depth of cell = $\log_r (n/(\# \text{ pts}))$

- Make cell Δ a leaf if Δ is inside convex hull(S)
- Let P** be the partition formed by the leaves

- Pf by adversary argument again
- Maintain a cell Δ_p for each pt p in S (initially, Δ_p = root)

Instance-Specific Lower Bd Pf (Cont'd)

- When alg'm tests $f(p,q)$? 0:
  - take the ~ r subcells Δ_p' of Δ_p
    ~ r subcells Δ_q' of Δ_q
  - among the ~ r^2 cells $\Delta_p' \times \Delta_q'$, $f = 0$ intersects $\tilde{O}(r^{2-\varepsilon})$ cells [since f is multilinear & line crossing # is $O(r^{1-\varepsilon})$]

  $\Rightarrow$ can set $\Delta_p \leftarrow$ one of Δ_p'
  $\Delta_q \leftarrow$ one of Δ_q' s.t. sign(f) is determined

- When Δ_p = a leaf, fix p to an unassigned pt in $S \cap \Delta_p$
- Minor note: don't let $> |S \cap \Delta'|$ pts go under a child Δ' ...

$\Rightarrow$ At the end, get a permutation of S

Instance-Specific Lower Bd Pf (Cont'd)

- Let depth(p) := depth of Δ_p in partition tree
- After T comparisons, total depth ≤ O(T)

- At the end, each Δ_p must be a leaf [otherwise convex hull could change]

$$\Rightarrow T \geq \Omega(\text{total depth})$$
$$\geq \Omega\left(\sum_{\text{leaf } \Delta} |S \cap \Delta| \text{ depth}(\Delta) \right)$$
$$\geq \Omega\left(\sum_{\text{leaf } \Delta} |S \cap \Delta| \log (n/|S \cap \Delta|) \right)$$
$$= \Omega(H(P^{**})) \geq \boxed{\Omega(H(S))}$$

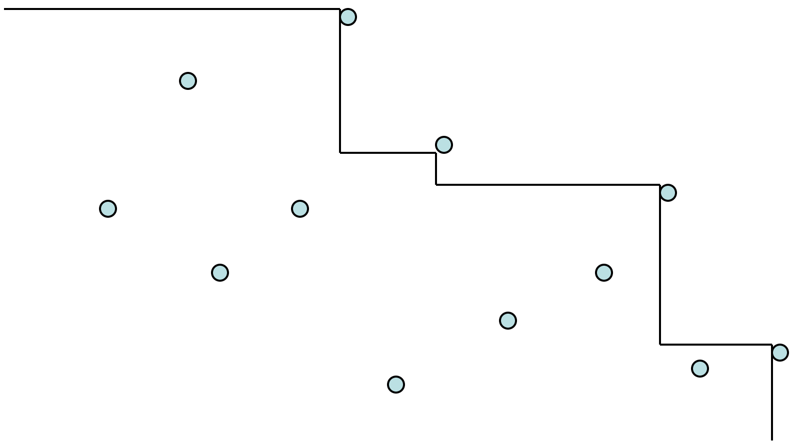
Q.E.D.

4. Other Applications

3D Convex Hull

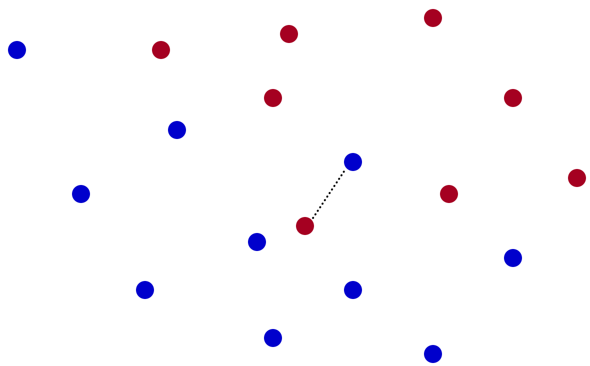
- Lower bd pf: same
- Upper bd: a new alg'm, using partition trees combined w. grouping [Chan'95] ...

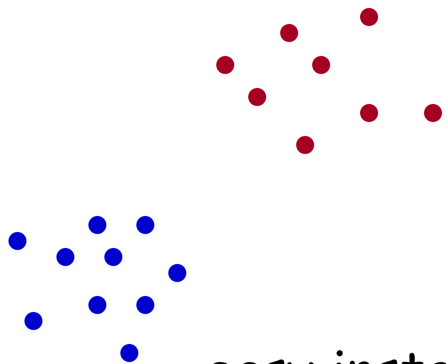
2D/3D Maxima



- Similar, except simpler: partition trees can be replaced by k-d trees

2D/3D Red-Blue Dominance

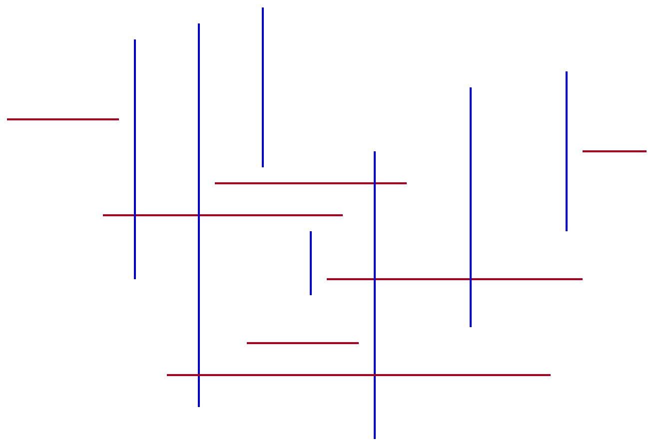


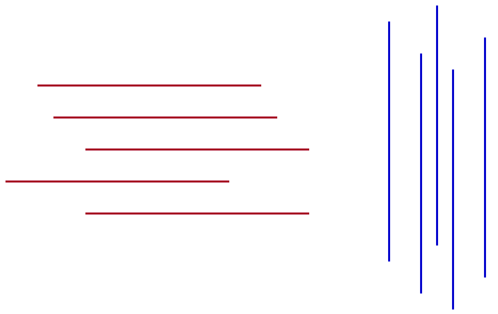


easy instance

- Consider a partition P of S into red/blue subsets S_i s.t.
  each red subset can be enclosed in a box where every pt in the box is dominated by exactly the same set of blue pts in S, & vice versa
- H(S) := min H(P) over all such partitions P

2D Orthogonal Segment Intersection

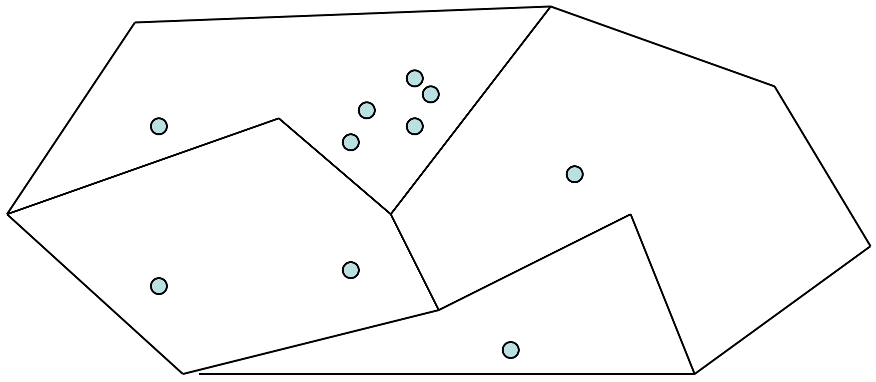




easy instance

- Lift each horizontal/vertical segment s into a red/blue point s* in 3D...

2D Offline Point Location



- Consider a partition P of S into subsets S_i s.t.
  each subset can be enclosed in a triangle completely inside one region
- H(S) := min H(P) over all such partitions P

2D Point Location Queries

$\Rightarrow$ Get a data structure with average query time O(H(S)/n), i.e., O(entropy)

- Note: re-proves known "distribution-sensitive" data structures [Arya,Malamatos,Mount'00; Iacono'01; etc.]

- Note: gets new distribution-sensitive data structures for many other query problems, e.g., 2D orthogonal range counting [answers open problem by Dujmovic,Howat,Morin'09], ...

Conclusions

- Specific open problems:
  - Nonorthogonal red-blue segment intersection
  - Diameter/width of a 2D point set
  - Beyond multilinear decision trees

- Other instance-optimal/adaptive models? [order-dependent?]

- Problems w. worst-case complexity worse than n log n??
  [e.g., offline simplex range search]