CS632 Homework 4

Qilin Ye

November 13, 2025

Solution to problem 1. (1) Let $k = n/3$. We consider the dual where S is transformed into L, a collection of lines. From this, we obtain the arrangement $A(L)$ and let Λ_k, Λ_{n-k} be the k- and $(n-k)$-levels of $A(L)$.

We can always only consider half planes H defined by lines that pass through points $x \in S$. A line through x with slope t corresponds, in the dual space, to the point where x^* meets the vertical line at t. Consequently, a primal point x is a center point iff its dual line x^* lies everywhere between the k- and the $(n-k)$-levels.

Based on this observation, the problem reduces to dealing with the arrangement $A(L)$. To obtain such a line x^* (squeezed between the k- and the $(n-k)$-levels), consider the family of lines $y = ax - b$ (duals to candidate primal points (a, b)). For every edge of the k-level (resp. the $(n-k)$-level) with x-range over $[u, v]$, we need to require $y = ax - b$ to lie weakly above (resp. below) that edge on $[u, v]$. Because the difference of two linear functions is linear, it suffices to enforce each requirement at endpoints u, v (or if unbounded such as $[u, \infty)$, enforcing it at $u, u+1$ suffices). This yields $\mathcal{O}(n^2)$ linear half spaces in the (a, b)-plane; let F be their intersection. Assuming center points exist, $F \neq \varnothing$; we return any $(a, b) \in F$ and output that point embedded in $\mathbb{R}^2$.

Finally, building $A(L)$ and tracing the two levels each take $\mathcal{O}(n^2)$ time; combined with the $\mathcal{O}(n^2)$ time used to find F, the total complexity is also $\mathcal{O}(n^2)$.

- We use ϵ-approximation. Fix a constant size c to be determined later. We choose a subset $R \subset S$ of size c, compute its center point $\hat{x}$ by the previous part, and simply return $\hat{x}$. Any half plane containing a true center point contains $n/3$ points of S. Our goal here is to find a point where any half plane containing $\hat{x}$ contains at least $n/4$ points. Hence we allow the sampling to have an error of $n/12$. In other words, we need

$$\left| \frac{|H \cap R|}{|R|} - \frac{|H \cap S|}{|S|} \right| < \frac{1}{12}$$

for half planes H. We apply it here with $\delta = 1/2$ and $\epsilon = 1/12$. Observe there is no dependency on $n = |S|$ but rather the VC-dimension of the collection of half planes in $\mathbb{R}^2$, which is constantly 3. Without going over the $\epsilon - \delta$ notations, as all conditions are satisfied, we claim there *will* exist sufficiently large *constant* size c that achieves the said objective.

Solution to problem 2. (1) A line ℓ intersects a segment pq iff p, q lie on different sizes of ℓ; in the dual, this happens iff the point ℓ^* is between the lines p^* and q^*. Let E be the multiset of $2n$ endpoints of the segments in S and build an arrangement A on the $2n$ dual lines. Then the value

$$g(F) = |\{\text{segment } pq \mid \text{face } F \text{ lies between } p^*, q^*\}|$$

is well defined and constant for every face F of A. We precompute $g(F)$ for all faces. For query, given ℓ, compute ℓ^*, and locate the face containing ℓ^* in A, and simply return the g-value of that face.

The arrangement takes $\mathcal{O}(n^2)$ storage. To precompute $g(F)$, we traverse through the dual arrangement faces by BFS/DFS. To do so, for each endpoint e (of a line in S), we maintain one bit recording whether the current face is above or below e^*. Maintain the current total g-value of the face. When crossing an edge supported by e^*, flip that single bit; for every segment incident to e, the XOR of its two endpoint bits toggles, so g changes by ± per such segment. Consequently, an edge crossing on e^* costs $\mathcal{O}(\deg(e))$ time. Line e^* is cut into $\mathcal{O}(n)$ edges by the other $2n-1$ lines, so the total work charged to e^* over the whole traversal is $\mathcal{O}(n \cdot \deg(e))$; summing over all endpoints, all $g(F)$ is computed in

$$\sum_{e\in E} \mathcal{O}(n \cdot \deg(e)) = \mathcal{O}(n \cdot \sum_{e\in E} \deg(e)) = \mathcal{O}(n \cdot 2n) = \mathcal{O}(n^2)$$

as each segment contributes 1 to the degree of each of its two endpoints. Trivially, storing one integer per face, along with the $\mathcal{O}(n^2)$ total for arrangement, takes $\mathcal{O}(n^2)$ space.

(1) Given a line ℓ, let H^+, H^- be the open half planes above and below ℓ respectively. Then $\ell_i = (p_i, q_i)$ crosses ℓ if p_i, q_i belong to different $H^\pm$. Define $a = |\{i : p_i \in H^+\}|, B = |\{i : q_i \in H^+\}|$, and $C = |\{i : p_i, q_i \in H^+\}|$. Then the quantity we seek equals $A + B - 2C$, so we just need to perform three queries to compute A, B, C. The first two are easy; the last one needs a two-level partition tree.

To build the first tree, let $Q = \{p_1, \cdots, p_n\}$. Build a partition tree T on P for half-plane counting. Space used is $\mathcal{O}(n)$ with height $\mathcal{O}(\log n)$; query time is $\mathcal{O}(n^{1/2+\epsilon})$, and the query returns both the count A *and* a canonical decomposition of $P \cap H^+$ into disjoint node-sets P_v, where each node v is associated with $\Delta_v \in H^+$. For a second level tree, let $Q = \{q_1, \cdots, q_n\}$. We build another partition tree T_Q on Q to count $q_i \in H^+$. This gives B.

It remains to compute C. For every node $v \in T$, let P_v be the point set, $S_v = \{\ell_i : p_i \in P_v\}$ be the segments whose first endpoint lies in P_v, and $Q_v = \{q_i : \ell_i \in S_v\}$ be their second endpoints. We build a partition tree T'_v on Q_v. Each segment appears in Q_v for all nodes v on the path from the root to the leaf containing p_i, so this takes $\mathcal{O}(\log n)$ complexity. Overall, $\sum_v |Q_v| = \mathcal{O}(n \log n)$, and the total space and preprocessing time for all second-level trees is $\mathcal{O}(n \log n)$.

Finally, we reiterate the query algorithm. Given a line ℓ and half-plane H^+ (recall this is the one above it), compute A and the canonical nodes. Querying T with H^+ gives A, as well as the node set $\mathcal{C}$ with $\Delta_v \subset H^+$. We then compute B by quering T_Q with H^+. Finally, we compute C using the second-level tree. For each $v \in \mathcal{C}$, all $p_i \in P_i$ are in H^+, so we count how many of their second endpoints are also in H^+ by quering T'_v on Q_v with half plane H^+. Adding over $\mathcal{C}$ gives the desired quantity.

Solution to problem 3. We first note a few observations. For a fixed line ℓ, the radius that minimizes $\max_{p\in S} |d(p, \ell) - r|$ is

$$r_\ell = \frac{1}{2}\Big(\max_{p\in S} d(p, \ell) + \min_{p\in S} d(p, \ell) \Big)$$

and the optimal value for this ℓ is

$$\Phi(\ell) = \frac{1}{2}\Big(\max_{p\in S} d(p, \ell) - \min_{p\in S} d(p, \ell) \Big),$$

So the problem reduces to choosing an ℓ that minimizes the above quantity $\max d(\cdot, \ell) - \min d(\cdot, \ell)$.
Next, we note that the problem can be linearized (by squaring them), and the conditions of the coreset theorem holds. Consider the decision version with parameters T: does there exist (ℓ, r) such that $|d(p, \ell) - r| \leqslant T$ for all $p \in S$? From lecture, this implies the existence of a coreset $Q \subset S$ of size $m = \epsilon^{\mathcal{O}(1)}$, computable in $\mathcal{O}(m+n)$ time, such that for every cylinder C,

$$(1 - \epsilon/3) \max_{p \in S} |d(p, \ell_C) - r_C| \leqslant \max_{q \in Q} |d(q, \ell_C) - r_C| \leqslant (1 + \epsilon/3) \max_{p \in S} \leqslant (1 + \epsilon/3) \max_{p \in S} |d(p, \ell_C) - r_C|.$$

This completes the proof.

Solution to problem 4. We write the problem as:

$$\min r \qquad \text{such that} \qquad \|x - p\| \leqslant r \text{ for all } p \in S.$$

We treat each point $p \in S$ as a constraint $\|x - p\| \leqslant r$. For any $R \subset S$, let $x^*(R) = (c_R, r_R)$ be the smallest enclosing ball of R. The only properties the randomized LP algorithm needs are : (i) monotonicity, $R \subset T$ implies $r_R \leqslant r_T$, (ii) existing of a basis of constant size that determines the optimum, and (iii) the ability to test violations and solve the subproblem on that basis. We verify these conditions; the rest follows from the lecture.
(i) is trivial. For (ii), let (c^*, r^*) be the optimum SEB of R. Let $A = \{p \in R : \|p - c^*\| = r^*\}$ be the set of tight optimal constraints. Optimality of c^* for the convex function $f(x) = \max_{p \in R} \|x - p\|$ yields

$$0 \in \operatorname{conv}\{(c^* - p)/\|c^* - p\| : p \in A\} \implies c^* \in \operatorname{conv}(A).$$

A result due to Caratheodory states that there exists $B \subset A$ with $|B| \leqslant d + 1$ and $c^* \in \operatorname{conv}(B)$. The unique sphere through B is exactly the SEB of R. Thus every optimum is determined by a basis of at most size $d+1$. This proves (ii). Finally, (iii) can be realized by the following primitive operations:

- testing for violation – given (c, r) and p, check if $\|p - c\| > r$;
- solving on a basis: for $|B| \leqslant d+1$, solve the linear system obtained by subtracting the equations $\|c - p_i\|^2 = r^2$ to recover c and r.

Now, to runtime, using the mixed LP algorithm, each time the step that adds violators to C succeeds, one new basis element is added. The expected number of such iterations is $2(d+1)$, and the resulting recurrence is $T(n) = 2(d+1)T(3d\sqrt{n}) + \mathcal{O}(dn)$, which solves to $T(n) = \mathcal{O}(dn)$. *I think this is the best we can do using lecture material? Not $\mathcal{O}(n)$?*