CMSC 754: Fall 2026 Dave Mount

Homework 6

Due by the start of class on Tuesday, Nov 10. (Submissions will be through Gradescope.) Late homework submissions are not accepted (unless an extension has been prearranged), so please turn in whatever you have completed by the due date. Unless otherwise specified, you may assume that all inputs are given in *general position*.

If you present an algorithmic solution based on (traditional) plane sweep through a line arrangement, you may assume (for the sake of running times) that it can be adapted to use *topological plane sweep* without explaining the adaptation.

Problem 1. In machine learning, it is often desirable to determine whether two point sets in the plane can be partitioned by a line. When this is not possible, the goal is to find a line that achieves the best possible split. Let A and B be two point sets in the plane each having n points. Given a nonvertical line ℓ, let ℓ^+ and ℓ^- denote the halfplanes lying above and below ℓ, respectively. Define ℓ's *separation defect* (see Fig. 1) to be

$$\delta(\ell) \;=\; \min\big(|A \cap \ell^+| + |B \cap \ell^-|,\;\; |A \cap \ell^-| + |B \cap \ell^+|\big).$$

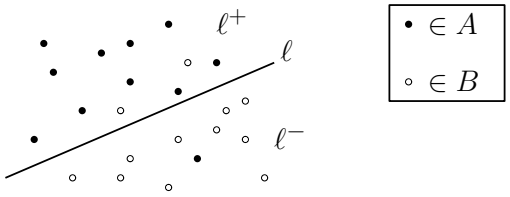


Figure 1: The line ℓ has separation defect 3, since there is one point of A below ℓ and two points of B above it.

Present an efficient algorithm that, given two n-element planar point sets A and B, computes a line ℓ that achieves the *minimum separation defect*. Justify your algorithm's correctness and derive its running time. (**Note:** There may generally be many such lines, and your algorithm can output any one of them. If points of A or B lie on ℓ, your algorithm may arbitrarily assign them as being above or below the line.) Aim for $O(n^2)$ running time and $O(n)$ space.

Problem 2. (a) Given a set $S = \{s_1, \ldots, s_n\}$ of non-intersecting line segments in the plane, does there exist a (nonvertical) line ℓ that *misses* (does not intersect) all of the segments of S such that there is at least one segment of S lying above ℓ and at least one lying below? (See Fig. 2(a).) Present an efficient algorithm for this problem. Justify your algorithm's correctness and derive its running time. Aim for $O(n^2)$ running time and $O(n)$ space.

(b) Consider the same setup as in part (a), but now the question is whether there exists a line ℓ that *hits* (intersects) all of the segments of S? (See Fig. 2(b).) (**Hint:** It suffices to explain how to modify your solution to (a).)

(c) Part (b) looks very similar to a problem we have seen before of stabbing *vertical segments*. We showed that problem could be solved in $O(n)$ time by linear programming. Explain (at an intuitive level) why LP cannot be applied to obtain an $O(n)$ time solution.

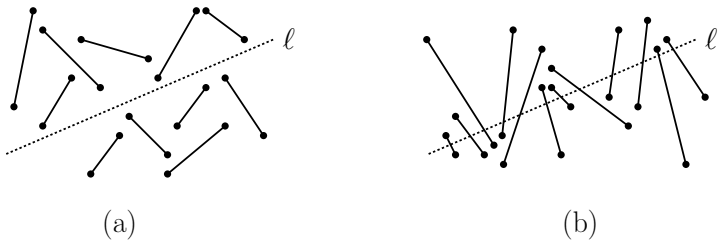


Figure 2: Stabbing segments.

Problem 3. A useful technique in the design of approximation algorithms involves computing a "sketch" of a given point set P. The objective is to compute a subset $Q \subseteq P$ with the property that no two points of Q are too close, and each point of P is not very far from its closest point in Q.

Given a point set P with n points and two positive reals, s and t, where $0 < s \leq t$, define an (s,t)-*Delone set* of P to be a subset Q of P such that:

Packing: For each pair of points $q, q' \in Q$, $|q - q'| \geq s$

Covering: For each $p \in P$, there exists $q \in Q$ such that $|p - q| \leq t$

Equivalently, the open balls of radius $s/2$ around each point of Q are disjoint, and the set of balls of radius t around each point of Q covers all the points of P (see Fig. 3).

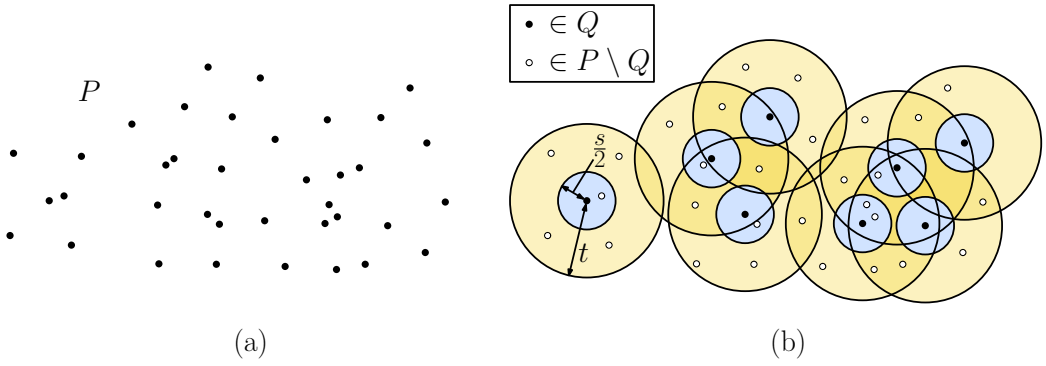


Figure 3: (s,t)-Delone set.

(a) Show that for any point set P in $\mathbb{R}^d$ and any $r > 0$, P has an (r,r)-Delone set. (**Hint:** Describe a simple greedy algorithm for computing the set. Your algorithm does not need to be efficient, since you only need to show existence.)

(b) Present an efficient algorithm that, given an n-element point set P in $\mathbb{R}^2$ and any $r > 0$, computes an (r, cr)-Delone set, where $c \geq 1$ is a constant (independent of r or n). Aim for a running time of $O(n)$. (**Hint:** Hash points into grid squares of an appropriate side length. You may assume that hashing takes $O(1)$ time per point. It is not too hard to obtain $c = 2\sqrt{2}$, but you might try for better.)

Problem 4. (Optional–Ungraded) As mentioned in class, a WSPD is an efficient (approximate) representation for the complete graph on a set of points P. Another important structure is the complete *bipartite graph* on a pair of point sets. In this problem, we will explore this topic.

You are given two sets of points in $\mathbb{R}^d$, called R (for red) and B (for blue). Throughout, let $n = |R| + |B|$. A *bichromatic pair* is any pair of points (p, q), where $p \in R$ and $q \in B$. Given a parameter $s > 0$, define a *bichromatic s-WSPD* is a collection of pairs of subsets $\{(R_1, B_1), (R_2, B_2), \ldots\}$ such that

(i) $R_i \subseteq R$ and $B_i \subseteq B$,

(ii) R_i and B_i are s-well separated, and

(iii) for every bichromatic pair (p, q) there is exactly one pair (R_i, B_i) such that $p \in R_i$ and $q \in B_i$.

Given this definition, answer the following questions:

(a) Explain how to modify the standard WSPD algorithm given in class to produce a bichromatic s-WSPD for the sets R and B. (You can explain what changes to make to the algorithm given in class.)

(b) Show that the asymptotic running time and total size of your bichromatic WSPD construction are the same as for the standard WSPD construction.

(c) Explain how to enhance your WSPD construction so that, with the same time and space requirements, every well-separated pair (R_i, B_i) stores a pair of *representative points* $\text{rep}(R_i) \in R_i$ and $\text{rep}(B_i) \in B_i$, and additionally, each WSPD stores the *sizes* $|R_i|$ and $|B_i|$.

Note: Challenge problems are not graded as part of the homework. The grades are recorded separately. After final grades have been computed, I may "bump up" a grade that is slightly below a cutoff threshold based on these extra points. (But there is no formal rule for this.)

Challenge Problem: Consider a set P of n points in the plane. For $k \leq \lfloor n/2 \rfloor$, point q (which may or may not be in P) is said to have *Tukey depth* (or simply *depth*) k if every line ℓ passing through q has at least k points of P lying on or above it and at least k points on or below it. (For example, for the $n = 12$ element point set Fig. 4, q has depth 4, since every line passing through q has at least 4 points lying on either side. But it does not have depth 5 since the horizontal line ℓ through q has only 4 points below it.)

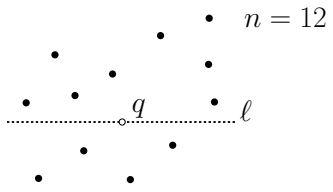


Figure 4: On Tukey depth.

(a) (Medium) Show that for all (sufficiently large) n, there exists a set of n points in the plane such that there is no point (in all of $\mathbb{R}^2$) of depth $\lceil n/2 \rceil$. (Your set should be in general position, and it should be clear how to generalize your construction to arbitrarily large n.)

(b) (Easy) Prove that for a given set P, there exists a point of depth k if and only if in the dual line arrangement, $\mathcal{A}(P^*)$, levels $\mathcal{L}_k$ and $\mathcal{L}_{n-k+1}$ can be *separated* by a line.

(c) (Medium) Describe an efficient algorithm that, given a point set P of size n and an integer k, either computes a point of depth k, or reports that no such point exists. Aim for $O(n^2)$ running time and $O(n)$ space.

(d) (Harder) Prove that for any set of n points in the plane, there exists a point in the plane of depth at least $\lfloor n/3 \rfloor$. (**Hint:** There are many ways to approach this. Here is one based on contradiction. If no such point exists, part (b) implies that $\mathcal{L}_k$ and $\mathcal{L}_{n-k+1}$ for $k = \lfloor n/3 \rfloor$ cannot be separated by a line. Show that this implies that there is a vertex of one of the two levels that lies in the interior of the convex hull of the other level. Use this fact to derive a contradiction.)