Dimensionality Reduction 1 — Maxima

Yufei Tao

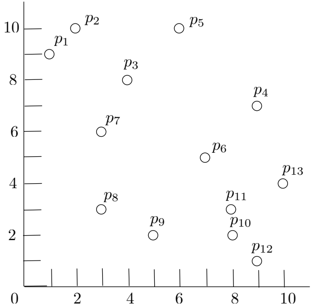
CSE Dept
Chinese University of Hong Kong

Many computational geometry problems are defined in Euclidean space $\mathbb{R}^d$ where the dimensionality d is an arbitrarily large constant. Often times, a problem of dimensionality d can be reduced to the same problem of dimensionality $d - 1$ efficiently. Today, we will demonstrate this by solving the maxima problem in arbitrary dimensionality.

Review: The Maxima Problem

A point p_1 **dominates** p_2 if the coordinate of p_1 is larger than or equal to that of p_2 in all dimensions, and strictly larger in at least one dimension.

Let P be a set of points in $\mathbb{R}^d$. A point $p \in P$ is a **maximal point** of P if it is not dominated by any other point in P.



The maximal points are p_4, p_5, and p_{13}.

Input: A set $P \subseteq \mathbb{R}^d$ of size $n = |P|$.
Output: All the maximal points of P.

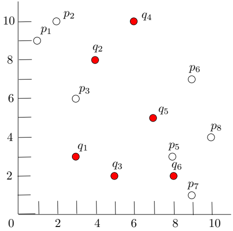
We will solve the problem in $O(n \log^{d-1} n)$ time.

Remark: This week's exercises will guide you to improve the time to $O(n \log^{d-2} n)$ for $d \geq 3$.

Dominance Screening

We will discuss a different problem:

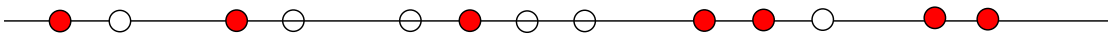
Let P and Q be sets of d-dimensional points in $\mathbb{R}^d$. In dominance screening problem, we want to report all the points in Q that are not dominated by any points in P. Set $n = |P| + |Q|$.



Suppose that P (or Q) is the set of white (or red, resp.) points. The result is $\{q_2, q_4\}$.

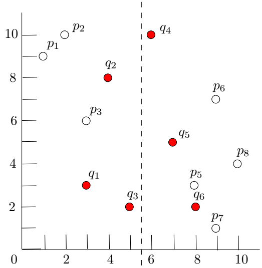
1D Dominance Screening

When $d = 1$, the problem can be easily solved in $O(n)$ time.



2D Dominance Screening

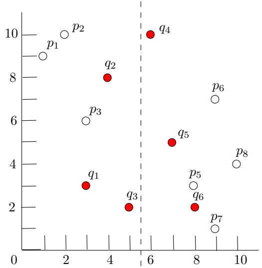
First, divide the input into two halves by x-coordinate:



Let P_1 (Q_1) be the set of white (or red, resp.) points on the left half (i.e., $P_1 = \{p_1, p_2, p_3\}$ and $Q_1 = \{q_1, q_2, q_3\}$). Define P_2 and Q_2 analogously with respect to the right half.

2D Dominance Screening

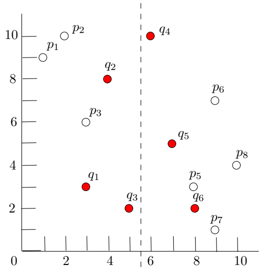
We have two instances of dominance screening: the first on P_1, Q_1, and the other on P_2, Q_2.



Solve each instance recursively. The left instance reports q_2, q_3, and the right instance reports q_4. Next, we will merge the two answers to obtain the final result.

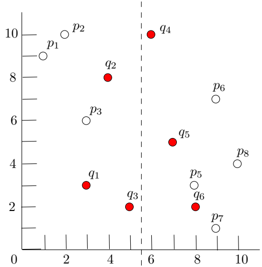
2D Dominance Screening

Observation 1: The right answer is definitely in the final result.
Observation 2: Let q be a point in the left answer. It is in the final result if and only if it is not dominated by any white point from the right instance.



2D Dominance Screening

We now resort to 1D dominance screening.



Let A_{left} be the left answer. Construct a 1D dominance screening problem with input sets P', Q' where

- P': obtained by projecting P_2 onto the y-axis
- Q': obtained by projecting A_{left} onto the y-axis.

2D Dominance Screening

Let us now analyze the running time. Let $f(n)$ be the time on $n = |P| + |Q|$ points. We have:

$$f(n) \leq 2 \cdot f(n/2) + O(n)$$

For $n \leq 2$, $f(n) = O(1)$.

Solving the recurrence gives: $f(n) = O(n \log n)$.

Dominance Screening in d-dimensional Space

1. Divide $P \cup Q$ into two equal halves by the first dimension. This yields two instances of d-dimensional dominance screening: (i) left instance P_1, Q_1, and (ii) right instance P_2, Q_2.
2. Solve the left and right instances, recursively. Let A_{left} and A_{right} be their answers, respectively.
3. Obtain a $(d-1)$-dimensional dominance screening problem P', Q' where P' (or Q') is the projection of P_2 (or A_{left}, resp.) onto dimensions $2, 3, ..., d$. Solve this instance to obtain its answer A'.
4. Return $A_{right} \cup A'$.

Dominance Screening in d-dimensional Space

Let us analyze the running time. Let $f(n)$ be the time on n points.

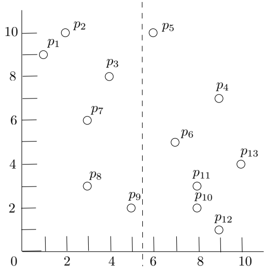
$$f(n) \leq 2 \cdot f(n/2) + g(n)$$

where $g(n)$ is the time of solving $(d-1)$-dimensional dominance screening. Solving the recurrence gives:

- when $d = 3$, $f(n) = O(n \log^2 n)$;
- when $d = 4$, $f(n) = O(n \log^3 n)$;
- ...
- in general, $f(n) = O(n \log^{d-1} n)$.

2D Maxima

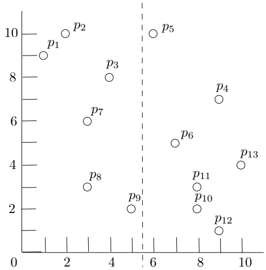
We now attack the maxima problem. First, divide the input set into two halves by x-coordinate:



Let P_1 (or P_2) be the set of points on the left (or right, resp.) half.

2D Maxima

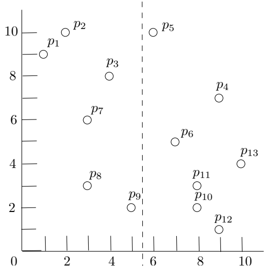
Recursively find the maximal points of P_1 and P_2.



The left instance returns $A_{left} = \{p_2, p_3, p_9\}$, and the right one returns $A_{right} = \{p_5, p_4, p_{13}\}$. The points in A_{right} must be in the final result.

2D Maxima

Observation: Let q be a point in A_{left}. It is in the final result if and only if it is not dominated by any point in A_{right}.



Clearly, now it suffices to solve a 1D dominance screening problem on A_{left} and A_{right}.

2D Maxima

Let us now analyze the running time of our algorithm. Let $f(n)$ be the time on $n = |P| + |Q|$ points. We have:

$$f(n) \leq 2 \cdot f(n/2) + O(n)$$

Solving the recurrence gives: $f(n) = O(n \log n)$.

Maxima in d-dimensional Space

We can solve the d-dimensional maxima problem in $O(n \log^{d-1} n)$ time with a reduction to $(d-1)$-dimensional dominance screening. The details should have become straightforward.