451/651 Lecture 19 – Geometric Algorithms

Solving low-dimensional problems involving geometric objects ... points, lines, circles, polygons, etc.

In addition to their practical applications, these problems and their algorithms lead to a lot of very interesting and generally applicable data structures and algorithmic techniques.

Part 1: 2-Dimensional Geometric Primitives.

A point: It's generally represented as a pair p = (x,y) of real numbers. (Other representations are possible.)

▶ Real Arithmetic Assumption: We allow arithmetic on real numbers and assume that it takes O(1) time.

This assumption is not to be abused.

Primitives, contd

Adding and subtracting points:

Cross Product:

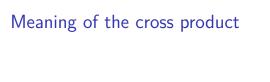
Dot Product:

Scalar Mult:

Length:

Normalization:

Rotation:



Meaning of the dot product

Of course we've seen this before. The objective function of an LP is a dot product.

The line through (0,0) perpendicular to the line ((0,0), v) is the set of points w such that $w \cdot v = 0$.

In general the set of points w such that $w \cdot v = c$ (some constant) is a line perpendicular to the vector v. some distance c/|v| from the origin. So if v is normalized $w \cdot v$ is the signed distance from the perpendicular through the origin.

More on the dot product

Given two vectors v and w we have that $v \cdot w = |v||w|\cos(\alpha)$ where α is the angle between v and w. This means that we can compute the angle between two vectors v and w by computing

$$\alpha = \cos^{-1}\left(\frac{v}{|v|}\frac{w}{|w|}\right)$$

Representing Lines: Two Standard Ways

$$Ax + By = C$$

where A, B and C are constants is the equation of a line. (A and B cannot both be zero.) This is the same as saying that $(A,B)\cdot(x,y)=C$.

Other method: Given a pair of distinct points p and q, it's the unique line through those two points.

Can convert between the two representations. Take the line through (p,q). Let v = rot 90cc(p-q). Now

$$v \cdot (x, y) = v \cdot p$$

is its dot product representation.

Distance to a line

Take the line *L*: Ax + By = C. Where the vector (A, B) is normalized, i.e. $A^2 + B^2 = 1$.)

Then for any point p, $(A, B) \cdot p$ is its signed distance from the line L.

Take the line L: through points a and b. Let v = (b-a)/|b-a|. Then the signed distance from p to L is

$$(p-a) \times v$$

Line Side Test

Given a line through points p and q, from $p \rightarrow q$ which side of that directed line is a on? LEFT, ON, RIGHT?

Let
$$v = q - p$$
, and $w = a - p$
Take $v \times w$.
If > 0 a is to the LEFT of $p \rightarrow q$
if < 0 a is to the RIGHT of $p \rightarrow q$

if = 0 q is ON $p \rightarrow q$

Convex Combination of Points

Given p_1, p_2, \ldots, p_n a convex combination of them is $\alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_n p_n$. Where the α s are non-negative scalars that sum to 1.

Example n = 2: It's the segment connecting the two points.

For n = 3 it fills the triangle. (show general case)

Convex Sets

A set C of points in \mathbb{R}^d is convex if for any points p,q in C the convex combination of p and q is also in C. (Examples of convex and non-convex sets)

The intersection of any two convex sets is convex (proof trivial from the definition)

Note: The half plane is convex, so the intersection of half planes is convex (i.e. a convex polytope, i.e. the feasible region of an LP.)

Part 2: The Convex Hull Problem

Given a set of points p_1, \ldots, p_n in 2D, compute the convex hull of the points. I.e. the bounary of the convex combination of the points.

Intuitively the convex hull can be computed with the following analog computer: Put nails in a board at the points p_1, \ldots, p_n . Now let a rubber band stretched around all of them snap in. The points it touches, and the order of them around that polygon is the convex hull of p_1, \ldots, p_n .

Convex Hull Contd.

Input: A set of points p_1, \ldots, p_n . We assume for clarity that no three of them are colinear.

Output: A sequence from among p_1, \ldots, p_n of the convex hull of the points in counter-clockwise order.

An $O(n^3)$ algorithm

Try all pairs to see if it's a convex hull boundary. Stitch the selected ones together.

An $O(n^2)$ algorithm

Graham Scan: an $O(n \log n)$ algorithm

