CS420+500: Advanced Algorithm Design and Analysis

Lectures: January 4 + January 6, 2017

Prof. Will Evans *Scribe: Alice Fredine*

In this lecture we:

- Reviewed the Syllabus
- Completed a Pre-Class Quiz
- Discussed Convex Hulls
- Discussed Jarvis March
- Discussed Graham's Scan

Handouts (posted on webpage): NONE.

Reading: NONE.

1 Syllabus

See the course website for updated syllabus information: `http://www.ugrad.cs.ubc.ca/~cs420/current/`

2 Quiz

The quiz was a refresher on the kinds of problems to be familiar with before taking the class. The quiz, and sample solutions, are available on piazza: `https://piazza.com/class/ixgle6zb9ze3pj?cid=8`.

3 Computational Problems Involving Geometry

- **Salad Dressing Example**
  Consider a salad dressing that is composed of oil, vinegar, and some other ingredients.

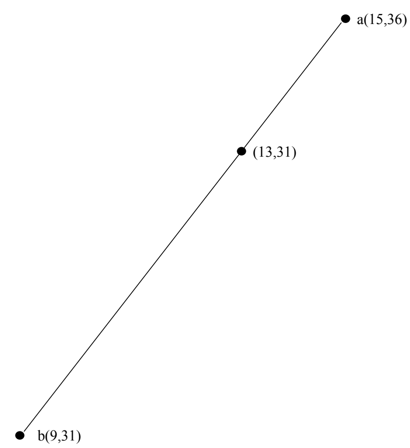
Dressing	Oil	Vinegar
a	15%	36%
b	9%	21%

Is is possible to combine these two bottles and achieve a dressing of 13% oil and 31% vinegar? Yes! We achieve this by combining 2 parts a, 1 part b.

$\frac{2}{3} * 15\% + \frac{1}{3} * 9\%, \frac{2}{3} * 36\% + \frac{1}{3} * 21\% = 10 + 3, 24 + 7 = 13, 31$

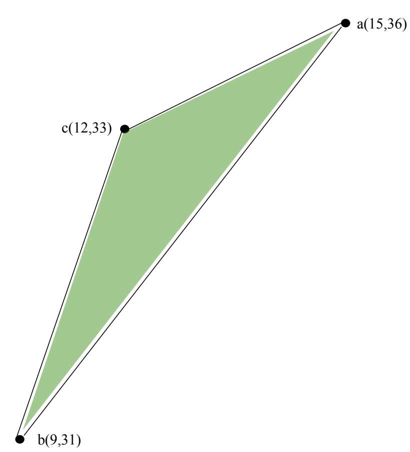
Is it then possible to produce a dressing of 12% oil and 30% vinegar?
It isn't... But how can we see that?

- **The Coordinate Plane**
  Consider the components as coordinates on a plane. What are the possible combinations?



If we treat a and b as points on a plane, we can make any combination that falls on the line. We get (13,31) on on the line. The point (12,30) does not fall on the line, and thus is not a valid combination.

What if we now add a third point (12,33)?



Now we can get any combination that falls in the triangle area between points a,b, and c.

Definition: a <u>mixture</u> is a <u>convex combination</u> of points representing the contents of the bottles.

- **Convex Hulls**

  A convex combination of points $p_1, p_2, ..., p_n$ is:

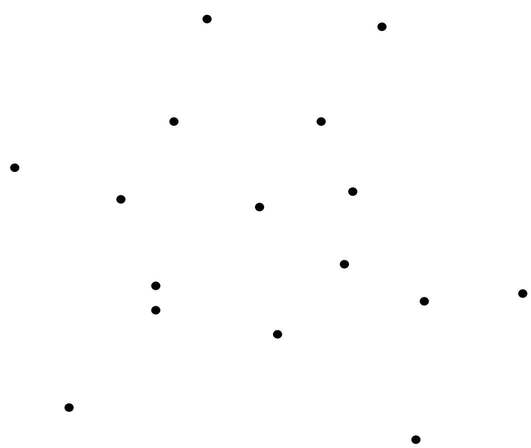
  $\sum_{i=1}^{n} \alpha_i p_i$

  where α_i is the proportion of example i used.

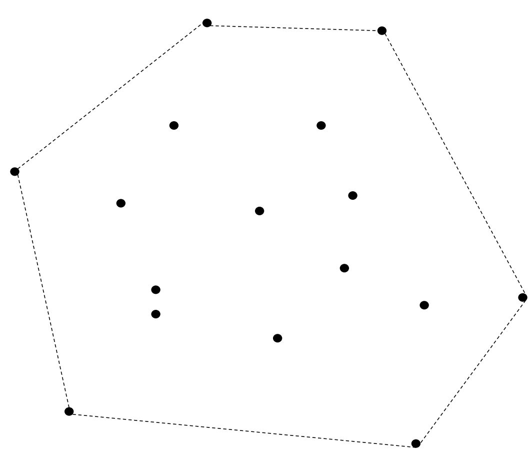
  Properties of α's:

  1. $\sum_{i=1}^{n} \alpha_i = 1$
  2. $\alpha_i \geq 0 \forall i$

  Let P be a set of points. What is the area of all convex combinations of this collection of points, P?

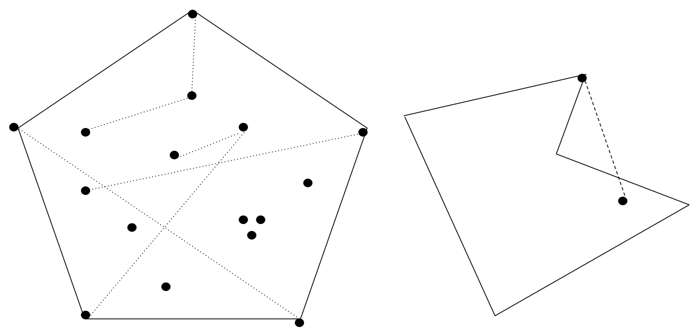


This is the set of all convex combinations of P (which forms the *Convex Hull of P*).



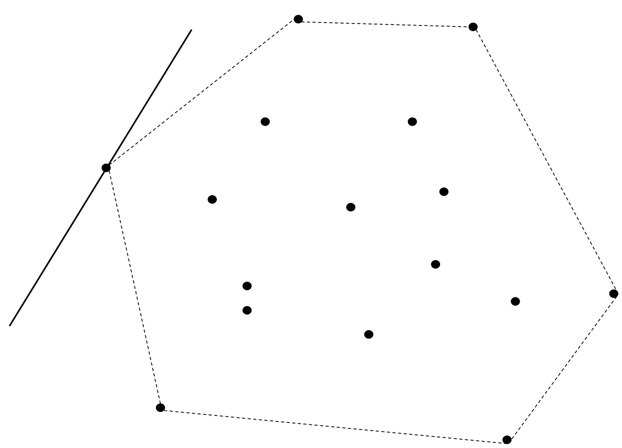
Definition: The Convex Hull of P is the smallest convex set that contains P.

Definition: A set, T, is convex if $\forall$ a,b ϵ T, segment $\overline{ab}$ is in T.



We see the left set is convex, as any line segment between any two points is within the set. However, the set on the right has a divot, and thus is not convex.

There are many ways to think about Convex Hulls. One way is noticing that the boundary points are points in the set such that $\exists$ a line through each boundary point such that all points in the set are one one side. This is called a supporting line.



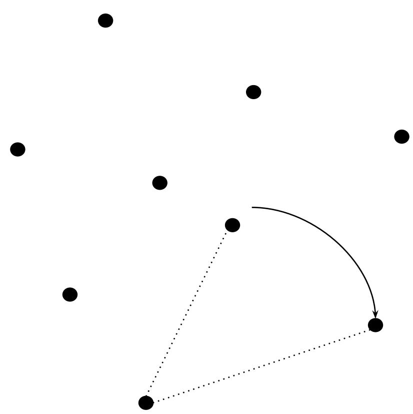
- **Jarvis March**
  Input set of points $P = p_1, p_2, ..., p_n$
  Output convex hull of P

  Intuition: Consider a board with nails in it. The convex hull can be found by putting an elastic band around all the nails.

Jarvis March Algorithm (Gift-Wrapping) 1973
Find a point that is certain to be on the hull, tie a string to it, and wrap around the rest of the points.

What input point is for sure on the boundary of the hull? Any extreme point, meaning any point with a supporting line (for example, a point with largest or smallest x or y coordinate).

Note: The next point in the hull from the starting point with be the one with the widest angle. We find this by doing a right turn test from point p_0. The intuition behind a right turn test if that if the next point is to the right, it is a better (wider angle).



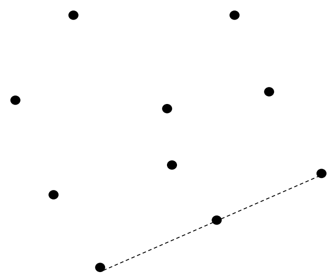
Jarvis March Pseudocode

Find a point p_0 in P with minimum y-coordinate.
$h = 0$ ▷ Index of point in the hull
repeat
  let $q \in P \setminus p_h$
  for all $p \in P$ **do** ▷ Invariant Goes Here
    if rightturn(p_h, q, p) **then** $q = p$
    end if
  end for
  $p_{h+1} = q$
  $h++$
until $p_h = p_0$

Exercise Left for the Student: What is the invariant for Jarvis March?

Runtime of Jarvis March: $\theta(n^2)$

In-class question: What about co-linear points?



As it stands, the middle point might be included in the representation of the hull, depending on the order of looking through points. Is this OK? Doesn't this make the size of the hull bigger? And since we want the smallest convex set, wouldn't that be wrong? First, it doesn't change the convex hull. The convex hull is the smallest convex shape that contains P. "Smallest" refers to the entire convex hull, which is a closed region in the plane. It includes all the points on the boundary and the interior; usually an infinite number of points. We typically *represent* a convex hull as a list or array of input points on its boundary. Should we include this middle point in the representation? We don't need to. It doesnt change the shape of the convex hull. By convention, we typically wouldn't include it in the representation, just as we wouldn't include duplicate boundary points. But the convex hull is still the same if we include it in the representation or not.

Does the middle point pass the supporting line condition? The usual definition of supporting line is: A supporting line for a set S of points in the plane contains a point in S and has all the other points of S on the same side of or on the line. So the middle point does pass the supporting line condition. But we still can choose not to include it in the representation of the hull. For example, we could say that the representative points on the hull boundary are input points that have a *strict* supporting line. (What does "strict" mean?)

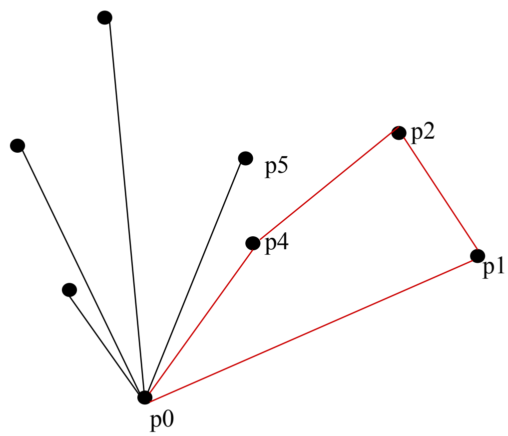
- **Graham's Scan Algorithm 1972**
  Graham's Scan is a faster, but trickier, algorithm.

  The problem with Jarvis March is that is keeps repeating work by examining all the points. The idea behind Graham's Scan is to first order the points (by angle relative to p_0), and then use that order to avoid the re-searching.

  1. Find a point p_0 in P with the smallest y-coordinate.
  2. Sort remaining points by angle around p_0 in counter-clockwise order.

  <u>Invariant</u>: at the i^{th} point, calculate the convex hull for everything up to p_i, using the convex hulls from prior points.

  Calculate successively: $CH(p_0, p_1, p_2)$, $CH(p_0, p_1, p_2, p_3)$, $CH(p_0, p_1, p_2, p_3, p_4)$, etc.



In order to calculate the convex hull for p_5 given the convex hull for p_4, we can do the right turn test. This works for an arbitrary number of points. We can do the right turn test until we reach the last one that should be in the convex hull by continuously popping points off and performing the test again.