Convex Hull Algorithms: Jarvis's March and Graham's Scan

Ramdas Singh

Convexity

Introduce the rubberband.

A subset $\mathcal{C} \subseteq \mathbb{R}^2$ is called *convex* if and only if for all $p, q \in S$, $(1-\lambda)p + \lambda q \in \mathcal{C}$ for all $\lambda \in [0, 1]$. Other equivalent definitions—

- A set $\mathcal{C} \subseteq \mathbb{R}^2$ is convex if and only if it can be expressed as the intersection of (possible infinitely many) closed half-spaces.

- A set $\mathcal{C} \subseteq \mathbb{R}^2$ is convex if and only if for every point $p \in \mathbb{R}^2 \backslash \mathcal{C}$, there exists a hyper plane diving the space into one open half-space containing p and one closed half-space containing $\mathcal{C}$.

Definition can be easily extended to higher dimensions.

Convex Hull

The *convex hull* $\mathcal{CH}(S)$ of a set $S \subseteq \mathbb{R}^2$ is the 'smallest' convex set that contains S. More precisely,

- The convex hull $\mathcal{CH}(S)$ of a set S is the intersection of all convex sets that contain S:

$$\mathcal{CH}(S) = \bigcap_{\lambda \in \Lambda} \{C \subseteq \mathbb{R}^n : S \subseteq C,\ C \text{ is convex}\}.$$

- The convex hull $\mathcal{CH}(S)$ of a set S consists of all convex combinations of finitely many points in S:

$$\mathcal{CH}(S) = \{\sum_{i=1}^{k} \lambda_i p_i : p_i \in S,\ \lambda_i \geq 0,\ \sum_{i=1}^{k} \lambda_i = 1,\ k \in \mathbb{N}\}$$

Vertices

The *vertices*, intuitively, are simply the vertices of the polygon formed by the convex hull. Formally, the *non-vertices* are defined as

$$\overline{V}(S) = \{p \in S : \mathcal{CH}(S) = \mathcal{CH}(S \backslash \{p\})\}$$

Thus, the vertices become

$$V(S) = S \backslash \overline{V}(S).$$

Problem Statement

Given a set $S = \{p_1, p_2, \ldots, p_n\}$ of points in $\mathbb{R}^2$, compute the ordered set $V(S)$, ordered in clockwise direction.

Interesting (and Helpful) Results

First introduce Jarvis and Graham; they developed fantastic convex hull algorithms for this problem statement.

Caratheodory's Theorem (1907)

If $p \in \mathcal{CH}(S)$ for $S \subseteq \mathbb{R}^2$, then p can be expressed as $p = \sum_{i=1}^{3} \lambda_i p_i$ where $\lambda_i \geq 0$ and $\sum_{i=1}^{k} \lambda_i$ for some $p_i \in S$.

All it says is, if p is in the convex hull, then there exists a triangle containing p whose vertices are formed by vertices in S. Gives us an idea that to determine if p is a vertex or not, we need only consider triangles; gave insight to Jarvis that only small subset of poitns is needed to represent a convex hull, and that they can be computed incrementally.

Radon's Theorem (1921)

Any set of 4 points in $\mathbb{R}^2$ can be partitioned into two sets whose convex hulls intersect. (Proof is a good exercise). Evoked the idea of divide and conquer.

Jarvis March (1973)

R. A. Jarvis. Involved in robotics, computer vision, path finding, and image processing.

Concept (Bruteforce wrapping)

1. Start with an empty set $V(S) = \emptyset$.
2. Consider all $(p, q) \in S \times S$.
3. If all $r \in S \setminus \{p, q\}$ exist to the right of the directed line $\ell(p, q)$, then set $V(S)$ to be $V(S) \cup \{p, q\}$.
4. Order clockwise, and return.

Pseudocode

Algorithm 1 Jarvis's March (Bruteforce Wrapping)

Require: A finite set of points $S \subset \mathbb{R}^2$.
Ensure: The set of extreme points $V(S)$, arranged in a clockwise order.

Initialize $V(S) \leftarrow \emptyset$.
for all distinct pairs $(p, q) \in S \times S$ with $p \neq q$ **do**
    Assume $\ell(p, q)$, the directed line through p and q, is a boundary of $V(S)$.
    for all points $r \in S \setminus \{p, q\}$ **do**
        if r lies strictly to the left of $\ell(p, q)$ **then**
            Discard $\ell(p, q)$ as a boundary.
        end if
    end for
    if $\ell(p, q)$ was never discarded **then**
        Include p and q in $V(S)$.
    end if
end for
Order the elements of $V(S)$ in a clockwise manner.
Return $V(S)$.

Time Complexity

$$O(n^2 h) + O(h \log h) = O(n^2 h), \quad h = \text{number of vertices}$$

Really, on the worst case possible when every point in S is a vertex, this is $O(n^3)$.

Revolutionary, but completely unoptimized.

Gift wrapping version is much better; the concept goes as follows:

Concept (Gift wrapping)

1. Start with an empty set $V(S) = \emptyset$.
2. Immediately add the 'leftmost' point, p_0 to $V(S)$.
3. You now only need to consider $(p_0, p) \in S \times S$ where $p \neq p_0$.

4. If all $r \in S \setminus \{p_0, p\}$ exist to the right of the directed line $\ell(p, q)$, then set $p_1 = p$ and add p_1 to $V(S)$.

5. Continue with $(p_1, p) \in S \times S$ where $p \neq p_1$ until you similarly find p_2. Keep continuing until p_h is forced to be p_0. Return.

Some Explanations

What happens if r lies on $\ell(p, q)$?

We could simply add it, increasing the value of h and increasing computation time. But a check for not adding it would preferably; this check would cost us $O(h)$, thus not increasing computation time by much.

How to check if r lies to the right of $\ell(p, q)$?

You could find the angle between $\ell(p, q)$ and $\ell(q, r)$. A better approach is to compute

$$D = \det \begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix}$$

The sign of D tells whether r lies to the right (and on) or the left of the line.