The Triangle Algorithm

Applications and Visualization

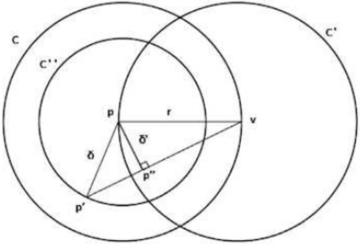
Colin Jarvis [1] , Dr. Bahman Kalantari [2]

[1]Macalester College

[2]Rutgers University, Dept. of Computer Science

July 20, 2012

Outline



Introduction

Definition

Given a set of points S, the convex hull is the minimal convex set containing S.

$$Conv(S) = \left\{ \sum_{i=1}^{k} \alpha_i x_i \middle| x_i \in S,\ \alpha_i > 0,\ \sum_{i=1}^{k} \alpha_i = 1 \right\}$$

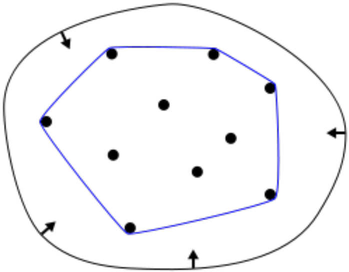


Image Source: http://en.wikipedia.org/wiki/Convex_hull

Convex Hull Decision Problem

Given a finite set of m-dimensional points, how can we determine whether a distinguished point p is in the convex hull or not?

Theorem

Let $S = \{v_1, ..., v_n\}$ be a set of points in $\mathbb{R}^m$. Given a point $p \in \mathbb{R}^m$, $p \in Conv(S) \iff$ for each $p' \in Conv(S) \setminus \{p\}$ $\exists$ $v_j \in S$ such that $d(p', v_j) \geq d(p, v_j)$.

Triangle Algorithm

A simple algorithm to approximate our target point p.

Triangle Algorithm

A simple algorithm to approximate our target point p.

1. Find v_j such that $d(p', v_j) \geq d(p, v_j)$. This is the *pivot*.

Triangle Algorithm

A simple algorithm to approximate our target point p.

1. Find v_j such that $d(p', v_j) \geq d(p, v_j)$. This is the *pivot*.
2. Compute the closest point on the line segment joining p' to v_j. This is the iterate.

Triangle Algorithm

A simple algorithm to approximate our target point p.

1. Find v_j such that $d(p', v_j) \geq d(p, v_j)$. This is the *pivot*.
2. Compute the closest point on the line segment joining p' to v_j. This is the iterate.
3. Now iterate until $|d(p_\epsilon, p)| < \epsilon$.

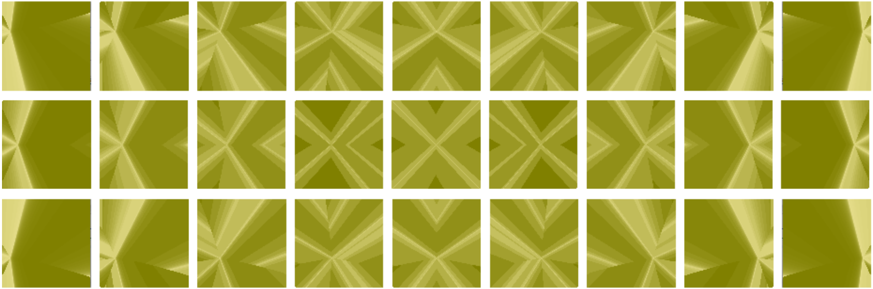
Triangle Algorithm

A simple algorithm to approximate our target point p.

1. Find v_j such that $d(p', v_j) \geq d(p, v_j)$. This is the *pivot*.
2. Compute the closest point on the line segment joining p' to v_j. This is the iterate.
3. Now iterate until $|d(p_\epsilon, p)| < \epsilon$.

Worst case arithmetic complexity of $O(nm\epsilon^{-2})$.

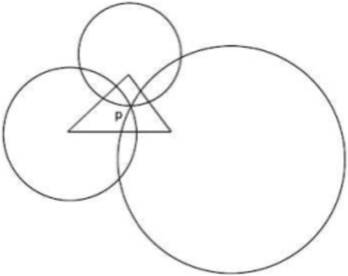
Visualization of a Trivial Case



Ambiguous Convex Hull Problem

The existence of p is unknown. All we are given is a set of distances, $\{r_1, ..., r_n\}$, where each r_i corresponds to its respective vertex in $S = \{v_1, ..., v_n\}$.

How can we then solve the Convex Hull Decision Problem and approximate our target (if appropriate)?



Blind Triangle Algorithm

Assuming p exists and $p \in Conv(S)$. Then

1. Find v_j such that $d(p', v_j) \geq r_j$. This is the *pivot*.

Blind Triangle Algorithm

Assuming p exists and $p \in Conv(S)$. Then

1. Find v_j such that $d(p', v_j) \geq r_j$. This is the *pivot*.
2. Compute the point on the line segment joining p' to v_j that is r_j away from the pivot. This is the iterate.

Blind Triangle Algorithm

Assuming p exists and $p \in Conv(S)$. Then

1. Find v_j such that $d(p', v_j) \geq r_j$. This is the *pivot*.
2. Compute the point on the line segment joining p' to v_j that is r_j away from the pivot. This is the iterate.
3. Iterate until $|d(p_\epsilon, v_i) - r_i| < \epsilon \ \forall \ i = 1, ..., n$.

Applications to Linear Programming

We can think about $Ax = b$ as $A\alpha = p$, where the columns of A are our vertices, alpha is a column of vertex coefficients, and p is our target point.

$$\begin{pmatrix} v_{1,1} & v_{2,1} & \cdots & v_{n,1} \\ v_{1,2} & v_{2,2} & \cdots & v_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1,m} & v_{2,m} & \cdots & v_{n,m} \end{pmatrix} \times \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{pmatrix}$$

$$(m \times n) \;\; (n \times 1) \;\; = \;\; m \times 1$$

Applications to Linear Programming

We can explicitly represent p' as a convex combination of v_i's.

$$p' = \sum_{i=1}^{n} \alpha_i v_i, \quad \sum_{i=1}^{n} \alpha_i = 1, \quad \alpha_i \geq 0$$

Based on how we calculate each iterate, we can track the vertex coefficients as we converge on the approximate solution

$$p'' = \sum_{i=1}^{n} \beta_i v_i, \quad \beta_j = (1-\alpha)\alpha_j + \alpha, \quad \beta_i = (1-\alpha)\alpha_i$$

The Next Steps

Reducing runtime

- Analysis of the midpoint method
- Continuous update midpoint approach
- Intelligently choosing the pivot

The Next Steps

Reducing runtime

- Analysis of the midpoint method
- Continuous update midpoint approach
- Intelligently choosing the pivot

Blind Triangle Algorithm

- Complexity analysis
- Proving some claims

The Next Steps

Reducing runtime

- Analysis of the midpoint method
- Continuous update midpoint approach
- Intelligently choosing the pivot

Blind Triangle Algorithm

- Complexity analysis
- Proving some claims

Linear Programming

- Approximating solutions
- Solving linear systems

References and Acknowledgments

- B. Kalantari. *A Characterization Theorem and an Algorithm for a Convex Hull Problem*.
- B. Kalantari. *Finding a Lost Treasure in Convex Hull of Points From Known Distances*. 4th Canadian Conference on Computational Geometry, 2012.

Special thanks to those who made my research experience possible:

- Bahman Kalantari
- Gene Fiorini
- Yusra Naqvi
- Kellen Myers