(ϵ, δ) hull algorithms

April 10, 2017

Suppose we are given a point set $P \in \mathbb{R}^d$ **with diameter** ≤ 1. An ϵ-approximate convex hull is a set S such that every point in P is within Euclidean distance ϵ from some point in S. We can think of ϵ-approximate convex hull in a slightly different way.

1 Definition of (ϵ, δ) hull

Definition 1.1. Given a vector v and a point set P, we define the **directional width** as

$$\omega_v(P) = \max_{p \in P} p \cdot v$$

Definition 1.2. If p is a point we define $\omega_v(p) = p \cdot v = \omega_v(\{p\})$

It is easy to see that if $S \subseteq P$ then for all v, $\omega_v(S) \leq \omega_v(P)$.

Definition 1.3. We say S **maximizes** P in v if

$$\omega_v(P) = \omega_v(S)$$

Note that as per definition 1.1, S can be either a single vector or a set of vectors.

Definition 1.4. A **convex hull** is the minimal sized set $S \subseteq P$ such that S maximizes P in all (unit) directions v.

Definition 1.5. We say S **ϵ-maximizes** P in direction v if v *is a unit vector* and

$$|\omega_v(P) - \omega_v(S)| \leq \epsilon$$

Note that as per definition 1.1, S can be either a single vector or a set of vectors.

Definition 1.6. An **ϵ-hull** is the minimal sized set $S \subseteq P$ such that S ϵ-maximizes P in all (unit) directions v. We define OPT to be $|S|$.

Intuitively, an ϵ-approximate convex hull approximates the original point set in all directions. Coming up with a streaming algorithm that is competitive within a constant factor of OPT (the batch optimal) for this problem appears to be difficult. An interesting relaxation proposed by Lin is to have a good approximation in *most directions*. In the sections that follow, we will assume that the algorithm has access to OPT and sets $k = \text{OPT}$. In practice, we do not know OPT so we would simply set k to be the largest value our computational resources permit. We would then have an (ϵ, δ)-approximation for all point sets where $\text{OPT} \leq k$.

Definition 1.7. An **(ϵ, δ)-hull** is the minimal sized set $S \subseteq P$ such that if we pick a vector v uniformly at random from the surface of the unit sphere, $\mathcal{S}^{d-1}$, S ϵ-maximizes P in direction v with probability at least $1 - \delta$, that is,

$$\Pr(|\omega_v(P) - \omega_v(S)| > \epsilon) \leq \delta$$

Our goal is to come up with streaming algorithms for (ϵ, δ)-hulls that are competitive with OPT (the batch optimal for ϵ-hulls).

2 Core Lemmas

Definition 2.1. Define E_s^S to be the set of all vectors in $\mathbb{R}^d$ (*not just unit vectors*) that s maximizes, that is,

$$E_s^S = \{v \mid v \cdot s = \omega_v(S)\}$$

Lemma 2.1 (ϵ-Maximization Lemma)**.** *Suppose $S \subseteq P$ is an ϵ-hull of P, and fix $s \in S$. Then s ϵ-maximizes P for all unit vectors $v \in E_s^S$.*

Proof. This is because for all unit vectors v, $|\omega_v(S) - \omega_v(P)| \leq \epsilon$ and for all vectors $v \in E_s^S$, $v \cdot s = \omega_v(S)$, so for all unit vectors $v \in E_s^S$, both properties hold. □

Lemma 2.2 (Covering Lemma)**.** *For all vectors $v \in \mathbb{R}^d$,*

$$v \in \bigcup_{s \in S} E_s^S$$

Proof. Given any vector v, set $s = \operatorname*{argmax}_{s' \in S} s' \cdot v$. Then $v \in E_s^S$. □

Lemma 2.3 (Conic Lemma)**.** *E_s^S is a cone, that is,* *1.* $0 \in E_s^S$

2. If $v \in E_s^S$ *and* $\alpha \in \mathbb{R}^+$ *then* $\alpha v \in E_s^S$.

3. If $v, w \in E_s^S$ *then* $v + w \in E_s^S$.

Proof. We prove each item,

1. $0 \cdot s = \max_{s' \in S} 0 \cdot s' = 0$

2. $(\alpha v) \cdot s = \alpha(v \cdot s) = \alpha(\max_{s' \in S} v \cdot s') = \max_{s' \in S}(\alpha v) \cdot s'$

3. $(v+w) \cdot s = (v \cdot s) + (w \cdot s) = \max_{s' \in S} v \cdot s' + \max_{s' \in S} w \cdot s' \geq \max_{s' \in S}(v+w) \cdot s'$

$\square$

Lemma 2.4 (Cutting Lemma)**.** *Given any 2 points* $a \neq b \in \mathbb{R}^d$*, let* $H = \{v \mid v \cdot a \geq v \cdot b\}$*. Then* H *is a closed halfspace cutting through the origin.*

Proof. Writing this in another way, $H = \{v \mid v \cdot (a - b) \geq 0\}$, which, if $a - b \neq 0$, is precisely the equation of a closed halfspace. The plane defining the boundary of the halfspace is defined by $P = \{v \mid v \cdot (a - b) = 0\}$ (that is, the set of all vectors perpendicular to $a - b$) which cuts through the origin. $\square$

Lemma 2.5 (Bounded Maximization Lemma)**.** *Suppose* S *has at least 2 distinct points. Then if* $s \in S$, E_s^S *is contained inside a closed halfspace passing through the origin.*

Proof. Choose $s' \in S$ with $s' \neq s$. Then, let $H = \{v \mid v \cdot s \geq v \cdot s'\}$. $E_s^S \subseteq H$ but by the cutting lemma, H is a closed halfspace passing through the origin. $\square$

3 2D-Algorithm

3.1 Algorithm

We give a deterministic algorithm that stores $O(\frac{k}{\delta})$ points and gives us an (ϵ, δ)-hull of a point set P, where k is the batch optimal for the ϵ-hull of P.

Choose $O(\frac{k}{\delta})$ equally separated unit vectors on the boundary of the unit circle. Going counter-clockwise by angle, the angle formed by any 2 consecutive vectors will be less than $\frac{2\pi\delta}{k}$. For each chosen vector v we store the point $p \in P$ s.t. $p \cdot v = \omega_v(P)$. This can be done in streaming - for a vector v, we keep an incoming point p iff $v \cdot p$ is greater than $v \cdot p'$ for the point p' we currently stored in direction v (or if we have not stored any point for direction v). Call the set of points our algorithm chooses T.

3.2 Proof

WLOG suppose that P has at least 2 distinct points (otherwise we can trivially solve the problem by storing the only point in P). Consider an optimal ϵ-hull $S \subseteq P$. WLOG suppose that S contains at least 2 points (otherwise we can simply add some point in P to S and our bounds will only change by a constant factor).

Partitioning: Pick a vector v uniformly at random on the boundary of the unit circle. By the covering lemma, $v \in E_s^S$ for some $s \in S$. It suffices to show that, conditional on this choice of s, the probability that T does not ϵ-approximate P is $\leq \frac{\delta}{k}$. Since $|S| = k$, this would imply that the unconditioned probability that T does not ϵ-approximate P is $\leq k\frac{\delta}{k} = \delta$.

Angular setup: Fix $s \in S$. S has at least 2 distinct points, so by the cutting lemma, E_s^S is contained in a half-space. So we can rotate space such that E_s^S does not contain the positive x axis. We measure angles counter-clockwise from the positive x-axis. From the conic lemma, we know that E_s^S is the set of all vectors with angles between θ_a and θ_b with $\theta_a < \theta_b$. Suppose that m of the vectors we chose, $v_1, v_2, ..., v_m$ were in E_s^S with angles $\theta_1 < \theta_2 < ... < \theta_m$. We also have that $\theta_a \leq \theta_1$ and $\theta_m \leq \theta_b$.

Lemma 3.1. *Consider some v_i. We choose a point p_i s.t. $p_i \cdot v_i = \omega_{v_i}(P)$ (that is, p_i is maximal in direction v_i). We will show that p_i ϵ-maximizes either all unit vectors with angles in the range $[\theta_a, \theta_i]$ or in $[\theta_i, \theta_b]$*

Proof. If $p_i = s$, then p_i ϵ-maximizes all unit directions in E_s^S, so we are done. Otherwise, suppose $p_i \neq s$. By the cutting lemma, there exists a closed half-space H passing through the origin, such that $p \cdot v \geq s \cdot v$ iff $v \in H$. By the ϵ-maximization lemma, for all unit vectors $v \in E_s$, $0 \leq \omega_v(P) - s \cdot v \leq \epsilon$. This implies that for all unit vectors $v \in E_s \cap H$, $0 \leq \omega_v(P) - p \cdot v \leq \omega_v(P) - s \cdot v \leq \epsilon$. In other words, p ϵ-maximizes all unit vectors $v \in E_s \cap H$.

Since E_s is itself contained in some halfspace passing through the origin, $E_s \cap H$ is either the set of vectors with angles in range $[\theta, \theta_b]$ or in range $[\theta_a, \theta]$. We note that $v_i \in E_s \cap H$ and has angle θ_i, so in either case the range contains θ_i. This proves the lemma. □

We say that v_i is *down* if p_i ϵ-maximizes P in all directions with angles in the range $[\theta_a, \theta_i]$ and *up* if p_i ϵ-maximizes P in all directions with angles in the range $[\theta_i, \theta_b]$. If v_1 is up, then p_1 ϵ-maximizes P in all directions with angles in $[\theta_1, \theta_b]$. The angle between θ_a and θ_1 is $\leq \frac{2\pi\delta}{k}$ because we chose vectors that were $\frac{2\pi\delta}{k}$ apart, so we are done. A similar argument applies if v_m is down, and if $m = 0$ (we did not choose any vectors between θ_a and θ_b. Otherwise, we consider the smallest i s.t. v_i is down but v_{i+1} is up. Then, p_i ϵ-maximizes P in all directions with angles in $[\theta_a, \theta_i]$ and p_{i+1} ϵ-maximizes P in all directions with angles in $[\theta_{i+1}, \theta_b]$. We might not ϵ-maximize P in directions with angles in the range $[\theta_i, \theta_{i+1}]$, but this angle is $\leq \frac{2\pi\delta}{k}$.

4 3D-Algorithm

5 Generalization