CPSC 536C: Algorithms for Convex Optimization
University of British Columbia

Cutting Plane Methods

Akshay Ram
January 5, 2025

These notes are subject to change and may contain errors.

1 GLS Oracle Model

1.1 Definitions

Definition 1 (Oracle Model)**.** *The following are algorithmic problems for closed convex set $K \subseteq \mathbb{R}^n$:*

MEM bership($x \in \mathbb{R}^n, K \subseteq \mathbb{R}^n$):
Output: YES if $x \in K$; NO otherwise.

SEP aration($x \in \mathbb{R}^n, K \subseteq \mathbb{R}^n$):
Output: YES if $x \in K$;
otherwise output separating hyperplane w such that $\langle w, x\rangle > \sup_{y \in K} \langle w, y\rangle$;

OPT imization($w \in \mathbb{R}^n, K \subseteq \mathbb{R}^n$)
Output: $h_K(w) = \sup_{x \in K} \langle w, x\rangle$
and optimizer $x_* \in K$ such that $h_K(w) = \langle w, x_*\rangle$;

VAL idity($w \in \mathbb{R}^n, b \in \mathbb{R}, K \subseteq \mathbb{R}^n$):
Output: YES if $\langle w, x\rangle \le b$ for all $x \in K$; NO otherwise.

VIOL ation($w \in \mathbb{R}^n, b \in \mathbb{R}, K \subseteq \mathbb{R}^n$):
Output: YES if $\langle w, x\rangle \le b$ for all $x \in K$;
otherwise output $x \in K$ such that $\langle w, x\rangle > b$.

We can also consider *approximate* versions of all these oracles. For the formal definitions, see [GLS 2.1]. Many of the algorithms we study in this class can be thought of as reductions between these algorithmic problems.

1.2 Relations between Oracles

The various oracles for K and its polar dual K° are related.

Proposition 2. *In the following, we assume $B(0, \varepsilon) \in int(K)$ and $B(0, \varepsilon) \in int(K^\circ)$.*

- *OPTimization oracle for K can be (approximately) implemented using VIOLation or VALidity oracle for K;*
- *MEMbership oracle for K is equivalent to VALidity oracle for K°; and vice-versa MEM(K°) is equivalent to VAL(K);*

- *SEParation oracle for K° can be implemented using a VIOLation oracle for K; VIOL(K) can be implemented using SEP(K°) and one of MEM(K) or VAL(K°).*

Proof:

- In the following we use VAL if we only need the optimum value, and VIOL if the optimizer is also required. Using the VAL/VIOL oracles we can check if $\exists x \in K : \langle w, x\rangle \geq b$. Therefore, to compute $h_K(w) = \sup_{x\in K}\langle w, x\rangle$ approximately, we perform binary search over b.

- We first implement MEM(K°) using VAL(K). Given input w we want to check if $w \in K^\circ$ iff $\sup_{x\in K}\langle w, x\rangle \leq 1$. For this we return the output of VAL$(w, 1, K)$.

  Conversely, we want to implement VAL(K) using MEM(K°). So given an input (w, b) we want to check if $\forall x \in K : \langle w, x\rangle \leq b$. First we use that $B(0, \varepsilon) \subseteq K$ so if $b \leq 0$ we output NO. Otherwise note that $v \in K^\circ$ iff $\sup_{x\in K}\langle v, x\rangle \leq 1$. So we can return the output of MEM$(w/b, K^\circ)$.

  The dual statement, swapping K and K°, follows due to bi-duality $(K^\circ)^\circ = K$.

- We first implement SEP(K°) using VIOL(K). Given input w we want to check if $w \in K^\circ$, and if not to output a separating hyperplane. Note $w \in K^\circ$ iff $\sup_{x\in K}\langle w, x\rangle \leq 1$, so we query VIOL$(w, 1, K)$; if the output is YES, we return YES; if the output is NO then the oracle returns $x \in K$ such that $\langle w, x\rangle > 1 \geq \sup_{v\in K^\circ}\langle v, x\rangle$, where the last step is by definition of K°, so we can output x as our separating hyperplane for K°.

  Conversely, we want to implement VIOL(K) using SEP(K°). So given an input (w, b) we want to check if $\sup_{x\in K}\langle w, x\rangle \leq b$, and if not to output $x \in K$ such that $\langle w, x\rangle > b$. First we use that $B(0, \varepsilon) \subseteq K$ so if $b \leq 0$ we output $0 \in K$ as $\langle w, 0\rangle = 0 > b$. Otherwise we query SEP$(w/b, K^\circ)$; if the output is YES then $\sup_{x\in K}\langle w/b, x\rangle \leq 1$ so we return YES; if the output is NO then the oracle returns separating hyperplane $x \in \mathbb{R}^n$ such that $\langle w/b, x\rangle > \sup_{v\in K^\circ}\langle v, x\rangle$. If we knew this right hand side $h_{K^\circ}(x) = \sup_{v\in K^\circ}\langle v, x\rangle$ then we could output $y := x/h_{K^\circ}(x) \in K^{\circ\circ} = K$ with $\langle w, y\rangle > b$ as our violating point. But we can approximately compute this value using binary search with the VALidity oracle for K° as shown in the first part. Equivalently, since $0 \in \text{int}(K)$, there is a point $y \in [0, x] \cap \partial K$ at the intersection of the line $[0, x]$ with the boundary of K, and this also gives a violating point $\langle w, y\rangle > b$ ($y \in \partial K$ implies $\sup_{v\in K^\circ}\langle v, y\rangle = 1$). And this boundary point we can approximately compute using binary search with the MEMbership oracle for K.

□

1.3 Function Oracles: TODO

What is the relation to function oracles? In particular, recall that we prove duality of convex functions by reducing to duality of the epigraph, which is a convex set.

Definition 3 (Fenchel dual)**.** *For convex $f : \mathbb{R}^n \to \mathbb{R}$, the Fenchel dual is*

$$f^*(w) := \sup_y \langle w, y\rangle - f(y).$$

Lemma 4. *For differentiable closed convex f, the supporting hyperplane for the epigraph of f at $(x, f(x))$ is given by slope $(\nabla f(x), -1)$ and value $f^*(\nabla f(x)) = \sup_y\langle \nabla f(x), y\rangle - f(y)$.*

Similarly, the supporting hyperplane for the sub-level set $L_{f(x)} := \{y \in \mathbb{R}^n \mid f(y) \leq f(x)\}$ is given by slope $\nabla f(x)$ and value $\langle \nabla f(x), x\rangle$.

Both these statements can be generalized to non-differentiable f using subgradients.

Proposition 5. *In the following, we assume $f : \mathbb{R}^n \to \mathbb{R}$ is a closed convex function.*

- *EVALuation oracle for f can be implemented using MEMbership oracle for epi(f);*
- *GRADient oracle for f can be implemented using SEParation oracle for epi(f);*
- *OPTimization oracle for epi(f) is equivalent to EVALuation for the Fenchel dual $f^*(w) := \sup_y \langle w, y \rangle - f(y)$.*

2 Cutting Plane Methods

Our eventual goal is to find an algorithm to solve convex programs $\min_{x \in K} f(x)$. Cutting plane methods solve this using just SEParation oracle for K and EVAL and GRAD oracles for f. We first show how to solve the much simpler FEASibility problem: given SEParation oracle for K, either find $x \in K$ or output that K is 'small'. Note that linear optimization over K can be reduced to this: test FEASibility for

$$K' := K \cap \{x \mid \langle w, x \rangle \geq b\}.$$

and run binary search on b. Note that SEP for K' can be easily implemented using SEP for K and testing the linear inequality $\langle w, x \rangle \geq b$.

The intuition for cutting plane methods comes from binary search: say we are attempting to find some point $x \in [0,1]^n$, and we have oracle access to coordinate queries $y_i \geq x_i$ or $y_i \leq x_i$. Then in each iteration, we should query the center of the remaining grid and eliminate the elements with chosen coordinate too large or small, depending on the output of the oracle. This procedure is optimal in the worst case, as it eliminates the maximum possible options in each iteration. This is the correct intuition for the Center of Gravity method, described below, which generalizes the procedure to arbitrary convex $K \subseteq \mathbb{R}^n$. In general, it is not clear how to query the 'center' of a convex body, so the next algorithm will maintain an Ellipsoid in each iteration to contain the feasible set, and update according to the separation oracle.

3 Center-of-Gravity Method

We first describe the algorithm to solve feasibility. In each iteration, we maintain a feasible convex set $K_t \supseteq K$; we query the separation oracle with the Center of Mass $c_t := \mathcal{E}_{x \sim K_t} x = \int_{x \in K_t} x dx$; if $c_t \in K$ then we are done; otherwise we get $\langle w, c_t \rangle > \sup_{x \in K} \langle w, x \rangle$; therefore, we know K is contained in the following halfspace

$$H_t := \{x \mid \langle w, x \rangle \leq \langle w, c_t \rangle,$$

so we update accordingly $K_{t+1} := K_t \cap H_t$. We claim that this solves feasibility:

Theorem 6. *Given $K \subseteq B(0, R) \subseteq \mathbb{R}^n$ via separation oracle, the center-of-gravity method requires $O(n \log(R/\varepsilon))$ iterations to either*

1. *find $x \in K$;*
2. *certify K does not contain a ball of radius ε, or certify vol$(K) \leq \varepsilon^n$.*

The key step in the analysis is the following beautiful result from convex geometry:

Theorem 7 (Grunbaum's Theorem)**.** *For convex $K \subseteq \mathbb{R}^n$, let $c := \mathcal{E}_{x \in K} x$ be the center-of-gravity of K. Then for any hyperplane $H \ni c$, the two halfspaces H_+, H_- satisfy*

$$\max\{vol(K \cap H_+), vol(K \cap H_-)\} \leq vol(K)(1 - 1/e).$$

We do not prove this in this class but refer to the excellent survey of Keith Ball: An Elementary Introduction to Modern Convex Geometry. With this result, the analysis is straightforward.

Proof: [Proof of theorem 6] If in any iteration we find $c_t \in K$ then we are done. Otherwise, in each iteration we reduce the volume by the factor stated above. Therefore in $T = O(n \log(R/\varepsilon))$ iterations we have

$$\mathrm{vol}(K_T) \leq \mathrm{vol}(K_0)(1 - 1/e)^T \leq \mathrm{vol}(B(0, R)) \exp(-n \log(R/\varepsilon)) \leq \mathrm{vol}(B(0, \varepsilon)),$$

where we used that $K \subseteq B(0, R)$. □

In the following section, we show that this is in fact the optimal query complexity possible for an algorithm using just a SEParation oracle. Of course, as stated it is not at all clear how to compute the center-of-gravity of K_t, and it turns out this is at least as hard as optimizing over K_t. Therefore in the following sections we will study the Ellipsoid algorithm, which requires more oracle queries but can be efficiently updated.

We next describe a very similar algorithm to solve general convex programs. In each iteration, we still maintain a feasible convex set $K_t \supseteq K$ and query the center c_t; if $c_t \notin K$ then we update $K_{t+1} = K_t \cap H_t$ just as in the previous algorithm; if $c_t \in K$ then we query the EVALuation and (sub)-GRADient oracle for f, to compute $(f(c_t), g)$; Finally we update $K_{t+1} = K_t \cap H_g$ where

$$H_g := \{x \mid \langle g, x - c_t \rangle \leq \}$$

i.e. the halfspace in the negative (sub-)gradient direction.

Theorem 8. *Given $K \subseteq B(0, R) \subseteq \mathbb{R}^n$ via SEParation oracle, and convex $f : \mathbb{R}^n \to \mathbb{R}$ via EVALuation and GRADient oracle, such that $\max_{x \in K} f(x) - \min_{x \in K} f(x) \leq F$; The center-of-gravity method requires $O(n \log(RF/\varepsilon))$ iterations to either*

1. *find $x \in K$ such that $f(x) \leq \min_{y \in K} f(x_*) + \varepsilon$;*
2. *certify $vol(K) \leq \varepsilon^n$.*

The proof rests on the following claim, showing the negative gradient gives a good update.

Claim 9. *For closed convex $f : \mathbb{R}^n \to \mathbb{R}$ and $x \in dom(f)$, any subgradient $g \in \partial f(x)$ gives a supporting hyperplane for the sub-level set*

$$L_{f(x)} := \{y \mid f(y) \leq f(x)\}.$$

We leave the proof as an exercise in the first problem set. But intuitively the negative subgradient halfspace always contains the optimizer $x_* := \arg\min_{x \in K} f(x)$, so our update maintains a body containing x_*. With this in hand, the analysis follows:

Proof: [Proof of theorem 8] We still manage to decrease the volume by a constant factor in each iteration, due to Grunbaums theorem. And the claim above shows that $K_t \ni x_*$ for all t. If in every iteration $c_t \notin K$, then by the previous feasibility analysis we can certify small volume in $T = O(n \log(R/\varepsilon))$ iterations. Otherwise, we show that when the volume gets small enough, one of the queried center points must have small function value.

For this we analyze the follows approximate optimal set:

$$X_\delta := (1-\delta)x_* + \delta K = \{(1-\delta)x_* + \delta x \mid x \in K\}.$$

Note that we can bound the function on this set as, for $x \in K$,

$$f((1-\delta)x_* + \delta x) \le (1-\delta)f(x_*) + \delta f(x) = f(x_*) + \delta(f(x) - f(x_*)) \le f(x_*) + \delta F,$$

where the first step was by convexity of f, and in the last step we used our bound $\max_{x\in K} f(x) - f(x_*) \le F$. Note that $\text{vol}(X_\delta) = \text{vol}(\delta K) = \delta^n \text{vol}(K)$. Therefore if we reach T such that $\text{vol}(K_T) < \text{vol}(X_\delta)$ then we must have $X_\delta/K_T \neq \emptyset$. Therefore in some iteration we must have queried c_t and cut off some part of X_δ. Let $c_t = (1-\lambda)x_* + \lambda x_\delta$ where $x_\delta \in X_\delta/K_t$ was cut off. Then

$$f(c_t) \le (1-\lambda)f(x_*) + \lambda f(x_\delta) \le f(x_\delta) \le f(x_*) + \delta F,$$

again by convexity and the derived bound on $f(x_\delta \in X_\delta)$. Choosing $\delta \lesssim \varepsilon/F$ gives the result. □

4 Lower Bound

Theorem 10. *Given $K \subseteq B(0,R) \subseteq \mathbb{R}^n$ via SEParation oracle, any algorithm solving the FEASibility problem requires $\Omega(n\log(R/\varepsilon))$ oracle calls to the SEP oracle.*

Proof: [Proof Sketch] Consider the grid example $K = x + [-\varepsilon,\varepsilon]^n \subseteq [0,1]^n$, and note that for each query, the adversay can always choose a separating hyperplane (even a coordinate hyperplane) to keep $\Omega(1)$ fraction of the mass. □

5 Ellipsoid Method [Khachiyan 1979]

Note that while the Center-of-Gravity method provably achieves optimal oracle complexity, as shown by the lower bound above, it is not clear how to implement the procedure in general: computing the centerpoint of a general convex body is at least as hard as optimizing over that body! And maintaining the sequence of intersections of halfspaces could become costly. Therefore, in this section we consider the Ellipsoid method:

Theorem 11. *Given $K \subseteq B(0,R) \subseteq \mathbb{R}^n$ via separation oracle, the Ellipsoid method requires $O(n^2\log(R/\varepsilon))$ iterations to either*

1. *find $x \in K$;*
2. *certify K does not contain a ball of radius ε.*

The strategy involves approximating our feasible set by an Ellipsoid. This can be maintained and updated efficiently. The key geometric lemma that allows us to make progress is as follows:

Lemma 12 (Lemma 2.3 in [Bubeck)**.** *] Let $\mathcal{E} := \{x \in \mathbb{R}^n \mid \langle (x-c), Q(x-c)\rangle \le 1\}$ be an Ellipsoid with $Q \succ 0$; and let $H := \{x \in \mathbb{R}^n \mid \langle w, x\rangle \le \langle w, c\rangle\}$ be a halfspace through the center of $\mathcal{E}$. Then there exists ellipsoid $\overline{\mathcal{E}}$ such that*

$$\overline{\mathcal{E}} \supset \mathcal{E} \cap H \quad \textit{and} \quad \textit{vol}(\overline{\mathcal{E}}) \le \textit{vol}(\mathcal{E})\exp(-1/2n).$$

Further, this update can be efficiently computed in terms of (c, Q) as

$$\overline{c} := c - \frac{1}{n+1}\frac{Q^{-1}w}{\sqrt{\langle w, Q^{-1}w\rangle}}, \qquad \overline{Q} := \left(1 - \frac{1}{n^2}\right)Q + \frac{2(n+1)}{n^2}\frac{ww^T}{\langle w, Q^{-1}w\rangle}.$$

In the following note we prove the the geometric result using a convex program. Given this result, the complexity follows from the same proof as the Center-of-Gravity algorithm, except with n factor worse oracle complexity.