Convex Minimization Problem Specification

Consider obtaining an ϵ-approximate minimum in the convex case. We allow a more general situation, with a constraint set G.

Definition 1 *A problem instance is a pair (f, G), with*

1. *$f : C = [-1,1]^n \to \mathbf{R}$ is convex, with $\max f(C) - \min f(C) \le 1$*
2. *$G \subset C$ is convex, with $G = \emptyset$, or $\mathrm{vol}\,(G) \ge \delta^n$.*

Call $\mathcal{F}$ the set of all such problem instances (f, G).

The algorithm produces $z \in \{*\} \cup C$. We define the error function

$$\epsilon(z, f, G) := \begin{cases} 0 & \text{if } z = * \text{ and } G = \emptyset \\ +\infty & \text{if } G \neq \emptyset \text{ and } z \notin G \\ f(z) - \min f(C) & \text{if } G \neq \emptyset \text{ and } z \in G \end{cases} \tag{1}$$

and define $\mathcal{N}_{\mathcal{F}}(\epsilon)$ as usual (the minimum number of steps required to guarantee an error less than ϵ for any problem instance in $\mathcal{F}$). The oracle provides

$$\begin{cases} \text{a separating/supporting hyperplane} & \text{if } x \notin G \\ \text{a function subgradient pair} & \text{if } x \in G. \end{cases}$$

Theorem 1 (Separating/Supporting Hyperplane) *If G is a closed convex subset of $\mathbf{R}^n$ and $x \notin \mathrm{int}\, G$, then there is a nonzero $v \in \mathbf{R}^n$ with $G \subset \{y : v^T(y - x) \le 0\}$.*

Subdifferential

Definition 2 *If $f : D \subset \mathbf{R}^n \to \mathbf{R}$ is convex, then for any $x \in D$, the subdifferential of f at x is*

$$\partial f(x) := \{g \in \mathbf{R}^n : f(y) \ge f(x) + g^T(y - x) \quad \forall y \in D\}, \tag{2}$$

i.e., the set of all subgradients of f at x.

Theorem 2 *If $f : D \to \mathbf{R}$ is convex, then for every $x \in D$, $\partial f(x)$ is a non-empty, convex compact subset of $\mathbf{R}^n$.*

Proof: Clearly ∂f is closed and convex. Choose $x \in \operatorname{int} D$ so that

$$(x, f(x)) \in \operatorname{epi} f := \{(y, \nu) : \nu \geq f(y)\} \subset \mathbf{R}^{n+1}.$$

It is easy to see that the epigraph is convex. Note that $(x, f(x)) \notin \operatorname{int} \operatorname{epi} f$, because $(x, f(x) - \epsilon) \notin \operatorname{epi} f$ for all $\epsilon > 0$. So there is a supporting hyperplane with normal $(g, \gamma) \in \mathbf{R}^{n+1}$ such that

$$g^T y + \gamma \nu \leq g^T x + \gamma f(x), \quad \forall (y, \nu) \in \operatorname{epi} f \tag{3}$$

Use $(x, f(x) + 1)$ to see $\gamma \leq 0$. If $\gamma = 0$, then $g^T y \leq g^T x$ for all $y \in D$ and $g \neq 0$. But this contradicts $x \in \operatorname{int} D$. By scaling, assume $\gamma = -1$, then (3) with $\nu = f(y)$ gives $f(y) \geq f(x) + g^T(y - x)$, and thus g is a subgradient (and hence $\partial f(x)$ is non-empty).

For boundedness, for each i, take $x + \epsilon\, e_i \in D$ for sufficiently small $\epsilon > 0$. Then $f(x + \epsilon\, e_i) \geq f(x) + g^T(\epsilon\, e_i)$ implies

$$e_i^T g \leq \frac{f(x + \epsilon\, e_i) - f(x)}{\epsilon},$$

and similarly, $y = x - \epsilon\, e_i \in D$ gives

$$e_i^T g \geq -\frac{f(x - \epsilon\, e_i) - f(x)}{\epsilon}.$$

Thus, $\partial f(x)$ is bounded. □

For problem instance (f, G), if we query the oracle at $x \in \mathbf{R}^n$:

If $x \notin G \cap \operatorname{int} C$	the oracle confirms this and returns a nonzero $v \in \mathbf{R}^n$ such that $v^T y \leq v^T x$ for all $y \in G$.
If $x \in G \cap \operatorname{int} C$	the oracle confirms this and returns $(f(x), g(x))$ with $g(x) \in \partial f(x)$.

We now want to get *lower* bounds on $\mathcal{N}_{\mathcal{F}}(\epsilon)$ (by constructing nasty examples) and *upper* bounds (by constructing algorithms). These will *match* up to a constant.

Convex Minimization Lower Bounds

For lower bounds, recall $G \neq \emptyset$ or $\operatorname{vol} G \geq \delta^n$, where δ is known.

Theorem 3

$$\mathcal{N}_{\mathcal{F}}(\epsilon) \geq n \left(\left\lceil \log_2 \frac{2}{\delta} \right\rceil - 1 \right).$$

Proof: We use (f, G) with $f = 0$ and G a small cube with side lengths δ. Let $h = \lceil \log_2(2/d) \rceil - 1$, meaning $2^h < 2/\delta$, and $\delta < \gamma := 2 \cdot 2^{-h}$. The algorithm's choices will determine a sequence $C = B_0 \supset B_1 \supset B_2 \supset \cdots$ of boxes, constructed as follows.

If the algorithm queries $x \in \mathbf{R}^n \setminus C$, the oracle returns

$$\begin{cases} e_i & \text{if } x^{(i)} > 1 \\ -e_i & \text{if } x^{(i)} < -1. \end{cases}$$

Otherwise we have boxes $B_0, \ldots, B_m$, and $j = \operatorname{argmax}\{i : x \in B_i\}$. If $j < m$, return v_j. These two cases cover queries one should not ask the oracle (due to redundancy or not using information properly).

If $x \in B_m =: \{y : p \leq y \leq q\}$, we choose $k = m \bmod n+1$ (cycling through the components), and if $x^{(k)} \geq \frac{1}{2}(p^{(k)} + q^{(k)})$, set

$$v_m = e_k \qquad B_{m+1} := \left\{ y \in B_m : y^{(k)} \leq \frac{1}{2}(p^{(k)} + q^{(k)}) \right\}.$$

Otherwise, if $x^{(k)} < \frac{1}{2}(p^{(k)} + q^{(k)})$, set

$$v_m = -e_k \qquad B_{m+1} := \left\{ y \in B_m : y^{(k)} > \frac{1}{2}(p^{(k)} + q^{(k)}) \right\}.$$

Example 1 *Let $n = 2$.*

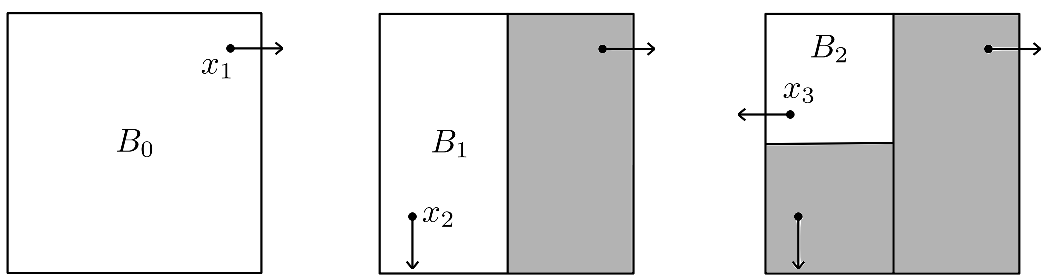

Figure 1: These answers are all consistent with *any* $G \subset B_m$, where B_m is the last box of any iteration.

After $nh - 1$ calls to the oracle, we have produced at must $nh - 1$ boxes, so components $i = 1 \ldots n-1$ have been bisected at most h times, while component n has been bisected at most $h - 1$ times.

So the current box contains a double cube of sides $\gamma = 2 \cdot 2^{-h}$, $n-1$ times, and $2 \cdot 2^{-h+1} = 2\gamma$, once.

Then the algorithm produces z, and now choose G to be a cube of side length δ in the interior of one half of the double cube not containing z.

So the algorithm has $z \notin G$, or $z = *$, while $G \neq \emptyset$, and therefore, $\epsilon(z, f, G = +\infty$.

Hence we need $\mathcal{N}_{\mathcal{F}}(\epsilon) > nh$. $\square$

Similar but more complicated arguments give

Theorem 4

$$\mathcal{N}_{\mathcal{F}}(\epsilon) \geq cn \log\left(\frac{1}{\epsilon}\right),$$

for some absolute constant c.

We will work towards an algorithm that requires

$$2.2\, n \left(\ln \frac{2}{d} + \ln \frac{1}{\epsilon}\right)$$

calls to the oracle to guarantee $\epsilon(z, f, G) \leq \epsilon$.

Classical Paradigm: at each iteration, an algorithm builds an approximation using the problem's data (e.g. QP or LP) and solves the subproblem. This approach requires analysis.

New Paradigm: at each iteration, an algorithm gains information about the location of the minimizer. This approach requires geometry.