Complexity of Convex Optimization

1 Solvability of Convex Optimization

Recall the convex optimization problem:

$$\begin{aligned} \min \ & f(\boldsymbol{x}) \\ \text{s.t. } & \boldsymbol{x} \in \mathcal{X} \end{aligned}$$

Where $\mathcal{X} \in \mathbb{R}^n$ is convex and closed, and $f : \mathbb{R}^n \to \mathbb{R}$ is also convex. Denote the length (in bits) of the description of the problem by $\langle I \rangle$.

We say that an algorithm **weakly solves** a convex optimization problem in polynomial time if the algorithm:

- Takes an approximation parameter $\epsilon > 0$
- Terminates in time $poly(\langle I \rangle, n, \log \frac{1}{\epsilon})$. Note that this is a pretty good bound, because it is logarithmic in the bit precision.
- Returns an ϵ-optimal $\boldsymbol{x} \in \mathcal{X}$, that is an $\boldsymbol{x}$ with

$$f(\boldsymbol{x}) \leq \min_{\boldsymbol{y} \in \mathcal{X}} f(\boldsymbol{y}) + \epsilon[\max_{\boldsymbol{y} \in \mathcal{X}} f(\boldsymbol{y}) - \min_{\boldsymbol{y} \in \mathcal{X}} f(\boldsymbol{y})]$$

Theorem 1. *Consider a family* Π *of convex optimization problems* $I = (f, \mathcal{X})$ *admitting the following operations in polynomial time:*

- *A separation oracle for* $\mathcal{X}$
- *A first-order oracle for* f*, that is an oracle which evaluates* $f(\boldsymbol{x})$ *and* $\nabla f(\boldsymbol{x})$
- *An algorithm which computes a starting ellipsoid* $E \supseteq \mathcal{X}$ *with* $vol(E)/vol(\mathcal{X}) = \mathcal{O}(\exp(\langle I \rangle, n))$

Then, there is a polynomial time algorithm which weakly solves Π.

Proof. By reduction to the Ellipsoid Algorithm. We start with a simplifying assumption, then prove the general case. Let's assume that we know $\min_{\boldsymbol{y} \in \mathcal{X}} f(\boldsymbol{y})$ and $\max_{\boldsymbol{y} \in \mathcal{X}} f(\boldsymbol{y})$; we can then rescale them to be 0 and 1. Then, our convex optimization problem becomes the convex feasibility problem:

$$\begin{aligned} \text{find } & \boldsymbol{x} \\ \text{s.t. } & \boldsymbol{x} \in \mathcal{X} \\ & f(\boldsymbol{x}) \leq \epsilon \end{aligned}$$

Note that the separation oracle for the new feasible set $K = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \leq \epsilon\}$ can be obtained from the separation oracle for $\mathcal{X}$ and the first-order oracle for f, and the starting ellipsoid is simply E because $K \subseteq \mathcal{X} \subseteq E$. But, we need to guarantee that $vol(E)/vol(K) \leq \exp(\langle I \rangle, n, \log \frac{1}{\epsilon})$.

Lemma 2. $vol(K) \geq \epsilon^n vol(\mathcal{X})$. *This shows that* $vol(K)$ *is only exponentially smaller than* $vol(\mathcal{X})$ *and therefore* $vol(E)$.

Proof. Assume wlog $\mathbf{0} \in \mathcal{X}$, $f(\mathbf{0}) = \min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) = 0$. Consider scaling $\mathcal{X}$ by ϵ to obtain $\epsilon\mathcal{X}$. Then $vol(\epsilon\mathcal{X}) = \epsilon^n vol(\mathcal{X})$ because $\det(\begin{bmatrix} \epsilon & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \epsilon \end{bmatrix}) = \epsilon^n$. Now, we show that $\epsilon\mathcal{X} \subseteq K$ by showing that $f(\boldsymbol{y}) \leq \epsilon \; \forall \boldsymbol{y} \in \epsilon\mathcal{X}$. Let $\boldsymbol{y} = \epsilon\boldsymbol{x}$ for some $\boldsymbol{x} \in \mathcal{X}$. Then by Jensen's Inequality,

$$f(\boldsymbol{y}) = f(\epsilon\boldsymbol{x} + (1-\epsilon)\mathbf{0}) \leq \epsilon f(\boldsymbol{x}) + (1-\epsilon) f(\mathbf{0}) \leq \epsilon$$

□

Since we established the lemma, we can just use the Ellipsoid Algorithm to solve the feasibility problem!

Now consider the general case. Let $L = \min_{\boldsymbol{y} \in \mathcal{X}} f(\boldsymbol{y})$ and $H = \max_{\boldsymbol{y} \in \mathcal{X}} f(\boldsymbol{y})$. If we know the target $T = L + \epsilon[H - L]$, we can reduce to solving the feasibility problem over $K = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \leq T\}$.

Remark 3. We don't need to know T, H, or L to simulate the execution of the Ellipsoid Algorithm on K!

Our convex feasibility problem is:

$$\begin{aligned} &\text{find } \boldsymbol{x} \\ &\text{s.t. } \boldsymbol{x} \in \mathcal{X} \\ &\qquad f(\boldsymbol{x}) \leq T \end{aligned}$$

We can simulate the execution of the Ellipsoid Algorithm on K for a polynomial number of iterations, after which we will have a point in K. Note that the action of the algorithm at each step other than the last can be described independently of T: if the center of the ellipsoid $\boldsymbol{c} \notin \mathcal{X}$, we obtain a separating hyperplane with $\mathcal{X}$; otherwise, we can use $\nabla f(\boldsymbol{c})$. We only cannot directly check $\boldsymbol{c} \in K$. Thus, T is only necessary for describing when to stop, but since we know the Ellipsoid Algorithm terminates after a polynomial number of iterations, we can just run it for "long enough" and look back at the transcript to obtain a point $\boldsymbol{x} \in K$. Thus we can find an ϵ-optimal solution to the convex program in polynomial time. □

2 Complexity of LP

Recall the LP problem:

$$\begin{aligned} \max \;& \boldsymbol{c}^\intercal \boldsymbol{x} \\ \text{s.t. } & \mathbf{A}\boldsymbol{x} \preceq \boldsymbol{b} \end{aligned}$$

Recall also that the optimal solution always occurs at a vertex by complementary slackness. Consider both explicit and implicit LPs; that is, we are either given $\mathbf{A}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ directly, or we are given $\boldsymbol{c}$ and a separation oracle for $\mathbf{A}\boldsymbol{x} \preceq \boldsymbol{b}$. In the explicit case, we require runtime $poly(\langle \mathbf{A} \rangle, \langle \boldsymbol{b} \rangle, \langle \boldsymbol{c} \rangle)$; in the implicit case, we require runtime polynomial in the bit complexity of the individual entries of $\mathbf{A}$, $\boldsymbol{b}$, and $\boldsymbol{c}$. Note that this requires all numbers to be rational.

Theorem 4. *There exists a polynomial time algorithm for LPs represented explicitly.*

Proof. Using our result for convex programs, we need:

1. A separation oracle for $\mathbf{A}\boldsymbol{x} \preceq \boldsymbol{b}$
2. A first-order oracle for $\boldsymbol{c}^\intercal \boldsymbol{x}$
3. A bounding ellipsoid with volume at most exponential relative to the feasible polyhedron
4. A method of "rounding" an optimal solution to an optimal vertex

1 and 2 are trivially given. The key to both 3 and 4 is the following lemma, which follows from Gaussian Elimination:

Lemma 5. *Let $\boldsymbol{v}$ be a vertex of the feasible polyhedron. It is the case that $\boldsymbol{v}$ has polynomial bit complexity; that is, $\langle \boldsymbol{v} \rangle \leq M$ where $M = \mathcal{O}(poly(\langle \mathbf{A} \rangle, \langle \boldsymbol{b} \rangle))$. In other words, the solution of a linear system has bit complexity polynomially related to that of the equations of the system.*

For the bounding ellipsoid, we can take E to be all vertices in the box $-2^M \leq \boldsymbol{x} \leq 2^M$, which is then contained in an ellipsoid of volume exponential in M and n. To guarantee the lower bound, we have to add ϵ to $\boldsymbol{b}$, just to ensure that the polyhedron is of full dimension.

For the rounding, note that if $\boldsymbol{y}$ is ϵ-optimal for the ϵ-relaxed LP, then for sufficiently small ϵ chosen carefully to polynomial in the description of the input, rounding to the nearest $\boldsymbol{x}$ with M bits recovers the vertex. The intuition here is easily described for integer programs: if you can solve to the nearest 0.1, then you can easily round to the nearest integer solution. □

Theorem 6. *Consider a family Π of LP problems $I = (\mathbf{A}, \boldsymbol{b}, \boldsymbol{c})$ admitting the following operations in $poly(\langle I \rangle, n)$:*

- *A separation oracle for the feasible polyhedron $\mathbf{A}\boldsymbol{x} \preceq \boldsymbol{b}$*
- *Explicit access to $\boldsymbol{c}$*

Moreover assume that every $\langle a_{ij} \rangle$, $\langle b_i \rangle$,$\langle c_j \rangle$ are at most $poly(\langle I \rangle, n)$. Then, there is a polynomial time algorithm for both the primals and duals in Π.

Proof. We prove the primal and leave the dual for next lecture. We need the same 4 elements as the above theorem, with the separation oracle and first order given. Since rounding to an optimal vertex is exactly the same as the last theorem, we are only concerned about a bounding ellipsoid. We can't just add ϵ into the feasible polyhedron anymore, since we don't have an explicit description, so we can't get a lower bound easily. It turns out this is OK, but is very difficult to prove. In essence, we need to figure out if the volume of the feasible polyhedron is 0 and jump into a lower-dimensional subspace (related to simultaneous Diophantine approximation). □