Stochastic Convex Optimization **Fall 2021**

Lecture 3: Cutting Plane Methods

Lecturer: Roi Livni *Scribe: Idan Amir*

In previous lecture we defined the standard setting of convex optimization. Today we will encounter and review a few *algorithms* to optimize convex problem, from a family of methods termed *cutting plane methods*.

Recall that we consider a standard setup where we want to minimize a convex function f, and we assume a *first order oracle*. Namely, given a parameter x we can obtain a (sub)gradient g of the function f at point x, and we start by defining an abstract algorithm that exploits a first order oracle to the function f in order to find a minimizer of f.

3.0.1 Center of Gravity Algorithm

Algorithm 1 Center of Gravity

Let $S_1 = \mathcal{W}$

for $t \geq 1$ **do**

Compute $c_t = \frac{1}{\text{vol}(S_t)} \int_{x \in S_t} x dx$

Query first-order oracle at c_t and obtain $g_t \in \partial f(c_t)$

Set $S_{t+1} = S_t \cap \{x \in R^n : g_t^T(x - c_t) \leq 0\}$

end for

If we stop after t_0 steps, use the zeroth-order oracle to find $\arg\min_{c_t \in \{c_1, \ldots, c_{t_0}\}} f(c_t)$

Theorem 3.1. *Suppose we ran the COG(Center of Gravity) algorithm for t iterations and $w_t = \arg\min_{c' \in \{c_1 \dots c_t\}} f(c')$ then,*

$$f(w_t) - \min_{w \in \mathcal{W}} f(w) \leq 2C \left(1 - \frac{1}{e}\right)^{\frac{t}{n}}$$

where $f : \mathcal{W} \to [-C, C]$.

Relying on theorem 3.1 we can find the condition on the number of iterations t that guarantees an ϵ-accurate solution.

$$f(w_t) \leq f(w^\star) + \epsilon$$

$$\Downarrow$$

$$2C \left(1 - \frac{1}{e}\right)^{\frac{t}{n}} \leq \epsilon$$

$$\Downarrow$$

$$n \log \frac{2C}{\epsilon} \log^{-1}\left(\frac{e}{e-1}\right) \leq t$$

therefore $t = O\left(n \log \frac{2C}{\epsilon}\right)$.

Lemma 3.2 (Grünbaum (without proof))**.** *Given a convex body $\mathcal{W}$ centered around 0, namely $\int_{x\in\mathcal{W}} x dx = 0$, then for any $v \in \mathbb{R}^n$:*

$$\text{vol}\left(\mathcal{W} \cap \{x \in \mathbb{R}^n : v^T x \geq 0\}\right) \geq \frac{1}{e}\text{vol}(\mathcal{W})$$

proof of theorem 3.1. Denote $w^\star \in \mathcal{W}$ such that $f(w^\star) = \min_{w\in\mathcal{W}} f(w)$. In round t we obtain $g_t \in \partial f(c_t)$ therefore,

$$\forall w \in \mathcal{W} \quad f(c_t) - f(w) \leq g_t^T(c_t - w)$$

We examine the set $S_t \setminus S_{t+1}$:

$$\begin{aligned} S_t \setminus S_{t+1} &\subset \{w \in \mathcal{W} : (w - c_t)^T g_t > 0\} \\ &\subseteq \{w \in \mathcal{W} : f(w) > f(c_t)\} \end{aligned}$$

this implies that $w^\star \in S_t$ for all t. Without loss of generality, assume that $g_t \neq 0$ otherwise $0 \in \partial f(c_t)$ and thus c_t is the minimum. Applying lemma 3.2 recursively we obtain,

$$\text{vol}(S_{t+1}) \leq \left(1 - \frac{1}{e}\right)^t \text{vol}(\mathcal{W})$$

Suppose $0 < \epsilon < 1$ and define,

$$X_\epsilon = \{(1-\epsilon)w^\star + \epsilon w : w \in \mathcal{W}\}$$

An immediate observation is that $\text{vol}(X_\epsilon) = \epsilon^n \text{vol}(\mathcal{W})$ and for $\epsilon > (1 - \frac{1}{e})^{\frac{t}{n}}$ we get,

$$\text{vol}(S_{t+1}) < \text{vol}(X_\epsilon)$$

This implies that there exists $\bar{w} \in X_\epsilon$ such that $\bar{w} \notin S_{t+1}$. It also means that there exists some round t' such that $\bar{w} \in S_{t'}$ and $\bar{w} \notin S_{t'+1}$. Thus, by the nature of our algorithm, we can deduce that $(\bar{w} - c_{t'})^T g_{t'} > 0$ and finally conclude,

$$\begin{aligned} f(c_{t'}) &< f(\bar{w}) \\ &= f((1-\epsilon)w^\star + \epsilon w) \\ &\leq (1-\epsilon) f(w^\star) + \epsilon C \\ &\leq f(w^\star) + 2\epsilon C \end{aligned}$$

This result guarantees that there exists a round t' for $\epsilon > (1-\frac{1}{e})^{\frac{t}{n}}$ such that $f(c_{t'}) - f(w^\star) \leq 2C \left(1 - \frac{1}{e}\right)^{\frac{t}{n}}$. $\square$

Note that in the center of gravity algorithm we need, at each iteration to compute the center-of-gravity. This is in many cases not feasible or relatively complex. But we were able to show that the algorithm is efficient in regard to number of oracle calls.

Exercise 3.1. *In the center of gravity method we do not choose the last iteration. Show an example of a convex function where the sequence of values in the centers* $f(c_1), \ldots, f(c_t)$ *is not monotonically decreasing: In particular, the last iteration need not be optimal.*

3.1 The Ellipsoid Method

3.1.1 Separation Oracle

As discussed the main drawback of the center of gravity method is that it assumes we can compute the center of gravity of an arbitrary set. The next algorithm we present avoids this computation, but requires a separation oracle for the convex set $\mathcal{W}$:

Definition 3.3 (Separation Oracle)**.** *Consider a convex set $\mathcal{W} \subseteq \mathbb{R}^n$. A separation oracle for the set $\mathcal{W}$ receives $x \in \mathbb{R}^n$ and states if $x \in \mathcal{W}$, and if $x \notin \mathcal{W}$ it provides v such that*

$$v^\top x > v^\top w \quad \forall w \in \mathcal{W}.$$

Example 3.1 (LP-Linear Programming)**.** *Linear programs are problems that take the form:*

$$\begin{aligned} \text{minimize} \quad & c^\top w \\ \text{s.t.} \quad & A^\top w \leq 0 \end{aligned}$$

For a given z, a separation oracle would check if $A^T z \leq 0$, and if not then there exists a row i such that $a_i^T z > 0$ and thus $g_t = a_i$.

Example 3.2 (SDP-Semidefinite Programming). *In SDP we minimize over the positive semidefinite matrices domain, namely* $S_+ = \{X \in \mathbb{R}^{n \times n} : X = X^T, X \succeq 0\}$. *Semidefinite programs take the form:*

$$\begin{aligned} \text{minimize} \quad & Tr(X \cdot C) \\ \text{s.t.} \quad & X \in S_+ \\ & Tr(X \cdot A_1) \leq b_1 \\ & \vdots \\ & Tr(X \cdot A_i) \leq b_i \end{aligned}$$

A well known SDP problem in learning is the matrix completion problem defined as:

$$\begin{aligned} \text{minimize} \quad & Tr(X) \\ \text{s.t.} \quad & X_{i,j} = Y_{i,j} \\ & \forall (i,j) \in \Omega \end{aligned}$$

When Ω is the set of all observed indices of the matrix X.

Exercise 3.2. *Show that an SDP is a convex program. In particular, consider $\mathbb{R}^{n^2}$ as the space of n by n matrices and show*

- *That the mapping $X \to Tr(X \cdot C)$ is a linear mapping for every C. In particular, convex.*
- *The set S_+ is convex.*
- *For every A and b the set $\{X : Tr(X \cdot A) \leq b\}$ is convex.*
- *If $\mathcal{W}_1, \ldots, \mathcal{W}_i$ are convex sets, then $\cap \mathcal{W}_i$ is a convex set.*
- *Conclude that an SDP is a convex program.*

Exercise 3.3. *Consider the space of matrices with the scalar product: $< A, B >= \sum_{i,j} A_{i,j} B_{i,j}$. In other words, we consider each matrix $M \in \mathbb{R}^{n \times n}$ as a vector in $\mathbb{R}^{n^2}$. Show that we can efficiently build a separation oracle for an SDP (the complexity may scale with number of constraints and dimensions).*

We begin to describe the ellipsoid method:

Definition 3.4. *An ellipsoid is a convex set of the form,*

$$\mathcal{E} = \{x \in \mathbb{R}^n : (x-c)^T H^{-1}(x-c) \leq 1\}$$

where $c \in \mathbb{R}^n$ *and* H *is a symmetric positive definite matrix.*

Lemma 3.5 (e.g, [1])**.** *Let* $\mathcal{E}_0 = \{x \in \mathbb{R}^n : (x-c_0)^T H_0^{-1}(x-c_0) \leq 1\}$*. For any* $g \in \mathbb{R}^n$*,* $g \neq 0$*, there exists an ellipsoid* $\mathcal{E}$ *such that:*

$$\mathcal{E} \supset \{x \in \mathcal{E}_0 : g^T(x-c_0) \leq 0\}$$

and

$$vol(\mathcal{E}) \leq e^{-\frac{1}{2n}} vol(\mathcal{E}_0)$$

Furthermore for $n \geq 2$ *one can take* $\mathcal{E} = \{x \in \mathbb{R}^n : (x-c)^T H^{-1}(x-c) \leq 1\}$ *where,*

$$c = c_0 - \frac{1}{n+1}\frac{H_0 g}{\sqrt{g^\top H_0 g}}$$
$$H = \frac{n^2}{n^2-1}\left(H_0 - \frac{2}{n+1}\frac{H_0 g g^\top H_0}{g^\top H_0 g}\right)$$

3.1.2 Ellipsoid Method Algorithm

Recall that a separation oracle receives $x \in \mathbb{R}^n$ and states if $x \in \mathcal{W}$ and if $x \notin \mathcal{W}$ it provides v such that $v \cdot x \geq v \cdot k$ for every $k \in \mathcal{W}$. We describe now the ellipsoid method, which assumes a separation oracle for the set $\mathcal{W}$. Let $\mathcal{E}_0 = B(0, R)$ be the ball of radius R that contains $\mathcal{W}$.

Algorithm 2 Ellipsoid Method

Let $c_0 = 0$, $H_0 = R^2 \cdot I_{n\times n}$, $\mathcal{E}_0 = B(0, R)$

for $t = 0, \ldots, t'$ **do**

if $c_t \notin \mathcal{W}$ **then**

Call separation oracle to obtain $g_t \in \mathbb{R}^n$ such that

$$\mathcal{W} \subset \{x : g_t^\top (x - c_t) \leq 0\}$$

else

Call first order oracle to obtain $g_t \in \partial f(c_t)$

end if

Let $\mathcal{E}_{t+1}$ be the ellipsoid given in Lemma 3.5 that contains $\{x \in \mathcal{E}_t : g_t^T (x - c_t) \leq 0\}$

end for

if $\{c_1, \ldots, c_{t'}\} \cap \mathcal{W} \neq \emptyset$ **then**

Use the zeroth order oracle to output

$$w_{t'} = \operatorname*{arg\,min}_{c \in \{c_1, \ldots, c_{t'}\} \cap \mathcal{W}} f(c)$$

end if

Theorem 3.6. *For $t' \geq 2n^2 \log(R/r)$ the ellipsoid method satisfies $\{c_1, \ldots, c_{t'}\} \cap \mathcal{W} \neq \emptyset$ and*

$$f(w_{t'}) - \min_{x \in \mathcal{W}} f(x) \leq \frac{2CR}{r} \exp\left(-\frac{t'}{2n^2}\right)$$

where $f : \mathcal{W} \to [-C, C]$.

We observe that the number of iterations needed to guarantee an ϵ-accurate solution is $O\left(n^2 \log(\frac{1}{\epsilon})\right)$, which is worse than the one of the center of gravity method. However, from a computational point of view we favor the ellipsoid method since that in many cases one can derive an efficient separation oracle, while the center of gravity method is basically always intractable.

Ellipsoid method and Center of Gravity are both parts of a family of methods called *cutting plane methods* (where they basically rely on the fact that at each iteration, we "cut" the search-space using a plane). An improvement over the Ellipsoid method, is Vaidya's cutting plane method that achieves oracle complexity of $\tilde{O}(n)$ (We use $\tilde{O}(n)$ notation to supress logarithmic factor i.e. $\tilde{O}(n) = O(n \cdot \text{poly}(\log n))$). Grötschel, Lorasz and Schrijver showed that under the membership oracle assumption one can achieve oracle complexity of $\tilde{O}(n^2)$ [2].

References

[1] S. Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[2] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.