The University of Texas at Austin
Department of Electrical and Computer Engineering

ECE 381V: Large-Scale Optimization II — Spring 2022

Lecture 9

Caramanis & Mokhtari Wednesday, February 16, 2022

Goal: In this lecture, we talk about projection-free methods, including the Frank-Wolfe algorithm and the gradient sliding algorithm.

1 Convex Constrained Optimization

Consider a general constrained convex optimization problem where the constraint set is a simple convex set with no functional constraints:

$$\begin{aligned} &\min f(\mathbf{x}) \\ &\text{s.t } \mathbf{x} \in \mathcal{Q}, \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable on the convex set $\mathcal{Q}$. **We further assume that the set $\mathcal{Q}$ is compact.**
We assume that the cost of projecting a point onto the set $\mathcal{Q}$ is not low. On the other hand, the cost of solving a linear program of the following form (for a given vector $\mathbf{a}$) is low.

$$\begin{aligned} \min_{\mathbf{v}} \quad & \mathbf{v}^\top \mathbf{a} \\ \text{s.t} \quad & \mathbf{v} \in \mathcal{Q}, \end{aligned} \tag{2}$$

2 Frank-Wolfe (Conditional Gradient) Algorithm

In projection gradient-based methods, we first take a gradient step and find a new point and then project back the obtained point onto the feasible set $\mathcal{Q}$ to obtain a feasible point. Unlike this approach, in the Frank-Wolfe algorithm, we first find a feasible point that moving to that point maximizes the function decrease and then pick our new point as a convex combination of that point and the previous point. To be more precise, consider $\mathbf{x}_k$ as the current iterate at step k that is a feasible point $\mathbf{x}_k \in \mathcal{Q}$. We first find a feasible point that has the smallest inner product with the current gradient

$$\mathbf{v}_k = \underset{\mathbf{v} \in \mathcal{Q}}{\operatorname{argmin}} \ \mathbf{v}^\top \nabla f(\mathbf{x}_k)$$

Then we compute the new point $\mathbf{x}_{k+1}$ as a convex combination of $\mathbf{v}_k$ and the previous $\mathbf{x}_k$, i.e.,

$$\mathbf{x}_{k+1} = (1 - \gamma_k)\mathbf{x}_k + \gamma_k \mathbf{v}_k$$

where $\gamma_k \in [0, 1]$.
To better understand the intuition behind the above update note that $\mathbf{v}_k$ can also be written as

$$\mathbf{v}_k = \underset{\mathbf{v} \in \mathcal{Q}}{\operatorname{argmin}} \ f(\mathbf{x}_k) + (\mathbf{v} - \mathbf{x}_k)^\top \nabla f(\mathbf{x}_k)$$

which shows that it is a feasible point that minimizes the linear approximation of the function. Note that $\mathbf{v}_k$ **may not be the same or parallel to** the negative of the gradient direction $-\nabla f(\mathbf{x}_k)$.

Remark 1. *A major advantage of the Frank-Wolfe algorithm is that it does not require projection steps to maintain feasible points, and simply requires solving a linear cost over the set $\mathcal{Q}$ at each iteration. Such subproblems could be easily solved in several applications. For example, when our constraint set is a polytope, projection to the polytope is much more costly than minimizing a linear loss over the polytope.*

2.1 Analysis for the Convex and Smooth Setting

Next, we analyze the Frank-Wolfe algorithm for the case of that the objective function f is convex and smooth and the feasible set $\mathcal{Q}$ is convex and compact.

Theorem 1. *Let $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$. If $\gamma_k = \min\{1, \frac{2}{k}\}$, then the iterates of Frank-Wolfe satisfy*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2LD^2}{k}$$

where D denotes the diameter of the set $\mathcal{Q}$.

Proof. Using the smoothness property of the objective function we have

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ &= f(\mathbf{x}_k) + \gamma_k \nabla f(\mathbf{x}_k)^\top (\mathbf{v}_k - \mathbf{x}_k) + \frac{L\gamma_k^2}{2}\|\mathbf{v}_k - \mathbf{x}_k\|^2 \end{aligned}$$

where the equality follows from the update of $\mathbf{x}_{k+1}$. Now using the fact that we have the inequality $\nabla f(\mathbf{x}_k)^\top \mathbf{v}_k \leq \nabla f(\mathbf{x}_k)^\top \mathbf{x}^*$ we can write

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \gamma_k \nabla f(\mathbf{x}_k)^\top (\mathbf{x}^* - \mathbf{x}_k) + \frac{L\gamma_k^2}{2}\|\mathbf{v}_k - \mathbf{x}_k\|^2 \\ &\leq f(\mathbf{x}_k) + \gamma_k (f(\mathbf{x}^*) - f(\mathbf{x}_k)) + \frac{L\gamma_k^2 D^2}{2} \end{aligned}$$

where the second inequality holds do to the convexity of f and the fact that $\mathcal{Q}$ is compact and has diameter D. Hence, we have

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) \leq (1 - \gamma_k)(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{L\gamma_k^2 D^2}{2}$$

Now using a simple induction argument one can show that

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2LD^2}{k}.$$

□

2.2 Lower Bound

Next, we state a lower bound for the class of algorithms that the iterates are generated according to linear combination of a set of points that are obtained by solving a linear solver. More precisely, consider the following class of algorithms

$$\mathbf{v}_k = \operatorname*{argmin}_{\mathbf{v}\in\mathcal{Q}} \mathbf{p}_k^\top \mathbf{v}$$

and

$$\mathbf{x}_{k+1} \in \mathbf{Span}\{\mathbf{x}_0, \mathbf{v}_0, \ldots, \mathbf{x}_k, \mathbf{v}_k\}$$

Observe the above class of algorithms can be quite general. Firstly, there are no restrictions regarding the definition of the linear function $\mathbf{p}_k^\top \mathbf{v}$. For example, then $\mathbf{p}_k$ can be defined as the gradient computed at some feasible solution or a linear combination of some previously computed gradients.

Theorem 2. *Let $\epsilon > 0$ be a given target accuracy. The number of iterations required by any method in the above class of algorithms to solve* (1) *when $f \in \mathcal{F}_L^{1,1}$, in the worst case, cannot be smaller than*

$$\left\lceil \min\left\{\frac{n}{2}, \frac{LD_{\mathcal{Q}}^2}{4\epsilon}\right\}\right\rceil - 1$$

Proof. Consider the function

$$f(\mathbf{x}) = \frac{L}{2}\sum_{i=1}^n x_i^2$$

and the feasible set

$$\mathcal{Q} = \left\{\mathbf{x}\in\mathbb{R}^n \mid \sum_{i=1}^n x_i = D, \mathbf{x}\geq \mathbf{0}\right\}$$

One can easily verify that

$$\mathbf{x}^* = \left(\frac{D}{n}, \ldots, \frac{D}{n}\right), \qquad f^* = \frac{LD^2}{n}$$

Without loss of generality, we assume that the initial point is given by $\mathbf{x}_0 = D\mathbf{e}_1$ where $\mathbf{e}_1$ is the unit vector. Now suppose our problem is to be solved by any algorithm described above. At the k-th iteration, this algorithm will call the linear oracle to compute a new search point $\mathbf{v}_k$ based on the input vector $\mathbf{p}_k$. We assume that the linear oracle is resisting in the sense that it always outputs an extreme point $\mathbf{v}_k \in \{\mathbf{e}1, \ldots, \mathbf{e}_n\}$ for $k \leq n$. Let's call e_{pi} as the extreme point selected at time i. Then, we have

$$f(\mathbf{x}_{k+1}) \geq \min_{\mathbf{x}\in\mathcal{Q}, \mathbf{x}\in\mathbf{Span}\{\mathbf{e}_{p0}\ldots,\mathbf{e}_{pk}\}} f(\mathbf{x}) \geq \frac{LD^2}{k+1}$$

Hence,

$$f(\mathbf{x}_{k+1}) - f^* \geq \frac{LD^2}{k+1} - \frac{LD^2}{n}$$

Further note that $D_{\mathcal{Q}} = \sqrt{2}D$. Hence,

$$f(\mathbf{x}_k) - f^* \geq \frac{LD_{\mathcal{Q}}^2}{2}\left(\frac{1}{k} - \frac{1}{n}\right)$$

Hence, by setting the right hand side to be ϵ, it can be easily verified that at least

$$\left\lceil \min\left\{\frac{n}{2}, \frac{LD_{\mathcal{Q}}^2}{4\epsilon}\right\}\right\rceil - 1$$

iterations is required to find an ϵ accurate solution. □

2.3 Observations

This result shows that using the class of algorithms mentioned above one can not achieve acceleration and a rate of $1/k^2$. Moreover, in the case of strongly convex setting, there exists an example for which achieving a linear rate is not possible. (Check Amir Beck's book on First-order methods)

3 Conditional Gradient Sliding

Note that the Frank-Wolfe algorithm requires $O(1/\epsilon)$ gradient evaluations and $O(1/\epsilon)$ linear optimizations for obtaining an ϵ-optimal solution in primal gap for convex smooth objective functions. In this section, we present the *Conditional Gradient Sliding algorithm (CGS)*, which reduces the required gradient evaluations to the minimal number of $O(1/\sqrt{\epsilon})$ without increasing the number of linear optimizations. Moreover, for the case of strongly-convex setting, it requires $O(\log(1\epsilon))$ gradient computations, which manifests a linear convergence rate, while requiring $O(1/\epsilon)$ linear optimizations. In order to achieve these results, CGS is essentially an implementation of Nesterov's accelerated projected gradient descent algorithm, using the vanilla Frank–Wolfe algorithm for performing approximate projections back into the feasible regions.
Let's first look at the idea of conditional gradient sliding for the projection gradient method. In the projection gradient method we follow the updates

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{Q}}\{\|\mathbf{x} - (\mathbf{x}_k - \eta_k \nabla f(\mathbf{x}_k))\|^2\}$$

Note that the projection step is a quadratic problem and one can try to solve it using a Frank-Wolfe update. Basically what we can do is at iteration k we follow the updates

$$\mathbf{x}_{k+1} = \text{the output of FW on the projection problem until we achieve an error of } \delta_k$$

Using the above approach we have avoided any projection steps, and we simply have approximately implemented the projection gradient method. Indeed, the correct implementation of the above method requires proper selection of the parameter δ_k.

Now one can use the same idea for the projected accelerated gradient method to achieve a faster convergence rate of $1/k^2$ in the convex setting and a linear convergence rate in the strongly convex and smooth setting.
To be more specifc, the update of Projected Accelerated Nesterov method can be written as

$$\begin{aligned}
\mathbf{y}_k &= (1-\gamma_k)\mathbf{z}_k + \gamma_k \mathbf{x}_k \\
\mathbf{x}_{k+1} &= \operatorname*{argmin}_{\mathbf{x}\in\mathcal{Q}} \left\{\nabla f(\mathbf{y}_k)^\top \mathbf{x} + \frac{\eta_k}{2}\|\mathbf{x} - \mathbf{x}_k\|^2\right\}. \\
\mathbf{z}_{k+1} &= \mathbf{z}_k + \gamma_k(\mathbf{x}_{k+1} - \mathbf{z}_k)
\end{aligned}$$

Instead of the above procedure, one can do

$$\mathbf{y}_k = (1-\gamma_k)\mathbf{z}_k + \gamma_k \mathbf{x}_k$$
$$\mathbf{x}_{k+1} = CG\left\{\nabla f(\mathbf{y}_k)^\top \mathbf{x} + \frac{\eta_k}{2}\|\mathbf{x}-\mathbf{x}_k\|^2; \delta_k\right\}.$$
$$\mathbf{z}_{k+1} = \mathbf{z}_k + \gamma_k(\mathbf{x}_{k+1} - \mathbf{z}_k)$$

Theorem 3. *Let $\mathcal{Q}$ be a compact convex set with diameter $D > 0$ and $f\colon \mathcal{X} \to \mathbb{R}$ be an L-smooth convex function in the Euclidean norm. Then CGS with $\gamma_k = 3/(k+3)$, $\eta_k = 3L/(k+2)$, and $\delta_k = LD^2/((k+1)(k+2))$ satisfies for all iterates $k \geq 1$,*

$$f(\mathbf{x}_t) - f(\mathbf{x}^*) \leq \frac{15LD^2}{2(k+1)(k+2)}.$$

The above result shows that the conditional gradient sliding method requires computing $\mathcal{O}\left(D\sqrt{\frac{L}{\epsilon}}\right)$ gradients of the objective function f and requires $\mathcal{O}\left(\frac{LD^2}{\epsilon}\right)$ calls to the linear oracle to find an ϵ accurate solution.