CS 726: Projection-Free (Frank-Wolfe) Methods for Constrained Convex Optimization

Jelena Diakonikolas

Fall 2022

In this lecture note, we discuss a class of methods (originally proposed by Marguerite Frank and Philip Wolfe in 1956 and still used today!) that can handle constrained optimization settings without using a projection operator. Instead, this method relies on a linear minimization oracle for the constraint set $\mathcal{X}$. This may seem like a strong requirement; however, there are many examples of convex sets where projections are expensive or even computationally intractable (e.g., because the set can only be described by exponentially many linear constraints/variables), but linear optimization can be done efficiently. Additionally, solutions produced by the Frank-Wolfe methods usually have additional desirable properties, such as sparse representation of the output solutions.

1 Setup

We begin by describing the setup we will be working with. Recall that our basic optimization problem is

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}). \tag{P}$$

For this lecture, we will additionally be assuming that:

- The norm of the space is Euclidean, $\|\cdot\| = \|\cdot\|_2$;
- f is L-smooth w.r.t. $\|\cdot\|_2$, convex, and minimized by some $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$;
- $\mathcal{X}$ is closed and convex. Additionally, for any $\mathbf{z} \in \mathbb{R}^d$,

  $$\min_{\mathbf{z}\in\mathcal{X}} \langle \mathbf{z}, \mathbf{x}\rangle$$

  is efficiently solvable. We further assume that $\mathcal{X}$ is bounded and denote by $D := \max_{\mathbf{x},\mathbf{y}\in\mathcal{X}} \|\mathbf{x}-\mathbf{y}\|_2$ its diameter.

Typical examples of feasible sets where linear optimization is "easier" than computing projections are polytopes: sets described by linear constraints. In particular, linear optimization over the probability simplex or the unit ℓ_1 ball can be done in $O(d)$ time, while projections take $O(d\log(d))$. While this is not a dramatic difference, the reason that Frank-Wolfe methods are sometimes employed for such sets is that they tend to produce sparse solution vectors (with few non-zero entries), which is usually a desirable property. We will see an example of this in Homework #5. Additionally, for some feasible sets linear optimization can be done efficiently (in polynomial time), whereas (exact) projections would be computationally intractable. A specific example is the perfect matching polytope for general graphs, which cannot be described by fewer than order $2^{|V|}$ constraints, where $|V|$ is the number of vertices in the graph, whereas linear optimization can be done in polynomial time by e.g., using Edmonds' algorithm from 1965.

2 Approximate Gap and the basic Frank-Wolfe Method

Basic Frank-Wolfe method can be summarized as follows. The method is initialized at a point $\mathbf{x}_0 \in \mathcal{X}$ and then for $k \geq 0$ performs the following updates:

$$\begin{aligned} \mathbf{v}_k &= \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} \langle \nabla f(\mathbf{x}_k), \mathbf{x}\rangle \\ \mathbf{x}_{k+1} &= \frac{A_{k-1}}{A_k}\mathbf{x}_k + \frac{a_k}{A_k}\mathbf{v}_k, \end{aligned} \tag{FW}$$

where $\{a_i\}_{i\geq 0}$ is a sequence of positive numbers and $A_k = \sum_{i=0}^k a_i$. There are different ways of choosing a_k (and, consequently, A_k), and in what follows we will see one specific choice. Observe that, as $\mathbf{v}_k \in \mathcal{X}$, by definition, it must be $\mathbf{x}_k \in \mathcal{X}$, $\forall k$. Hence, the points $\mathbf{x}_k$ produced and output by (FW) are in the feasible set $\mathcal{X}$.

It is possible to justify the updates of (FW) using the approximate gap analysis we saw in previous lectures. On the upper bound side, nothing too interesting happens: we choose $U_k = f(\mathbf{x}_{k+1})$, in the same way as we did before, and $\mathbf{x}_{k+1}$ is our output point.

More interesting things happen on the lower bound side. Recall that by convexity of f, we have

$$f(\mathbf{x}^*) \geq \frac{1}{A_k}\sum_{i=0}^k a_i\big(f(\mathbf{x}_i) + \langle \nabla f(\mathbf{x}_i), \mathbf{x}^* - \mathbf{x}_i\rangle\big).$$

Since we are assuming that the diameter D of $\mathcal{X}$ is bounded, in this case we still get a useful lower bound on $f(\mathbf{x}^*)$ if we replace $\mathbf{x}^*$ by the minimizer of the right-hand side of the last inequality. However, from the analysis perspective, it is not clear how to work with such a lower bound. Instead, what turns out to be more useful (or at least easier to work with) is relaxing the lower bound further, as follows:

$$\begin{aligned} f(\mathbf{x}^*) &\geq \min_{\mathbf{x}\in\mathcal{X}} \Big\{\frac{1}{A_k}\sum_{i=0}^k a_i\big(f(\mathbf{x}_i) + \langle \nabla f(\mathbf{x}_i), \mathbf{x} - \mathbf{x}_i\rangle\big)\Big\} \\ &\geq \frac{1}{A_k}\sum_{i=0}^k a_i f(\mathbf{x}_i) + \frac{1}{A_k}\sum_{i=0}^k a_i \min_{\mathbf{x}\in\mathcal{X}} \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}_i\rangle \\ &= \frac{1}{A_k}\sum_{i=0}^k a_i f(\mathbf{x}_i) + \frac{1}{A_k}\sum_{i=0}^k a_i \langle \nabla f(\mathbf{x}), \mathbf{v}_i - \mathbf{x}_i\rangle =: L_k. \end{aligned} \tag{1}$$

Recalling that $G_k = U_k - L_k$ and combining the definition of $U_k = f(\mathbf{x}_{k+1})$ with (1), we now have:

$$\begin{aligned} A_kG_k - A_{k-1}G_{k-1} &\leq A_k(f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)) - a_k\langle \nabla f(\mathbf{x}_k), \mathbf{v}_k - \mathbf{x}_k\rangle \\ &\leq A_k\langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \frac{A_kL}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 - a_k\langle \nabla f(\mathbf{x}_k), \mathbf{v}_k - \mathbf{x}_k\rangle, \end{aligned}$$

where the last inequality is by smoothness of f. The easiest way to get rid of the inner product terms in the last inequality is by setting $\mathbf{x}_{k+1} = \frac{A_{k-1}}{A_k}\mathbf{x}_k + \frac{a_k}{A_k}\mathbf{v}_k$, which is precisely what (FW) does. Doing so leaves us with

$$A_kG_k - A_{k-1}G_{k-1} \leq \frac{{a_k}^2L}{2A_k}\|\mathbf{v}_k - \mathbf{x}_k\|_2^2 \leq \frac{{a_k}^2L}{2A_k}D^2, \tag{2}$$

as, by assumption, the diameter of $\mathcal{X}$ is bounded by $D < \infty$. Using the same argument, it is not hard to show that also

$$A_0G_0 \leq \frac{{a_0}^2L}{2A_0}D^2. \tag{3}$$

Hence, combining (2) and (3), we have

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) \leq G_k \leq \frac{LD^2}{2A_k}\sum_{i=0}^k \frac{{a_i}^2}{A_i}.$$

To complete the analysis, it remains to choose the sequence $\{a_i\}_{i\geq 0}$. Different choices work here, but as I told you in class, whenever you see something like $\frac{{a_i}^2}{A_i}$, you should try $a_i \propto \frac{i+1}{2}$, as it ensures $\frac{{a_i}^2}{A_i} \approx 1$. In particular, if we take $a_i = \frac{i+1}{2}$, then $A_i = \frac{(i+1)(i+2)}{2}$ and $\frac{{a_i}^2}{A_i} \leq 1$, and thus we get

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) \leq \frac{2LD^2}{k+2}. \tag{4}$$

This bound is very similar to what we had from using projected gradient descent, but without the use of projections.

3 Can We Do Better?

It is natural to ask whether we can get a faster convergence rate for (some variant of) the Frank-Wolfe method, either using acceleration or assuming more about our objective function (e.g., that it is strongly convex). It turns out that the answer is negative, as long as we are only relying on linear minimization oracles to access information about $\mathcal{X}$. This is summarized in the following lemma.

Lemma 3.1. *Given an L-smooth function f and a closed convex set $\mathcal{X}$, any algorithm that accesses the feasible set $\mathcal{X}$ only via a linear minimization oracle requires at least*

$$\min\left\{\frac{d}{2}, \frac{LD^2}{16\epsilon}\right\}$$

iterations (calls to the linear minimization oracle) to construct a point $\hat{\mathbf{x}} \in \mathcal{X}$ such that $f(\hat{\mathbf{x}}) - \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \leq \epsilon$, for any $\epsilon > 0$. This lower bound applies even if f is strongly convex.

Proof. To prove the lemma, we only need to construct one example that would force any algorithm that learns about $\mathcal{X}$ only via a linear minimization oracle to make $\min\left\{d, \frac{LD^2}{\epsilon}\right\}$ queries to the oracle before it can guarantee that $f(\hat{\mathbf{x}}) - \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \leq \epsilon$, for any $\epsilon > 0$.

Consider $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$ and $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{x} \geq \mathbf{0}, \langle \mathbf{1}, \mathbf{x}\rangle = 1\}$ (that is, $\mathcal{X}$ is the probability simplex). It is not hard to see that f is both 1-smooth and 1-strongly convex and its unique minimizer is $\mathbf{x}^* = \frac{1}{d}\mathbf{1}$. The minimum function value is thus $f(\mathbf{x}^*) = \frac{1}{2d}$. The diameter of the probability simplex is $D = 2$.

Since we are assuming that the algorithm can only access $\mathcal{X}$ using a linear minimization oracle, we can assume that to construct an initial point, the algorithm queries the oracle. For any linear optimization problem over a bounded polytope, a solution lies at a vertex. Thus, we can assume that every query to the oracle returns a vertex of the simplex (which is one of the standard basis vectors). After k queries to the oracle, the algorithm has learned at most k different basis vectors. To guarantee that any constructed solution lies in the feasible set, the best the algorithm can do is construct convex combinations of these basis vectors. The lowest function value that can be obtained in this way is by assigning the same weight to all distinct basis vectors, thus for any point $\hat{\mathbf{x}}$ that the algorithm outputs it must be $f(\hat{\mathbf{x}}) \geq \frac{1}{2\min\{k,d\}}$ (the largest number of distinct standard basis vectors in $\mathbb{R}^d$ is d). Hence,

$$f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \geq \frac{1}{2\min\{k, d\}} - \frac{1}{2d}.$$

Consider $\epsilon > 0$. If $\frac{LD^2}{16\epsilon} = \frac{1}{4\epsilon} \geq \frac{d}{2}$, then for $k \leq \frac{LD^2}{8\epsilon}$ we have

$$f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \geq \frac{1}{4k} \geq \epsilon.$$

If $\frac{LD^2}{16\epsilon} = \frac{1}{4\epsilon} < \frac{d}{2}$, then for $k \leq \frac{d}{2}$, we have

$$f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \geq \frac{1}{2d} > \epsilon.$$

□

Exercises

1. Write the closed form expressions for solutions $\mathbf{v} = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{z}, \mathbf{x}\rangle$ to linear minimization problems for a given $\mathbf{z} \in \mathbb{R}^d$ and the following feasible sets:

1. Euclidean ball: $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 \leq 1\}$;
2. ℓ_1 ball: $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_1 \leq 1\}$;
3. Probability simplex: $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{x} \geq \mathbf{0}, \langle \mathbf{1}, \mathbf{x}\rangle = 1\}$;

4. Hyperrectangle: $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{a} \leq \mathbf{x} \leq \mathbf{b}\}$, where $\mathbf{a} \leq \mathbf{b}$ element-wise.

2. Let $f : \mathbb{R}^d \to \mathbb{R}$ be an L-smooth convex function. Let $\mathcal{X} \subset \mathbb{R}^d$ be a closed, convex, and bounded set (with diameter $D = \max_{\mathbf{x},\mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|$). Suppose that you are given a method with the following guarantee:

$$(\forall k \geq 0): \quad f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \leq -\frac{\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{v}_k \rangle^2}{2L\|\mathbf{v}_k - \mathbf{x}_k\|_2^2}, \tag{5}$$

where $\mathbf{v}_k = \operatorname{argmin}_{\mathbf{u} \in \mathcal{X}} \langle \nabla f(\mathbf{x}_k), \mathbf{u} - \mathbf{x}_k \rangle$. Prove that, for $k \geq 1$, the method converges as:

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

3. Prove that the Frank-Wolfe method (FW) with the step size a_k chosen using line search to minimize

$$\langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2$$

leads to (5), and thus (FW) with line search converges as $f(\mathbf{x}_k) - f(\mathbf{x}^*) = O\Big(\frac{LD^2}{k}\Big)$.