DSCC/CSC 435 & ECE 412 Optimization for Machine Learning — Lecture 8

Frank-Wolfe Method

Lecturer: Jiaming Liang — *October 16, 2025*

1 Frank-Wolfe method

Consider the problem $\min\{f(x) : x \in Q\}$ where f is convex and $Q \subseteq \operatorname{dom} f$ is convex and compact. We also assume f is differentiable over $\operatorname{dom} f$. One method can be employed is the projected gradient method

$$x_{k+1} = \operatorname{proj}_Q(x_k - t_k \nabla f(x_k)),$$

which is equivalent to

$$x_{k+1} = \operatorname{argmin}\left\{\ell_f(x; x_k) + \frac{1}{2t_k}\|x - x_k\|^2 : x \in Q\right\}.$$

In this lecture, we will present an alternative approach that does not require the projection operator proj_Q. The idea is to minimize the linearization of f (without the quadratic term) over Q

$$y_k = \operatorname{argmin}\left\{\ell_f(x; x_k) : x \in Q\right\} = \operatorname{argmin}\left\{\langle \nabla f(x_k), x\rangle : x \in Q\right\},$$

and then take a convex combination

$$x_{k+1} = x_k + t_k(y_k - x_k), \quad t_k \in [0, 1].$$

This algorithm is called Frank-Wolfe method, a.k.a., conditional gradient method.

Algorithm 1 Frank-Wolfe method

Input: Initial point $x_0 \in Q$
for $k \geq 0$ **do**
  Step 1. Compute $y_k = \operatorname{argmin}_{y \in Q} \langle y, \nabla f(x_k)\rangle$.
  Step 2. Choose $t_k \in [0, 1]$ and set $x_{k+1} = x_k + t_k(y_k - x_k)$.
end for

This is a projection-free method since we minimize a linear function over Q. In many case, linear optimization over Q is simpler than projection onto Q. For example, consider the following Q and associated linear optimization and projection:

- Unit simplex: linear optimization reduces to selecting the coordinate with the largest gradient component $\mathcal{O}(n)$, whereas projection requires sorting $\mathcal{O}(n \log n)$;

- ℓ_1-ball: linear optimization is $\mathcal{O}(n)$, while projection again involves sorting $\mathcal{O}(n \log n)$;
- Nuclear-norm ball: linear optimization requires computing only the largest singular value/vector, while projection requires a full SVD;
- Spectrahedron: linear optimization can be performed by a power method to find the leading eigenvector, while projection demands a full eigen-decomposition;
- Flow and matching polytopes: linear optimization reduces to standard combinatorial subproblems such as shortest path, maximum flow, or bipartite matching, for which highly efficient specialized solvers exist. In contrast, computing the projection onto Q typically lacks a closed-form solution and often requires solving a quadratic program using iterative methods, Lagrange multipliers, or cutting-plane techniques.

Frank-Wolfe method satisfies an even more important property: it produces sparse iterates. More precisely, consider the situation where $Q \subset \mathbb{R}^n$ is a polytope, that is the convex hull of a finite set of points (vertices). Then Carathéodory's theorem states that any point $x \in Q \subset \mathbb{R}^n$ can be written as a convex combination of at most $n+1$ vertices of Q. On the other hand, by step 2 of Frank-Wolfe, one knows that the k-th iterate x_k can be written as a convex combination of $k+1$ vertices (assuming that x_0 is a vertex). Thanks to the dimension-free rate of convergence, we are interested in the regime where $k \ll n$, and thus we see that the iterates of Frank-Wolfe are very sparse in their vertex representation.

Let us consider the general composite opimization problem.

$$\min\{\phi(x) := f(x) + h(x)\}. \tag{1}$$

- h is closed and convex and $\operatorname{dom} h$ is compact;
- f is closed and convex, $\operatorname{dom} h \subseteq \operatorname{dom} f$, and f is L-smooth over some set $\operatorname{dom} h$, i.e.,

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\|, \quad \forall x, y \in \operatorname{dom} h;$$

- the optimal set X_* is nonempty.

It is not difficult to deduce that the last condition is implied by the first two conditions.

The three properties of Frank-Wolfe method are projection-free (prox-free), norm-free, and sparse iterates.

In the rest of the lecture, we will consider the following generalized Frank-Wolfe method.

Algorithm 2 Generalized Frank-Wolfe method

Input: Initial point $x_0 \in \operatorname{dom} h$

for $k \geq 0$ **do**

Step 1. Compute $y_k = \operatorname{argmin}_{y \in \mathbb{R}^n} \{\langle y, \nabla f(x_k) \rangle + h(y)\}$.

Step 2. Choose $t_k \in [0, 1]$ and set $x_{k+1} = (1 - t_k)x_k + t_k y_k$.

end for

2 Convergence analysis

Definition 1. *The Wolfe gap is the function* $S(x) : \operatorname{dom} f \to \mathbb{R}$ *given by*

$$S(x) = \max_{y \in \mathbb{R}^n} \{\langle \nabla f(x), x - y \rangle + h(x) - h(y)\}.$$

Lemma 1. *The following statements hold:*

(a) $S(x) \geq 0$ *for any* $x \in \operatorname{dom} f$*;*

(b) $S(x_*) = 0$ *if and only if* $-\nabla f(x_*) \in \partial h(x_*)$*, that is, if and only if* x_* *is a stationary point of* (1)*.*

The above lemma gives the importance of the Wolfe gap $S(x)$, which can be (and is indeed) used to analyze the convergence of Frank-Wolfe for nonconvex optimization.

Lemma 2. *Let* $x \in \operatorname{dom} h$ *and* $t \in [0, 1]$*. Then, we have*

$$\phi((1 - t)x + ty) \leq \phi(x) - tS(x) + \frac{t^2 L}{2} \|y - x\|^2, \tag{2}$$

where $y = \operatorname{argmin}_{u \in \mathbb{R}^n} \{\langle u, \nabla f(x) \rangle + h(u)\}$*.*

Proof. Let $x^+ = (1 - t)x + ty$. Then, using the smoothness of f and the convexity of h, we easily show

$$\begin{aligned}
\phi(x^+) &= f(x^+) + h(x^+) \\
&\leq f(x) - t\langle \nabla f(x), x - y \rangle + \frac{t^2 L}{2} \|y - x\|^2 + h(x^+) \\
&\leq f(x) - t\langle \nabla f(x), x - y \rangle + \frac{t^2 L}{2} \|y - x\|^2 + (1 - t)h(x) + th(y) \\
&= \phi(x) - t\left[\langle \nabla f(x), x - y \rangle + h(x) - h(y)\right] + \frac{t^2 L}{2} \|y - x\|^2 \\
&= \phi(x) - tS(x) + \frac{t^2 L}{2} \|y - x\|^2.
\end{aligned}$$

□

Note that so far, we do not use the convexity of f yet.

Three stepsize rules

1) predefined diminishing stepsize:
$$\alpha_k = \frac{2}{k+2};$$

2) adaptive stepsize:
$$\beta_k = \min\left\{1, \frac{S(x_k)}{L\|y_k - x_k\|^2}\right\};$$

3) exact minimization/line search:
$$\eta_k \in \operatorname{argmin}_{t\in[0,1]} \phi\left((1-t)x_k + ty_k\right).$$

The intuition of the adaptive stepsize is β_k minimizes the right-hand side of (2) w.r.t. $t \in [0,1]$ when $x = x_k$. It is clear the exact minimization rule chooses $t_k = \eta_k$ to minimize the left-hand side of (2). The intuition of the first rule α_k is more involved and is given in Section 3.

The following lemma uses the convexity of f for the first time.

Lemma 3. *For any $x \in \operatorname{dom} f$, we have*
$$S(x) \geq \phi(x) - \phi_*.$$

Proof. Let $y = \operatorname{argmin}_{u\in\mathbb{R}^n}\{\langle u, \nabla f(x)\rangle + h(u)\}$. Then, we easily show
$$\begin{aligned}
S(x) &= \langle \nabla f(x), x - y\rangle + h(x) - h(y) \\
&= \langle \nabla f(x), x\rangle + h(x) - [\langle \nabla f(x), y\rangle + h(y)] \\
&\geq \langle \nabla f(x), x\rangle + h(x) - [\langle \nabla f(x), x_*\rangle + h(x_*)] \\
&= \langle \nabla f(x), x - x_*\rangle + h(x) - h(x_*) \\
&\geq f(x) - f(x_*) + h(x) - h(x_*) \\
&= \phi(x) - \phi_*.
\end{aligned}$$

□

Theorem 1. *The generalized Frank-Wolfe method with any of the three stepsize rules satisfies*
$$\phi(x_k) - \phi_* \leq \frac{2LD^2}{k} \tag{3}$$
where D is the diameter of $\operatorname{dom} h$.

Proof. Using Lemma 2 with $t = t_k$ and $x = x_k$, we have

$$\phi((1-t_k)x_k + t_k y_k) \le \phi(x_k) - t_k S(x_k) + \frac{t_k^2 L}{2}\|y_k - x_k\|^2.$$

1) If the predefined stepsize is used, i.e., $t_k = \alpha_k$, then

$$\phi((1-\alpha_k)x_k + \alpha_k y_k) \le \phi(x_k) - \alpha_k S(x_k) + \frac{\alpha_k^2 L}{2}\|y_k - x_k\|^2.$$

2) If the adaptive stepsize is used, i.e., $t_k = \beta_k$, then

$$\beta_k = \operatorname{argmin}_{t\in[0,1]} \left\{ -tS\left(x_k\right) + \frac{t^2 L}{2}\left\|y_k - x_k\right\|^2 \right\},$$

and hence

$$\begin{aligned}\phi((1-\beta_k)x_k + \beta_k y_k) &\le \phi(x_k) - \beta_k S(x_k) + \frac{\beta_k^2 L}{2}\|y_k - x_k\|^2 \\ &\le \phi(x_k) - \alpha_k S(x_k) + \frac{\alpha_k^2 L}{2}\|y_k - x_k\|^2.\end{aligned}$$

3) If the exact minimization/line search is used, i.e., $t_k = \eta_k$, then

$$\begin{aligned}\phi((1-\eta_k)x_k + \eta_k y_k) &\le \phi((1-\alpha_k)x_k + \alpha_k y_k) \\ &\le \phi(x_k) - \alpha_k S(x_k) + \frac{\alpha_k^2 L}{2}\|y_k - x_k\|^2.\end{aligned}$$

In any case, we have

$$\phi(x_{k+1}) \le \phi(x_k) - \alpha_k S(x_k) + \frac{\alpha_k^2 L}{2}\|y_k - x_k\|^2.$$

Using Lemma 3, we have

$$\phi(x_{k+1}) \le \phi(x_k) - \alpha_k[\phi(x_k) - \phi_*] + \frac{\alpha_k^2 L}{2}\|y_k - x_k\|^2.$$

$$\phi(x_{k+1}) - \phi_* \le (1-\alpha_k)[\phi(x_k) - \phi_*] + \frac{\alpha_k^2 L D^2}{2}.$$

We prove (3) by induction. It follows from the definition of α_k and the above inequality with $k = 0$ that $\alpha_0 = 1$ and

$$\phi(x_1) - \phi_* \le \frac{LD^2}{2}.$$

Thus, (3) holds with $k = 0$. Suppose (3) holds for some $k \geq 0$.

$$\begin{aligned}
\phi(x_{k+1}) - \phi_* &\leq (1 - \alpha_k)[\phi(x_k) - \phi_*] + \frac{\alpha_k^2 L D^2}{2} \\
&= \frac{k}{k+2}[\phi(x_k) - \phi_*] + \frac{2LD^2}{(k+2)^2} \\
&\leq \frac{k}{k+2}\frac{2LD^2}{k} + \frac{2LD^2}{(k+2)^2} \\
&= \frac{2(k+3)LD^2}{(k+2)^2} \leq \frac{2LD^2}{k+1}.
\end{aligned}$$

□

3 Frank-Wolfe as an ACG method without acceleration

In this section, we explore an alternative presentation of the Frank-Wolfe method from the perspective of the accelerated composite gradient (ACG) framework with the AT rule (see Lecture 7). We show that Frank-Wolfe is very close ACG except that we minimize a linear approximation instead of a quadratic approximation as in ACG. Hence, we only get $\mathcal{O}(1/k)$ convergence rate but not $\mathcal{O}(1/k^2)$ as in ACG.

Algorithm 3 Alternative presentation of Frank-Wolfe

Input: Initial point $x_0 \in \operatorname{dom} h$, set $A_0 = 0$

for $k \geq 0$ **do**

Step 1. Compute

$$a_k = \frac{1 + \sqrt{1 + 4LA_k}}{2L}, \quad A_{k+1} = A_k + a_k \tag{4}$$

Step 2. Compute

$$y_k = \operatorname*{argmin}_{u \in \mathbb{R}^n} \{\ell_f(u; x_k) + h(u)\} \tag{5}$$

and

$$x_{k+1} = \frac{A_k x_k + a_k y_k}{A_{k+1}}. \tag{6}$$

end for

Note that the sequences $\{a_k\}$ and $\{A_k\}$ are the same as those in Lecture 7 with $L_k = L$. Hence, Lemma 2 of Lecture 7 holds here. That is

$$A_{k+1} = La_k^2, \quad A_k \geq \frac{k^2}{4L}. \tag{7}$$

Theorem 2. *For every $k \geq 1$, we have*

$$\phi(x_k) - \phi_* \leq \frac{2LD^2}{k}.$$

Proof. Let $\gamma_k(\cdot) = \ell_f(\cdot; x_k) + h(\cdot)$. Using (5), (6), and (7), we have

$$\begin{aligned}
A_k\gamma_k(x_k) + a_k\gamma_k(u) + \frac{1}{2}\|y_k - x_k\|^2 &\geq A_k\gamma_k(x_k) + a_k\gamma_k(y_k) + \frac{1}{2}\|y_k - x_k\|^2 \\
&\geq A_{k+1}\gamma_k(x_{k+1}) + \frac{A_{k+1}L}{2}\|x_{k+1} - x_k\|^2 = A_{k+1}\left[\gamma_k(x_{k+1}) + \frac{L}{2}\|x_{k+1} - x_k\|^2\right] \\
&\geq A_{k+1}\phi(x_{k+1})
\end{aligned}$$

where the last inequality is due to the smoothness of f. Taking $u = x_*$ and using the fact that $\gamma_k \leq \phi$, we have

$$\begin{aligned}
A_{k+1}\phi(x_{k+1}) &\leq A_k\gamma_k(x_k) + a_k\gamma_k(x_*) + \frac{1}{2}\|y_k - x_k\|^2 \\
&\leq A_k\phi(x_k) + a_k\phi_* + \frac{1}{2}\|y_k - x_k\|^2.
\end{aligned}$$

Rearranging the terms and using the boundedness of $\operatorname{dom} h$, we have

$$A_{k+1}[\phi(x_{k+1}) - \phi_*] \leq A_k[\phi(x_k) - \phi_*] + \frac{D^2}{2}.$$

Finally, we have

$$A_k[\phi(x_k) - \phi_*] \leq A_0[\phi(x_0) - \phi_*] + \frac{kD^2}{2} = \frac{kD^2}{2},$$

which together with the bound on A_k implies that

$$\phi(x_k) - \phi_* \leq \frac{kD^2}{2A_k} \leq \frac{2LD^2}{k}.$$

$\square$

Indeed, the convergence rate of Frank-Wolfe can be improved to $\mathcal{O}(1/k^2)$ for $\min\{f(x) : x \in Q\}$ if we assume Q is a strongly convex set.

To conclude this section, we finally shed some light on the intuition of the predefined stepsize α_k from the perspective of ACG.

Lemma 4. *For every $k \geq 0$, let*

$$s_k = \frac{A_{k+1}}{a_k}.$$

Then, we have for every $k \geq 0$,

(a)

$$s_{k+1} = \frac{1 + \sqrt{1 + 4s_k^2}}{2};$$

(b) $s_0 = 1$ *and*

$$s_k \geq \frac{k+2}{2}.$$

Proof. (a) Recall that we have

$$A_{k+1} = A_k + a_k = La_k^2.$$

Hence, it follows

$$La_{k+1}^2 - a_{k+1} - A_{k+1} = 0$$

and

$$L\left(\frac{A_{k+2}}{La_{k+1}}\right)^2 - \frac{A_{k+2}}{La_{k+1}} - \frac{A_{k+1}^2}{La_k^2} = 0.$$

In terms of s_k, it reads

$$s_{k+1}^2 - s_{k+1} - s_k^2 = 0.$$

Therefore, the solution s_{k+1} satisfies statement (a).

(b) First, it follows from the definition that

$$s_0 = \frac{A_1}{a_0} = \frac{a_0}{a_0} = 1.$$

It easily follows from (a) that

$$s_{k+1} = \frac{1 + \sqrt{1 + 4s_k^2}}{2} \geq \frac{1 + 2s_k}{2},$$

and hence that

$$2s_{k+1} \geq 1 + 2s_k.$$

So we have

$$2s_k \geq k + 2s_0 = k + 2.$$

$\square$

Noting that $s_k^{-1} = a_k/A_{k+1}$ is the weight t_k used in Algorithm 1, hence the bound given by Lemma 4, i.e.,

$$s_k^{-1} \leq \frac{2}{k+2},$$

explains the choice of $t_k = \alpha_k$.

A final remark is that following from the above bound on s_k, we can derive slightly tighter bounds on A_k. Since

$$La_k = \frac{A_{k+1}}{a_k} = s_k \geq \frac{k+2}{2},$$

we have

$$A_{k+1} = La_k^2 = \frac{(La_k)^2}{L} \geq \frac{(k+2)^2}{4L}, \quad A_k \geq \frac{(k+1)^2}{4L} \geq \frac{k^2}{4L},$$

or

$$A_k = a_0 + \ldots + a_{k-1} \geq \frac{1}{2L}[2 + \ldots + (k+1)] = \frac{(k+3)k}{4L} \geq \frac{k^2}{4L}.$$