The Frank-Wolfe algorithm basics

Karl Stratos

1 Problem

A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be in differentiability class $C^k$ if the $k$-th derivative $f^{(k)}$ exists and is furthermore continuous. For $f \in C^k$, the value of $f(x)$ around $a \in \mathbb{R}^d$ is approximated by the $k$-th order Taylor series $F_{a,k} : \mathbb{R}^d \to \mathbb{R}$ defined as (using the “function-input” tensor notation for higher moments):

$$F_{a,k}(x) = f(a) + f'(a)(x - a) + \frac{1}{2} f''(a)(x-a,x-a) + \cdots$$

up to an additive error that vanishes as $x$ approaches $a$.

Let $D \subseteq \mathbb{R}^d$ be a compact convex set and $f \in C^1$ be a convex function. We consider a constrained convex optimization problem of the form:

$$x^* = \arg \min_{x \in D} f(x)$$  \hspace{1cm} (1)

2 Algorithm

A standard version of the Frank-Wolfe algorithm initializes some $x^{(0)} \in D$ and repeats for $t = 1, 2, \ldots$

1. Instead of (1), solve the following constrained linear optimization problem:

$$y_t = \arg \min_{y \in D} f(x^{(t-1)}) + f'(x^{(t-1)})(y - x^{(t-1)})$$

2. Choose the step size $\gamma_t = 2/(t + 1)$.

3. Update the estimate:

$$x^{(t)} = \gamma_t y_t + (1 - \gamma_t)x^{(t-1)}$$

Step 1 is often easy\(^1\) and yields sparse updates. Step 2 is deterministically given so that no tuning is needed.\(^2\) Step 3 always yields an estimate inside $D$ due to its convexity.

\(^1\)There are other variants of the Frank-Wolfe algorithm to handle cases where it's not.

\(^2\)Another variant of the algorithm performs the line search and finds

$$\gamma_t = \arg \min_{\gamma \in [0,1]} f(\gamma y_t + (1 - \gamma)x^{(t-1)})$$

which is also often given in a closed form solution.
3 Example (with line search)

Define \( f(x) := (1/2) \|b - Ax\|^2 \) for some \( b \in \mathbb{R}^m \) and \( A \in \mathbb{R}^{m \times d} \). Define \( D := \{ x \in \mathbb{R}^d : x \geq 0, \sum_i x_i = 1 \} \). Then we initialize \( x^{(0)} = 1/d \) and at each step \( t = 1, 2, \ldots \) compute:

\[
\begin{align*}
y_t &= e_i^* \text{ where } i^* = \arg\min_{i=1,\ldots,d} A^T (Ax^{(t-1)} - b)_i, \\
\gamma_t &= \min \left( 0, \max \left( 1, \frac{(Ax^{(t-1)} - Ae_{i^*})^T (Ax^{(t-1)} - b)}{\|Ax^{(t-1)} - Ae_{i^*}\|^2} \right) \right) \\
x^{(t)} &= \gamma_t y_t + (1 - \gamma_t)x^{(t-1)}
\end{align*}
\]

4 Duality gap

\( F_{a,1}(x) \) is linear and tangent with \( f(x) \) at \( a \) and, so the convexity of \( f \) implies that \( F_{a,1}(x) \leq f(x) \) for all \( x \in \mathbb{R}^d \). Thus

\[
\begin{align*}
f(x^{(t)}) + f'(x^{(t)})(y - x^{(t)}) &\leq f(y) \\
\min_{y \in D} f'(x^{(t)})(y - x^{(t)}) &\leq f(x^*) - f(x^{(t)}) \\
\max_{y \in D} f'(x^{(t)})(x^{(t)} - y) &\geq f(x^{(t)}) - f(x^*) \\
f'(x^{(t)})(x^{(t)} - y_{t+1}) &\geq f(x^{(t)}) - f(x^*)
\end{align*}
\]

The right-hand side

\[
h(x^{(t)}) := f(x^{(t)}) - f(x^*)
\]

is the (unknown) "true error" of \( x^{(t)} \). The left-hand side

\[
g(x^{(t)}) := f'(x^{(t)})(x^{(t)} - y_{t+1})
\]

is called the "duality gap" for a connection to Fenchel duality (which we won’t go into). Since \( h(x^{(t)}) \leq g(x^{(t)}) \) always and \( g(x^{(t)}) \) is given for free as part of the algorithm (Step 1), we can use the duality gap as a stopping criterion.

5 Convergence rate

To derive how fast the algorithm converges, we need to define a notion of non-linearity of \( f \). Let \( C_f \) be a constant such that for all \( x, a \in D \) and \( \gamma \in [0, 1] \),

\[
f((1 - \gamma)x + \gamma a) \leq f(x) + \gamma f'(x)(a - x) + \frac{\gamma^2}{2} C_f
\]

Intuitively, the more "curved" \( f \) is in \( D \), the larger \( C_f \) needs to be. With this constant, we first prove the following lemma:

Lemma 5.1. \( f(x^{(t)}) \leq f(x^{(t-1)}) + \gamma_t g(x^{(t-1)}) + \frac{\gamma^2}{2} C_f \) for \( t \geq 1 \).
Proof.

\[ f(x(t)) = f((1 - \gamma t)x(t - 1) + \gamma ty_t) \]
\[ \quad \leq f(x(t-1)) + \gamma tf'(x(t-1))(y_t - x(t-1)) + \frac{\gamma^2}{2} C_f \]
\[ = f(x(t-1)) - \gamma g(x(t-1)) + \frac{\gamma^2}{2} C_f \]

The following theorem states that the true error at step \( t \) is bounded above as \( O(1/t) \). So the algorithm has a linear convergence rate.

**Theorem 5.2** (Frank and Wolfe, 1956). \( h(x(t)) \leq \frac{2C_f}{t^2} \) for \( t \geq 1 \).

**Proof.** By Lemma 5.1,

\[ f(x(t)) \leq f(x(t-1)) - \gamma g(x(t-1)) + \frac{\gamma^2}{2} C_f \]
\[ f(x(t)) - f(x^*) \leq f(x(t-1)) - f(x^*) - \gamma g(x(t-1)) + \frac{\gamma^2}{2} C_f \]
\[ h(x(t)) \leq h(x(t-1)) - \gamma g(x(t-1)) + \frac{\gamma^2}{2} C_f \]
\[ \quad \leq h(x(t-1)) - \gamma h(x(t-1)) + \frac{\gamma^2}{2} C_f \]
\[ \quad \leq (1 - \gamma)h(x(t-1)) + \frac{\gamma^2}{2} C_f \]

When \( t = 1 \), using \( \gamma_1 = 2/(1 + 1) = 1 \) we have \( h(x(1)) \leq \frac{1}{2} C_f \leq \frac{2}{3} C_f \).

When \( t > 1 \), using \( \gamma_t = 2/(t + 1) \) we have

\[ h(x(t)) \leq \left(1 - \frac{2}{t + 1}\right) h(x(t-1)) + \frac{4C_f}{2(t + 1)^2} \]
\[ \leq \left(1 - \frac{2}{t + 1}\right) \frac{2C_f}{t + 1} + \frac{2C_f}{(t + 1)^2} \]
\[ = \frac{2C_f}{t + 1} \left(1 - \frac{1}{t + 1}\right) \]
\[ = \frac{2C_f}{t + 1} \left(\frac{t}{t + 1}\right) \]
\[ \leq \frac{2C_f}{t + 1} \left(\frac{t + 1}{t + 2}\right) = \frac{2C_f}{t + 2} \]

□