Course Notes for EE227C (Spring 2018): Convex Optimization and Approximation

Instructor: Moritz Hardt

Email: hardt+ee227c@berkeley.edu

Graduate Instructor: Max Simchowitz

Email: msimchow+ee227c@berkeley.edu

April 29, 2018

5 Conditional gradient method

In this lecture we discuss the conditional gradient method, also known as the Frank-Wolfe (FW) algorithm [FW56]. The motivation for this approach is that the projection step in projected gradient descent can be computationally inefficient in certain scenarios. The conditional gradient method provides an appealing alternative.

5.1 The algorithm

Conditional gradient side steps the projection step using a clever idea.

We start from some point $x_0 \in \Omega$. Then, for time steps $t = 1$ to T, where T is our final time step, we set

$$x_{t+1} = x_t + \eta_t(\bar{x}_t - x_t)$$

where

$$\bar{x}_t = \arg\min_{x \in \Omega} f(x_t) + \nabla f(x_t)^\top (x - x_t).$$

This expression simplifies to:

$$\bar{x}_t = \arg\min_{x \in \Omega} \nabla f(x_t)^\top x$$

Note that we need step size $\eta_t \in [0, 1]$ to guarantee $x_{t+1} \in \Omega$.

So, rather than taking a gradient step and projecting onto the constraint set. We optimize a liner function (defined by the gradient) inside the constraint set as summarized in Figure 1.

Starting from $x_0 \in \Omega$, repeat:

$$\bar{x}_t = \arg\min_{x \in \Omega} \nabla f(x_t)^\top x \qquad \text{(linear optimization)}$$

$$x_{t+1} = x_t + \eta_t(\bar{x}_t - x_t) \qquad \text{(update step)}$$

Figure 1: Conditional gradient

5.2 Conditional gradient convergence analysis

As it turns out, conditional gradient enjoys a convergence guarantee similar to the one we saw for projected gradient descent.

Theorem 5.1 (Convergence Analysis)**.** *Assume we have a function $f : \Omega \to \mathbb{R}$ that is convex, β-smooth and attains its global minimum at a point $x^* \in \Omega$. Then, Frank-Wolfe achieves*

$$f(x_t) - f(x^*) \leqslant \frac{2\beta D^2}{t+2}$$

with step size

$$\eta_t = \frac{2}{t+2}.$$

Here, D is the diameter of Ω, defined as $D = \max_{x-y \in \Omega} \|x - y\|$.

Note that we can trade our assumption of the existence of x^* for a dependence on L, the Lipschitz constant, in our bound.

Proof of Theorem 5.1. By smoothness and convexity, we have

$$f(y) \leqslant f(x) + \nabla f(x)^\top (x - x_t) + \frac{\beta}{2}\|x - y\|^2$$

Letting $y = x_{t+1}$ and $x = x_t$, combined with the progress rule of conditional gradient descent, the above equation yields:

$$f(x_{t+1}) \leqslant f(x_t) + \eta_t \nabla f(x_t)^\top (\bar{x}_t - x_t) + \frac{\eta_t^2 \beta}{2}\|\bar{x}_t - x_t\|^2$$

We now recall the definition of D from Theorem 5.1 and observe that $\|\bar{x}_t - x_t\|^2 \leqslant D^2$. Thus, we rewrite the inequality:

$$f(x_{t+1}) \leqslant f(x_t) + \eta_t \nabla f(x_t)^\top (x_t^* - x_t) + \frac{\eta_t^2 \beta D^2}{2}$$

Because of convexity, we also have that

$$\nabla f(x_t)^\top (x^* - x_t) \leqslant f(x^*) - f(x_t)$$

Thus,

$$f(x_{t+1}) - f(x^*) \leqslant (1-\eta_t)(f(x_t) - f(x^*)) + \frac{\eta_t^2 \beta D^2}{2} \tag{1}$$

We use induction in order to prove $f(x_t) - f(x^*) \leqslant \frac{2\beta D^2}{t+2}$ based on Equation 1 above.

Base case $t = 0$**.** Since $f(x_{t+1}) - f(x^*) \leqslant (1-\eta_t)(f(x_t) - f(x^*)) + \frac{\eta_t^2 \beta D^2}{2}$, when $t = 0$, we have $\eta_t = \frac{2}{0+2} = 1$. Hence,

$$\begin{aligned}
f(x_1) - f(x^*) &\leqslant (1-\eta_t)(f(x_t) - f(x^*)) + \frac{\beta}{2}\|x_1 - x^*\|^2 \\
&= (1-1)(f(x_t) - f(x^*)) + \frac{\beta}{2}\|x_1 - x^*\|^2 \\
&\leqslant \frac{\beta D^2}{2} \\
&\leqslant \frac{2\beta D^2}{3}
\end{aligned}$$

Thus, the induction hypothesis holds for our base case.

Inductive step. Proceeding by induction, we assume that $f(x_t) - f(x^*) \leqslant \frac{2\beta D^2}{t+2}$ holds for all integers up to t and we show the claim for $t+1$.

By Equation 1,

$$\begin{aligned}
f(x_{t+1}) - f(x^*) &\leqslant \left(1 - \frac{2}{t+2}\right)(f(x_t) - f(x^*)) + \frac{4}{2(t+2)}\beta D^2 \\
&\leqslant \left(1 - \frac{2}{t+2}\right)\frac{2\beta D^2}{t+2} + \frac{4}{2(t+2)}\beta D^2 \\
&= \beta D^2 \left(\frac{2t}{(t+2)^2} + \frac{2}{(t+2)^2}\right) \\
&= 2\beta D^2 \cdot \frac{t+1}{(t+2)^2} \\
&= 2\beta D^2 \cdot \frac{t+1}{t+2} \cdot \frac{1}{t+2} \\
&\leqslant 2\beta D^2 \cdot \frac{t+2}{t+3} \cdot \frac{1}{t+2} \\
&= 2\beta D^2 \frac{1}{t+3}
\end{aligned}$$

Thus, the inequality also holds for the $t+1$ case.

■

5.3 Application to nuclear norm optimization problems

The code for the following examples can be found here.

5.3.1 Nuclear norm projection

The *nuclear norm* (sometimes called *Schatten* 1*-norm* or *trace norm*) of a matrix A, denoted $\|A\|_*$, is defined as the sum of its singular values

$$\|A\|_* = \sum_i \sigma_i(A)\,.$$

The norm can be computed from the singular value decomposition of A. We denote the unit ball of the nuclear norm by

$$B_*^{m\times n} = \{A \in \mathbb{R}^{m\times n} \mid \|A\|_* \leqslant 1\}.$$

How can we project a matrix A onto B_*? Formally, we want to solve

$$\min_{X\in B_*} \|A - X\|_F^2$$

Due to the rotational invariance of the Frobenius norm, the solution is obtained by projecting the singular values onto the unit simplex. This operation corresponds to shifting all singular values by the same parameter θ and clipping values at 0 so that the sum of the shifted and clipped values is equal to 1. This algorithm can be found in [DSSSC08].

5.3.2 Low-rank matrix completion

Suppose we have a partially observable matrix Y, of which the missing entries are filled with 0 and we would like to find its completion form projected on a nuclear norm ball. Formally we have the objective function

$$\min_{X\in B_*} \frac{1}{2}\|Y - P_O(X)\|_F^2$$

where P_O is a linear projection onto a subset of coordinates of X specified by O. In this example $P_O(X)$ will generate a matrix with corresponding observable entries as in Y while other entries being 0. We can have $P_O(X) = X \odot O$ where O is a matrix with binary entries. Calculating the gradient of this function, we have

$$\nabla f(X) = Y - X \odot O\,.$$

We can use projected gradient descent to solve this problem but it is more efficient to use Frank-Wolfe algorithm. We need to solve the linear optimization oracle

$$\bar{X}_t \in \operatorname*{argmin}_{X\in B_*} \nabla f(X_t)^\top X$$

To simplify this problem, we need a simple fact that follows from the singular value decomposition.

Fact 5.2. *The unit ball of the nuclear norm is the convex hull of rank-1 matrices*

$$\operatorname{conv}\{uv^\top \,|\, \|u\| = \|v\| = 1, u \in \mathbb{R}^m, v \in \mathbb{R}^n\} = \{X \in \mathbb{R}^{m\times n} \mid \|X\|_* = 1\}\,.$$

From this fact it follows that the minimum of $\nabla f(X_t)^\top X$ is attained at a rank-1 matrix $uv^\top$ for unit vectors u and v. Equivalently, we can maximize $-\nabla f(X_t)^\top uv^\top$ over all unit vectors u and v. Put $Z = -\nabla f(X_T)$ and note that

$$Z^\top uv^\top = \operatorname{tr}(Z^\top uv^\top) = \operatorname{tr}(u^\top Zv) = u^\top Zv\,.$$

Another way to see this is to note that the dual norm of a nuclear norm is operator norm,

$$\|Z\| = \max_{\|X\|_* \leqslant 1} \langle Z, X\rangle\,.$$

Either way, we see that to run Frank-Wolfe over the nuclear norm ball we only need a way to compute the top left and singular vectors of a matrix. One way of doing this is using the classical power method described in Figure 2.

- Pick a random unit vector x_1 and let $y_1 = A^\top x / \|A^\top x\|$.
- From $k = 1$ to $k = T - 1$:
  - Put $x_{k+1} = \frac{Ay_k}{\|Ay_k\|}$
  - Put $y_{k+1} = \frac{A^\top x_{k+1}}{\|A^\top x_{k+1}\|}$
- Return x_T and y_T as approximate top left and right singular vectors.

Figure 2: Power method

References

[DSSSC08] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proc.* 25*th ICML*, pages 272–279. ACM, 2008.

[FW56] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.