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

February 1, 2018

5 Lecture 5: Conditional Gradient (Frank-Wolfe)

In this lecture we discussed about the conditional gradient method, also known as the Frank-Wolfe (FW) algorithm [FW56]. The motivation of using this approach is the projected gradient descent can be computationally inefficient under certain scenarios.

5.1 Intuition behind Frank-Wolfe algorithm

We motivate this lecture by the computational inefficiency that projected gradient can have. With conditional gradient, we are able to sidestep some of these inefficiencies. An intuitive idea of the FW algorithm is as follows.

We start from x_0. 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

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

Since we hope to minimize with respect to $x \in \Omega$, we can simplify the equation.

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

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

5.2 Theorem: conditional gradient convergence analysis

Theorem 5.1 (Convergence Analysis). *Assume we have a function $f \colon \Omega \to \mathbb{R}$ that is convex and β-smooth. Then, Frank-Wolfe achieves*

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

with step size

$$\eta_t = \frac{2}{t+2}, (t \geqslant 0)$$

where D is the diameter of Ω, defined as $D = \max_{x-y\in\Omega} \|x-y\|$, x^* is defined as $x^* = \arg\min_{x\in\Omega} f(x)$

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.
First step: 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, $\eta_t = \frac{2}{0+2} = 1, then$

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

Thus, the induction hypothesis holds for our base case.

Here we need to note that we cannot directly use $f(x_{t+1}) - f(x^*) \leqslant (1-\eta_t)(f(x_t) - f(x^*)) + \frac{\eta_t^2 \beta D^2}{2}$ directly to derivate the base case out. Because $(1-\eta_t)(f(x_t) - f(x^*))$ is greater than 0 and cannot be directly proved to be bounded.

Second step: We assume that $f(x_t) - f(x^*) \leqslant \frac{2\beta D^2}{t+2}$ holds $\forall t$ and can show that the induction hyposthesis holds in general.

General case: given $f(x_t)$ holds for our destinated inequality, we'd like to show it also holds for $f(x_{t+1})$. By using Equation 1,

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

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

■

5.3 Examples

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. Calculate the gradient of this function we will 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

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

This will lead to a rank-1 matrix which is decomposed from $-\nabla f(X_t)$. This can be derived from Lemma 5.2. Now we can have the update rule for the conditional gradient as

$$X_{t+1} = X_t + \eta_t(-u_1 v_1^\top - X_t)$$

where u_1 and v_1 are the top left and right singular vectors.

Lemma 5.2.

$$\min_{X\in B_*}\langle \nabla f(X_t)^\top, X\rangle, \quad B_* = \{X | \|X\|_* \leqslant 1\}$$

The optimal result is

$$-\|\nabla f(X_t)\|$$

with $X = -u_1 v_1^\top$ *being one of the* X *that minimize the function where* u_1 *and* v_1 *are left and right singular vectors corresponding to the max singular value,* $\|\nabla f(X_t)\| = \max_{\|X\|_*\leqslant 1}\langle \nabla f(X_t), X\rangle$.

Proof of Lemma 5.2.

Fact 5.3. *The unit ball of the nuclear norm is the convex hull of rank-1 matrices*

$$\mathrm{Conv}\{uv^\top | \|uv^\top\| \leqslant 1, u \in \mathbb{R}^m, v \in \mathbb{R}^n\} = \{X \in \mathbb{R}^{m\times n} | \sum_i \sigma(X)_i \leqslant 1\}$$

From Fact 5.3 we can tell that the minimum is attained at a vertex, corresponding to a rank-1 matrix. And by Cauchy-Schwarz, we have

$$\langle \nabla f(X_t)^\top, X\rangle \geqslant -\|\nabla f(X_t)\|$$

The optimizer X^* is a rank-1 matrix which approximates $-\nabla f(X_t)$ and we can set its corresponding singular value to 1 so it is projected on the unit ball of the nuclear norm.

Here we provide another approach to prove Lemma 5.2. Since the dual norm of a nuclear norm is operator norm,

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

we can easily see that the optimal result of the objective function in Lemma 5.2 being

$$-\|Y\| = -\nabla f(X_t) = -\sigma_{\max}.$$

Then we show by Singular value decomposition to prove that $X = -u_1 v_1^\top$ is one of the solutions that minimize the function. From SVD, $\nabla f(X_t) = U\Sigma V^\top$, then $\Sigma = U^\top Y V$, we set $X = -u_1 v_1^\top$, thus

$$\langle \nabla f(X_t), X\rangle = \nabla f(X_t)^\top X = -V\Sigma U^\top u_1 v_1^\top = -\sigma_1$$

■

According to the lemma, we can see that only the leading singular value and corresponding vectors are used to get the optimizer. Thus we can do the rank-1 approximation of this function which makes no difference to final result. Power method works well in approximate the leading singular value and corresponding left and right vectors. Here is the introduction of the power method for solving the leading singular vectors.

- Randomly take a vector x_1', $x_1 = \frac{x_1'}{\|x_1'|}$, $y_1' = A^\top x$, $y_1 = \frac{y_1'}{\|y_1'\|}$;
- when $k \geqslant 1$, do the iterate process until k=number steps: $x_{k+1} = \frac{Ay_k}{\|Ay_k\|}$, $y_{k+1} = \frac{A^\top x_{k+1}}{\|A^\top x_{k+1}\|}$;
- return x_k and y_k as left and right singular vectors corresponding to the leading singular value, we can also get the approximation of σ_1 through $\sqrt{\frac{y_k}{y_{k+1}}}$;

The basic idea behind is to use singular value. The proof of convergence can be found in many papers and sources.

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 *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 272–279, New York, New York, USA, 2008. ACM Press.

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