Columbia University in the City of New York
Optimization Methods for Machine Learning
Instructors: Satyen Kale
Authors: Chao Qin
Email: cq2199@columbia.edu

SCRIBE

16

Wrap-up Analysis of SVRG and Frank-Wolfe Algorithm

In this lecture, we first wrap up the analysis of SVRG. Then we introduce the Frank-Wolfe algorithm.

1. Wrap up Analysis of SVRG

In the last lecture, we arrived at

$$2\eta\mathbb{E}\left[\sum_{t=0}^{T-1} f(\mathbf{x}_{t+1}) - f(\mathbf{x}^*)\right]$$
$$\leq (1-\eta\alpha)\left\| \mathbf{x}_0 - \mathbf{x}^* \right\|^2 + \sum_{t=0}^{T-1} 4\beta\eta^2 \left\{\mathbb{E}\left[f(\mathbf{x}_t) - f(\mathbf{x}^*) \right] + (f(\mathbf{x}_0) - f(\mathbf{x}^*))\right\}$$

where we fix epoch k (i.e. condition on $\mathbf{x}_0$). This leads to

$$\begin{aligned} 2\eta(1-2\beta\eta)\mathbb{E}\left[\sum_{t=0}^{T-1} f(\mathbf{x}_{t+1}) - f(\mathbf{x}^*)\right] &\leq (1-\eta\alpha)\left\| \mathbf{x}_0 - \mathbf{x}^* \right\|^2 + (T+1)4\beta\eta^2[f(\mathbf{x}_0) - f(\mathbf{x}^*)] \\ &\leq \left[\frac{2(1-\eta\alpha)}{\alpha} + (T+1)4\beta\eta^2\right](f(\mathbf{x}_0) - f(\mathbf{x}^*)) \end{aligned}$$

where the last inequality follows from $f(\mathbf{x}_0) - f(\mathbf{x}^*) \geq \frac{\alpha}{2}\left\| \mathbf{x}_0 - \mathbf{x}^* \right\|^2$ by α-strong convexity. On the other hand, by Jensen's inequality,

$$\text{LHS} \geq T2\eta(1-2\beta\eta)\mathbb{E}\left[f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t\right) - f(\mathbf{x}^*)\right],$$

Using the fact that $\mathbf{x}_0^{(k+1)} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t$ and $\mathbf{x}_0^{(k)} = \mathbf{x}_0$, and taking expectation to remove the conditioning on $\mathbf{x}_0$, we have

$$\mathbb{E}\left[f\left(\mathbf{x}_0^{(k+1)}\right) - f(\mathbf{x}^*)\right] \leq \left[\frac{2(1-\eta\alpha)}{\alpha T2\eta(1-2\beta\eta)} + \frac{(T+1)4\beta\eta^2}{T2\eta(1-2\beta\eta)}\right]\mathbb{E}\left[f\left(\mathbf{x}_0^{(k)}\right) - f(\mathbf{x}^*)\right]$$

By choosing $\eta = \frac{1}{\beta}$ and $T = \frac{40\beta}{\alpha}$, we obtain

$$\mathbb{E}\left[f\left(\mathbf{x}_0^{(k+1)}\right) - f(\mathbf{x}^*)\right] \leq \frac{14}{15}\mathbb{E}\left[f\left(\mathbf{x}_0^{(k)}\right) - f(\mathbf{x}^*)\right].$$

This leads to

$$\mathbb{E}\left[f\left(\mathbf{x}_0^{(K)}\right) - f(\mathbf{x}^*)\right] \leq \left(\frac{14}{15}\right)^K\left(f\left(\mathbf{x}_0^{(0)}\right) - f(\mathbf{x}^*)\right).$$

Then we have

$$\mathbb{E}\left[f\left(\mathbf{x}_0^{(K)}\right) - f(\mathbf{x}^*) \right] \leq \epsilon$$

if we set

$$K = \frac{\log\left(\frac{f(\mathbf{x}_0)-f(\mathbf{x}^*)}{\epsilon}\right)}{\log\left(\frac{15}{14}\right)}.$$

Notice that $f(\mathbf{x}^*)$ is unknown in general. We can replace $f(\mathbf{x}^*)$ with a known lower bound of it in the above equation. The first-order complexity of SVRG is

$$Kn + KT = O\left(\left(n + \frac{\beta}{\alpha}\right)\log\left(\frac{1}{\epsilon}\right)\right).$$

since $T = O\left(\frac{\beta}{\alpha}\right)$.

2. Frank-Wolfe Algorithm

Recall the following convex optimization problem

$$\min f(\mathbf{x})$$
$$\text{s.t. } \mathbf{x} \in K$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is convex and K is a convex set of $\mathbb{R}^d$. We have learned how to use projected gradient descent to solve this problem. Although calculating the projection onto l_2 ball or l_∞ ball is easy due to the closed-form solution, it is in general computationally hard. Instead, we can replace the projection with the so-called linear optimization (LP) oracle: given $\mathbf{v} \in \mathbb{R}^d$, finds $\text{argmax}_{\mathbf{x}\in K}\mathbf{v}^\top\mathbf{x}$. Generally, the LP oracle is computationally easier to implement than a projection oracle, expecially when K is a polytope. Based on LP oracle, we can propose the Frank-Wolfe algorithm, which is also called conditional gradient method.

Algorithm 1: Frank-Wolfe Algorithm/Conditional Gradient Method

1. Start with arbitrary $\mathbf{x}_0 \in K$
2. For $t = 0, 1, \ldots, T-1\ldots$
   (a) Compute $\mathbf{v}_t = \text{argmax}_{\mathbf{x}\in K} - \nabla f(\mathbf{x}_t)^\top \mathbf{x}$
   (b) $\mathbf{x}_{t+1} = \frac{2}{t+2}\mathbf{v}_t + \left(1 - \frac{2}{t+2}\right)\mathbf{x}_t$
3. Output $\mathbf{x}_T$