10-725/36-725: Convex Optimization **Spring 2015**

Lecture 25: April 20th

Lecturer: Aaditya Ramdas/Ryan Tibshirani *Scribes: Fan Yang*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

25.1 Introduction

Consider a problem that minimize a finite sum:

$$\min_w F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w) + \frac{\lambda}{2}\|w\|^2 \tag{25.1}$$

Assume f_i's are differentiable and convex. The objective is strongly convex because of the regularizer $\frac{\lambda}{2}\|w\|^2$. Further assume that ∇f is L-lipschitz.

Question: What are some existing methods that solve this problem?

Answer: gradient descent, newton's method, stochastic gradient descent, etc.

25.2 (Stochastic) Gradient Descent (GD/SGD)

Gradient descent will require a lot computation if N is huge. Stochastic gradient descent is an alternative if N is huge. However, SGD has slower convergence rate. Though at each iteration, the SGD uses an unbiased estimator of the gradient ∇F, it has extremely high variance. A variation based on this observation is to use mini-batch SGD.

In the strongly convex case, SGD has a function error suboptimality of $O(1/T)$, and of $O(1/\sqrt{T})$ in the convex setting, where T is the time budget.

There are lower bounds that prove, in black box setting for convex optimization, the previously mentioned rates for minimizing F is essentially optimal in T. So, are we done because we have matched the lower bounds? Is it possible to get the best of both gradient descent and SGD? I.e. good convergence rate and efficient computation?

25.3 The suboptimality of SGD for minimizing finite sums

A paper published in 2012 proposed an algorithm SAG that beats the black box stochastic gradient methods. It has $O(1/T)$ rates for convex functions, and $O(\rho^T)$ rates for strongly convex functions for some $\rho < 1$.

The key insights that help them get around the black box lower bounds are the following:

1. The objective function we are minimizing is a **finite** sum. Since we know more about the objective function, we can have better bounds than the black box analysis.

2. We are restricted in choosing the stochastic gradient as $\nabla f_i(w_i)$, we can choose any g_i as long as $\mathbb{E}g_t = \nabla F(w_t)$.

3. Also, we can have unbiased gradients, as long as the biases are tolerable in some sense. In fact, the SAG updates are biased.

There is a trade-off between bias and variance when choosing the stochastic gradient. We are more interested in reduce the variance here, since it's the variance that hurts the convergence rates!

25.4 SDCA, SVRG, SAG, SAGA, etc

SDCA: stochastic dual coordinate ascent
SVRG: stochastic variance-reduced gradient
SAG: stochastic average gradient
SAGA: stochastic averaged gradient (another version)

The above are the most relevant work in stochastic gradient descent. The paper can be easily found using Google. The SAGA paper has a comprehensive related work section that puts things together.

We will focus on an example using SDCA algorithm. The objective function is

$$\min_w F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w) + \frac{\lambda}{2}\|w\|^2 \tag{25.2}$$

where f_i is convex with L-smooth gradients. Note that at optimality,

$$w^* = -\frac{1}{\lambda n}\sum_{i=1}^{n} \nabla f_i(w^*) \tag{25.3}$$

and what we will do is maintain "dual vectors" $\alpha_1, \ldots, \alpha_n$ such that

$$w^{(t)} = \frac{1}{\lambda n}\sum_{i=1}^{n} \alpha_i^{(t)} \tag{25.4}$$

The SDCA algorithm will make $w^{(t)} \to w^*$ and simultaneously $\alpha_i^{(t)} \to -\nabla f_i(w^*) =: \alpha_i^*$.

We initialize there dual vectors arbitrarily, and then at each time step, we uniform randomly pick an index from 1 to n, and perform

$$\alpha_i^{(t)} = \alpha_i^{(t-1)} - \eta\lambda n(\nabla f_i(w^{(t-1)}) + \alpha_i^{(t-1)}) \tag{25.5}$$

where the step size $\eta < 1/\lambda n$.

And correspondingly,

$$w^{(t)} = w^{(t-1)} - \eta(\nabla f_i(w^{(t-1)}) + \alpha_i^{(t-1)}) := w^{(t-1)} - \eta g_t \tag{25.6}$$

The algorithm is simple and straightforward.

We will check that SDCA has unbiased graident:

$$\mathbb{E}g_t = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(w^{(t-1)}) + \frac{1}{n}\sum_{i=1}^{n}\alpha_i^{(t-1)} = \nabla F(w^{(t-1)})$$

This is because we always maintain $w^{(t-1)} = \frac{1}{\lambda n}\sum_{i=1}^{n}\alpha_i^{t-1}$.

We've checked that SDCA is unbiased, does it also have lower variance than SGD? It's provable that the variance goes to zero.

Proof:

$$\begin{aligned}\mathbb{E}\|g_t\|^2 &= \mathbb{E}\|\nabla f_i(w^{(t-1)}) + \alpha_i^* - \alpha_i^* + \alpha_i^{(t-1)}\|^2 \\ &\leq 2\mathbb{E}\|\nabla f_i(w^{(t-1)}) + \alpha_i^*\|^2 + 2\mathbb{E}\| - \alpha_i^* + \alpha_i^{(t-1)}\|^2\end{aligned}$$

Since $\alpha_i^* = -\nabla f_i(w^*)$, the first term $\|\nabla f_i(w^{(t-1)}) + \alpha_i^*\|^2 \leq L^{\|}w^{(t-1)} - w^*\|^2$. There is a theorem that implies that both the first term and second term go to zero. We only cite this technical theorem here.

■

Hence, the SDCA eventually behaves like GD, but starts like SGD.

25.5 Optimality of various methods

SDCA has much better convergence rate than naive stochastic algorithm and vanilla gradient descent. The total complexity of SDCA is $(\frac{L}{\lambda} + n)d\log(1/\epsilon)$, since each iteration takes time d. This is better than the $n\frac{L}{\lambda}d\log(1/\epsilon)$ complexity of gradient descent.

Though this convergence rate is great, it is not provably optimal. There are many possibilities in this research area:

- accelerated and proximal extension of the algorithm
- these algorithms may not be optimal in all ranges of the condition number
- maybe these algorithms are indeed optimal
- one may be able to prove tighter lower bounds