CS 596: Theoretical Machine Learning

Lecturer: Pranjal Awasthi
Scribe: Charles Shvartsman

Lecture #21
December 1, 2016

In this lecture we will look at stochastic optimization. So far we have studied minimizing convex functions by gradient descent where we have the following guarantees

- If f is convex and G-bounded ($||\nabla g(x)|| \leq G$), $\frac{RG}{\sqrt{T}}$ is the rate of convergence with $R = ||x_0 - x||$.
- If f is convex and β-Lipschitz ($||\nabla f(x) - \nabla f(y)|| \leq \beta||x - y||$) or, in other words, smooth, $\frac{\beta R^2}{T}$ is the rate, or $\frac{\beta R^2}{T^2}$ by accelerated gradient descent.

In the above theorems we made two crucial assumptions which are often not true in practice:

- The first assumption is that we have access to exact gradients, meaning there is no noise in the process of obtaining these gradients.
- The second assumption is that computing gradients has no cost. (Before we only worried about the number of iterations of gradient descent without worrying about the cost of computing the gradient).

Our goal is to address these two assumptions to obtain a more practical version of gradient descent. Looking at the first assumption, we must ask, "In what cases do we not have exact gradients?" One case is where we don't know what to optimize, for example bandit optimization. Looking at the second assumption, notice that often, the function to minimize is the sum of the many functions. This is especially true in machine learning settings. For example, $f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$ where m is large. Running gradient descent in this case corresponds to $w_{t+1} \leftarrow w_t - h\nabla f(w_t)$, where $\nabla f(w_t) = \frac{1}{m}\sum_{i=1}^{m} f_i(w_t)$. Thus, we would need to compute gradients of all the smaller functions. This will cost a lot - on the order of m operations. Therefore, in practice, ignoring the cost of computing the gradient is a bad idea. How might we deal with this? Instead of computing the true gradient, we will compute noisy gradients. This leads us to our main topic: Stochastic Gradient Descent.

1 Stochastic Gradient Descent

In this setting, when a data point x_t is queried to an oracle, the output is $\hat{g}(x_t)$, the noisy gradient:

$$\hat{\nabla} f(x_t) = \nabla f(x_t) + e_t$$

. We will assume that the noisy gradient is an unbiased estimator of the true gradient, i.e., $E[e_t|x_t] = 0$. For machine learning problems computing the noisy gradient is a much cheaper operation. For instance if $f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$, we can choose i at random and evaluate the gradient of that particular function:

input: x_t
output: Choose $i \in [m]$ uniformly at random. Output $\hat{\nabla} f(x_t) = \nabla f_i(x_t)$

Note that this is an unbiased estimate because $E[\hat{\nabla} f(x_t)] = \frac{1}{m}\sum_{i=1}^{m} \nabla f_i(x_t)$. We will prove the following two guarantees for such a method.

- If f is convex and G-bounded, the rate is $E[\frac{1}{T}\sum_{t=1}^{T} f(x_t) - f(x^*)] \leq \frac{R(\sigma+G)}{\sqrt{T}}$ where σ is the variance of the noise. $\sigma^2 = \max_x[||\hat{\nabla}f(x) - \nabla f(x)||^2]$
- If f is convex and β-Lipschitz, $E[\frac{1}{T}\sum_{t=1}^{T} f(x_t) - f(x^*)] \leq \frac{\beta R^2}{T} + \frac{\sigma R}{\sqrt{T}}$. In stochastic gradient descent, we lose the speed up obtained by gradient descent for Lipschitz functions. Intuitively we lose the speed up because as we get closer to the optimum in the algorithm, the gradient gets smaller, but the noise stays the same. Thus, in the noisy setting we can't take large steps otherwise the variance will become problematic.

Theorem 1. *If f is convex and G-bounded then after T steps of stochastic gradient descent with $\eta = \frac{R(\sigma+G)}{\sqrt{T}}$ we have, $E[\frac{1}{T}\sum_{t=0}^{T-1} f(x_t) - f(x^*)] \leq \frac{R(\sigma+G)}{\sqrt{T}}$ where $\sigma^2 = \max_x[||\hat{\nabla}f(x) - \nabla f(x)||^2]$ and $R = \|x_0 - x^*\|$.*

Proof. By convexity, $f(x^*) \geq f(x_t) + \nabla f(x_t)(x^* - x_t)$. This implies

$$f(x_t) - f(x^*) \leq \nabla f(x_t)(x_t - x^*) = \hat{\nabla} f(x_t)(x_t - x^*) + (\nabla f(x_t) - \hat{\nabla} f(x_t))(x_t - x^*)$$
$$= \frac{1}{\eta}(x_t - x_{t+1})(x_t - x^*) + (\nabla f(x_t) - \hat{\nabla} f(x_t))(x_t - x^*)$$

Using the fact that $(a-b)\cdot(a-c) = \frac{1}{2}[||a-b||^2 + ||a-c||^2 - ||b-c||^2]$ we get that

$$f(x_t) - f(x^*) \leq \frac{1}{2\eta}[||x_t - x^*||^2 + ||x_t - x_{t+1}||^2 - ||x_{t+1} - x^*||^2] + (\nabla f(x_t) - \hat{\nabla} f(x_t))(x_t - x^*)$$

$$= \frac{1}{2\eta}[||x_t - x^*||^2 - ||x_{t+1} - x^*||^2] + \frac{1}{2\eta}||x_t - x_{t+1}||^2 + (\nabla f(x_t) - \hat{\nabla} f(x_t))(x_t - x^*)$$

Since $x_{t+1} = x_t - \eta\hat{\nabla} f(x_t)$ we get that

$$f(x_t) - f(x^*) \leq \frac{1}{2\eta}[||x_t - x^*||^2 - ||x_{t+1} - x^*||^2] + \frac{\eta}{2}||\hat{\nabla} f(x_t)||^2 + (\nabla f(x_t) - \hat{\nabla} f(x_t))(x_t - x^*)$$

Now take expectation on both sides. The last term will have an expected value of 0[1].

$$E[f(x_t) - f(x^*)] \leq \frac{1}{2\eta}[||x_t - x^*||^2 - ||x_{t+1} - x^*||^2] + \frac{\eta}{2}E[||\hat{\nabla} f(x_t)||^2] + 0$$

Using $E[y^2] = E[y]^2 + \sigma^2$,

$$E[f(x_t) - f(x)] \leq \frac{1}{2\eta}[||x_t - x^*||^2 - ||x_{t+1} - x^*||^2] + \frac{\eta}{2}(\sigma^2 + G^2)$$

Summing up from $t = 0$ to $T - 1$ we get

$$E[\frac{1}{T}\sum_{t=0}^{T-1} f(x_t) - f(x^*)] \leq \frac{1}{2\eta}\frac{||x_0 - x^*||^2}{T} + \frac{\eta}{2}(\sigma^2 + G^2)$$

$$\leq \frac{R^2}{2\eta T} + \frac{\eta}{2}(\sigma^2 + G^2)$$

Setting $\eta = \frac{R}{\sqrt{T(\sigma^2+G^2)}}$, we get that the RHS is at most

$$\leq \frac{R(\sigma + G)}{\sqrt{T}}$$

□

[1]Technically we need to first take expectation conditioned on x_t and then take expectation w.r.t. x_t.

Theorem 2. *If f is convex and β-Lipschitz then after T steps of stochastic gradient descent with $\eta = \frac{1}{\beta+\sigma\frac{\sqrt{T}}{R}}$ we have, $E[\frac{1}{T}\sum_{t=0}^{T-1} f(x_t) - f(x^*)] \leq \frac{\beta R^2}{T} + \frac{\sigma R}{\sqrt{T}}$, where $\sigma^2 = \max_x[||\hat{\nabla} f(x) - \nabla f(x)||^2]$ and $R = \|x_0 - x^*\|$.*

Proof. We will use the property of Lipschitz functions that $f(y) \leq f(x) + \nabla f(x)(y-x) + \frac{\beta}{2}||x-y||^2$. We have

$$f(x_{t+1}) \leq f(x_t) + \nabla f(x_t)(x_{t+1} - x_t) + \frac{\beta}{2}||x_{t+1} - x_t||^2.$$

Also by convexity, $f(x^*) \geq f(x_t) + \nabla f(x_t)(x^* - x_t)$. From this we get that

$$f(x_{t+1}) - f(x^*) \leq \nabla f(x_t)(x_t - x^*) + \nabla f(x_t)(x_{t+1} - x_t) + \frac{\beta}{2}||x_{t+1} - x_t||^2$$

Writing this in terms of the noisy gradient:

$$f(x_{t+1}) - f(x^*) \leq \nabla f(x_t)(x_t - x^*) - \eta \nabla f(x_t)\hat{\nabla} f(x_t) + \frac{\beta\eta^2}{2}||\hat{\nabla} f(x_t)||^2$$

$$\leq \hat{\nabla} f(x_t)(x_t - x^*) - \eta||\hat{\nabla} f(x_t)||^2 + \frac{\beta\eta^2}{2}||\hat{\nabla} f(x_t)||$$

$$+(\nabla f(x_t) - \hat{\nabla} f(x_t))(x_t - x^*) - \eta(\nabla f(x_t) - \hat{\nabla} f(x_t))\hat{\nabla} f(x_t)$$

Note that we had the first three terms above when there was no noise. The last two terms above are new in the noisy case. When you take the expectation, the new noisy case terms are 0 and $\eta\sigma^2$, respectively. Lets rewrite the above equation with this in mind. We get

$$f(x_{t+1}) - f(x^*) \leq \hat{\nabla} f(x_t)(x_t - x^*) - \eta||\hat{\nabla} f(x_t)||^2 + \frac{\beta\eta^2}{2}||\hat{\nabla} f(x_t)||$$

$$+\eta\sigma^2 + \frac{1}{2\eta}||x_t - x^*||^2 - \frac{1}{2\eta}||x_t - x^*||^2$$

$$\leq -\frac{1}{2\eta}[||x_t - x^* - \eta\hat{\nabla} f(x_t)||^2 - \frac{\eta}{2}||\hat{\nabla} f(x_t)||^2 + \frac{\beta\eta}{2}||\hat{\nabla} f(x_t)||^2 + \eta\sigma^2 + \frac{1}{2\eta}[||x_t - x^*||^2]$$

$$\frac{1}{2\eta}[||x_t - x^*||^2 - ||x_{t+1} - x^*||^2] + \eta\sigma^2$$

Since $\frac{\eta}{2} > \frac{\beta\eta^2}{2}$, then $\eta < \frac{1}{\beta}$ and we get that

$$E[\frac{1}{T}\sum_{t=0}^{T-1} f(x_{t+1}) - f(x^*)] \leq \frac{R^2}{2\eta T} + \eta\sigma^2$$

Set $\eta = \frac{1}{\beta + \frac{\sigma\sqrt{T}}{R}}$

$$E[\frac{1}{T}\sum_{t=0}^{T-1} f(x_{t+1}) - f(x^*)] \leq \frac{\beta R^2}{T} + \frac{\sigma R}{\sqrt{T}}$$

□

1.1 ML Problems

Let's take a look at the case where we know the function we are optimizing but we don't want to take all the gradients.

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$$

The natural approach towards quickly computing a noisy gradient is to choose $i \sim [m]$ uniformly at random and let $\hat{\nabla} f(x_t) = \nabla f_i(x_t)$. Here is an extension known as *mini batch SGD* that has a lower variance. Choose B indices at random:

$$\hat{\nabla} f(x_t) = \frac{1}{B}\sum_{i=1}^{B} \nabla f_i(x_t)$$

$$\hat{g}(x_t) \leftarrow \sigma$$

It is easy to see that if σ^2 is the variance of the original estimator based on a single random sample, then the variance of the new estimator is $\frac{\sigma^2}{\beta}$. Using this to compute a noisy gradient is known as Mini Batch Stochastic Gradient Descent. The rate of convergence for β-Lipschitz functions is $\frac{\beta R^2}{T} + \frac{\sigma R}{B\sqrt{T}}$. To recover the original rate of gradient descent, set $B \approx \sqrt{T}$

$$Rate \approx \frac{(\beta + \sigma)R^2}{T}$$

Mini Batch SGD works well when you can utilize parallel processors. Each processor picks a random gradient to calculate. Instead of performing $\approx B$ calculations, you perform a constant number of calculations.

Q; Can one get rid of σ (the variance)? Can you design algorithms that provide noisy gradients where the variance approaches 0 as we approach the optimal solution? In fact, you can on the condition that you know the function.

1.2 Stochastic Variance Reduced Gradient Descent (SVRGD)

Ideally we want to output $\nabla f(x_t)$, the true gradient but it's expensive to compute. Re-write the true gradient as

$$\nabla f(x_t) = \nabla f_i(x_t) + [\nabla f(x_t) - \nabla f_i(x_t)]$$

Now pick a point y that has been produced at an earlier statge by our algorithm and define

$$\hat{\nabla} f(x_t) = \nabla f_i(x_t) + [\nabla f(y) - \nabla f_i(y)]$$

It is easy to see that this is also an unbiased estimator. Let's look at the variance of the new estimator:

$$||\hat{\nabla} f(x_t) - \nabla f(x_t)||^2 = ||\nabla f_i(x_t) - \nabla f_i(y) + \nabla f(y) - \nabla f(x_t)||^2$$

The hope is that as optimization proceeds and we approach x^*, x_t and y are getting closer to each other and hence the variance is approaching 0. However, we don't want to compute

$\nabla f(y)$ many times since it is expensive. Instead, we will compute it (update it) every once in a while. Here is the algorithm for SVRGD:

Algorithm 1: SVRGD

$y_1 = 0$;
for $s = 1, 2, ..., S$ **do**
  $x_1 = y_s$;
  for $t = 1, ..., T$ **do**
    Choose $i \in [m]$ at random and define $\hat{\nabla} f(x_t) = \nabla f_i(x_t) + \nabla f(y) - \nabla f_i(y)$;
    $x_{t+1} \leftarrow x_t - \eta \hat{\nabla} f(x_t)$;
  $y_{s+1} = \frac{1}{T} \sum_{t=1}^{T} x_t$;

Note: we want values of y and current x to stay close to each other.
What guarantee can we get?

Theorem 3. *If f_i are β-Lipschitz, $\eta = \frac{1}{10\beta}$, then in S epochs $E[\frac{1}{S} \sum_{s=1}^{s} f(y_s) - f(x^*)] \leq \frac{\beta R^2}{T} + (.9)^S \beta R^2$.*

Notice that the variance term is going down exponentially fast in S.

2 Additional Readings

- The book by Boyd and Vandenberghe. `http://stanford.edu/~boyd/cvxbook/`.
- Lecture notes by Sébastian Bubeck. `https://blogs.princeton.edu/imabandit/orf523-the-complexities-of-optimization/`.
- SVRGD paper. `https://papers.nips.cc/paper/4937-accelerating-stochastic-gradient-descen pdf`.