10-725: Convex Optimization **Spring 2023**

Lecture 8: February 9th

Lecturer: Siva Balakrishnan

Today we'll talk about the stochastic gradient descent (SGD) algorithm.

8.1 Stochastic Gradient Descent

SGD has a long, rich history and the basic algorithm has been reinvented many times. The algorithm in roughly the form we study it today is usually attributed to Robbins and Munro who were trying to find roots of functions with noisy function access (very similar to function optimization, with noisy gradient access).

As usual we're trying to minimize a function f (for now, lets suppose that f is differentiable, and that we're interested in solving an unconstrained problem). In many examples (we'll see several in this lecture), it will be natural to hypothesize that rather than obtaining the exact gradient value $\nabla f(x)$, at some point x, we are able to compute a vector $g(x, \xi)$ which is a function of a random variable ξ and has the property that,

$$\mathbb{E}_{\xi}[g(x, \xi)] = \nabla f(x).$$

We'll generally suppose that ξ has distribution P. The expectation above is saying that the stochastic gradient $g(x, \xi)$ is an unbiased estimate of the actual gradient $\nabla f(x)$.

The SGD algorithm then simply follows the iterates:

$$x^{t+1} = x^t - \eta_t g(x^t, \xi^t),$$

where ξ^t has distribution P, and is drawn independently of everything else. Often computing $g(x^t, \xi^t)$ will be much faster than computing $\nabla f(x^t)$, but in general it can be a very noisy estimate of $\nabla f(x^t)$ (i.e. it might for instance have high variance).

Similar to the subgradient method we'll need to be careful about our choice of step-size, i.e. for instance it's easy to see that (unlike for GD in the smooth case) fixed step-size choices don't usually work. If we're at the optimum x^*, the gradient might be zero, but the stochastic gradient might still be non-zero (variance), and we'll need to carefully choose the step-size to decay to ensure convergence.

8.2 Some Examples of SGD Algorithms

8.2.1 Noisy Gradients

Maybe the most intuitive setting is where rather than have access to gradients, we have access to a noisy oracle, i.e. we can make measurements of the gradient which are corrupted by additive noise, i.e.

$$g(x, \xi) = \nabla f(x) + \xi,$$

where $\mathbb{E}[\xi] = 0$.

8.2.2 Incremental Gradient Method

This is the idea behind many algorithms (including things like backpropogation), where we are attempting to minimize a function:

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

Computing the gradient of f then requires, computing the gradient of each of $f_1, \ldots, f_n$, and the incremental gradient method instead just cycles through the functions, using the updates:

$$x^{t+1} = x^t - \eta_t \nabla f_{i^t}(x^t),$$

where $i^t = t \bmod n + 1$. This isn't really an SGD algorithm (and can be reasonably difficult to analyze) but one can instead imagine the randomized version – at each iteration we choose an index i^t uniformly at random from $\{1, \ldots, n\}$, and we repeat the same iteration as above.

Here ξ^t denotes the (random) choice of index, and $g(x, \xi^t) = \nabla f_{\xi^t}(x)$. It's easy to check that,

$$\mathbb{E}[g(x, \xi)] = \nabla f(x),$$

so this randomized variant is indeed an SGD algorithm.

8.2.3 ERM and Population Risk Minimization

Empirical Risk Minimization is one of the standard lenses through which we come up with algorithms (and statistical analysis) in machine learning. The so-called "statistical learning" setup, is that we have a loss function, and an associated risk, i.e. the expected loss. Our goal,

broadly speaking, is to find rules/functions/... which have small risk (i.e. small expected loss) given some "training samples" from a distribution.

We are typically given samples $\{(X_1, y_1), \ldots, (X_n, y_n)\} \sim P_{Xy}$ and given a rule/classifier/regressor f, we evaluate it via:

$$R(f) = \mathbb{E}_{X,y\sim P_{Xy}}[\ell(f(X), y)],$$

where ℓ measures the loss for making a prediction $f(X)$ when the true label is y.

We cannot directly minimize R to find the best rule since we don't have access to the distribution P_{Xy} so a standard idea is to instead attempt to minimize:

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(X_i), y_i),$$

over candidate functions f.

One can now see at least two connections to SGD:

1. We can apply the randomized incremental gradient algorithm to the *empirical risk*. Usually our rules f will be parametrized by some parameters, and we'll be doing SGD on those parameters but for now the exact details are not so important.

2. Perhaps less obviously – if we only make one pass over the training samples, then we can view the incremental gradient algorithm as *directly* minimizing the *population risk*, i.e. $\mathbb{E}[\nabla\ell(f(X_i), y_i)] = \nabla R(f)$.

   This connection is very interesting, i.e. if we can show that one-pass SGD works in this setting (i.e. converges to something close to the minimizer of $R(f)$ at some rate) – then we can obtain "generalization bounds" directly.

To summarize, in the "statistical learning" setting one can view SGD either as an algorithm for minimizing the empirical risk (in which case, we'll need to separately reason about how good the ERM solution is, i.e. separately prove a generalization bound). Alternatively, one can view it as an algorithm for minimizing the population risk directly.

8.2.4 Mini-Batch SGD

In the ERM setting, or in the incremental gradient setting we are not restricted to using a single sample (or single function) to compute our stochastic gradient. Often in practice (due to various communication, data-manipulation bottlenecks) it will be faster to choose subsets $I_t \subset \{1, \ldots, n\}$ of size m (say), and compute:

$$x^{t+1} = x^t - \eta_t \frac{1}{m}\sum_{i\in I_t} \nabla f_i(x^t).$$

If the subsets are chosen uniformly at random from $\{1, \ldots, n\}$ then this is a valid stochastic gradient. It has a variance which is a factor of m smaller (but can be m times more expensive to compute). In practice, m is a hyperparameter which needs to be tuned carefully.

8.2.5 Randomized Coordinate Descent

Another example of an SGD algorithm comes from (randomized) coordinate descent. In coordinate descent, at each iteration we select a variable $i^t \in \{1, \ldots, d\}$ and do a gradient step updating only that variable (keeping all others fixed), i.e.:

$$x^{t+1} = x^t - d\eta_t \nabla_{i^t} f(x^t),$$

where $\nabla_{i^t} f(x)$ denotes the partial derivative of f with respect to i^t-th variable.

Now, it is easy to see that if the variable to update is chosen uniformly at random from $\{1, \ldots, d\}$ then,

$$d\mathbb{E}\nabla_i f(x) = \nabla f(x),$$

so this is an example of a stochastic gradient descent algorithm.

8.3 A Warm-Up Example

We'd like to develop an understanding of the rates of convergence of the SGD algorithm, and perhaps some insights on step-size choices, and some insights on the role of the variance (at least intuitively, it should be the case that the variance of the stochastic gradients affects how fast the algorithm converges).

Suppose our goal is to optimize a very simple quadratic objective:

$$\min_x \frac{1}{2n} \sum_{i=1}^{n} \|X_i - x\|_2^2.$$

Suppose we start at $x^0 = 0$. Now, the incremental gradient algorithm would use the updates for $t = \{0, \ldots, n-1\}$.

$$x^{t+1} = x^t - \eta_t(x^t - X_{t+1}) = (1 - \eta_t)x^t + \eta_t X_{t+1}.$$

If we use the step-size $\eta_t = \frac{1}{t+1}$, then we have that,

$$x^{t+1} = \frac{tx^t + X_{t+1}}{t+1}.$$

After n iterations the incremental gradient algorithm would converge to the optimal solution (just the average of $X_1, \ldots, X_n$). One maybe shouldn't take too much away from this example (it's not even an SGD algorithm) but notice that even in this extremely favorable case (smooth, strongly convex objective) we needed to take our step-sizes to decay at the rate $1/(t+1)$.

One-pass SGD is a bit more interesting to study. Suppose we are interested in optimizing the population objective:

$$\min_x \frac{1}{2}\mathbb{E}_{X\sim P}\|X - x\|_2^2.$$

We obtain samples $X_1, \ldots, X_n$ from P. Lets suppose that P has mean μ and variance σ^2. From each sample, we can compute a stochastic gradient $g(x^t, X_i) = X_i - x^t$, and use this in an SGD algorithm. Suppose we use step-sizes $\eta_t = 1/(t+1)$, and $x^0 = 0$ as above. In this case, after n iterations we obtain the solution,

$$\widehat{x} = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

Now, we can evaluate the quality of $\widehat{x}$ via its objective value,

$$\frac{1}{2}\mathbb{E}_{X\sim P}\|X - \widehat{x}\|_2^2 = \frac{\sigma^2}{2} + \frac{\sigma^2}{2n}.$$

On the other hand the optimal solution x^* is the population mean, which achieves the objective value,

$$\frac{1}{2}\mathbb{E}_{X\sim P}\|X - \mu\|_2^2 = \frac{\sigma^2}{2}.$$

So we see that,

$$f(\widehat{x}) - f(x^*) = \frac{\sigma^2}{2n}.$$

Notice that:

1. Even in this favorable case (smooth, strongly convex objective) we obtain $1/k$-type rates of convergence.
2. Furthermore, we know that this cannot be improved in this case. Standard statistical lower bounds will tell us that the sample mean is the best possible estimator here, and that it's excess error scales exactly like $\sigma^2/2n$.
3. In this case, the SGD algorithm which processes a single sample at a time, and makes a step after each sample, is as good as any estimator which uses all of the samples $X_1, \ldots, X_n$ at once.

8.4 SGD for Lipschitz Convex Functions

We will now turn our attention to some formal results for the SGD algorithm. We'll analyze SGD for non-smooth functions, and here the hypothesis will be that,

$$\mathbb{E}_\xi[g(x,\xi)] \in \partial f(x).$$

Theorem 8.1 *Suppose that f is convex, our initialization satisfies $\|x^0 - x^*\|_2 \leq R$ (for some, not necessarily unique, minimizer x^* which is fixed throughout the proof), and the stochastic gradients satisfy,*

$$\mathbb{E}\|g(x,\xi)\|_2^2 \leq G^2 \quad \textit{for all } x,$$

then if we choose $\eta = \frac{\sqrt{R}}{\sqrt{Gk}}$, we have the guarantee that,

$$\mathbb{E}f\left(\frac{1}{k}\sum_{t=1}^{k} x^t\right) - f(x^*) \leq \frac{RG}{\sqrt{k}}.$$

Notice, the main differences between our earlier result for the subgradient method and this result:

1. We obtain a guarantee that holds in expectation, and we obtain a guarantee for the averaged iterate (similar bounds hold in high-probability and for the last iterate but are a bit more difficult to prove).
2. We make a different hypothesis, essentially that the stochastic gradients are bounded. This in some sense bounds the variance of the stochastic gradients (as well as the magnitude of the actual gradients).
3. The SGD algorithm here can be much faster than the subgradient method (at least in the ERM type problems we discussed earlier). It achieves the same rate of convergence as a function of k but each iteration of SGD can be much faster than a corresponding iteration of the sub-gradient method.

Proof: The proof is very similar to that of the subgradient method, except that we use expectations (and conditional expectations) at various points. We're using a fixed step-size across iterations. As usual we have that,

$$\|x^{t+1} - x^*\|_2^2 = \|x^t - x^*\|_2^2 + \eta^2\|g(x^t,\xi)\|_2^2 - 2\eta(x^t - x^*)^T g(x^t,\xi).$$

Now, suppose we consider:

$$\begin{aligned}\mathbb{E}\left[\|x^{t+1} - x^*\|_2^2 | x^t\right] &= \|x^t - x^*\|_2^2 + \eta^2\mathbb{E}\left[\|g(x^t,\xi)\|_2^2 | x^t\right] - 2\eta(x^t - x^*)^T\mathbb{E}[g(x^t,\xi)|x^t] \\ &\leq \|x^t - x^*\|_2^2 + \eta^2 G^2 - 2\eta(x^t - x^*)^T g_{x^t},\end{aligned}$$

where $g_{x^t} \in \partial f(x^t)$. Now, we use convexity on the last term to obtain that,

$$\mathbb{E}\left[\|x^{t+1} - x^*\|_2^2 | x^t\right] \leq \|x^t - x^*\|_2^2 + \eta^2 G^2 + 2\eta(f(x^*) - f(x^t)),$$

and therefore by the tower property of conditional expectations,

$$\mathbb{E}\left[\|x^{t+1} - x^*\|_2^2\right] \leq \mathbb{E}\left[\|x^t - x^*\|_2^2\right] + \eta^2 G^2 + 2\eta(f(x^*) - \mathbb{E}f(x^t)).$$

Re-arranging and telescoping the sum we obtain that,

$$\frac{1}{k}\sum_{t=1}^{k} \mathbb{E}f(x^t) - f(x^*) \leq \frac{G^2\eta}{2} + \frac{\|x^0 - x^*\|_2^2}{2k\eta}.$$

Now, by convexity we know that,

$$f\left(\frac{1}{k}\sum_{t=1}^{k} x^t\right) \leq \sum_{t=1}^{k} f(x^t), \tag{8.1}$$

so we obtain that,

$$\mathbb{E}f\left(\frac{1}{k}\sum_{t=1}^{k} x^t\right) - f(x^*) \leq \frac{G^2\eta}{2} + \frac{\|x^0 - x^*\|_2^2}{2k\eta},$$

and using our choice of step-size this gives the desired result.

■

8.5 SGD for Strongly Convex Functions

The key takeaway from this section is that for strongly convex functions, SGD does not achieve a linear rate of convergence (and additionally assuming smoothness makes no difference). This is primarily due to the variance of the stochastic gradients, and in some later lecture we might discuss tools for variance reduction in SGD (which do in some cases yield algorithms with linear convergence rates for structured smooth and strongly convex functions).

Theorem 8.2 *Suppose f is α-strongly convex, and the stochastic gradients satisfy,*

$$\mathbb{E}\|g(x,\xi)\|_2^2 \leq G^2 \quad \textit{for all } x.$$

Then,

1. *For a fixed step-size $\eta < 1/\alpha$, we obtain,*

$$\mathbb{E}\|x^k - x^*\|_2^2 \leq (1 - \alpha\eta)^k \|x^0 - x^*\|_2^2 + \frac{\eta G^2}{\alpha}.$$

2. *For $\eta_t = \frac{1}{\alpha(t+1)}$,*

$$\mathbb{E} f\left(\frac{1}{k}\sum_{t=1}^{k} x^t\right) - f(x^*) \leq \frac{G^2(1 + \log k)}{2\alpha k}.$$

It is worth noticing:

1. The first result suggests that SGD iterates with a fixed step-size, will converge rapidly to some fixed ball around x^* and then bounce around there. This in turn suggests a very common practical epoch-based heuristic for SGD step-sizes – run it with some fixed step-size, when it seems like the iterates are bouncing around (or you stop making progress in function value), then decay it by some factor and continue running it.
2. In the second case, one can remove the extra log factor with some work – for instance, if you use an SGD variant where rather than average all the iterates you only average the last half the log factor can be eliminated.

Proof: Suppose we follow our earlier proof to obtain that,

$$\begin{aligned}\mathbb{E}\left[\|x^{t+1} - x^*\|_2^2 | x^t\right] &= \|x^t - x^*\|_2^2 + \eta_t^2 \mathbb{E}\left[\|g(x^t, \xi)\|_2^2 | x^t\right] - 2\eta_t (x^t - x^*)^T \mathbb{E}[g(x^t, \xi) | x^t] \\ &\leq \|x^t - x^*\|_2^2 + \eta_t^2 G^2 - 2\eta_t (x^t - x^*)^T \nabla f(x^t).\end{aligned}$$

The key point to notice here is that previously we would have used the descent lemma (a consequence of smoothness) to bound the squared norm of the gradient. However, in the current stochastic gradient setup, the expected squared norm of the gradient includes two contributions: one which is roughly the squared norm of the expected gradient which we could hope to control by smoothness, and the second which is the variance of the stochastic gradients. This latter term, we should not in general expect to decrease as we get close to the optimum.

Now, using strong convexity on the last term we obtain that,

$$\mathbb{E}\left[\|x^{t+1} - x^*\|_2^2 | x^t\right] \leq \|x^t - x^*\|_2^2 + \eta_t^2 G^2 - \alpha\eta_t \|x^t - x^*\|_2^2 + 2\eta_t (f(x^*) - f(x^t)). \tag{8.2}$$

Proof of Claim 1: Now, to prove the first claim we use a fixed step-size η and see that,

$$\mathbb{E}\left[\|x^{t+1} - x^*\|_2^2\right] \leq (1 - \alpha\eta)\mathbb{E}[\|x^t - x^*\|_2^2] + \eta^2 G^2,$$

and so provided that $\alpha\eta < 1$ we can unroll this recursion to obtain,

$$\mathbb{E}\left[\|x^k - x^*\|_2^2\right] \leq (1-\alpha\eta)^k \|x^0 - x^*\|_2^2 + \frac{\eta G^2}{\alpha}.$$

Proof of Claim 2: Rearranging (8.2), and using the tower property, we see that,

$$\mathbb{E}f(x^t) - f(x^*) \leq \frac{\mathbb{E}\left[\|x^t - x^*\|_2^2\right] - \mathbb{E}[\|x^{t+1} - x^*\|_2^2]}{2\eta_t} + \frac{\eta_t G^2}{2} - \frac{\alpha}{2}\mathbb{E}[\|x^t - x^*\|_2^2].$$

Now, one can verify that with our choice of step-sizes $\eta_t = 1/\alpha(t+1)$ the first two and last terms together telescope, and we are left with $-\alpha k\|x^{k+1} - x^*\|_2^2$ which is negative and can be dropped. Thus we obtain the bound,

$$\sum_{t=0}^{k}\left[\mathbb{E}f(x^t) - f(x^*)\right] \leq \frac{G^2}{2\alpha}\sum_{t=0}^{k}\eta_t \leq \frac{G^2(1+\log k)}{2\alpha}.$$

Using the same idea as in (8.1) we obtain the final bound.

■