CME 307/MSE 311: Optimization

Acceleration, Stochastic Gradient Descent, and Variance Reduction

Professor Udell

Management Science and Engineering
Stanford

April 26, 2023

Convergence of gradient descent

unconstrained minimization: find $x \in \mathbb{R}^n$ to solve

$$\text{minimize} \quad f(x) \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable

we analyzed gradient descent (GD) on this problem:

- a point x is ϵ-*suboptimal* if $f(x) - f^\star \leq \epsilon$
- when f is L-smooth and μ-PL (or μ-strongly convex), we showed GD converges to sub-optimality ϵ in at most

$$T = \mathcal{O}\left(\kappa \log\left(\frac{1}{\epsilon}\right)\right) \text{ iterations,}$$

where $\kappa := \frac{L}{\mu}$ is the condition number

Acceleration: motivation

Definition

a *first-order method* uses only a first-order oracle for $f : \mathbb{R}^n \to \mathbb{R}$ (*i.e.*, gradient and function evaluation) to minimize $f(x)$

GD $x \leftarrow x - \alpha \nabla f(x)$ is a first-order method

Acceleration: motivation

Definition

a *first-order method* uses only a first-order oracle for $f : \mathbb{R}^n \to \mathbb{R}$ (*i.e.*, gradient and function evaluation) to minimize $f(x)$

GD $x \leftarrow x - \alpha \nabla f(x)$ is a first-order method

Q: is GD the best first-order method for L-smooth, μ-strongly convex functions?

Acceleration: motivation

Definition

a *first-order method* uses only a first-order oracle for $f : \mathbb{R}^n \to \mathbb{R}$ (*i.e.*, gradient and function evaluation) to minimize $f(x)$

GD $x \leftarrow x - \alpha \nabla f(x)$ is a first-order method

Q: is GD the best first-order method for L-smooth, μ-strongly convex functions?
A: no! Nemirovski and Yudin showed a *lower-bound* of

$$T_{\mathrm{opt}} = \Omega\left(\sqrt{\kappa}\log\left(\frac{1}{\epsilon}\right)\right) \text{ iterations}$$

to find an ϵ-suboptimal point of *any* L-smooth, μ-strongly convex function
notice: same rate as CG if f is quadratic

A worst-case quadratic function

the lower bound can be obtained by constructing a particularly hard problem instance using quadratic functions

A worst-case quadratic function

the lower bound can be obtained by constructing a particularly hard problem instance using quadratic functions

- easier to work in the infinite dimensional-space $\ell^2(\mathbb{R})$, which consists of vectors x of infinite length, satisfying

$$\|x\|^2 = \sum_{j=1}^{\infty} x_j^2 < \infty$$

A worst-case quadratic function

the lower bound can be obtained by constructing a particularly hard problem instance using quadratic functions

- easier to work in the infinite dimensional-space $\ell^2(\mathbb{R})$, which consists of vectors x of infinite length, satisfying

$$\|x\|^2 = \sum_{j=1}^{\infty} x_j^2 < \infty$$

- the evil quadratic function is then given by

$$f(x) = \frac{\mu(\kappa_f - 1)}{8} \left(x_1^2 + \sum_{j=1}^{\infty} (x_j - x_{j+1})^2 - 2x_1 \right) + \frac{\mu}{2}\|x\|^2,$$

  where $\mu > 0$ and $\kappa_f > 1$
- above example actually gives a family of hard quadratic functions parametrized by μ, κ_f

source: Section 2.1, Nesterov, 2018

The lower bound

Using the family of quadratics on the preceding slide, the following theorem may be shown

Theorem (Nesterov Theorem 2.1.13)

Let $\mu > 0$, $\kappa_f > 1$. Suppose $\mathcal{M}$ is a first-order method such that for any input function f, $\mathcal{M}$ generates a sequence satisfying

$$x_k \in x_0 + \mathrm{span}\{\nabla f(x_0), \ldots, \nabla f(x_k)\}, \quad \forall k$$

Then there exists a L-smooth, μ-strongly convex function with $L/\mu = \kappa_f$ such that the sequence output by $\mathcal{M}$ applied to f satisfies

$$\|x_k - x_\star\|^2 \geq \left(\frac{\sqrt{\kappa_f} - 1}{\sqrt{\kappa_f} + 1}\right)^{2k} \|x_0 - x_\star\|^2,$$

$$f(x_k) - f(x_\star) \geq \frac{\mu}{2} \left(\frac{\sqrt{\kappa_f} - 1}{\sqrt{\kappa_f} + 1}\right)^{2k} \|x_0 - x_\star\|^2$$

Accelerated Gradient Descent

Nesterov's accelerated gradient method (AGD)

- a first-order method
- that matches the lower bound

so, converges faster than GD (esp. on ill-conditioned functions)

(one variant of) Nesterov's AGD:

1. Choose $x_0, y_0 \in \mathbb{R}^n$
2. for $k = 0, 1, \ldots, T$,

$$\begin{aligned} x_{k+1} &= y_k - \alpha \nabla f(y_k) \\ y_{k+1} &= x_{k+1} + \beta \left(x_{k+1} - x_k\right) \end{aligned}$$

3. Return x_{k+1}

achieves lower bound when $\alpha = \frac{1}{L}$, $\beta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$

source: Section 2.2, Nesterov, 2018

GD vs. AGD: numerical example

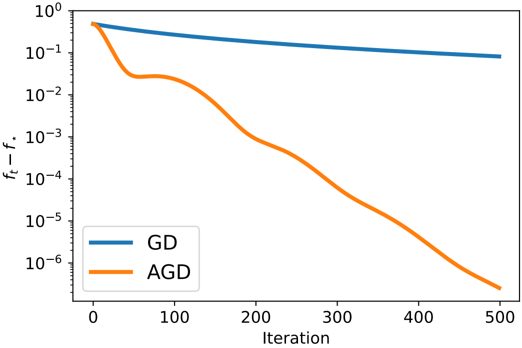
goal is to solve the logistic regression problem

$$\text{minimize} \quad \frac{1}{m}\sum_{i=1}^{m} \log\left(1 + \exp\left(-b_i a_i^T x\right)\right) + \frac{1}{m}\|x\|^2$$

with variable x on rcv1 dataset, with data matrix $A \in \mathbb{R}^{20{,}242 \times 47{,}236}$ and labels $b \in \mathbb{R}^{20{,}242}$

- GD and AGD both use theoretically-chosen stepsizes:
  - GD is run with stepsize $\frac{1}{L}$, which for this example equals 4
  - AGD is run with $\alpha = \frac{1}{L}$ and $\beta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$
- here strong convexity $\mu = \frac{1}{m}$ from the regularizer

GD vs. AGD results



AGD summary and closing remarks

- AGD is theoretically optimal among first-order methods for L-smooth and μ-strongly convex functions
- converges to ϵ-suboptimality in at most

$$\mathcal{O}\left(\sqrt{\kappa}\log\left(\frac{1}{\epsilon}\right)\right) \text{ iterations}$$

- despite its elegance, AGD is rarely used in practice (quasi-Newton methods work better and are more stable)
- however, it forms the basis for more useful accelerated gradient methods like FISTA and Katyusha

Outline

Stochastic optimization

Finite sum minimization

Minimizing a sum

finite sum minimization: solve

$$\text{minimize} \quad \frac{1}{m}\sum_{i=1}^{m} f_i(x)$$

examples:

- least squares: $f_i(x) = (a_i^T x - b_i)^2$
- logistic regression: $f_i(x) = \log(1 + \exp\left(-b_i a_i^T x\right))$
- maximum likelihood estimation: $f_i(x)$ is -loglik of observation i given parameter x
- machine learning: f_i is misfit of model x on example i

Minimizing a sum

finite sum minimization: solve

$$\text{minimize} \quad \frac{1}{m}\sum_{i=1}^{m} f_i(x)$$

with variable $x \in \mathbb{R}^n$

quandary:

- solving a problem with *more data* should be *easier*
- but complexity of algorithms increases with m!

goal: find algorithms that work *better* given *more* data
(or at least, not worse)

Minimizing a sum

finite sum minimization: solve

$$\text{minimize} \quad \frac{1}{m}\sum_{i=1}^{m} f_i(x)$$

with variable $x \in \mathbb{R}^n$

quandary:

- solving a problem with *more data* should be *easier*
- but complexity of algorithms increases with m!

goal: find algorithms that work *better* given *more* data
(or at least, not worse)
idea:

Minimizing a sum

finite sum minimization: solve

$$\text{minimize} \quad \frac{1}{m}\sum_{i=1}^{m} f_i(x)$$

with variable $x \in \mathbb{R}^n$

quandary:

- solving a problem with *more data* should be *easier*
- but complexity of algorithms increases with m!

goal: find algorithms that work *better* given *more* data
(or at least, not worse)
idea: throw away data! (cleverly)

Minimizing an expectation

Stochastic optimization: solve

$$\text{minimize} \quad \mathbb{E}f(x) = \mathbb{E}_\omega f(x;\omega)$$

with variable $x \in \mathbb{R}^n$

- random loss function f
- or equivalently, function $f(\cdot;\omega)$ of random variable ω

Minimizing an expectation

Stochastic optimization: solve

$$\text{minimize} \quad \mathbb{E}f(x) = \mathbb{E}_\omega f(x;\omega)$$

with variable $x \in \mathbb{R}^n$

- random loss function f
- or equivalently, function $f(\cdot;\omega)$ of random variable ω

examples: *data* $\omega = (a, b)$ is random

- least squares: $f(x;\omega) = (a^T x - b)^2$
- logistic regression: $f(x;\omega) = \log(1 + \exp\left(-ba^T x\right))$
- maximum likelihood estimation: $f(x;\omega)$ is -loglik of observation ω given parameter x
- machine learning: $f(x;\omega)$ is misfit of model x on example ω

minimize test loss, not just training loss

Stochastic optimization: what distribution?

stochastic optimization problem

$$\begin{array}{ll} \text{minimize} & \mathbb{E}_{\omega\sim\mu_\Omega}\left[f(\omega,x)\right] \\ \text{variable} & x \in \mathbb{R}^n \end{array} \tag{2}$$

with $f(\omega,x) : \Omega \times \mathbb{R}^n$ convex, $\Omega \subseteq \mathbb{R}^n$, ω a random variable distributed according to probability measure μ_Ω

objective is expected cost under the randomness due to ω:

$$\mathbb{E}_{\omega\sim\mu_\Omega}\left[f(\omega,x)\right] = \int_\Omega f(\omega;x)d\mu_\Omega(\omega)$$

Stochastic optimization: examples

1. $n = 1, \Omega = \mathbb{R}$, and $f(\omega, x) = (x - \omega)^2$.

$$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}}\left[(x-\omega)^2\right]$$

Stochastic optimization: examples

1. $n = 1, \Omega = \mathbb{R}$, and $f(\omega, x) = (x - \omega)^2$.

$$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}} \left[(x - \omega)^2 \right]$$

then $x_\star =$

Stochastic optimization: examples

1. $n = 1, \Omega = \mathbb{R}$, and $f(\omega, x) = (x - \omega)^2$.

$$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}} \left[(x - \omega)^2 \right]$$

then $x_\star = \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}}[\omega]$ and $f_\star = \mathrm{Var}_{\omega \sim \mu_{\mathbb{R}}}[\omega]$.

Stochastic optimization: examples

1. $n = 1, \Omega = \mathbb{R}$, and $f(\omega, x) = (x - \omega)^2$.

$$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}} \left[(x - \omega)^2 \right]$$

   then $x_\star = \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}}[\omega]$ and $f_\star = \mathrm{Var}_{\omega \sim \mu_{\mathbb{R}}}[\omega]$.

2. $n = 1, \Omega = \mathbb{R}$, and $f(\omega, x) = |x - \omega|$.

$$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}} \left[|x - \omega| \right]$$

Stochastic optimization: examples

1. $n = 1, \Omega = \mathbb{R}$, and $f(\omega, x) = (x - \omega)^2$.

$$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}} \left[(x - \omega)^2 \right]$$

   then $x_\star = \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}}[\omega]$ and $f_\star = \mathrm{Var}_{\omega \sim \mu_{\mathbb{R}}}[\omega]$.

2. $n = 1, \Omega = \mathbb{R}$, and $f(\omega, x) = |x - \omega|$.

$$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}} \left[|x - \omega| \right]$$

   then $x_\star =$

Stochastic optimization: examples

1. $n = 1, \Omega = \mathrm{I\!R}$, and $f(\omega, x) = (x - \omega)^2$.

$$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathrm{I\!R}}} \left[(x - \omega)^2 \right]$$

   then $x_\star = \mathbb{E}_{\omega \sim \mu_{\mathrm{I\!R}}}[\omega]$ and $f_\star = \mathrm{Var}_{\omega \sim \mu_{\mathrm{I\!R}}}[\omega]$.

2. $n = 1, \Omega = \mathrm{I\!R}$, and $f(\omega, x) = |x - \omega|$.

$$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathrm{I\!R}}} \left[|x - \omega| \right]$$

   then $x_\star =$ the median of $\mu_{\mathrm{I\!R}}$

Stochastic optimization: examples

1. $n = 1, \Omega = \mathbb{R}$, and $f(\omega, x) = (x - \omega)^2$.

   $$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}} \left[(x - \omega)^2 \right]$$

   then $x_\star = \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}}[\omega]$ and $f_\star = \mathrm{Var}_{\omega \sim \mu_{\mathbb{R}}}[\omega]$.

2. $n = 1, \Omega = \mathbb{R}$, and $f(\omega, x) = |x - \omega|$.

   $$\text{minimize} \quad \mathbb{E}_{\omega \sim \mu_{\mathbb{R}}} [|x - \omega|]$$

   then $x_\star =$ the median of $\mu_{\mathbb{R}}$

3. $\Omega = \mathbb{R}^n$, $\mu_{\mathbb{R}^n} = \frac{1}{m} \sum_{i=1}^m \delta_{\omega_i}$. we get the finite sum minimization problem

   $$\text{minimize} \quad \frac{1}{m} \sum_{i=1}^m f(\omega_i, x).$$

Stochastic gradient oracle

Definition

a *stochastic gradient oracle* $\mathcal{G}$, when queried at $x \in \mathbb{R}^n$ produces $g(\omega; x) \in \mathbb{R}^n$ satisfying

$$\mathbb{E}_{\omega \sim \mu_\Omega} [g(\omega; x)] = \nabla F(x)$$

i.e., $\mathcal{G}$ produces an unbiased estimate of the true gradient $\nabla F(x)$

Stochastic gradient oracle

Definition

a *stochastic gradient oracle* $\mathcal{G}$, when queried at $x \in \mathrm{I\!R}^n$ produces $g(\omega; x) \in \mathrm{I\!R}^n$ satisfying

$$\mathbb{E}_{\omega \sim \mu_\Omega} \left[g(\omega; x) \right] = \nabla F(x)$$

i.e., $\mathcal{G}$ produces an unbiased estimate of the true gradient $\nabla F(x)$

Stochastic gradient oracle

Definition

a *stochastic gradient oracle* $\mathcal{G}$, when queried at $x \in \mathbb{R}^n$ produces $g(\omega; x) \in \mathbb{R}^n$ satisfying

$$\mathbb{E}_{\omega \sim \mu_\Omega} [g(\omega; x)] = \nabla F(x)$$

i.e., $\mathcal{G}$ produces an unbiased estimate of the true gradient $\nabla F(x)$

Q: examples of stochastic gradient oracle?

Stochastic gradient oracle

Definition

a *stochastic gradient oracle* $\mathcal{G}$, when queried at $x \in \mathbb{R}^n$ produces $g(\omega; x) \in \mathbb{R}^n$ satisfying

$$\mathbb{E}_{\omega \sim \mu_\Omega} [g(\omega; x)] = \nabla F(x)$$

i.e., $\mathcal{G}$ produces an unbiased estimate of the true gradient $\nabla F(x)$

Q: examples of stochastic gradient oracle?
A: minibatch gradient

$$\frac{1}{|S|} \sum_{\omega \in S} \nabla f_i(\omega, x)$$

notation: use $\hat{\nabla} f(x)$ to denote stochastic gradient at x

Stochastic gradient descent (SGD)

SGD:

1. Choose $x_0 \in \mathbb{R}^n$
2. for $k = 0, 1, \dots$
   1. query $\mathcal{G}$ at x_k to obtain $g(\omega_k, x_k)$
   2. compute update:

$$x_{k+1} = x_k - \eta_k g(\omega_k, x_k)$$

- SGD is not a descent method!
- SGD exactly the same as GD, except that it uses a stochastic gradient $g(\omega_k, x_k)$ rather than the true gradient
- selection of stepsize η_k is challenging!

A typical convergence result

Theorem (General SGD convergence)

Consider (2) *with smooth and strongly convex f and stochastic gradient oracle satisfying*

$$\mathbb{E}_\omega \|g(\omega, x)\|^2 \leq M_1 + M_2 \|\nabla F(\omega, x)\|^2.$$

1. *for an appropriate fixed stepsize $\eta_k = O(1)$,*

$$\lim_{k \to \infty} \mathbb{E}[f(\omega_k, x_k)] - f_\star = O(1)$$

2. *for decreasing stepsizes $\eta_k = O(1/k)$,*

$$\mathbb{E}[f(\omega_k, x_k)] - f_\star = O(1/k)$$

SGD convergence: discussion

- with fixed stepsize, the algorithm converges to ϵ-sublevel set
- convergence of SGD requires a decreasing stepsize $\implies$ slow!

contrast to GD, which converges to the exact optimum even with fixed stepsize

analysis is tight: there is a matching lower bound.
Agarwal et al., 2012 shows that for strongly convex problems, any algorithm using a stochastic gradient oracle must make at least $\Omega(1/\epsilon)$ queries to obtain an ϵ-suboptimal point

SGD convergence: discussion

- with fixed stepsize, the algorithm converges to ϵ-sublevel set
- convergence of SGD requires a decreasing stepsize $\implies$ slow!

contrast to GD, which converges to the exact optimum even with fixed stepsize

analysis is tight: there is a matching lower bound.
Agarwal et al., 2012 shows that for strongly convex problems, any algorithm using a stochastic gradient oracle must make at least $\Omega(1/\epsilon)$ queries to obtain an ϵ-suboptimal point

don't despair: add more assumptions!

Outline

Stochastic optimization

Finite sum minimization

Finite-sum minimization

return to finite sum problem:

$$\text{minimize} \quad \frac{1}{m}\sum_{i=1}^{m} f_i(x), \tag{3}$$

where each f_i is L_i-smooth and convex

why use SGD for finite sum minimization?

- evaluating minibatch gradient is cheaper per iteration
- converges faster than GD b/c each iteration is faster

Convergence of SGD

prove SGD minimizes finite sum (3):

Convergence of SGD

prove SGD minimizes finite sum (3):

$$\begin{aligned}\|x_{k+1} - x_\star\|^2 &= \|x_k - x_\star - \eta\widehat{\nabla} f(x_k)\|^2 \\ &= \|x_k - x_\star\|^2 - 2\eta\langle x_k - x_\star, \widehat{\nabla} f(x_k)\rangle + \eta^2\|\widehat{\nabla} f(x_k)\|^2.\end{aligned}$$

Convergence of SGD

prove SGD minimizes finite sum (3):

$$\begin{aligned}\|x_{k+1} - x_\star\|^2 &= \|x_k - x_\star - \eta\widehat{\nabla} f(x_k)\|^2 \\ &= \|x_k - x_\star\|^2 - 2\eta\langle x_k - x_\star, \widehat{\nabla} f(x_k)\rangle + \eta^2\|\widehat{\nabla} f(x_k)\|^2.\end{aligned}$$

take expectation wrt $\hat{\nabla} f(x_k)$:

$$\begin{aligned}\mathbb{E}_k\|x_{k+1} - x_\star\|^2 &= \|x_k - x_\star\|^2 - 2\eta\langle x_k - x_\star, \nabla f(x_k)\rangle + \eta^2\mathbb{E}_k\|\widehat{\nabla} f(x_k)\|^2 \\ &\leq (1 - \eta\mu)\|x_k - x_\star\|^2 - 2\eta\left(f(x_k) - f(x_\star)\right) \\ &\quad + \eta^2\mathbb{E}_k\|\widehat{\nabla} f(x_k)\|^2\end{aligned}$$

using strong convexity:

$$f(x_\star) \geq f(x_k) + \nabla f(x_k)^T(x_\star - x_k) + \frac{\mu}{2}\|x_\star - x_k\|^2.$$

One-step lemma

we have shown the following progress bound for one step of SGD

Lemma

at iteration k of SGD,

$$
\begin{aligned}
&\mathbb{E}_k\|x_{k+1} - x_\star\|^2 \\
&\quad \leq (1 - \eta\mu)\|x_k - x_\star\|^2 - 2\eta\left(f(x_k) - f(x_\star)\right) + \eta^2\mathbb{E}_k\|\widehat{\nabla} f(x_k)\|^2
\end{aligned}
$$

to show convergence, we must bound $\mathbb{E}_k\|\widehat{\nabla} f(x_k)\|^2$

One-step lemma

we have shown the following progress bound for one step of SGD

Lemma

at iteration k of SGD,

$$\begin{aligned} &\mathbb{E}_k\|x_{k+1} - x_\star\|^2 \\ &\quad \le (1 - \eta\mu)\|x_k - x_\star\|^2 - 2\eta\left(f(x_k) - f(x_\star)\right) + \eta^2\mathbb{E}_k\|\widehat{\nabla} f(x_k)\|^2 \end{aligned}$$

to show convergence, we must bound $\mathbb{E}_k\|\widehat{\nabla} f(x_k)\|^2$

we will follow the approach of Gower et al., 2019

Expected smoothness

Definition (Expected smoothness)

f satisfies *L-expected smoothness* (L-ES) if $\exists L > 0$ such that

$$\mathbb{E}\|\widehat{\nabla} f(x) - \widehat{\nabla} f(x_\star)\|^2 \leq 2L(f(x) - f(x_\star))$$

reduces to L-smoothness if we replace $\widehat{\nabla}$ by ∇:

$$f(x) - f(x_\star) \geq \frac{1}{2L}\|\nabla f(x) - \nabla f(x_\star)\|^2$$

Expected smoothness

Definition (Expected smoothness)

f satisfies *L-expected smoothness* (L-ES) if $\exists L > 0$ such that

$$\mathbb{E}\|\widehat{\nabla} f(x) - \widehat{\nabla} f(x_\star)\|^2 \leq 2L(f(x) - f(x_\star))$$

reduces to L-smoothness if we replace $\widehat{\nabla}$ by ∇:

$$f(x) - f(x_\star) \geq \frac{1}{2L}\|\nabla f(x) - \nabla f(x_\star)\|^2$$

Corollary

define $\sigma^2 := \mathbb{E}\|\widehat{\nabla} f(x_\star)\|^2$*. then*

$$\mathbb{E}\|\widehat{\nabla} f(x)\|^2 \leq 4L(f(x) - f(x_\star)) + 2\sigma^2, \qquad \forall x$$

Expected smoothness

Definition (Expected smoothness)

f satisfies *L-expected smoothness* (L-ES) if $\exists L > 0$ such that

$$\mathbb{E}\|\widehat{\nabla} f(x) - \widehat{\nabla} f(x_\star)\|^2 \leq 2L(f(x) - f(x_\star))$$

reduces to L-smoothness if we replace $\widehat{\nabla}$ by ∇:

$$f(x) - f(x_\star) \geq \frac{1}{2L}\|\nabla f(x) - \nabla f(x_\star)\|^2$$

Corollary

define $\sigma^2 := \mathbb{E}\|\widehat{\nabla} f(x_\star)\|^2$. *then*

$$\mathbb{E}\|\widehat{\nabla} f(x)\|^2 \leq 4L(f(x) - f(x_\star)) + 2\sigma^2, \qquad \forall x$$

under ES, gradient variance is controlled by suboptimality and variance of the gradient at the optimum

L-ES condition for smooth convex functions

Theorem (special case of Gower et al., 2019)

Suppose each f_i is L_i-smooth and convex. Consider mini-batch stochastic gradients $\widehat{\nabla} f = \frac{1}{|S|} \sum_{i \in S} f_i(x)$ with batch-size $b_g = |S|$. Then

$$\mathbb{E}\|\widehat{\nabla} f(x)\|^2 \leq 4L(f(x) - f(x_\star)) + 2\sigma^2,$$

with

$$L = \frac{m(b_g - 1)}{b_g(m-1)} \frac{1}{m} \sum_{i=1}^{m} L_i + \frac{m - b_g}{b_g(m-1)} \max_{1 \leq i \leq m} L_i$$

and

$$\sigma^2 = \frac{m - b_g}{b_g(m-1)} \frac{1}{m} \sum_{i=1}^{m} \|\nabla f_i(x_\star)\|^2$$

sanity check: $\sigma^2 \to 0$ as $b_g \to n$

Back to SGD convergence

using the one-step lemma with μ-strong convexity and L-ES, we find

$$\begin{aligned}\mathbb{E}_k\|x_{k+1} - x_\star\|^2 \leq (1 - \eta\mu)\|x_k - x_\star\|^2 &+ 2\eta(2\eta L - 1)\left(f(x_k) - f(x_\star)\right) \\ &+ \eta^2 2\sigma^2\end{aligned}$$

so, choosing stepsize $\eta \leq \frac{1}{2L}$,

$$\mathbb{E}_k\|x_{k+1} - x_\star\|^2 \leq (1 - \eta\mu)\|x_k - x_\star\|^2 + \eta^2 2\sigma^2$$

SGD convergence contd

apply induction + take total expectation to get

$$\mathbb{E}\|x_{k+1} - x_\star\|^2 \le (1-\eta\mu)^{k+1}\|x_0 - x_\star\|^2 + \left(\sum_{j=0}^{k}(1-\eta\mu)^j\right)\eta^2 2\sigma^2$$

$$\le (1-\eta\mu)^{k+1}\|x_0 - x_\star\|^2 + \frac{\eta 2\sigma^2}{\mu}$$

by summing the geometric series. choose $\eta \le \frac{\mu\epsilon}{4\sigma^2}$, so

$$\mathbb{E}\|x_{k+1} - x_\star\|^2 \le (1-\eta\mu)^{k+1}\|x_0 - x_\star\|^2 + \frac{\epsilon}{2}$$

we can solve for k to find how many iterations are needed to reach error $\frac{\epsilon}{2}$:

$$k \ge (\eta\mu)^{-1}\log\left(\frac{2(f(x_0) - f(x_\star))}{\epsilon}\right)$$

SGD convergence with fixed stepsize

we have shown

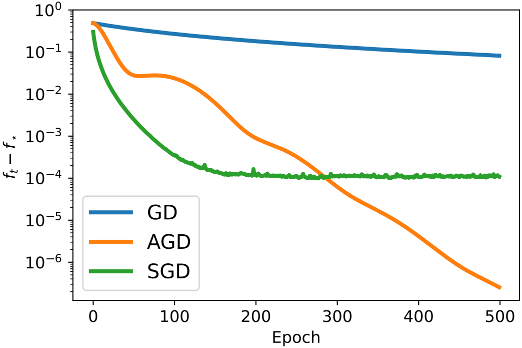
Theorem

Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is μ-strongly convex, with an L-ES stochastic gradient oracle. Run SGD with batchsize b_g and fixed stepsize $\eta = \min\left\{\frac{1}{2L}, \frac{\epsilon\mu}{4\sigma^2}\right\}$. Then for $k \geq (\eta\mu)^{-1} \log\left(\frac{2(f(x_0)-f(x_\star))}{\epsilon}\right)$ iterations,

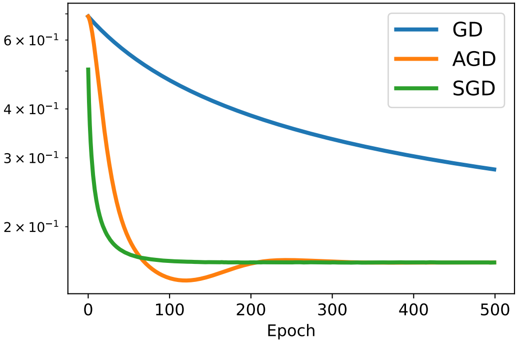
$$\mathbb{E}\|x_k - x_\star\|^2 \leq \epsilon$$

- same convergence rate as we'd get with decreasing stepsize sequence $\eta = \mathcal{O}(1/k)$
- but motivates variance reduction, which will give linear convergence!

Results: Optimization error



Results: Test error



The gradient is too noisy!

the expected smoothness condition shows the gradient is noisy,

$$\mathbb{E}\|\widehat{\nabla} f(x)\|^2 \leq 4L(f(x) - f(x_\star)) + 2\sigma^2,$$

even at $x_\star$

- good news: $f(x) - f^\star \to 0$ as $x \to x_\star$
- bad news: $\sigma^2 > 0$ even near $x_\star$

can we design an algorithm that eliminates this noise as $x \to x_\star$?

Stochastic Variance Reduced Gradient

Stochastic Variance Reduced Gradient (SVRG) uses a different stochastic gradient

$$g(x) = \widehat{\nabla} f(x) - \widehat{\nabla} f(x_s) + \nabla f(x_s)$$

where

- $\widehat{\nabla}$ still denotes the minibatch gradient
- $x_s \in \mathbb{R}^n$ is a reference point
- $\nabla f(x_s) - \widehat{\nabla} f(x_s)$ is a control variate introduced to reduce variance

$g(x) \in \mathbb{R}^n$ is a stochastic gradient at $x \in \mathbb{R}^n$:

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

Some useful identities

recall the following two identities for random variables X, Y:

1. $\mathbb{E}\|X + Y\|^2 \leq 2\mathbb{E}\|X\|^2 + 2\mathbb{E}\|Y\|^2$
2. $\mathbb{E}\|X - \mathbb{E}[X]\|^2 \leq \mathbb{E}\|X\|^2$

Some useful identities

recall the following two identities for random variables X, Y:

1. $\mathbb{E}\|X + Y\|^2 \leq 2\mathbb{E}\|X\|^2 + 2\mathbb{E}\|Y\|^2$
2. $\mathbb{E}\|X - \mathbb{E}[X]\|^2 \leq \mathbb{E}\|X\|^2$

(exercise: prove these!)

SVRG reduces variance

variance of $g(x)$ depends on suboptimality of x and x_s

$$\begin{aligned}
\mathbb{E}\|g(x)\|^2 &= \mathbb{E}\|g(x) - \widehat{\nabla} f(x_\star) + \widehat{\nabla} f(x_\star)\|^2 \\
&= \mathbb{E}\|\widehat{\nabla} f(x) - \widehat{\nabla} f(x_\star) + \widehat{\nabla} f(x_\star) - \widehat{\nabla} f(x_s) + \nabla f(x_s)\|^2 \\
&\leq 2\mathbb{E}\|\widehat{\nabla} f(x) - \widehat{\nabla} f(x_\star)\|^2 \\
&\quad +2\mathbb{E}\|\widehat{\nabla} f(x_s) - \widehat{\nabla} f(x_\star) - \nabla f(x_s)\|^2 \\
&= 2\mathbb{E}\|\widehat{\nabla} f(x) - \widehat{\nabla} f(x_\star)\|^2 \\
&\quad +2\mathbb{E}\|\widehat{\nabla} f(x_s) - \widehat{\nabla} f(x_\star) - \mathbb{E}[\widehat{\nabla} f(x_s) - \widehat{\nabla} f(x_\star)]\|^2 \\
&= 2\mathbb{E}\|\widehat{\nabla} f(x) - \widehat{\nabla} f(x_\star)\|^2 + 2\mathbb{E}\|\widehat{\nabla} f(x_s) - \widehat{\nabla} f(x_\star)\|^2 \\
&= 4L[f(x) - f(x_\star) + f(x_s) - f(x_\star)]
\end{aligned}$$

hence $\mathrm{Var}(g(x)) \to 0$ as $f(x) \to f_\star$, $f(x_s) \to f_\star$

How to select x_s?

to ensure x, $x_s \to x_\star$ (and so $\mathrm{Var}(g(x)) \to 0$)

- update x_s as we make progress (so $f(x_s) \to f(x_\star)$)
- don't update too often, as computing $\nabla f(x_s)$ is expensive

SVRG algorithm

```
1. initialize at $x_0$ and set $x_s = x_0$
2. for $s = 0, \ldots, S$
    2.1 compute and store $\nabla f(x_s)$
    2.2 for $k = 0, \ldots, m-1$
            $x_{k+1}^{(s)} = x_k^{(s)} - \eta \left( \widehat{\nabla} f(x_k^{(s)}) - \widehat{\nabla} f(x_s) + \nabla f(x_s) \right)$
    2.3 select $x_{s+1}$ by uniformly sampling at random from
        $\{x_0^{(s)}, \ldots, x_{m-1}^{(s)}\}$
    2.4 set $x_0^{(s+1)} = x_{s+1}$
3. return $x_S$
```

- notice that $\mathbb{E} f_{s+1} = \frac{1}{m} \sum_{i=1}^{m} f(x_i^{(s)}$ (needed for proof)
- in practice, fine to set $f_{s+1} = f(x_m^{(s)}$ (last iterate)

SVRG convergence

Theorem

Run SVRG with $S = \mathcal{O}\left(\log\left(\frac{1}{\epsilon}\right)\right)$ outer iterations, $m = O(\kappa)$ inner iterations, and fixed stepsize $\eta = O(1/L)$. Then

$$\mathbb{E}[f(x_S)] - f(x_\star) \leq \epsilon.$$

The number of gradient oracle calls is bounded by

$$\mathcal{O}\left((n + \kappa b_g)\log\left(\frac{1}{\epsilon}\right)\right).$$

SVRG convergence

Theorem

Run SVRG with $S = \mathcal{O}\left(\log\left(\frac{1}{\epsilon}\right)\right)$ outer iterations, $m = O(\kappa)$ inner iterations, and fixed stepsize $\eta = O(1/L)$. Then

$$\mathbb{E}[f(x_S)] - f(x_\star) \leq \epsilon.$$

The number of gradient oracle calls is bounded by

$$\mathcal{O}\left((n + \kappa b_g)\log\left(\frac{1}{\epsilon}\right)\right).$$

- unlike SGD, SVRG converges linearly to the optimum
- when $\kappa = \mathcal{O}(n)$, SVRG makes only $\widetilde{\mathcal{O}}(nb_g)$ oracle calls, while GD makes $\widetilde{\mathcal{O}}(n^2)$ calls. so SVRG reduces the number of calls by n/b_g!

Proof of SVRG convergence

the argument may be broken down into two lemmas. We begin with the following one-step progress bound for outer-iteration s

Lemma (One-step lemma)

Suppose we are at iteration k of outer-iteration s. Then

$$\begin{aligned}\mathbb{E}_k\|x_{k+1}^{(s)} - x_\star\|^2 \le \|x_k^{(s)} - x_\star\|^2 + 2\eta\,(2\eta L - 1)\,[f(x_k^{(s)}) - f(x_\star)] \\ + 4\eta^2 L[f(x_s) - f(x_\star)]\end{aligned}$$

Proof of One-step lemma

$$\mathbb{E}_k\|x_{k+1}^{(s)} - x_\star\|^2 =$$

$$\begin{aligned}
&\|x_k^{(s)} - x_\star\|^2 - 2\eta\langle\nabla f(x_k), x_k - x_\star\rangle + \eta^2\mathbb{E}_k\|\ g(x_k)\|^2 \\
&\leq \|x_k^{(s)} - x_\star\|^2 - 2\eta\left(f(x_k) - f(x_\star)\right) + \eta^2\mathbb{E}_k\|\ g(x_k)\|^2 \\
&\leq \|x_k^{(s)} - x_\star\|^2 - 2\eta\left(f(x_k) - f(x_\star)\right) + \\
&4\eta^2 L[f(x) - f(x_\star) + f(x_s) - f(x_\star),]
\end{aligned}$$

where the first inequality uses convexity

$$f(x_k) - f(x_\star) \leq \langle\nabla f(x_k), x_k - x_\star\rangle$$

so, after rearranging

$$\begin{aligned}
\mathbb{E}_k\|x_{k+1}^{(s)} - x_\star\|^2 \leq \|x_k^{(s)} - x_\star\|^2 + 2\eta\left(2\eta L - 1\right)[f(x_k^{(s)}) - f(x_\star)] \\
+ 4\eta^2 L[f(x_s) - f(x_\star)]
\end{aligned}$$

Outer iteration contraction

the next step is show to the follow contraction result for the outer-iterations.

Lemma (Outer iteration contraction)

Suppose we are in outer iteration s. Then

$$\mathbb{E}_{0:s-1}[f(x_s)] - f(x_\star) \leq \left[\frac{1}{\eta\mu(1-2\eta L)m} + \frac{2}{1-2\eta L}\right](f(x_{s-1}) - f(x_\star)),$$

where $\mathbb{E}_{0:s-1}$ denotes the expectation conditioned on outer-iterations 0 through $s-1$.

Proof of outer iteration contraction

summing the inequality in the one-step lemma from $k = 0, \ldots, m-1$,

$$\sum_{k=1}^{m} \mathbb{E}_k \|x_{k+1}^{(s)} - x_\star\|^2 \leq \sum_{k=0}^{m-1} \|x_k^{(s)} - x_\star\|^2 +$$

$$2\eta m \left(2\eta L - 1\right) \frac{1}{m} \sum_{k=0}^{m-1} [f(x_k^{(s)}) - f(x_\star)] + 4m\eta^2 [f(x_{s-1}) - f(x_\star)].$$

Proof of outer iteration contraction

summing the inequality in the one-step lemma from $k = 0, \ldots, m-1$,

$$\sum_{k=1}^{m} \mathbb{E}_k \|x_{k+1}^{(s)} - x_\star\|^2 \leq \sum_{k=0}^{m-1} \|x_k^{(s)} - x_\star\|^2 +$$

$$2\eta m\left(2\eta L - 1\right)\frac{1}{m}\sum_{k=0}^{m-1}[f(x_k^{(s)}) - f(x_\star)] + 4m\eta^2[f(x_{s-1}) - f(x_\star)].$$

taking the expectation over all inner-iterations conditioned on outer-iterations 0 through $s-1$ + cancellation, yields

$$\mathbb{E}_{0:s-1}\|x_m^{(s)} - x_\star\|^2 \leq \|x_{s-1} - x_\star\|^2 +$$
$$+ 2\eta m\left(2\eta L - 1\right)\left(\mathbb{E}_{0:s-1}\left[f(x_s)\right] - f(x_\star)\right) + 4m\eta^2 L[f(x_{s-1}) - f(x_\star)].$$

Proof contd.

rearranging gives

$$\begin{aligned}&\mathbb{E}_{0:s-1}\|x_s - x_\star\|^2 + 2\eta m\,(1-2\eta L)\,(\mathbb{E}_{0:s-1}\,[f(x_s)] - f(x_\star))\\ &\leq 2\left(\frac{1}{\mu} + 2m\eta^2 L\right)[f(x_{s-1}) - f(x_\star)],\end{aligned}$$

where we used strong convexity of f

$$\|x_{s-1} - x_\star\|^2 \leq \frac{2}{\mu}\,(f(x_{s-1}) - f(x_\star))$$

hence (dropping $\mathbb{E}_{0:s-1}\|x_s - x_\star\|^2 \geq 0$)

$$\begin{aligned}&2\eta m\,(1-2\eta L)\,(\mathbb{E}_{0:s-1}\,[f(x_s)] - f(x_\star))\\ &\leq 2\left(\frac{1}{\mu} + 2m\eta^2 L\right)[f(x_{s-1}) - f(x_\star)],\end{aligned}$$

and so the claim follows by rearrangement

Finishing the proof

$$\mathbb{E}_{0:s-1}[f(x_{s+1})] - f(x_\star) \leq \left[\frac{1}{\eta\mu(1-2\eta L)m} + \frac{2}{1-2\eta L}\right](f(x_s) - f(x_\star))$$

setting $\eta = \frac{1}{10L}$ and $m = 20\frac{\mathcal{L}}{\mu}$, we find

$$\mathbb{E}_{0:s-1}[f(x_s)] - f(x_\star) \leq \frac{1}{2}(f(x_{s-1}) - f(x_\star))$$

now taking expectations over all outer iterations and recursing,

$$\mathbb{E}[f(x_s)] - f(x_\star) \leq \left(\frac{1}{2}\right)^s (f(x_0) - f(x_\star)),$$

which gives the theorem after setting $s = O(\log(1/\epsilon))$

Practical questions for SVRG

Practical questions for SVRG

Q: how to select update frequency m?

Practical questions for SVRG

Q: how to select update frequency m?
A: not obvious, as the dependence upon L/μ is loose. In practice, use m n/b_g update every 1–2 epochs

Practical questions for SVRG

Q: how to select update frequency m?
A: not obvious, as the dependence upon L/μ is loose. In practice, use m n/b_g update every 1–2 epochs
Q: how to choose step-size η?

Practical questions for SVRG

Q: how to select update frequency m?
A: not obvious, as the dependence upon L/μ is loose. In practice, use m n/b_g update every 1–2 epochs
Q: how to choose step-size η?
A: monitor convergence. theoretical step-size often too small

Practical questions for SVRG

Q: how to select update frequency m?
A: not obvious, as the dependence upon L/μ is loose. In practice, use $m \; n/b_g$ update every 1–2 epochs
Q: how to choose step-size η?
A: monitor convergence. theoretical step-size often too small
Q: does SVRG work for non-convex problems like deep learning?

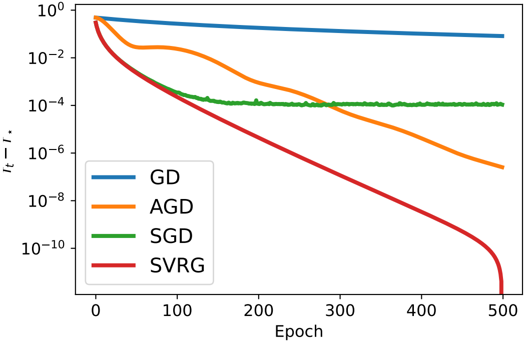
Practical questions for SVRG

Q: how to select update frequency m?
A: not obvious, as the dependence upon L/μ is loose. In practice, use m n/b_g update every 1–2 epochs
Q: how to choose step-size η?
A: monitor convergence. theoretical step-size often too small
Q: does SVRG work for non-convex problems like deep learning?
A: alas, no: variance reduction may worsen performance for nonconvex problems!

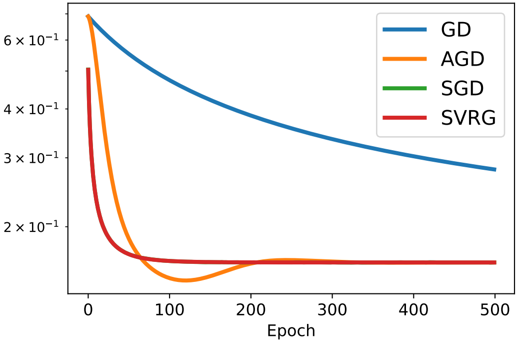
SVRG numerical performance

- revisit the same logistic regression example
- run SVRG with step-size $\eta = 4$
- update snapshot every epoch

Results: Optimization error



Results: Test loss



SVRG: Final comments

- variance reduction is a powerful tool for convex finite-sum optimization, as it delivers linear convergence
- SVRG has motivated the development of better (usually) variance reduced algorithms such as SAGA and Katyusha
- outside of finite-sum convex optimization, variance reduction hasn't proven to be terribly useful