ML and Optimization  Lecture 13

- Newton's Method: uses curvature information, so needs fewer iterations to reach high-quality solutions; each iteration is expensive.

- Stochastic Gradient Descent: touches fewer data points, so each iteration is faster, but convergence to high-quality solutions is slower.
Newton's Method

Gradient Descent Update:

\[ x_{t+1} = \arg\min_x f(x_t) + \langle \nabla f(x_t), x - x_t \rangle + \frac{1}{2} \lVert x - x_t \rVert^2 \]

\[ = x_t - \alpha_t \nabla f(x_t) \]

we model the curvature of \( f \) locally at \( x_t \) using

\[ \nabla^2 f(x_t) \approx \frac{1}{\alpha_t} I \]

Newton's method: use the optimal local quadratic approximation

\[ x_{t+1} = \arg\min_x f(x_t) + \langle \nabla f(x_t), x - x_t \rangle + \frac{1}{2} (x - x_t)^T \nabla^2 f(x_t) (x - x_t) \]

\[ = x_t - \nabla^2 f(x_t)^{-1} \nabla f(x_t) \]

\[ = m(x) \]

because \( x^* \) satisfies \( \nabla m(x^*) = 0 = \nabla f(x_t) + \nabla^2 f(x_t) (x - x_t) \)

"Unregularized Newton method/update"
We generally take our search direction
\[ g_t = - \left[ \nabla^2 f(x_t) \right]^{-1} \nabla f(x_t) \]
and judiciously choose our step size \( \alpha_t \) to ensure
\[ f(x_{t+1}) = f(x_t + \alpha_t g_t) \leq f(x_t) \]
c.g. with backtracking line search. This is called the
guarded Newton's method.

There are two phases to this algorithm:

- "damped phase" (when \( \alpha_t < 1 \)): suboptimality reduces
  linearly:
  \[ f(x_t) - f(x_*) \leq \varepsilon \text{ after } t = O\left(\log\left(\frac{1}{\varepsilon}\right)\right) \]

- "purely Newton phase" (when \( \alpha_t = 1 \)): quadratic convergence
  at each iteration, the number of digits of precision doubles!
  \[ f(x_t) - f(x_*) \leq \varepsilon \text{ after } t = O\left(\log\log\frac{1}{\varepsilon}\right) \]
**Ex.** Consider an ill-conditioned linear regression problem

\[ x^* = \arg\min_x \frac{1}{2} \|Ax-b\|^2 \]

\[ \Rightarrow \nabla^2 f(x) = AA^T \]

and

\[ \lambda_{\text{min}}(AA^T) \cdot I \leq AA^T \leq \lambda_{\text{max}}(AA^T) \cdot I \]

so

\[ K_f = \frac{\lambda_{\text{max}}(AA^T)}{\lambda_{\text{min}}(AA^T)} \]

**assume**

\[ \lambda_{\text{max}}(AA^T) \gg 1, \quad \lambda_{\text{min}}(AA^T) \ll 1 \]

and

\[ K_f = e^C \quad \text{where } C > 0 \text{ is large} \]
Option 1
Use GD with stepsize \( \frac{1}{\beta} \) where \( \beta = \lambda_{\text{max}}(A^TA) \gg 1 \)

\[ f(x_T) - f(x_*) \leq \varepsilon \quad \text{when} \quad T = O\left(\frac{1}{\varepsilon} \log\left(\frac{1}{\varepsilon}\right)\right) \gg 1 \]

Option 2
Use Newton's method with unit step size

\[ x_1 = x_0 - (A^TA)^{-1}A^T(Ax_0 - b) \]
\[ = x_0 - (x_0 - (A^TA)^{-1}A^Tb) \]
\[ = (A^TA)^{-1}A^Tb \]
\[ = x_0 \quad \text{one iteration of Newton's method} \]
\[ \Rightarrow \text{exact solution} \]

Takeaway: we converge faster if we account for curvature
Drawbacks to Newton's Method

- need to form $\nabla^2 f(x_t) \in \mathbb{R}^{d \times d}$
  and solve a linear system

In general this costs $O(d^2)$ memory and $O(d^3)$ operations to compute the Newton search direction
so when $d$ is large, Newton's method is infeasible.
GD can be expensive per iteration

Ex: GD to solve an ERM problem

\[ f(\omega) = \frac{1}{n} \sum_{i=1}^{n} l_i(\omega) \]

\[ \nabla f(\omega) = \frac{1}{n} \sum_{i=1}^{n} \nabla l_i(\omega) \]

\[ l_i(\omega) \text{ is loss on on } i\text{th datapoint}, \text{ e.g.} \]
\[ l_i(\omega) = \log(1 + \exp(-y_i(\omega^T x_i))) \]
\[ \text{or} \]
\[ l_i(\omega) = (\omega^T x_i - y_i)^2 \]

Per iteration to compute the full gradient

\[ \nabla f(\omega) \text{ requires } \textbf{touching } n \gg 1 \text{ data points!} \]

\[ \mathcal{O}(nd) \text{ operations} \]
**Stochastic Gradient Descent**

We want an algorithm that's cheaper per iteration than GD, preferably $O(d)$ per iteration.

Idea: notice that $\nabla f(\omega_t)$ is an approximate direction to search for decrease

$$f(\omega) \approx f(\omega_t) + \langle \nabla f(\omega_t), \omega - \omega_t \rangle + \frac{1}{2d_t} \| \omega - \omega_t \|^2_2$$

where

$$f(\omega) = \frac{1}{n} \sum_{i=1}^{n} l_i(\omega)$$

$$\Rightarrow \quad \nabla f(\omega) = \frac{1}{n} \sum_{i=1}^{n} \nabla l_i(\omega)$$
\( \nabla f(\omega) \) is already a sample average that approximates a population average.

Treat \( \nabla f(\omega) \) itself as a population average, and approximate it with a sample average.

- Select a random subset \( I \subset \{1, \ldots, n^3\} \) of size \( k \) and form an approximation

\[
\hat{g} = \frac{1}{k} \sum_{i \in I} \nabla l_i(\omega) \quad \text{stochastic gradient estimate}
\]

If we choose \( I \) appropriately, then

\[
\mathbb{E} \hat{g} = \nabla f(\omega) \quad \text{the stochastic gradient estimate is an unbiased estimator of the gradient}
\]

The set \( I \) is called a minibatch.
Minibatch Stochastic Gradient Descent

\[ x_\ast \in \arg\min_x f(x) \]

Choose a minibatch size \( k \)

for \( t = 0, \ldots, T-1 \)

- sample \( I \subseteq [n] \) of size \( k \) (independently between iterations)

- \( g_t \leftarrow \frac{1}{k} \sum_{i \in I} \nabla l_i(x_t) \)

- \( x_{t+1} = x_t - \alpha_t g_t \)

return:

- \( x_T \)
- or

\[- \frac{1}{T} \sum_{i=1}^{T} x_i \]
In practice, to efficiently use our training data, we use **Randomized Re-shuffling**:

In each epoch:
- randomly shuffle the training data
- everytime we sample a minibatch, use the first \( K \) unused points

Notice: there are \( \frac{N}{K} \) minibatches per epoch.
RP Minibatch SGD

Inputs: \( O \) of oracle, \( \alpha_t \) step sizes, \( k \) minibatch size, \( E \) epochs

Alg
\[ z_0 = x_0 \]
for \( e = 1, \ldots, E \)
\[ (x, y) \leftarrow \text{random reshuffle} \]
\[ x_0 = z_{e-1} \]
for \( t = 0, \ldots, \frac{D}{k} - 1 \)
\[ I \leftarrow \text{first } k \text{ unused samples in this epoch} \]
\[ g_t = \frac{1}{k} \sum_{i \in I} \nabla l_i(x_t) \]
\[ x_{t+1} = x_t - \alpha g_t \]
end
\[ z_e = x \frac{n}{K} \]
end
return \( z_E \)
How does Minibatch SGD perform?

- We don’t care about $k$, per se. Instead we assume
  \[ \mathbb{E}[\|g_t - \nabla f(x_t)\|^2] \leq \sigma^2 \]
  or
  \[ \mathbb{E}\|g_t\|^2 < \sigma^2 \]

- We always assume $\mathbb{E}[g_t | x_t] = \nabla f(x_t)$

- We will look at performance for $\mu$-strongly convex objectives to understand impact of noise level $\sigma^2$

- We will look at convergence guarantees for nonconvex optimization as well
Monotonicity

If \( f \) is a \( \mu \)-strongly convex function then

\[
\forall x, y \in \text{dom}(f): \quad \langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \mu \|x - y\|_2^2
\]

i.e. the gradient operator is \textit{monotone}.

Proof

\[
f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2} \|x - y\|_2^2
\]
\[
f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|x - y\|_2^2
\]

\[
\Rightarrow f(x) + f(y) \geq f(y) + f(x) + \langle \nabla f(y) - \nabla f(x), x - y \rangle + \frac{\mu}{2} \|x - y\|_2^2
\]

\[
\Rightarrow \langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \mu \|x - y\|_2^2
\]
Consider $f$ to be $\mu$-strongly convex to see the impact of the noisy gradient estimate on the convergence of $x_t$.

If $\alpha < \frac{1}{2\mu}$ and $E\|g\|_2^2 \leq \sigma^2$, then

$$E\|x_t - x_\ast\|_2^2 \leq (1 - 2\alpha \mu)^t \|x_0 - x_\ast\|_2^2 + \frac{\alpha \sigma^2}{2\mu}$$

so SGD converges to within a radius of the minimizer $x_\ast$ that is determined by the noise level $\sigma^2$.

Takeaway: to converge to $x_\ast$, unlike in GD, you must use a shrinking stepsize.
\[ \text{Prf} \]

\[ x_{tf1} - x_* = x_t - \alpha g_t - x_* \]
\[ = x_t - x_* - \alpha g_t \]

so

\[ \mathbb{E} \left\| x_{tf1} - x_* \right\|_2^2 = \mathbb{E} \left\| x_t - x_* - \alpha g_t \right\|_2^2 \]
\[ = \mathbb{E} \left[ \| x_t - x_* \|_2^2 - 2 \langle x_t - x_* , \alpha g_t \rangle + \alpha^2 \| g_t \|_2^2 \right] \]
\[ = \mathbb{E} \| x_t - x_* \|_2^2 + \alpha^2 \sigma^2 - 2 \alpha \mathbb{E} \left[ \langle x_t - x_* , \nabla f(x_t) \rangle \right] \]
\[ = \mathbb{E} \| x_t - x_* \|_2^2 + \alpha^2 \sigma^2 - 2 \alpha \mathbb{E} \langle x_t - x_* , \nabla f(x_t) - \nabla f(x) \rangle \]
Recall by monotonicity,
\[ \langle x_t - x^* \rangle, \nabla f(x_t) - \nabla f(x) \rangle \geq \mu \| x_t - x^* \|^2 \]
\[ \Rightarrow \quad \mathbb{E} \| x_{t+1} - x^* \|^2 \leq \mathbb{E} \| x_t - x^* \|^2 + \sigma^2 \Delta_t^2 - 2\mu \mathbb{E} \| x_t - x^* \|^2 \]
\[ = (1 - 2\mu \alpha) \mathbb{E} \| x_t - x^* \|^2 + \alpha^2 \sigma^2 \]

To solve this recursion, let

\[ \Delta_t = \mathbb{E} \| x_t - x^* \|^2 \] and \[ c = (1 - 2\mu \alpha) \] \[ b = \alpha^2 \sigma^2 \]

then

\[ \Delta_{t+1} \leq c \Delta_t + b \]
\[ \Rightarrow \quad \Delta_1 \leq c \Delta_0 + b \]
\[ \Delta_2 \leq c \Delta_1 + b \leq c^2 \Delta_0 + cb + b \]
\[ \Delta_3 \leq c \Delta_2 + b \leq c^3 \Delta_0 + c^2 b + cb + b \]
In general, \( \Delta_T \leq c^T \Delta_0 + \left( \sum_{i=0}^{T-1} c_i \right) b \)

\[ \leq c^T \Delta_0 + \frac{1}{1-c} b \]

\[ = c^T \Delta_0 + \frac{1}{\Delta_0} \alpha^2 \sigma^2 \]

\[ = c^T \Delta_0 + \frac{\alpha \sigma^2}{\Delta_0} \]

so

\[ E \| x_T - x^* \|_2^2 \leq (1 - 2\alpha \mu)^T E \| x_0 - x^* \|_2^2 + \frac{\alpha \sigma^2}{\Delta_0} \]
Proof of convergence in pure Newton phase (not covered in class)

Newton's method locally converges quadratically fast, even for nonconvex objectives.

Assume the Hessian of $f$ is $L$-Lipschitz continuous,

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L \|x-y\|_2$$

and strictly positive at $x^*$ where $\nabla f(x^*)=0$,

$$\nabla^2 f(x^*) \succ \mu I$$

for some $\mu > 0$.

Then, if the first iterate is within the basin of attraction,

$$\|x_0 - x^*\|_2 \leq \frac{\mu}{2L}$$

Newton's method is well-defined (i.e. $\nabla^2 f(x_k)$ is invertible), the iterates remain within the basin of attraction, and they converge to $x^*$ at a quadratic rate:

$$\|x_{k+1} - x^*\|_2 \leq \frac{1}{\mu} \|x_k - x^*\|_2^2$$
\textbf{Proof}

It suffices to show that if $x_t$ is in the basin of attraction then
\[ \|x_{t+1} - x^*\|_a \leq \frac{b}{\mu} \|x_t - x^*\|_a^2 \] (1)
because this implies
\[ \|x_{t+1} - x^*\|_a \leq \frac{b}{\mu} \cdot \frac{1}{2L} \|x_t - x^*\|_a \leq \frac{1}{2} \|x_t - x^*\|_a \leq \frac{\mu}{2L} \]

so $x_{t+1}$ is again in the basin of attraction.

To show (1), we compute
\[
x_{t+1} - x^* = x_t - \nabla f(x_t)^{-1} \nabla f(x_t) - x^*
\]
\[
= x_t - \nabla^2 f(x_t)^{-1} [\nabla f(x_t) - \nabla f(x^*)] - x^*
\]
\[
= \nabla^2 f(x_t)^{-1} [\nabla f(x^*) - \nabla f(x_t) - \nabla^2 f(x_t)(x^* - x_t)]
\]
Now we expect two things intuitively:

1) because \( x_t \) is close to \( x^* \) and the Hessian is Lipschitz, \( \nabla^2 f(x_t) \) should also be invertible (so the method is well-defined)

2) by a T.S. argument, because \( x_t \) is close to \( x^* \),

\[
\nabla f(x^*) - \nabla f(x_t) - \nabla^2 f(x_t)(x^* - x_t) = O(\|x_t - x^*\|_2^2)
\]

together these argue that we can expect

\[
\|x_{t+1} - x^*\|_2 \leq O(\|x_t - x^*\|_2^2)
\]

First, let's quantify the first point. In general, if \( A, B \) are matrices,

\[
-\|A - B\|_2 \cdot I \leq A - B \leq \|A - B\|_2 \cdot I
\]

which implies

\[
A \geq B - \|A - B\|_2 \cdot I
\]
Apply this fact to see that
\[ \nabla^2 f(x_t) \succeq \nabla^2 f(x^*) - \| \nabla f(x_t) - \nabla f(x^*) \|_2 \cdot I \]
\[ \Rightarrow \mu \cdot I - I \| x_t - x^* \|_2 \cdot I \]
\[ \Rightarrow \mu \cdot I - \frac{1}{2\lambda} \frac{\mu}{\lambda} \cdot I \]
\[ = \frac{\mu}{2} \cdot I \]

so indeed \( \nabla^2 f(x_t) \) is invertible, and \( \nabla^2 f(x_t)^{-1} \leq \frac{2}{\mu} \cdot I \).

Next, we quantify the second claim with a T.S. argument.
Define
\[ g(u) = \nabla f(x_t + u(x^* - x_t)) \], so
\[ g(u) - g(0) = \nabla f(x^*) - \nabla f(x_t) = \int_0^1 \nabla^2 f(x_t + u(x^* - x_t)) \cdot (x^* - x_t) \, du \]
\[ \nabla f(x^*) - \nabla f(x_t) - \nabla^2 f(x_t)(x^* - x_t) = \int_0^1 \left[ \nabla^2 f(x_t + u(x^* - x_t)) \right] (x^* - x_t) \, du, \]

so

\[ \| \nabla f(x^*) - \nabla f(x_t) - \nabla^2 f(x_t)(x^* - x_t) \|_2 \leq \int_0^1 L \| u(x^* - x_t) \|_2 \| x^* - x_t \|_2 \, du \]

= \frac{L}{2} \| x^* - x_t \|_2^2
Putting the pieces together,

\[ x_{t+1} - x^* = \nabla f(x_t)^{-1} \left[ \nabla f(x^*) - \nabla f(x_t) - \nabla^2 f(x_t)(x^* - x_t) \right] \]

so

\[ \| x_{t+1} - x^* \|_2 \leq \| \nabla f(x_t)^{-1} \|_2 \cdot \| \nabla f(x^*) - \nabla f(x_t) - \nabla^2 f(x_t)(x^* - x_t) \|_2 \]

\[ \leq \frac{\alpha}{\mu} \cdot \frac{1}{2} \| x_t - x^* \|_2^2 \]

\[ = \frac{1}{\mu} \cdot \| x_t - x^* \|_2^2 , \]

as claimed.