Subgradient Descent Algorithm

\[ x^* = \arg\min_x f(x) \]

**Inputs:**
- \( x_0 \) starting point
- \( T \) number of iterations
- \( \alpha_0, \ldots, \alpha_{T-1} \) step sizes

**Algorithm:**

\[
\begin{align*}
&\text{for } t = 0, \ldots, T-1: \\
&\quad g_t \leftarrow \text{a vector in } \partial f(x_t) \\
&\quad x_{t+1} = x_t - \alpha_t g_t
\end{align*}
\]

**Return**

\[ \hat{x} = \frac{1}{T} \sum_{t=1}^{T} x_t \Rightarrow f(\hat{x}) - f(x^*) \leq \left( \frac{\log T}{T} \right) \]

when \( \alpha_t = \frac{1}{\mu t} \)

\( \mu \)-strong convexity parameter
Problem: gradient descent/subgradient descent is expensive if \( n \) is large

b/c the cost of computing \( \nabla f(x_t) \) or \( g_t \in \partial f(x_t) \) is \( O(n) \). Typically cost \( O(nd) \)

**Ex:**

\[
f(\omega) = \frac{1}{2} \| X\omega - y \|_2^2
\]

\[
\Rightarrow \nabla f(\omega) = X^T(X\omega - y)
\]

and if we can't store \( X^TX \) (b/c if we can, we can compute \( \nabla f(\omega) \) in \( O(d^2) \) time)

then cost of computing \( \nabla f(\omega) \) is

\[
= \text{cost of computing } z = X\omega - y, \quad O(nd) \Rightarrow O(nd)
\]

+ cost of computing \( X^Tz \), \( O(nd) \)
We want a faster algorithm, one where the search direction is cheaper to compute, e.g. $O(d)$

Idea: notice that $\nabla f(\omega_t)$ as a search direction is already approximate:

$$f(\omega) \approx f(\omega_t) + \langle \nabla f(\omega_t), \omega - \omega_t \rangle$$

so we may as well approximate $\nabla f(\omega_t)$ itself:

if

$$f(\omega) = \frac{1}{n} \sum_{i=1}^{n} l(\omega^T x_i, y_i)$$

then

$$\nabla f(\omega) = \frac{1}{n} \sum_{i=1}^{n} x_i l'(\omega^T x_i, y_i)$$
Idea: select a random subset \( I \subset [n] \) of size \( k \) and choose an approximation

\[
g = \frac{1}{k} \sum_{i \in I} x_i l'(\omega^T x_i > y_i)
\]

\[\Rightarrow \mathbb{E}g = \nabla f(\omega)\]

The set \( I \) is called a minibatch
Stochastic gradient descent

\[ x^* = \arg\min_x \mathcal{F}(x) \]

Choose a batch size \( k \)

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

- sample \( I \) of size \( k \)

\[ g_t = \frac{1}{k} \sum_{i \in I} \nabla l_i(x_t) \]

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

return \( \bar{x} = \frac{1}{T} \sum_{t=1}^{T} x_t \)
In practice:
- randomize the order of your training data
  \((x_1, y_1), \ldots, (x_n, y_n)\) was original training data
  \[\text{randomly permute these examples to get your new training data}\]
  \((x_1', y_1'), \ldots, (x_n', y_n')\)

- every time you sample a minibatch use the first \(k\) points you haven't already used

\[t = 0\quad t = 1\quad t = \frac{n}{k}\]
— once you’ve touched all of your data
(this is called an ‘epoch’, after \( \frac{n}{k} \) iterations
if minibatch size = \( k \))
then reshuffle your training data and
repeat the same process for sampling
your minibatches
Stochastic subgradient descent (Non-smooth case)

for $t = 0, \ldots, T-1$:
- sample $I$
- $g_t \leftarrow \frac{1}{|I|} \sum_{i \in I} g_i, t$
- $x_{t+1} \leftarrow x_t - \alpha_t g_t$

return $x = \frac{1}{T} \sum_{i=1}^{T} x_i$

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

$l(\omega^T x_i, y_i)$

or

$f(x) = x \log x + e^{-x}$

$-x$

$= \ell_1(x) + \ell_2(x) + \ell_3(x)$

are valid applications

(Notice stochastic gradient descent is exactly stochastic subgradient descent applied to smooth functions)
Analysis of stochastic gradient descent (SGD)

Assumptions:

1) $f$ is strongly convex:

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

"$f$ is curved upwards" \quad $f$ is $\mu$-strongly convex
b) the estimate of the gradient $\nabla f(x_t)$ given by $g_t$ satisfies two properties:

1) $E[g_t | x_0, \ldots, x_t] = \nabla f(x_t)$ "unbiased"

2) $E[\|g_t\|_2^2] \leq B^2$ "finite variance" condition
With those assumptions we want to show
\[ f(x_t) - f(x^*) \to 0 \quad \text{(and get the rate)} \]

First
\[
\| x_t - x^* \|^2 = \| (x_{t-1} - \alpha_{t-1}g_{t-1}) - x^* \|^2 \\
= \| x_{t-1} - x^* \|^2 + \alpha_{t-1}^2 \| g_{t-1} \|^2 \\
- 2 \langle x_{t-1} - x^* , \alpha_{t-1}g_{t-1} \rangle
\]

\[ \mathbb{E} \left[ \| x_t - x^* \|^2 \bigg| x_0, \ldots, x_{t-1} \right] \]
\[ = \| x_{t-1} - x^* \|^2 + \alpha_{t-1}^2 \mathbb{E} \left[ \| g_{t-1} \|^2 \bigg| x_0, \ldots, x_{t-1} \right] \\
- 2 \langle x_{t-1} - x^* , \alpha_{t-1} \mathbb{E} [ g_{t-1} | x_0, \ldots, x_{t-1} ] \rangle \]
Recall \( \mathbb{E} \left[ g_{t-1} \mid x_0, \ldots, x_{t-1} \right] = \nabla f(x_{t-1}) \geq 0 \)

\[
\mathbb{E} \left[ \| x_t - x^* \|^2 \mid x_0, \ldots, x_{t-1} \right]
= \| x_{t-1} - x^* \|^2 + \alpha_{t-1}^2 \mathbb{E} \left[ \| g_{t-1} \|^2 \mid x_0, \ldots, x_{t-1} \right]
+ 2 \alpha_{t-1} \langle x^* - x_{t-1}, \nabla f(x_{t-1}) \rangle
\]

Now note by strong convexity of \( f \) that

\[
f(x^*) \geq f(x_{t-1}) + \langle \nabla f(x_{t-1}), x^* - x_{t-1} \rangle
+ \frac{\alpha}{2} \| x_{t-1} - x^* \|^2
\]

\[
\Rightarrow \quad f(x^*) - f(x_{t-1}) - \frac{\alpha}{2} \| x_{t-1} - x^* \|^2 \geq \langle \nabla f(x_{t-1}), x^* - x_{t-1} \rangle
\]
\[ \begin{align*}
\mathbb{E}
& \left[ \| x_t - x^* \|^2 \mid x_0, \ldots, x_{t-1} \right] \\
& \leq \| x_{t-1} - x^* \|^2 \\
& + \alpha_{t-1}^2 \mathbb{E} \left[ \| g_{t-1} \|^2 \mid x_0, \ldots, x_{t-1} \right] \\
& + 2\alpha_{t-1} \left( f(x^*) - f(x_{t-1}) - \frac{\mu}{2} \| x_{t-1} - x^* \|^2 \right) \\
\end{align*} \]

- This is negative (b/c \( f(x^*) \) is minimum function value).
- So if it is large enough to counteract gradient length squared, then we expect \( x_t \) to be closer to \( x^* \) than \( x_{t-1} \).
Now we use the fact the subgradients are bounded in size
\[ \mathbb{E}[\|g_{t-1}\|_2^2 \mid x_0, \ldots, x_{t-1}] \leq B^2 \]

\[ \Rightarrow \mathbb{E} [\|x_t - x^*\|_2^2 \mid x_0, \ldots, x_{t-1}] \leq \|x_{t-1} - x^*\|_2^2 + \lambda_{t-1}^2 B^2 \\
+ 2\alpha_{t-1} (f(x^*) - f(x_{t-1}) - \frac{\mu}{2} \|x_{t-1} - x^*\|_2^2) \]

Now take the expectations w.r.t. \( x_0, \ldots, x_{t-1} \):

\[ \mathbb{E} \|x_t - x^*\|_2^2 \leq (1 - \alpha_{t-1} \mu) \mathbb{E} \|x_{t-1} - x^*\|_2^2 \\
+ \lambda_{t-1}^2 \mathbb{E} B^2 \\
+ 2\alpha_{t-1} \mathbb{E} [f(x^*) - f(x_{t-1})] \]
\[ E \left[ f(x_{t-1}) - f(x^*) \right] \leq \alpha_{t-1} \frac{B^2}{2} + \left( \frac{\alpha_{t-1} - \mu}{\mu t} \right) E \| x_{t-1} - x^* \|_2^2 - \frac{\alpha_{t-1}^2}{2} E \| x_t - x^* \|_2^2 \]

Now we carefully choose our stepsize, \( \alpha_{t-1} = \frac{1}{\mu t} \), then

\[ E \left[ f(x_{t-1}) - f(x^*) \right] \leq B^2 \frac{1}{2 \mu t} + \frac{\mu (t-1)}{2} E \| x_{t-1} - x^* \|_2^2 - \mu t E \| x_t - x^* \|_2^2 \]

Why did we choose \( \alpha_{t-1} = \frac{1}{\mu t} \)?
So we can get a telescoping sum by adding these inequalities over $t=1, \ldots, T$:

$$\sum_{t=1}^{T} \mathbb{E} \left[ f(x_{t-1}) - f(x^*_t) \right] \leq \frac{B^2}{2\mu} \sum_{t=1}^{T} \frac{1}{t} + \frac{O - \frac{MT}{2} \| \mathbf{w}_T - \mathbf{w}^* \|^2}{2}$$

Now averaging over $T$,

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left[ f(x_{t-1}) \right] \leq f(x^*_T) + \frac{B^2}{2\mu T} \log(T)$$

since

$$\frac{1}{T} \sum_{t=1}^{T} \frac{1}{t} \leq \log(T)$$

Now apply Jensen's inequality to the expression on the left-hand:

$$f \left( \frac{1}{T} \sum_{t=1}^{T} x_{t-1} \right) \leq \frac{1}{T} \sum_{t=1}^{T} f(x_{t-1})$$

where $f$ is convex and consider

c.v. $X = x_i$ for $i = 0, \ldots, T-1$ w.p. $\frac{1}{T}$
This gives that \( \bar{x} = \frac{1}{T} \sum_{i=1}^{T} x_{i-1} \) satisfies
\[
f(\bar{x}) \leq \frac{1}{T} \sum_{i=1}^{T} \mathbb{E}[f(x_{i-1})] \leq f(x_{*}) + \frac{B^2}{2\mu T} \log(T)
\]
so to get to \( \epsilon \) suboptimality, take \( T \) large enough that
\[
\frac{B^2}{2\mu T} \log(T) \leq \epsilon
\]
then will have
\[
f(\bar{x}) \leq f(x_{*}) + \epsilon
\]
This says that if subgradients are large \( (B \gg 1) \) or curvature is small \( (\mu \ll 1) \) then you will take longer to converge w/ stepsize choice \( \alpha_{t-1} = \frac{1}{\mu t} \).
N.B.: We typically don't want to or can't store all the $x_t$, so we maintain $\hat{x}$ as a running average. Note that if we define

$$\hat{x} = \frac{1}{T} \sum_{t=1}^{T} x_{t-1}$$

then

$$\hat{x}_{T+1} = \frac{1}{T+1} \left[ \sum_{t=1}^{T} x_{t-1} + x_T \right]$$

$$= \frac{T}{T+1} \left[ \frac{1}{T} \sum_{t=1}^{T} x_{t-1} \right] + \frac{1}{T+1} x_T$$

$$= \frac{T}{T+1} \hat{x} + \frac{1}{T+1} x_T$$

so we can simply update a running average as we go:

$$\hat{x} = 0 \text{ initially, then } \hat{x}_t = \frac{t}{t+1} \hat{x}_{t-1} + \frac{1}{t+1} x_t \text{ for } t=0, \ldots, T-1$$

then return $\hat{x}$ after $t=T-1$
NB: This analysis goes through without changes for convergence rate of subgradient descent

**Stochastic Subgradient Descent** (with Polyak-Ruppert averaging)

\[ \hat{x}_0 = 0 \]
for \( t = 0, \ldots, T-1 \):

\[ g_t \leftarrow \text{an estimate of a vector in } \nabla f(x_t) \]

\[ x_{t+1} \leftarrow x_t - \alpha_t g_t \]

\[ x \leftarrow \frac{t}{t+1} \hat{x} + \frac{x_{t+1}}{t+1} \]

return \( \hat{x} \)

or an estimate of \( \nabla f(x_t) \) if \( f \) is differentiable
NB: Another approach is to maintain

$$x_{\text{best}} = \arg\min_{x_t} f(x_t)$$

over the course of the optimization and return $x_{\text{best}}$.

By definition:

$$f(x_{\text{best}}) \leq \frac{1}{T} \sum_{t=1}^{T} f(x_{t-1})$$

$$\mathbb{E}f(x_{\text{best}}) \leq \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}f(x_{t-1})$$

$$\leq \frac{B^2 \log(T)}{2\mu T}$$

Note that both sides of this inequality are random variables because the $x_t$ depend on the random subgradient estimates.

Q: Why might we not track $x_{\text{best}}$ in a setting where we choose to use SGD?