Subgradient Descent Algorithm (Unconstrained non-smooth cvx optim)

$$x_* = \operatorname*{argmin}_x f(x)$$

Inputs: x_0, starting point
T, number of iterations
$\alpha_0, \dots, \alpha_{T-1}$ stepsizes

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

Algorithm:

for $t = 0, \dots, T-1$:

$g_t \leftarrow$ a vector in $\partial f(x_t)$

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

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

when $\alpha_t = \frac{1}{\mu t}$ μ - strong cvxty param

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 $O(d^2)$ times)

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

= cost of computing $z = X\omega - y$, $O(nd)$

+ cost of computing X^Tz, $O(nd)$

$\Rightarrow O(nd)$

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

$$\Rightarrow \quad \nabla f(\omega) = \frac{1}{n} \sum_{i=1}^{n} x_i l'(\omega^T x_i, y_i)$$

Idea: select a random subset $I \subset [n]$ of size k and choose an approximation

$$g = \frac{1}{k} \sum_{i \in I} x_i \ell'(w^T x_i, y_i)$$

$$\Rightarrow \mathbb{E} g = \nabla f(w)$$

The set I is called a <u>minibatch</u>

Stochastic gradient descent

(Unconstrained Smooth cvx optim)

$$x_* = \operatorname{argmin}_x f(x)$$

→ where $f(x) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(x)$

Choose a batch size k

for $t = 0, \dots, T-1$

- sample I of size k
- $g_t = \frac{1}{k}\sum_{i \in I} \nabla \ell_i(x_t)$ ← much cheaper than computing $\nabla f(x_t)$ by a factor of $\frac{k}{n}$
- $x_{t+1} = x_t - \alpha_t g_t$

return

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

In practice:

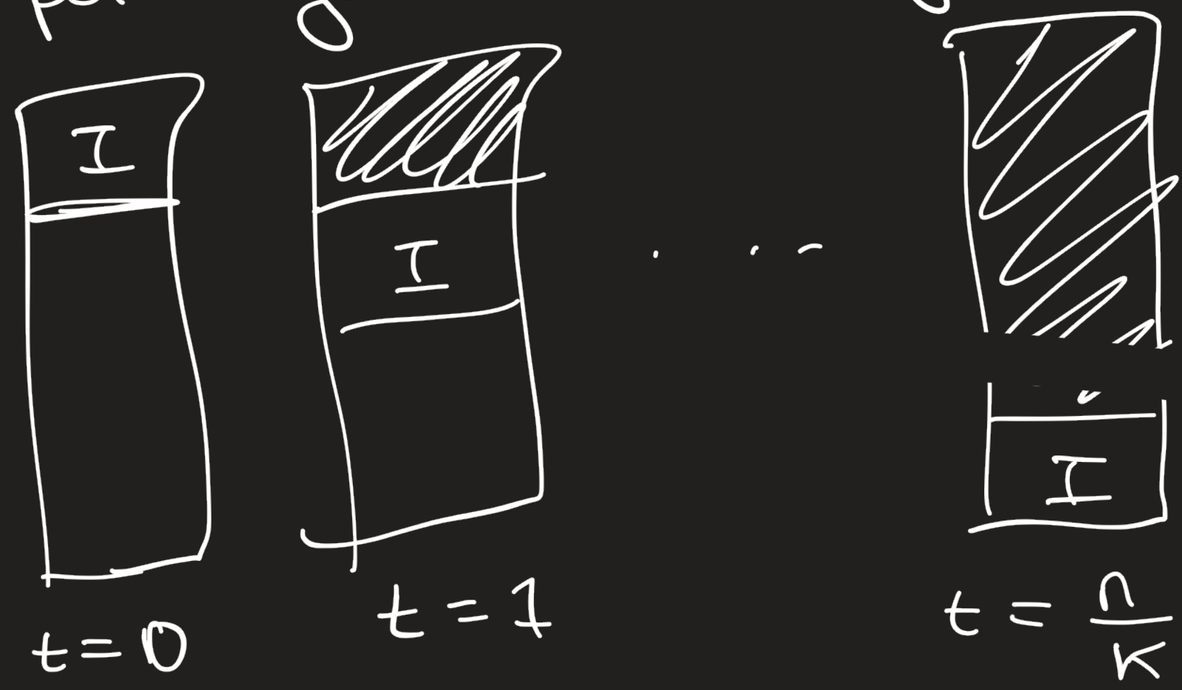
— randomize the order of your training data

$(x_1, y_1), \ldots, (x_n, y_n)$ was original training data

↓ randomly permute these examples to get your new training data

$(x_1, y_1) \quad \ldots \quad (x_n, y_n)$

— every time you sample a minibatch use the first k points you haven't already used



- 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 cvx optim)

for $t = 0, \dots, T-1$:

- sample I
- $g_t \leftarrow \frac{1}{K} \sum_{i \in I} g_{i,t}$

where $g_{i,t} \in \partial l_i(x_t)$

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

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

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

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

or

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

$$-x$$

$$= l_1(x) + l_2(x) + l_3(x)$$

are valid applications

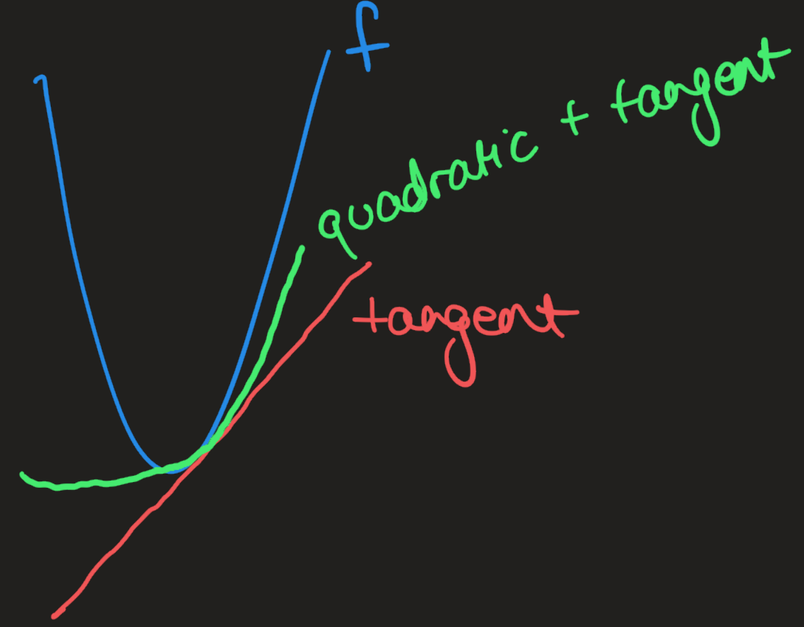
(Notice stochastic gradient descent is exactly stochastic subgradient descent applied to smooth functions)

Analysis of stochastic gradient descent (SGD)

Assumptions:

1) f is strongly convex:

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



"f is curved upwards" f is μ-strongly convex

b) the estimate of the gradient $\nabla f(x_t)$ given by g_t satisfies two properties:

1) $\mathbb{E}[g_t \mid x_0, \dots, x_t] = \nabla f(x_t)$ "unbiased"

2) $\mathbb{E}\{\|g_t\|_2^2\} \leq B^2$ "finite variance" condition

With those assumptions we want to show

$$f(x_t) - f(x_*) \to 0 \qquad \text{(and get the rate)}$$

First

$$\underline{\| x_t - x^* \|_2^2} = \| (x_{t-1} - \alpha_{t-1} g_{t-1}) - x^* \|_2^2$$

$$= \underline{\| x_{t-1} - x_* \|_2^2} + \alpha_{t-1}^2 \| g_{t-1} \|_2^2$$

$$- 2 \langle x_{t-1} - x_* , \alpha_{t-1} g_{t-1} \rangle$$

$$\mathbb{E}\left[\| x_t - x^* \|_2^2 \mid x_0, \dots, \underline{x_{t-1}} \right]$$

$$= \| x_{t-1} - x_* \|_2^2 + \alpha_{t-1}^2 \, \mathbb{E}\left[\| g_{t-1} \|_2^2 \mid x_0, \dots, x_{t-1} \right]$$

$$- 2 \langle x_{t-1} - x_* , \alpha_{t-1} \, \mathbb{E}\left[g_{t-1} \mid x_0, \dots, x_{t-1} \right] \rangle$$

Recall $\mathbb{E}[g_{t-1} | x_0, \cdots, x_{t-1}] = \nabla f(x_{t-1})$, so

$$\mathbb{E}\left[\|x_t - x^*\|^2 \mid x_0, \cdots, x_{t-1}\right]$$

$$= \|x_{t-1} - x_*\|_2^2 + \alpha_{t-1}^2 \, \mathbb{E}\left[\|g_{t-1}\|_2^2 \mid x_0, \cdots, x_{t-1}\right]$$

$$+ 2\alpha_{t-1} \langle x_* - x_{t-1}, \nabla f(x_{t-1}) \rangle$$

Now note by strong convexity of f it

$$f(x_*) \geq f(x_{t-1}) + \langle \nabla f(x_{t-1}), x^* - x_{t-1} \rangle + \frac{\mu}{2} \|x_{t-1} - x_*\|_2^2$$

$$\Rightarrow \quad f(x_*) - f(x_{t-1}) - \frac{\mu}{2} \|x_{t-1} - x_*\|_2^2 \geq \langle \nabla f(x_{t-1}), x^* - x_{t-1} \rangle$$

$$\Rightarrow \mathbb{E}\left[\|x_t - x^*\|_2^2 \mid x_0, \dots, x_{t-1}\right] \leq \|x_{t-1} - x_*\|_2^2$$

$$+ \alpha_{t-1}^2 \, \mathbb{E}\left[\|g_{t-1}\|_2^2 \mid x_0, \dots, x_{t-1}\right]$$

$$+ 2\alpha_{t-1}\left(f(x^*) - f(x_{t-1}) - \frac{\mu}{2}\|x_{t-1} - x^*\|_2^2\right)$$

this is negative (b/c $f(x_*)$ is minimum function value)

so if 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

$$\mathbb{E}\left[\|g_{t-1}\|_2^2 \mid x_0, \dots, x_{t-1}\right] \leq B^2$$

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

Now take the expectations w.r.t. $x_0, \dots, x_{t-1}$:

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

$$\Rightarrow$$

$$\mathbb{E}\left[f(x_{t-1}) - f(x^*)\right] \leq$$

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

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

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

Why did we choose $\alpha_{t-1} = \frac{1}{\mu t}$?

So we can get a telescoping sum by adding these inequalities over $t = 1, \dots, T$:

$$\sum_{t=1}^{T} \mathbb{E}\left[f(x_{t-1}) - f(x_*)\right] \leq \frac{B^2}{2\mu} \sum_{t=1}^{T} \frac{1}{t} + \boxed{0 - \frac{\mu T}{2} \mathbb{E}\|w_T - w^*\|_2^2}$$

Now averaging over T,

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

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

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

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

f is cvx and consider r.v. $X = x_i$ for $i = 0, \dots, T-1$ w.p. $\frac{1}{T}$

This gives that $\hat{x} = \frac{1}{T}\sum_{i=1}^{T} x_{i-1}$ satisfies

called Polyak-Ruppert averaging

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

so to get to ε-suboptimality, take T large enough that

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

then will have

$$f(\hat{x}) \leq f(x_*) + \varepsilon$$

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 $\alpha_{t-1} = \frac{1}{\mu t}$

NB: We typically don't want to or can't store all the x_t's so we maintain $\hat{x}$ as a running average. Note that

if we define $\hat{x} \equiv \hat{x}_T \equiv \frac{1}{T}\sum_{t=1}^{T} x_{t-1}$

then

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

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

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

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

$\hat{x} = 0$ initially, then $\hat{x} = \frac{t}{t+1}\hat{x} + \frac{1}{t+1}x_{t+1}$ for $t = 0, \dots, T-1$

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)

$\hat{x} = 0$,

for $t = 0, \dots, T-1$:

$g_t \leftarrow$ an estimate of a vector in $\partial f(x_t)$

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

$\hat{x} \leftarrow \frac{t}{t+1}\hat{x} + \frac{x_{t+1}}{t+1}$

return $\hat{x}$

or an estimate of $\nabla f(x_t)$ if f is differentiable

NB: Another approach is to maintain

$$x_{best} = \underset{x_t}{\operatorname{argmin}}\, f(x_t)$$

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

By definition:

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

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

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

(note that both sides of this ineq are r.v.s because the x_t depend on the random subgrad estimates)

Q: why might we not track x_{best} in a setting where we choose to use sgd?