CMPT 409/981: Optimization for Machine Learning

Lecture 14

Sharan Vaswani

October 29, 2024

Recap

- For G-Lipschitz functions, for all $x, y \in \mathcal{D}$, $|f(y) - f(x)| \leq G \|x - y\|$. Equivalently, $\|\nabla f(w)\| \leq G$. *Example*: Hinge loss: $f(w) = \max\{0, 1 - y\langle w, x\rangle\}$ is $\|y\, x\|$-Lipschitz.
- **Subgradient**: For a convex function f, the subgradient of f at $x \in \mathcal{D}$ is a vector g that satisfies the inequality for all y, $f(y) \geq f(x) + \langle g, y - x\rangle$. *Example*: For $f(w) = |w|$ at $w = 0$, vectors with slope in $[-1, 1]$ and passing through the origin are subgradients.
- **Subdifferential**: The set of subgradients of f at $w \in \mathcal{D}$ is referred to as the subdifferential and denoted by $\partial f(w)$. Formally, $\partial f(w) = \{g | \forall y \in \mathcal{D}; f(y) \geq f(w) + \langle g, y - w\rangle\}$.
- For unconstrained minimization of convex, non-smooth functions, w^* is the minimizer of f iff $0 \in \partial f(w^*)$ (this is analogous to the smooth case).
- For Lipschitz functions, we cannot relate the subgradient norm to the suboptimality in the function values. *Example*: For $f(w) = |w|$, for all $w > 0$ (including $w = 0^+$), $\|g\| = 1$.
- **Projected Subgradient Descent**: $w_{k+1} = \Pi_{\mathcal{D}}[w_k - \eta_k g_k]$, where $g_k \in \partial f(w_k)$.
- Since the sub-gradient norm does not necessarily decrease closer to the solution, to converge to the minimizer, we need to explicitly decrease the step-size.

Minimizing convex, Lipschitz functions using Subgradient Descent

For simplicity, let us assume that $\mathcal{D} = \mathbb{R}^d$ and analyze the convergence of subgradient descent.

Claim: For G-Lipschitz, convex functions, for $\eta > 0$, T iterations of subgradient descent with $\eta_k = \eta/\sqrt{k+1}$ converges as follows, where $\bar{w}_T = \sum_{k=0}^{T-1} w_k/T$,

$$f(\bar{w}_T) - f(w^*) \leq \frac{1}{\sqrt{T}} \left[\frac{\|w_0 - w^*\|^2}{2\eta} + \frac{G^2\eta\,[1 + \log(T)]}{2}\right].$$

Proof: Similar to the previous proofs, using the update $w_{k+1} = w_k - \eta_k g_k$ where $g_k \in \partial f(w_k)$,

$$\|w_{k+1} - w^*\|^2 = \|w_k - w^*\|^2 - 2\eta_k \langle g_k, w_k - w^* \rangle + \eta_k^2 \|g_k\|^2$$

$$\leq \|w_k - w^*\|^2 - 2\eta_k [f(w_k) - f(w^*)] + \eta_k^2 \|g_k\|^2$$

(Definition of subgradient with $x = w_k$, $y = w^*$)

$$\leq \|w_k - w^*\|^2 - 2\eta_k [f(w_k) - f(w^*)] + \eta_k^2 G^2$$

(Since f is G-Lipschitz)

$$\implies \eta_k [f(w_k) - f(w^*)] \leq \frac{\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2}{2} + \frac{\eta_k^2 G^2}{2}$$

Minimizing convex, Lipschitz functions using Subgradient Descent

Recall that $\eta_k[f(w_k) - f(w^*)] \leq \frac{\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2}{2} + \frac{\eta_k^2 G^2}{2}$,

$$\implies \eta_{\min} \sum_{k=0}^{T-1} [f(w_k) - f(w^*)] \leq \sum_{k=0}^{T-1} \left[\frac{\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2}{2} \right] + \frac{G^2}{2} \sum_{k=0}^{T-1} \eta_k^2$$

$$\leq \frac{\|w_0 - w^*\|^2}{2} + \frac{G^2}{2} \sum_{k=0}^{T-1} \eta_k^2$$

$$\implies \frac{\eta}{\sqrt{T}} \sum_{k=0}^{T-1} [f(w_k) - f(w^*)] \leq \frac{\|w_0 - w^*\|^2}{2} + \frac{G^2 \eta^2}{2} \sum_{k=1}^{T} \frac{1}{k} \qquad \text{(Since } \eta_k = \eta / \sqrt{k+1}\text{)}$$

$$\implies \frac{\sum_{k=0}^{T-1} [f(w_k) - f(w^*)]}{T} \leq \frac{1}{\sqrt{T}} \left[\frac{\|w_0 - w^*\|^2}{2\eta} + \frac{G^2 \eta \, [1 + \log(T)]}{2} \right]$$

$$\implies f(\bar{w}_T) - f(w^*) \leq \frac{1}{\sqrt{T}} \left[\frac{\|w_0 - w^*\|^2}{2\eta} + \frac{G^2 \eta \, [1 + \log(T)]}{2} \right]$$

(Using Jensen's inequality on the LHS, and by definition of $\bar{w}_T$.)

Minimizing convex, Lipschitz functions using Subgradient Descent

Recall that $f(\bar{w}_T) - f(w^*) \leq \frac{1}{\sqrt{T}} \left[\frac{\|w_0 - w^*\|^2}{2\eta} + \frac{G^2 \eta \, [1 + \log(T)]}{2}\right]$. The above proof works for any value of η and we can modify the proof to set the "best" value of η.

For this, let us use a constant step-size $\eta_k = \eta$. Following the same proof as before,

$$\eta_{\min} \sum_{k=0}^{T-1} [f(w_k) - f(w^*)] \leq \frac{\|w_0 - w^*\|^2}{2} + \frac{G^2}{2} \sum_{k=0}^{T-1} \eta_k^2$$

$$\implies \sum_{k=0}^{T-1} [f(w_k) - f(w^*)] \leq \frac{\|w_0 - w^*\|^2}{2\eta} + \frac{G^2 T \eta}{2} \qquad \text{(Since } \eta_k = \eta\text{)}$$

Setting $\eta = \frac{\|w_0 - w^*\|}{G\sqrt{T}}$, dividing by T and using Jensen's inequality on the LHS,

$$f(\bar{w}_T) - f(w^*) \leq \frac{G \, \|w_0 - w^*\|}{\sqrt{T}}$$

For Lipschitz, convex functions, the above $O(1/\epsilon^2)$ rate is optimal, but we require knowledge of $G, \|w_0 - w^*\|, T$ to set the step-size.

Minimizing convex, Lipschitz functions using Subgradient Descent

Recall that for smooth, convex functions, we could use Nesterov acceleration to obtain a faster $O(1/\sqrt{\epsilon})$ rate. On the other hand, for Lipschitz, convex functions, subgradient descent is optimal.

In order to get the $\frac{G\|w_0 - w^*\|}{\sqrt{T}}$ rate, we needed knowledge of G and $\|w_0 - w^*\|$ to set the step-size. There are various techniques to set the step-size in an adaptive manner.

- AdaGrad [DHS11] is adaptive to G, but still requires knowing a quantity related $\|w_0 - w^*\|$ to select the "best" step-size. This influences the practical performance of AdaGrad.
- Polyak step-size [HK19] attains the desired rate without knowledge of G or $\|w_0 - w^*\|$, but requires knowing f^*.
- Coin-Betting [OP16] does not require knowledge of $\|w_0 - w^*\|$. It only requires an estimate of G and is robust to its misspecification in theory (but not quite in practice).

Minimizing convex, Lipschitz functions using Subgradient Descent

For Lipschitz, strongly-convex functions, subgradient descent attains an $\Theta\left(\frac{1}{\epsilon}\right)$ rate. For this, the step-size depends on μ and the proof is similar to the one in (Slide 6, Lecture 10).

Subgradient descent is also optimal for Lipschitz, strongly-convex functions.

For Lipschitz functions, the convergence rates for SGD are the same as GD (with similar proofs).

Function class	L-smooth + convex	L-smooth + μ-strongly convex	G-Lipschitz + convex	G-Lipschitz + μ-strongly convex
GD	$O\left(1/\epsilon\right)$	$O\left(\kappa \log\left(1/\epsilon\right)\right)$	$\Theta\left(1/\epsilon^2\right)$	$\Theta\left(1/\epsilon\right)$
SGD	$\Theta\left(1/\epsilon^2\right)$	$\Theta\left(1/\epsilon\right)$	$\Theta\left(1/\epsilon^2\right)$	$\Theta\left(1/\epsilon\right)$

Table 1: Number of iterations required for obtaining an ϵ-sub-optimality.

Questions?

Online Optimization

```
Online Optimization (w_0, Algorithm 𝒜, Convex set 𝒞)
for k = 1, ..., T do
    Algorithm 𝒜 chooses point (decision) w_k ∈ 𝒞
    Environment chooses and reveals the (potentially adversarial) loss function f_k : 𝒞 → ℝ
    Algorithm suffers a cost f_k(w_k)
end for
```

Application: **Prediction from Expert Advice**: Given n experts, $\mathcal{C} = \Delta_n = \{w_i | w_i \geq 0 \; ; \sum_{i=1}^{n} w_i = 1\}$ and $f_k(w_k) = \langle c_k, w_k \rangle$ where $c_k \in \mathbb{R}^n$ is the loss vector.

Application: **Imitation Learning**: Given access to an expert that knows what action $a \in [A]$ to take in each state $s \in [S]$, learn a policy $\pi : [S] \to [A]$ that imitates the expert, i.e. we want that $\pi(a|s) \approx \pi_{\text{expert}}(a|s)$. Here, $w = \pi$ and $\mathcal{C} = \Delta_A \times \Delta_A \ldots \Delta_A$ (simplex for each state) and f_k is a measure of discrepancy between π_k and π_{expert}.

References i

John Duchi, Elad Hazan, and Yoram Singer, *Adaptive subgradient methods for online learning and stochastic optimization.*, Journal of machine learning research **12** (2011), no. 7.

Elad Hazan and Sham Kakade, *Revisiting the polyak step size*, arXiv preprint arXiv:1905.00313 (2019).

Francesco Orabona and Dávid Pál, *Coin betting and parameter-free online learning*, Advances in Neural Information Processing Systems **29** (2016).