CSCI699: Theory of Machine Learning Fall 2021

Lecture 16

Instructor: Vatsal Sharan *Scribe: Di Zhang*

Theorem 1 (convergence of SGD)**.** *Let f be a convex, ρ-Lipschitz function. Let $w^* = \arg\min_{w \in \mathbb{R}^d} f(x)$ and $\|w^* - w_1\| = B$ (where w_1 is the initialization). Suppose we run SGD for T steps with step size $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$ and $\|v_t\| \leq \rho$ for all t. Let $\overline{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$. Then,*

$$\mathbb{E}[f(\overline{w})] - f(w^*) \leq \frac{B\rho}{\sqrt{T}}.$$

Proof. Let $v_{1:t}$ denote the sequence $v_1, \ldots, v_t$. By convexity,

$$\mathbb{E}_{v_{1:T}}[f(\overline{w}) - f(w^*)] \leq \mathbb{E}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T} f(w_t) - f(w^*)\right].$$

By the iterative update Lemma,

$$\mathbb{E}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T} \langle w_t - w^*, v_t\rangle\right] \leq \frac{B\rho}{\sqrt{T}}.$$

So we only need to show

$$\mathbb{E}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T} f(w_t) - f(w^*)\right] \leq \mathbb{E}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T} \langle w_t - w^*, v_t\rangle\right].$$

$$\begin{aligned}\mathbb{E}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T} f(w_t) - f(w^*)\right] &= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{v_{1:T}}\left[f(w_t) - f(w^*)\right] \\ &= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{v_{1:t-1}}\left[f(w_t) - f(w^*)\right].\end{aligned}$$

SGD requires $\mathbb{E}_{v_t}[v_t \mid w_t] = \nabla f(w_t)$.

Since w_t only depends on $v_{1:t-1}$,

$$\mathbb{E}_{v_t}[v_t \mid v_{1:t-1}] = \nabla f(w_t).$$

$$\begin{aligned}
\mathbb{E}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T} f(w_t) - f(w^*)\right] &\leq \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{v_{1:t-1}}\left[\langle w_t - w^*, \mathbb{E}_{v_t}[v_t \mid v_{1:t-1}]\rangle\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{v_{1:t-1}}\left[\mathbb{E}_{v_t}\left[\langle w_t - w^*, v_t\rangle \mid v_{1:t-1}\right]\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{v_{1:t-1}}[\langle w_t - w^*, v_t\rangle] \\
&\text{(law of iterated expectations)} \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{v_{1:T}}[\langle w_t - w^*, v_t\rangle] \\
&= \mathbb{E}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T} \langle w_t - w^*, v_t\rangle\right],
\end{aligned}$$

and we are done. □

Some more points about the convergence rate,

- Same convergence rate as GD!
- The convergence rate does not improve with the β-smoothness assumption.
- For strongly convex functions, the convergence rate goes from $\frac{1}{\sqrt{T}}$ to $\frac{1}{T}$, we can't get the $e^{-T/K}$ rates.

So some tradeoffs in terms of cost per iteration and number of iterations.

1 Learning with SGD

Suppose we are interested in minimizing the risk

$$R(w) = \mathbb{E}_{z\sim D}[\ell(w, z)].$$

Using ERM, we sample n datapoints and minimize training error. Use generalization bounds to ensure small test error.

SGD gives a different way to look at this,

$$\nabla R(w_t) = \nabla \mathbb{E}_{z\sim D}[\ell(w_t, z)] = \mathbb{E}_{z\sim D}[\underbrace{\nabla \ell(w_t, z)}_{\text{gradient at } z}].$$

If we set $v_t = \nabla \ell(w_t, z)$ (for $z \sim D$),

$$\mathbb{E}[v_t] = \nabla R(w_t).$$

SGD for minmizing $R(w)$

1. Initialize w_1
2. For $t = 1, \ldots, T$:
3. $\quad z \leftarrow \mathrm{EX}(c, D)$
4. $\quad$ Get $v_t = \nabla \ell(w_t, z)$
5. $\quad$ Update $w_{t+1} = w_t - \eta v_t$
6. Output $\overline{w} = \dfrac{1}{T} \displaystyle\sum_{t=1}^{T} w_t$

Corollary 2. *Consider a convex, ρ-Lipschitz function $\ell(w, z)$. Let $w^* = \arg\min_{w \in \mathcal{H}} R(w)$. Let $\|w^* - w_1\| \leq B$. Then if we run SGD for T iterations with $\eta = \sqrt{\dfrac{B^2}{\rho^2 T}}$ where $T \geq \dfrac{B^2 \rho^2}{\epsilon^2}$, then the output $\overline{w}$ satisfies,*

$$\mathbb{E}[R(\overline{w})] \leq \min_{w \in \mathcal{H}} R(w) + \epsilon.$$

Note: SGD is also implementable in the SQ model (we need to extend SQ model to allow real-valued queries).

2 Online Learning

In PAC learning/statistical learning we ask the learn to do well under probabilistic assumptions on data (train/test data are drawn from the same distribution). We developed a theory of generalization to understand how much an algorithm's test accuracy can differ from its training accuracy.

In *online learning*, we make no probabilistic assumptions on the data. The goal is to predict well on datapoints as we see them.

Example: Weather forecasting

- We're interested in predicting rain/no rain.
- Every night, make prediction about next day, based on current conditions.
- Next day, we see whether or not it rained.

Note that there's no train/test split. Every example is both a training example and a test example. Formally:

At every time step t,

- Learner receives an input $x_t \in \mathcal{X}$.
- Makes prediction $p_t \in \mathcal{Y}$.
- Sees true label $y_t \in \mathcal{Y}$. Suffers loss $\ell(p_t, y_t)$.

For most of our discussion, think $\mathcal{Y} = \{0, 1\}$, $\ell(p_t, y_t) = \mathbf{1}\{p_t \neq y_t\}$.

Realizability

As we did in PAC learning/statistical learning, we begin with the realizability assumption on the sequence, which says that there is some hypothesis in the hypothesis classes which correctly labels all datapoints.

Definition 3 (Mistake bound model)**.** *Let $\mathcal{H}$ be a hypothesis class and A be a online learning algorithm. Given any sequence $S = (x_1, h^*(x_1)), \ldots, (x_T, h^*(x_T))$ of T labelled datapoints where $h^* \in \mathcal{H}$, let $M_A(S)$ be the number of mistakes A makes on the sequence S. We denote by $M_A(\mathcal{H})$ to the supremum of $M_A(S)$ over all possible S.*

*If there exists an algorithm A that satisfies a **mistake bound** of the form $M_A(\mathcal{H}) \leq B < \infty$, we say $\mathcal{H}$ is **online learnable** in the mistake bound model.*

Note: B should be independent of length of sequence T. As $T \to \infty$, average number of mistakes $(\leq B/T) \to 0$.

In PAC learning, we saw that the ERM algorithm which chooses any consistent hypothesis over the training set does well, as long as the size of the training set is large enough that the generalization error is small. We start by defining an analogous algorithm for the online learning setup.

Alg: Consistent

1. Initialize $V_1 = \mathcal{H}$
2. For $t = 1, \ldots, T$:
3. Receive x_t
4. Choose any $h \in V_t$
5. Predict $p_t = h(x_t)$
6. Receive $y_t = h^*(x_t)$, loss $\mathbf{1}(p_t \neq y_t)$
7. Update $v_{t+1} = \{h \in V_t : h(x_t) = y_t\}$

Proposition 4. *Let $\mathcal{H}$ be a finite hypothesis class. The above algorithm gets a mistake bound*

$$M_{consistent}(\mathcal{H}) \leq |\mathcal{H}| - 1.$$

Can we do better? Yes, by quite a lot. If we refine the above algorithm to make its predictions p_t in a smarter way than choosing any consistent hypothesis, then we can improve exponentially on the above mistake bound.

Alg: Halving

1. Initialize $V_1 = \mathcal{H}$
2. For $t = 1, \ldots, T$:

3. Receive x_t
4. Predict $p_t = \arg\max_{r \in \{0,1\}} |\{h \in V_t : h(x_t) = r\}|$ (if tie, $p_t = 1$)
5. Receive $y_t = h^*(x_t)$, loss $\mathbf{1}(p_t \neq y_t)$
6. Update $v_{t+1} = \{h \in V_t : h(x_t) = y_t\}$

Proposition 5. *Let $\mathcal{H}$ be a finite hypothesis class. Then halving algorithm satisfies the mistake bound $M_{Halving}(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$.*

Proof. Whenever the algorithm errors, $|V_{t+1}| \leq \frac{|V_t|}{2}$. If M is the number of mistakes

$$|V_{T+1}| \leq |\mathcal{H}| 2^{-M}$$

as $|V_{T+1}| \geq 1 \implies M \leq \log_2(|\mathcal{H}|)$. □