Optimization Theory and Algorithm II **October 9, 2022**

Lecture 8

Lecturer:Xiangyu Chang *Scribe: Xiangyu Chang*

Edited by: Xiangyu Chang

1 Stochastic Gradient Descent

1.1 Motivation of SGD

Suppose that we have a dataset $\{\mathbf{a}_i, b_i\}_{i=1}^m, \mathbf{a}_i \in \mathcal{A} \subseteq \mathbb{R}^n, b_i \in \mathcal{B} \subseteq \mathbb{R}$. A supervised learning process is to find a function to present the relationship between $\mathcal{A}$ and $\mathcal{B}$, that is $h : \mathcal{A} \to \mathcal{B}$.

- If $\mathbb{B} = \{-1, 1\}$, then it is called "binary classification problem".
- If $\mathcal{B} = \mathbb{R}$, then it is called "regression problem".

To estimate h, generally adopting the so-called expected risk minimization method:

$$\min_{h \in \mathcal{F}} \mathbb{E}_\rho[\ell(b, h(\mathbf{a}))], \tag{1}$$

where $\mathcal{F}$ is a function space (also called Hypothesis Space), ℓ is a loss function and $(\mathbf{a}, b)$ is generated from an unknown distribution ρ.

Example 1 *For a classification problem, a natural loss funtion is* $0-1$ *loss,*

$$\ell_{01}(b, h(\mathbf{a})) = \begin{cases} 0, & b = h(\mathbf{a}), \\ 1, & b \neq h(\mathbf{a}). \end{cases} \tag{2}$$

Then, we can derive that

$$\begin{aligned} \min_{h \in \mathcal{F}} \mathbb{E}[\ell_{01}(b, h(\mathbf{a}))|\mathbf{a}] &= \min_{h \in \mathcal{F}} \{1 \cdot \mathbb{P}(b \neq h(\mathbf{a}|\mathbf{a}) + 0 \cdot \mathbb{P}(b = h(\mathbf{a})|\mathbf{a})\} \\ &= \min_{h \in \mathcal{F}} \{\mathbb{P}(b \neq h(\mathbf{a})|\mathbf{a}\} \\ &= \min_{h \in \mathcal{F}} \mathbb{E}[\mathbf{1}(h(\mathbf{a}) \neq b)|\mathbf{a}]. \end{aligned}$$

Let us consider a more general case, define $g : \mathcal{A} \to \mathbb{R}$*, then we use*

$$h(\mathbf{a}) = (g(\mathbf{a})) = \begin{cases} 1, & g(\mathbf{a}) > 0, \\ 0, & g(\mathbf{a}) = 0, \\ -1, & g(\mathbf{a}) < 0. \end{cases} \tag{3}$$

Thus,

$$\ell_{01}(b, h(\mathbf{a})) = \begin{cases} 0, & b = h(\mathbf{a}), \\ 1, & b \neq h(\mathbf{a}), \end{cases} = \ell_{01}(b, g(\mathbf{a})) = \begin{cases} 0, & bg(\mathbf{a}) > 0, \\ 1, & bg(\mathbf{a}) \leq 0. \end{cases} \tag{4}$$

The quantity $bg(\mathbf{a})$ *is called "margin" in the supervised learning.*

Because that the distribution ρ is unknown, the expected risk minimization (1) cannot be directly computed. The following *empirical risk minimization* (ERM for short) approach is used to replace the expected risk minimization as

$$\min_{h\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} \ell(b_i, h(\mathbf{a}_i)). \tag{5}$$

Obviously, the loss function (2) is discontinuous and non-convex. So, we cannot directly use the convex optimization trick to handle the problem (5) with the 0-1 loss. The so-called convex **surrogate function** is adopted to overcome this hurdle (see Figure 1).

Consider the margin $u = bg(\mathbf{a})$, then general loss on the margin has the property: $\ell(u) \to 0$ as $u \to +\infty$ and $\ell(u)$ increasing as $u \to -\infty$. Thus, we list the commonly used surrogate function as follows:

- Logistic loss: $\ell(u) = \log(1 + \exp(-u))$ for the logistic regression.
- Hinge loss: $\ell(u) = (1-u)_+ = \max(1-u, 0)$ for the SVM.
- Exponential loss: $\ell(u) = \exp(-u)$ for the AdaBoost.
- Square loss: $\ell(u) = (1-u)^2/2$ for the least squares SVM.

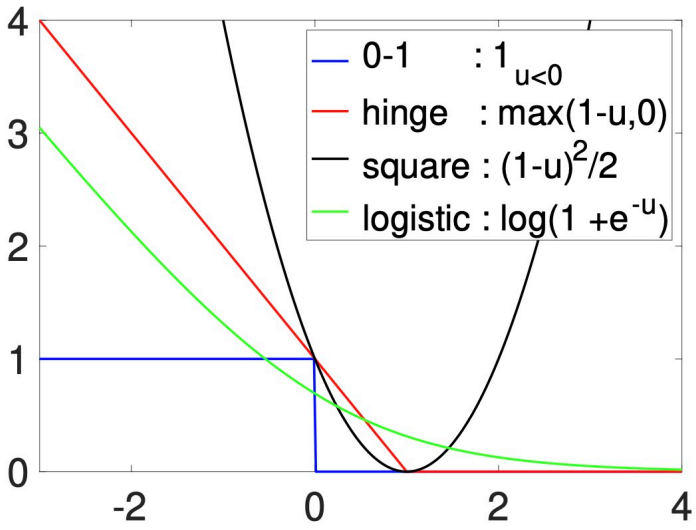

Figure 1: Classical convex surrogates for binary classification with the 0-1 loss.

Example 2 *(Logistic Regression Again) Consider* $\mathbb{P}(b=1|\mathbf{a}) = p$ *and* $\mathbb{P}(b=-1|\mathbf{a}) = 1-p$, *then suppose that*

$$\mathbb{P}(b=1|\mathbf{a}) = \frac{\exp(\mathbf{a}^\top \mathbf{x})}{1+\exp(\mathbf{a}^\top \mathbf{x})},$$

and

$$\mathbb{P}(b=-1|\mathbf{a}) = \frac{1}{1+\exp(\mathbf{a}^\top \mathbf{x})}.$$

Thus,

$$\mathbb{P}(b|\mathbf{a}) = \frac{1}{1+\exp(-b\mathbf{a}^\top \mathbf{x})}. \tag{6}$$

Log-likelihood is

$$\sum_i \ell(b_i, \mathbf{a}_i) = -\sum_i \log(1 + \exp(-b_i \mathbf{a}_i^\top \mathbf{x})).$$

MLE is equivalent to

$$\min_{\mathbf{x}} \frac{1}{m} \sum_i \underbrace{\log(1 + \exp(-b_i \mathbf{a}_i^\top \mathbf{x}))}_{logistic\ loss}.$$

Example 3 *(Regression) Let us consider the following regression cases*

- *Suppose that* $\mathcal{F} = \{h | h(\mathbf{a}) = \mathbf{a}^\top \mathbf{x}\}$ *and* $\ell(b, h(\mathbf{a})) = \frac{1}{2}(b - h(\mathbf{a}))^2$*, then ERM is equivalent to*

$$\min_{\mathbf{x}} \frac{1}{2m} \sum_i (b_i - h(\mathbf{a}_i))^2 = \frac{1}{2m} \|A\mathbf{x} - \mathbf{b}\|^2.$$

- *Ridge Regression:*

$$\min_{\mathbf{x}} \frac{1}{m} \sum_i (b_i - h(\mathbf{a}_i))^2 + \lambda \|\mathbf{x}\|^2 = \frac{1}{m} \sum_i \{(b_i - h(\mathbf{a}_i))^2 + \lambda \|\mathbf{x}\|^2\}.$$

- *Nonlinear case:*

$$\min_{\mathbf{x}} \frac{1}{m} \sum_i (b_i - h_{\mathbf{x}}(\mathbf{a}_i))^2,$$

  where $h_{\mathbf{x}}$ *is a nonlinear function, e.g., deep nets.*

Definition 1 *Define that* ***finite-sum optimization*** *problem as*

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{m} \sum_i^m f_i(\mathbf{x}), \tag{7}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$.

Let $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ be a dataset, $\mathcal{F} = \{h_{\mathbf{x}} | h_{\mathbf{x}} : \mathcal{A} \to \mathcal{B}, \mathbf{x} \in \mathbb{R}^n\}$ be a class of predictor function and ℓ be a loss function. Then we can find that the corresponding ERM framework is a finite-sum optimization, that is

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{m} \sum_i^m f_i(\mathbf{x}) = \frac{1}{m} \sum_i^m \ell(b_i, h_{\mathbf{x}}(\mathbf{a}_i)). \tag{8}$$

One key property of the formulation (7) is that every term in the finite sum optimization only involves one sample from the dataset.

If we use the gradient descent algorithm to solve it:

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \frac{s_t}{m} \sum_i \nabla f_i(\mathbf{x}^t) = \mathbf{x}^t - \frac{s_t}{m} \sum_i \nabla_{\mathbf{x}} \ell(b_i, h_{\mathbf{x}^t}(\mathbf{a}_i)).$$

From this update, we see that one iteration of gradient descent requires to go over the entire dataset in order to compute the gradient vector. In a big data setting where the number of samples m is very huge, this cost can be prohibitive.

Algorithm 1 Stochastic Gradient Descent

1: **Input:** Given an initial starting point $\mathbf{x}^0 \in \mathbb{R}^n$, and $t = 0$
2: **for** $t = 0, 1, \ldots, T-1$ **do**
3: Compute a stepsize or learning rate $s_t > 0$.
4: Draw a random index $i_t \in \{1, \ldots, m\}$.
5: $\mathbf{x}^{t+1} = \mathbf{x}^t - s_t \nabla f_{i_t}(\mathbf{x}^t)$ and $t := t + 1$.
6: **end for**
7: **Output:** $\mathbf{x}^T$.

1.2 SGD

The core idea of SGD is assuming each component function f_i is differential, the method picks an index i_t randomly and takes a step in the direction of the negative gradient of the component function f_{i_t}. Then we have the unbiased estimation of the gradient of ERM problem:

$$\mathbb{E}[\nabla f_{i_t}(\mathbf{x})] = \sum_{i=1}^{m} \mathbb{P}(i_t = i) \nabla f_i(\mathbf{x}) = \frac{1}{m} \sum_i \nabla f_i(\mathbf{x}).$$

The key motivation for this process is the using a signal data point at a time results in updates that are m times cheaper than a full gradient step. Note that using a signal component does not necessarily lead to convergence, even cannot guarantee the decreasing of objective function.

Example 4

$$\min_x f_1(x) + f_2(x),$$

where $f_1(x) = 2x^2$ *and* $f_2(x) = -x^2$. *If* $x^t > 0$, *and* $i_t = 2$, *then the SGD update will necessarily lead to an increase in the objective function value.*

Batch SGD:

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \frac{s_t}{|D_t|} \sum_{i_t \in D_t} \nabla f_{i_t}(\mathbf{x}^t),$$

where D_t is a subset of $\{1, 2, \ldots, m\}$ called "Batch". If $|D_t| = m$, it is GD. If $|D_t| = 1$, it is SGD. If $|D_t| << m$, it is the mini-batch stochastic gradient descent algorithm.

1.2.1 Convergence

Assumption 1 *(A1) Objective function* f *is* β*-smooth,*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq \beta \|\mathbf{x} - \mathbf{y}\|.$$

Assumption 2 *(A2)*

(1) The index i_t *does not depended from the previous* $i_0, i_1, \ldots, i_{t-1}$.

(2) $\mathbb{E}_{i_t}[\nabla f_{i_t}(\mathbf{x}^t)] = \nabla f(\mathbf{x}^t)$ *(Unbiased Estimation).*

(3) $\mathbb{E}_{i_t}[\|\nabla f_{i_t}(\mathbf{x}^t)\|^2] \leq \sigma^2 + \|\nabla f(\mathbf{x}^t)\|^2$ *(control the variance).*

Assumption 3 *(A3) The objective function* f *is* α*-strong convex*

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

Lemma 1 *Under A1, consider the SGD, then*

$$\begin{aligned}\mathbb{E}_{i_t}[f(\mathbf{x}^{t+1})] &:= \mathbb{E}[f(\mathbf{x}^{t+1})|\mathbf{x}^t] \\ &\leq f(\mathbf{x}^t) - s_t\langle\nabla f(\mathbf{x}^t), \mathbb{E}_{i_t}[\nabla f_{i_t}(\mathbf{x}^t)]\rangle + \frac{\beta s_t^2}{2}\mathbb{E}_{i_t}[\|\nabla f_{i_t}(\mathbf{x}^t)\|^2].\end{aligned}$$

Proof 1 *We know that*

$$\begin{aligned}f(\mathbf{x}^{t+1}) &\leq f(\mathbf{x}^t) + \langle\nabla f(\mathbf{x}^t), \mathbf{x}^{t+1} - \mathbf{x}^t]\rangle + \frac{\beta}{2}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \\ &= f(\mathbf{x}^t) - s_t\langle\nabla f(\mathbf{x}^t), \nabla f_{i_t}(\mathbf{x}^t)\rangle + \frac{\beta s_t^2}{2}\|\nabla f_{i_t}(\mathbf{x}^t)\|^2.\end{aligned}$$

Taking the expectation of the above inequality leads to the results.

Lemma 2 *Based on A1 and A2, it has*

$$\mathbb{E}_{i_t}[f(\mathbf{x}^{t+1}) - f(\mathbf{x}^t)] \leq \frac{\beta s_t^2}{2}\sigma^2 - s_t(1 - \frac{\beta s_t}{2})\|\nabla f(\mathbf{x}^t)\|^2.$$

Proof 2 *According Lemma 1, A1 and A2,*

$$\begin{aligned}\mathbb{E}_{i_t}[f(\mathbf{x}^{t+1}) - f(\mathbf{x}^t)] &\leq \frac{\beta s_t^2}{2}\mathbb{E}_{i_t}[\|\nabla f_{i_t}(\mathbf{x}^t)\|^2] - s_t\langle\nabla f(\mathbf{x}^t), \mathbb{E}_{i_t}[\nabla f_{i_t}(\mathbf{x}^t)]\rangle \\ &\leq \frac{\beta s_t^2}{2}(\sigma^2 + \|\nabla f(\mathbf{x}^t)\|^2) - s_t\|\nabla f(\mathbf{x}^t)\|^2 \\ &= \frac{\beta s_t^2}{2}\sigma^2 - s_t(1 - \frac{\beta s_t}{2})\|\nabla f(\mathbf{x}^t)\|^2.\end{aligned}$$

Lemma 3 *Suppose A3 holds, then*

$$f(\mathbf{x}) - f^* \leq \frac{1}{2\alpha}\|\nabla f(\mathbf{x})\|^2.$$

Non-convex and β-smooth objective functions:

SGD is a commonly accepted method for training neural networks, which are usually non-convex and smooth optimization problems. For GD, we have known that

$$\min_{0\leq t\leq T-1}\|\nabla f(\mathbf{x}^t)\| = O(\frac{1}{\sqrt{T}}).$$

What about SGD?

Theorem 1 *(Fixed Learning Rate)*

Suppose that A1 and A2 hold. Let $s_t = s \in (0, 1/\beta]$, then

$$\mathbb{E}[1/T\sum_{t=0}^{T-1}\|\nabla f(\mathbf{x}^t)\|^2] \leq s\beta\sigma^2 + \frac{2(f(\mathbf{x}^0) - f^*)}{Ts}.$$

References