Machine Learning: Optimization 2

Hao Tang

February 4, 2024

Convexity on more points

If a function f is convex,

$$f(\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3) \leq \alpha_1 f(x_1) + \alpha_2 f(x_2) + \alpha_3 f(x_3) \tag{1}$$

for $\alpha_1, \alpha_2, \alpha_3 \geq 0$ and $\alpha_1 + \alpha_2 + \alpha_3 = 1$.

Convexity on more points

If a function f is convex,

$$f\left(\sum_{i=1}^{n} \alpha_i x_i\right) \leq \sum_{i=1}^{n} \alpha_i f(x_i) \tag{2}$$

for $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i = 1$.

Jensen's inequality

If a function f is convex,

$$f(\mathbb{E}_{x\sim p(x)}[x]) \leq \mathbb{E}_{x\sim p(x)}[f(x)]. \quad (3)$$

- For log loss

$$L = \sum_{i=1}^{n} \log \left(1 + \exp(-y_i w^\top \phi(x_i)) \right) \tag{4}$$

we cannot even solve $\nabla_w L = 0$.

- How do we find the optimal solution?

Gradient descent

- Gradient descent is an iterative algorithm, consisting of the steps

$$w_{t+1} = w_t - \eta_t \nabla L(w_t). \tag{5}$$

- The variable $\eta_t > 0$ is called the **step size** (or learning rate), and can depend on t.

Gradient descent on log loss

- The log loss in the binary case

$$L = \sum_{i=1}^{n} \log \left(1 + \exp(-y_i w^\top x_i) \right). \tag{6}$$

- We have shown that L is convex in w.

Gradient descent on log loss

$$\frac{\partial L}{\partial w} = \sum_{i=1}^{n} \frac{\exp(-y_i w^\top x_i)}{1+\exp(-y_i w^\top x_i)}(-y_i x_i) \tag{7}$$

$$= \sum_{i=1}^{n} \left(1 - \frac{1}{1+\exp(-y_i w^\top x_i)}\right)(-y_i x_i) \tag{8}$$

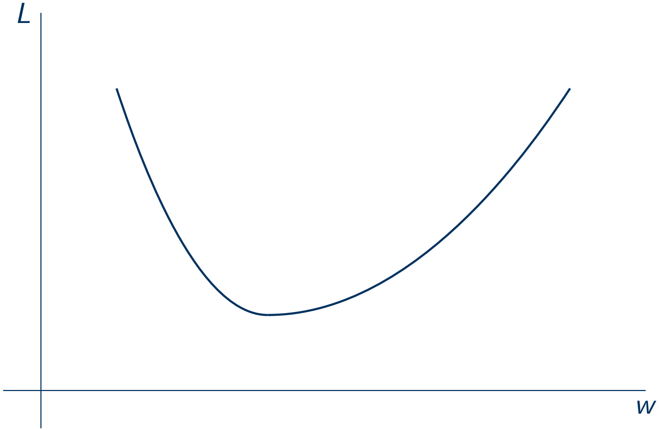
$$= \sum_{i=1}^{n} (1 - p(y_i|x_i))\,(-y_i x_i) \tag{9}$$

Gradient descent on log loss

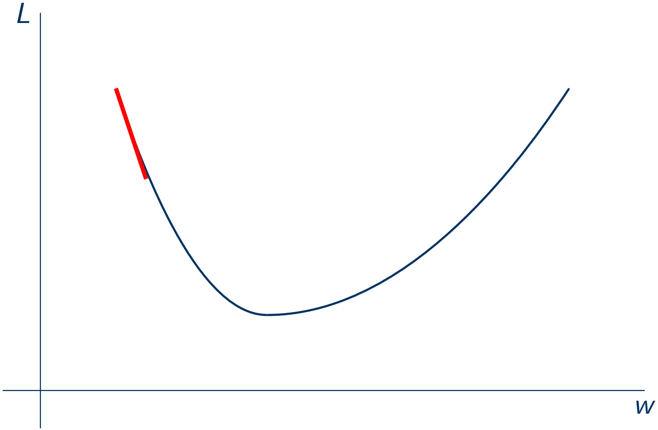
$$w_{t+1} = w_t - \eta_t \nabla L(w_t) \tag{10}$$

$$w_{t+1} = w_t - \eta_t \sum_{i=1}^{n} \left(1 - \frac{1}{1 + \exp(-y_i w_t^\top x_i)}\right)(-y_i x_i) \tag{11}$$

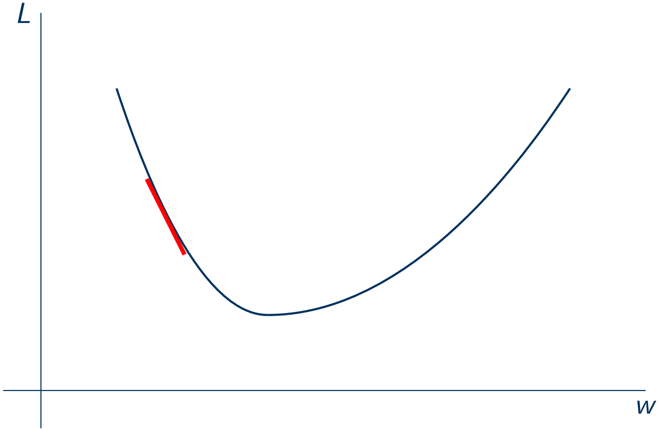
Gradient descent



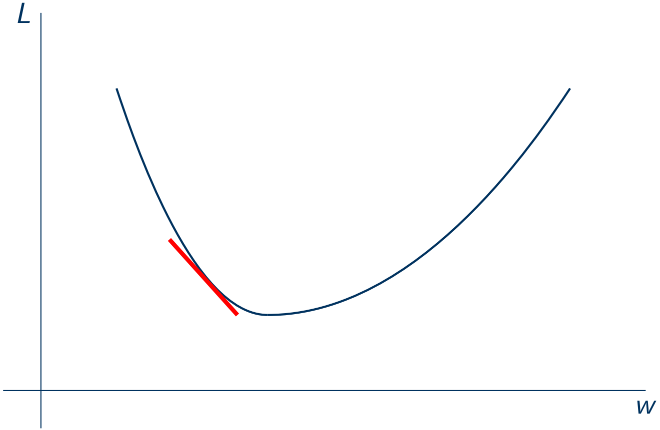
Gradient descent



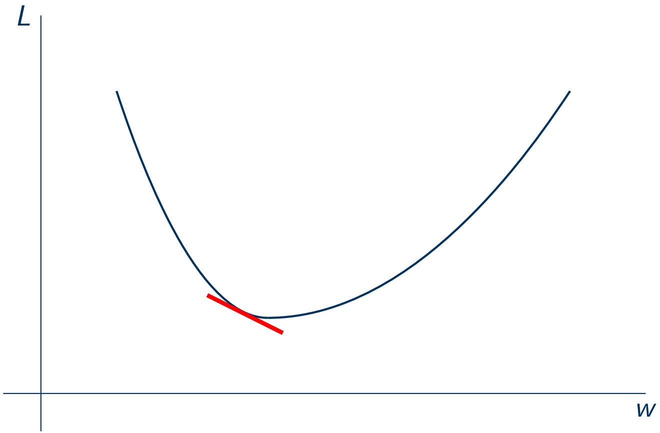
Gradient descent



Gradient descent



Gradient descent



Approximate solutions in optimization

- We say that $\hat{x}$ is an approximate solution of the minimizer x^* if, for a given $\epsilon > 0$,

$$f(\hat{x}) - f(x^*) < \epsilon. \tag{12}$$

- Note that it is close in function value, not close in the input.

Approximate solutions for iterative algorithms

- An iterative algorithm creates a sequence $x_1, \ldots, x_t$.
- How many updates do we need to achieve an approximate solution?
- Given $\epsilon > 0$, how large does t needs to be to achieve

$$f(x_t) - f(x^*) < \epsilon? \tag{13}$$

- We want to express ϵ as a function of t.

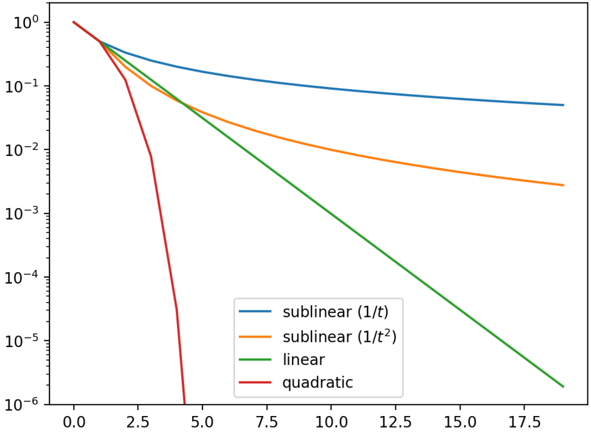
Potential results

- Sublinear
  - $f(x_t) - f(x^*) \leq \frac{c}{t^2}$
- Linear
  - $f(x_t) - f(x^*) \leq cr^t$ for $0 < r < 1$
- Quadratic
  - $f(x_t) - f(x^*) \leq cr^{2^t}$ for $0 < r < 1$

Potential results

- Sublinear
  - $f(x_t) - f(x^*) \leq \dfrac{c}{t^2}$
  - $\epsilon = O\left(\frac{1}{t^2}\right)$ or $t = O(\frac{1}{\sqrt{\epsilon}})$
- Linear
  - $f(x_t) - f(x^*) \leq cr^t$ for $0 < r < 1$
  - $\epsilon = O\left(2^{-t}\right)$ or $t = O(\log \frac{1}{\epsilon})$
- Quadratic
  - $f(x_t) - f(x^*) \leq cr^{2^t}$ for $0 < r < 1$
  - $\epsilon = O\left(2^{-2^t}\right)$ or $t = O(\log \log \frac{1}{\epsilon})$

Convergence rates



Gradient descent on mean-squared error

- The mean-squared error is

$$L = \|Xw - y\|_2^2. \tag{14}$$

- We have shown that L is convex in w.
- We have shown that the optimal solution is $(X^\top X)^{-1} X^\top y$.

Gradient descent on mean-squared error

$$\nabla L = 2(X^\top X w - X^\top y) \tag{15}$$

$$w_{t+1} = w_t - \eta_t \nabla L(w_t) = w_t - \eta_t 2(X^\top X w_t - X^\top y) \tag{16}$$

Rumtime comparison for solving mean-squared error

- The runtime of $(X^\top X)^{-1} X^\top y$ is $O(nd^2)$ or $O(d^3)$ (whichever dominates).

Rumtime comparison for solving mean-squared error

- The runtime of $(X^\top X)^{-1}X^\top y$ is $O(nd^2)$ or $O(d^3)$ (whichever dominates).

Rumtime comparison for solving mean-squared error

- The runtime of $(X^\top X)^{-1} X^\top y$ is $O(nd^2)$ or $O(d^3)$ (whichever dominates).
- The runtim of a single gradient step $w_{t+1} = w_t - \eta_t 2(X^\top X w_t - X^\top y)$ is $O(nd)$.

Convergence rates of gradient descent on mean-squared error

- If we run gradient descent on mean-squared error, we have

$$L(w_t) - L(w^*) = \frac{1}{2}(w_0 - w^*)^\top (I - \eta H)^{2t} H (w_0 - w^*) \tag{17}$$

  where $H = X^\top X$ is the Hessian of L.

- If we choose $\eta = \frac{1}{\lambda_{\max}}$, where $\lambda_{\max}$ is the largest eigenvalue of H, the convergence is linear.

Stochastic gradient descent

1. Sample x_t, y_t from a data set S.
2. $w_{t+1} = w_t - \eta_t \nabla \ell(w_t; x_t, y_t)$
   - Per sample L_2 loss $\ell(w; x, y) = (w^\top x_t - y_t)^2$
   - Per sample log loss $\ell(w; x, y) = \log(1 + \exp(-y_t w^\top x_t))$
3. Go to 1 until the solution is satisfactory.

Stochastic gradient descent

- $\nabla \ell(w_t; x_t, y_t)$ is now random, because x_t and y_t is random.
- The expectation

$$\mathbb{E}_{x,y \sim U(S)}[\nabla \ell(w; x, y)] = \nabla L(w) \tag{18}$$

where $U(S)$ is the uniform distribution over the samples in S.

Guarantee for stochastic gradient descent

- If we do SGD on an convex function,

$$\mathbb{E}_{x,y \sim U(S)}[L(\bar{w}_t)] - L(w^*) \leq \frac{\|w_0 - w^*\|_2^2}{2\eta t} + \frac{\eta B^2}{2}. \tag{19}$$

- $\|\nabla \ell(w_t; x, y)\|_2 \leq B$ for any t, x, and y
- $\bar{w}_t = \frac{w_1 + \cdots + w_t}{t}$
- The runtime is $O(1/\sqrt{t})$ if we choose $\eta = \frac{\|w_0 - w^*\|_2}{B\sqrt{T}}$, independent of the data set size n!

Mini-batch stochastic gradient descent

1. Sample a subset S_t from a data set S.
2. $w_{t+1} = w_t - \eta_t \nabla \frac{1}{|S_t|} \sum_{x,y \in S_t} \ell(w_t; x, y)$
   - The random sampling maintains $\mathbb{E}_{S_t \sim U(S)^t} \left[\nabla \frac{1}{|S_t|} \sum_{x,y \in S_t} \ell(w; x, y) \right] = \nabla L(w)$.
3. Go to 1 until the solution is satisfactory.

Common practice and terminology

Common practice and terminology

- When doing SGD, permute the data and then go in serial.

Common practice and terminology

- When doing SGD, permute the data and then go in serial.
- A pass over the data set is called an **epoch**.

Common practice and terminology

- When doing SGD, permute the data and then go in serial.
- A pass over the data set is called an **epoch**.
- When doing mini-batch SGD, remember to normalize by the batch size.

Common practice and terminology

- When doing SGD, permute the data and then go in serial.
- A pass over the data set is called an **epoch**.
- When doing mini-batch SGD, remember to normalize by the batch size.
- With a larger batch size, we go over an epoch faster.

Common practice and terminology

- When doing SGD, permute the data and then go in serial.
- A pass over the data set is called an **epoch**.
- When doing mini-batch SGD, remember to normalize by the batch size.
- With a larger batch size, we go over an epoch faster.
- Use the largest batch size you can afford.