MIT 9.520/6.860, Fall 2018
Statistical Learning Theory and Applications

Class 06: Learning with Stochastic Gradients

Sasha Rakhlin

Why Optimization?

Much (but not all) of Machine Learning: write down objective function involving data and parameters, find good (or optimal) parameters through optimization.

Key idea: find a near-optimal solution by iteratively using only local information about the objective (e.g. gradient, Hessian).

Motivating example: Newton's Method

Newton's method in 1d:

$$w_{t+1} = w_t - (f''(w_t))^{-1} f'(w_t)$$

Example (parabola):

$$f(w) = aw^2 + bw + c$$

Start with *any* w_1. Then Newton's Method gives

$$w_2 = w_1 - (2a)^{-1}(2aw_1 + b)$$

which means $w_2 = -b/(2a)$. Finds minimum of f in 1 step, *no matter where you start*!

Newton's Method in multiple dim:

$$w_{t+1} = w_t - [\nabla^2 f(w_t)]^{-1} \nabla f(w_t)$$

(here $\nabla^2 f(w_t)$ is the Hessian, assume invertible)

Recalling Least Squares

Least Squares objective (without $1/n$ normalization)

$$f(w) = \sum_{i=1}^{n} (y_i - x_i^\top w)^2 = \|Y - Xw\|^2$$

Calculate: $\nabla^2 f(w) = 2X^\top X$ and $\nabla f(w) = -2X^\top (Y - Xw)$.

Taking $w_1 = 0$, the Newton's Method gives

$$w_2 = 0 + (2X^\top X)^{-1} 2X^\top (Y - X0) = (X^\top X)^{-1} X^\top Y$$

which is *the least-squares solution* (global min). Again, 1 step is enough.

Verify: if $f(w) = \|Y - Xw\|^2 + \lambda \|w\|^2$, $(X^\top X)$ becomes $(X^\top X + \lambda)$

What do we do if data $(x_1, y_1), \ldots, (x_n, y_n), \ldots$ are streaming? Can we incorporate data on the fly without having to re-compute inverse $(X^\top X)$ at every step?

$\longrightarrow$ Online Learning

Let $w_1 = 0$. Let w_t be least-squares solution after seeing $t-1$ data points. Can we get w_t from w_{t-1} cheaply? Newton's Method will do it in 1 step (since objective is quadratic).

Let $C_t = \sum_{i=1}^{t} x_i x_i^\mathsf{T}$ (or $+\lambda I$) and $X_t = [x_1, \ldots, x_t]^\mathsf{T}$, $Y_t = [y_1, \ldots, y_t]^\mathsf{T}$. Newton's method gives

$$w_{t+1} = w_t + C_t^{-1} X_t^\mathsf{T} (Y_t - X_t w_t)$$

This can be simplified to

$$w_{t+1} = w_t + C_t^{-1} x_t (y_t - x_t^\mathsf{T} w_t)$$

since residuals up to $t-1$ are orthogonal to columns of X_{t-1}.

The bottleneck is computing C_t^{-1}. Can we update it quickly from C_{t-1}^{-1}?

Sherman-Morrison formula: for invertible square A and any u, v

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top A^{-1}u}$$

Hence

$$C_t^{-1} = C_{t-1}^{-1} - \frac{C_{t-1}^{-1}x_t x_t^\top C_{t-1}^{-1}}{1 + x_t^\top C_{t-1}^{-1}x_t}$$

and (do the calculation)

$$C_t^{-1}x_t = C_{t-1}^{-1}x_t \cdot \frac{1}{1 + x_t^\top C_{t-1}^{-1}x_t}$$

Computation required: $d \times d$ matrix C_t^{-1} times a $d \times 1$ vector $= O(d^2)$ time to incorporate new datapoint. Memory: $O(d^2)$. Unlike full regression from scratch, does not depend on amount of data t.

Recursive Least Squares (cont.)

Recap: recursive least squares is

$$w_{t+1} = w_t + C_t^{-1} x_t (y_t - x_t^{\mathsf{T}} w_t)$$

with a rank-one update of C_{t-1}^{-1} to get C_t^{-1}.

Consider throwing away second derivative information, replacing with scalar:

$$w_{t+1} = w_t + \eta_t x_t (y_t - x_t^{\mathsf{T}} w_t).$$

where η_t is a decreasing sequence.

Online Least Squares

The algorithm

$$w_{t+1} = w_t + \eta_t x_t (y_t - x_t^\top w_t).$$

- is recursive;
- does not require storing the matrix C_t^{-1};
- does not require updating the inverse, but only vector/vector multiplication.

However, we are not guaranteed convergence in 1 step. How many? How to choose η_t?

First, recognize that

$$-\nabla(y_t - x_t^\top w)^2 = 2x_t[y_t - x_t^\top w].$$

Hence, proposed method is gradient descent. Let us study it abstractly and then come back to least-squares.

Lemma: Let f be convex G-Lipschitz. Let $w^* \in \operatorname{argmin}_w f(w)$ and $\|w^*\| \leq B$. Then gradient descent

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

with $\eta = \frac{B}{G\sqrt{T}}$ and $w_1 = 0$ yields a sequence of iterates such that the average $\bar{w}_T = \frac{1}{T}\sum_{t=1}^{T} w_t$ of trajectory satisfies

$$f(\bar{w}_T) - f(w^*) \leq \frac{BG}{\sqrt{T}}.$$

Proof:

$$\begin{aligned} \|w_{t+1} - w^*\|^2 &= \|w_t - \eta \nabla f(w_t) - w^*\|^2 \\ &= \|w_t - w^*\|^2 + \eta^2 \|\nabla f(w_t)\|^2 - 2\eta \nabla f(w_t)^\intercal (w_t - w^*) \end{aligned}$$

Rearrange:

$$2\eta \nabla f(w_t)^\intercal (w_t - w^*) = \|w_t - w^*\|^2 - \|w_{t+1} - w^*\|^2 + \eta^2 \|\nabla f(w_t)\|^2 .$$

Note: Lipschitzness of f is equivalent to $\|\nabla f(w)\| \leq G$.

Summing over $t = 1, \ldots, T$, telescoping, dropping negative term, using $w_1 = 0$, and dividing both sides by 2η,

$$\sum_{t=1}^{T} \nabla f(w_t)^\intercal (w_t - w^*) \leq \frac{1}{2\eta} \|w^*\|^2 + \frac{\eta}{2} T G^2 \leq \sqrt{BGT}.$$

Convexity of f means

$$f(w_t) - f(w^*) \leq \nabla f(w_t)^\intercal (w_t - w^*)$$

and so

$$\frac{1}{T} \sum_{t=1}^{T} f(w_t) - f(w^*) \leq \frac{1}{T} \sum_{t=1}^{T} \nabla f(w_t)^\intercal (w_t - w^*) \leq \frac{BG}{\sqrt{T}}$$

Lemma follows by convexity of f and Jensen's inequality. (end of proof)

Gradient descent can be written as

$$w_{t+1} = \underset{w}{\operatorname{argmin}} \; \eta \left\{ f(w_t) + \nabla f(w_t)^\mathsf{T}(w - w_t) \right\} + \frac{1}{2} \left\| w - w_t \right\|^2$$

which can be interpreted as minimizing a linear approximation but staying close to previous solution.

Alternatively, can interpret it as building a second-order model locally (since cannot fully trust the local information – unlike our first parabola example).

Remarks:

- Gradient descent for non-smooth functions does not guarantee actual descent of the iterates w_t (only their average).
- For constrained optimization problems over a set K, do projected gradient step

$$w_{t+1} = \mathrm{Proj}_K \left(w_t - \eta \nabla f(w_t) \right)$$

  Proof essentially the same.
- Can take stepsize $\eta_t = \frac{BG}{\sqrt{t}}$ to make it horizon-independent.
- Knowledge of G and B not necessary (with appropriate changes).
- Faster convergence under additional assumptions on f (smoothness, strong convexity).
- Last class: for smooth functions (gradient is L-Lipschitz), constant step size $1/L$ gives faster $O(1/T)$ convergence.
- Gradients can be replaced with stochastic gradients (unbiased estimates).

Stochastic Gradients

Suppose we only have access to an unbiased estimate ∇_t of $\nabla f(w_t)$ at step t. That is, $\mathbb{E}[\nabla_t | w_t] = \nabla f(w_t)$. Then Stochastic Gradient Descent (SGD)

$$w_{t+1} = w_t - \eta \nabla_t$$

enjoys the guarantee

$$\mathbb{E}[f(\bar{w}_T)] - f(w^*) \leq \frac{BG}{\sqrt{n}}$$

where G is such that $\mathbb{E}[\|\nabla_t\|^2] \leq G^2$ for all t.

Kind of amazing: at each step go in the direction that is wrong (but correct on average) and still converge.

Stochastic Gradients

Setting #1:

Empirical loss can be written as

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, w^\mathsf{T} x_i) = \mathbb{E}_{I\sim \text{unif}[1:n]} \ell(y_I, w^\mathsf{T} x_I)$$

Then $\nabla_t = \nabla \ell(y_I, w_t^\mathsf{T} x_I)$ is an unbiased gradient:

$$\mathbb{E}[\nabla_t | w_t] = \mathbb{E}[\nabla \ell(y_I, w_t^\mathsf{T} x_I) | w_t] = \nabla \mathbb{E}[\ell(y_I, w_t^\mathsf{T} x_I) | w_t] = \nabla f(w_t)$$

Conclusion: if we pick index I uniformly at random from dataset and make gradient step $\nabla \ell(y_I, w_t^\mathsf{T} x_I)$, then we are performing SGD on empirical loss objective.

Stochastic Gradients

Setting #2:

Expected loss can be written as

$$f(w) = \mathbb{E}\ell(Y, w^\intercal X)$$

where (X, Y) is drawn i.i.d. from population $P_{X\times Y}$.

Then $\nabla_t = \nabla\ell(Y, w_t^\intercal X)$ is an unbiased gradient:

$$\mathbb{E}[\nabla_t | w_t] = \mathbb{E}[\nabla\ell(Y, w_t^\intercal X)|w_t] = \nabla\mathbb{E}[\ell(Y, w_t^\intercal X)|w_t] = \nabla f(w_t)$$

Conclusion: if we pick example (X, Y) from distribution $P_{X\times Y}$ and make gradient step $\nabla\ell(Y, w_t^\intercal X)$, then we are performing SGD on expected loss objective. Equivalent to going through a dataset once.

Stochastic Gradients

Say we are in Setting #2 and we go through dataset once. The guarantee is

$$\mathbb{E}[f(\bar{w})] - f(w^*) \leq \frac{BG}{\sqrt{T}}$$

after T iterations. So, time complexity to find ϵ-minimizer of expected objective $\mathbb{E}\ell(w^\mathsf{T} X, Y)$ is independent of the dataset size n!! Suitable for large-scale problems.

Stochastic Gradients

In practice, we cycle through the dataset several times (which is somewhere between Setting #1 and #2).

Appendix

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if

$$f(\alpha u + (1-\alpha)v) \leq \alpha f(u) + (1-\alpha)f(v)$$

for any $\alpha \in [0,1]$ and $u, v \in \mathbb{R}^d$ (or restricted to a convex set). For a differentiable function, convexity is equivalent to monotonicity

$$\langle \nabla f(u) - \nabla f(v), u - v \rangle \geq 0. \tag{1}$$

where

$$\nabla f(u) = \left(\frac{\partial f(u)}{\partial u_1}, \ldots, \frac{\partial f(u)}{\partial u_d} \right).$$

Appendix

It holds that for a convex differentiable function

$$f(u) \geq f(v) + \langle \nabla f(v), u - v \rangle . \tag{2}$$

A subdifferential set is defined (for a given v) precisely as the set of all vectors ∇ such that

$$f(u) \geq f(v) + \langle \nabla, u - v \rangle . \tag{3}$$

for all u. The subdifferential set is denoted by $\partial f(v)$. A subdifferential will often substitute the gradient, even if we don't specify it.

Appendix

If $f(v) = \max_i f_i(v)$ for convex differentiable f_i, then, for a given v, whenever $i \in \operatorname*{argmax}_i f_i(v)$, it holds that

$$\nabla f_i(v) \in \partial f(v).$$

(Prove it!) We conclude that the subdifferential of the hinge loss $\max\{0, 1 - y_t \langle w, x_t \rangle\}$ with respect to w is

$$-y_t x_t \cdot \mathbf{1}\{y_t \langle w, x_t \rangle < 1\}\,. \tag{4}$$

Appendix

A function f is L-Lipschitz over a set S with respect to a norm $\|\cdot\|$ if

$$\|f(u) - f(v)\| \leq L \|u - v\|$$

for all $u, v \in S$. A function f is β-smooth if its gradient maps are Lipschitz

$$\|\nabla f(v) - \nabla f(u)\| \leq \beta \|u - v\|,$$

which implies

$$f(u) \leq f(v) + \langle \nabla f(v), u - v \rangle + \frac{\beta}{2} \|u - v\|^2.$$

(Prove that the other implication also holds.) The dual notion to smoothness is that of strong convexity. A function f is σ-strongly convex if

$$f(\alpha u + (1-\alpha)v) \leq \alpha f(u) + (1-\alpha)f(v) - \frac{\sigma}{2}\alpha(1-\alpha)\|u - v\|^2,$$

which means

$$f(u) \geq f(v) + \langle u - v, \nabla f(v) \rangle + \frac{\sigma}{2} \|u - v\|^2.$$