COMPSCI 514: Algorithms for Data Science

Cameron Musco

University of Massachusetts Amherst. Fall 2024.

Lecture 23

Logistics

- Problem Set 5 can be turned in up to 12/12 (next Thursday) at 11:59pm with no penalty. No extensions will be granted beyond this. The challenge problem is optional extra credit.
- After today you will be able to solve every problem on it.
- Additional final review office hours will be posted soon.

Summary

Last Class:

- Multivariable calculus review
- Introduction to gradient descent. Motivation as a greedy algorithm.
- Convex functions
- Lipschitz functions

This Class:

- Analysis of gradient descent for convex Lipschitz functions
- Extension to projected gradient descent for constrained optimization.
- Start on online/stochastic gradient descent?

Definition – Convex Function: A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if, for any $\vec{\theta}_1, \vec{\theta}_2 \in \mathbb{R}^d$ and $\lambda \in [0,1]$:

$$(1-\lambda) \cdot f(\vec{\theta}_1) + \lambda \cdot f(\vec{\theta}_2) \geq f\left((1-\lambda) \cdot \vec{\theta}_1 + \lambda \cdot \vec{\theta}_2\right)$$

Corollary – Convex Function: A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if, for any $\vec{\theta}_1, \vec{\theta}_2 \in \mathbb{R}^d$ and $\lambda \in [0,1]$:

$$f(\vec{\theta}_2) - f(\vec{\theta}_1) \geq \vec{\nabla} f(\vec{\theta}_1)^T \left(\vec{\theta}_2 - \vec{\theta}_1\right)$$

Definition – Lipschitz Function: A function $f : \mathbb{R}^d \to \mathbb{R}$ is G-Lipschitz if $\|\vec{\nabla} f(\vec{\theta})\|_2 \leq G$ for all $\vec{\theta}$.

GD Analysis – Convex Functions

Assume that:

- f is convex.
- f is G-Lipschitz.
- $\|\vec{\theta}_1 - \vec{\theta}_*\|_2 \leq R$ where $\vec{\theta}_1$ is the initialization point.

Gradient Descent

- Choose some initialization $\vec{\theta}_1$ and set $\eta = \frac{R}{G\sqrt{t}}$.
- For $i = 1, \ldots, t-1$
    - $\vec{\theta}_{i+1} = \vec{\theta}_i - \eta \vec{\nabla} f(\vec{\theta}_i)$
- Return $\hat{\theta} = \arg\min_{\vec{\theta}_1, \ldots, \vec{\theta}_t} f(\vec{\theta}_i)$.

Theorem – GD on Convex Lipschitz Functions: For convex G-Lipschitz function f, GD run with $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon.$$

Step 1: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \vec{\theta}_*\|_2^2 - \|\vec{\theta}_{i+1} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$. Visually:

GD Analysis Proof

Theorem – GD on Convex Lipschitz Functions: For convex G-Lipschitz function f, GD run with $t \geq \frac{R^2G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon.$$

Step 1: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \theta_*\|_2^2 - \|\vec{\theta}_{i+1} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$. Formally:

GD Analysis Proof

Theorem – GD on Convex Lipschitz Functions: For convex G-Lipschitz function f, GD run with $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon.$$

Step 1: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \vec{\theta}_*\|_2^2 - \|\vec{\theta}_{i+1} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$.

Step 1.1: $\vec{\nabla} f(\vec{\theta}_i)^T (\vec{\theta}_i - \vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \vec{\theta}_*\|_2^2 - \|\vec{\theta}_{i+1} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$ $\implies$ **Step 1 by convexity.**

GD Analysis Proof

Theorem – GD on Convex Lipschitz Functions: For convex G-Lipschitz function f, GD run with $t \geq \frac{R^2G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon.$$

Step 1: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \vec{\theta}_*\|_2^2 - \|\vec{\theta}_{i+1} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \implies$

Step 2: $\frac{1}{t}\sum_{i=1}^{t} f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{R^2}{2\eta \cdot t} + \frac{\eta G^2}{2}$.

GD Analysis Proof

Theorem – GD on Convex Lipschitz Functions: For convex G-Lipschitz function f, GD run with $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon.$$

Step 2: $\frac{1}{t}\sum_{i=1}^{t} f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{R^2}{2\eta \cdot t} + \frac{\eta G^2}{2}$.

GD Analysis Proof

Theorem – GD on Convex Lipschitz Functions: For convex G-Lipschitz function f, GD run with $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon.$$

Step 2: $\frac{1}{t}\sum_{i=1}^{t} f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{R^2}{2\eta \cdot t} + \frac{\eta G^2}{2}$.

Constrained Convex Optimization

Often want to perform convex optimization with convex constraints.

$$\vec{\theta}^* = \arg\min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}),$$

where $\mathcal{S}$ is a convex set.

Definition – Convex Set: A set $\mathcal{S} \subseteq \mathbb{R}^d$ is convex if and only if, for any $\vec{\theta}_1, \vec{\theta}_2 \in \mathcal{S}$ and $\lambda \in [0, 1]$:

$$(1 - \lambda)\vec{\theta}_1 + \lambda \cdot \vec{\theta}_2 \in \mathcal{S}$$

E.g. $\mathcal{S} = \{\vec{\theta} \in \mathbb{R}^d : \|\vec{\theta}\|_2 \leq 1\}$.

Projected Gradient Descent

For any convex set let $P_{\mathcal{S}}(\cdot)$ denote the projection function onto $\mathcal{S}$.

- $P_{\mathcal{S}}(\vec{y}) = \arg\min_{\vec{\theta} \in \mathcal{S}} \|\vec{\theta} - \vec{y}\|_2$.
- For $\mathcal{S} = \{\vec{\theta} \in \mathbb{R}^d : \|\vec{\theta}\|_2 \leq 1\}$ what is $P_{\mathcal{S}}(\vec{y})$?
- For $\mathcal{S}$ being a k dimensional subspace of $\mathbb{R}^d$, what is $P_{\mathcal{S}}(\vec{y})$?

Projected Gradient Descent

- Choose some initialization $\vec{\theta}_1$ and set $\eta = \frac{R}{G\sqrt{t}}$.
- For $i = 1, \ldots, t-1$
  - $\vec{\theta}_{i+1}^{(out)} = \vec{\theta}_i - \eta \cdot \vec{\nabla} f(\vec{\theta}_i)$
  - $\vec{\theta}_{i+1} = P_{\mathcal{S}}(\vec{\theta}_{i+1}^{(out)})$.
- Return $\hat{\theta} = \arg\min_{\vec{\theta}_i} f(\vec{\theta}_i)$.

Convex Projections

Projected gradient descent can be analyzed identically to gradient descent!

Theorem – Projection to a convex set: For any convex set $\mathcal{S} \subseteq \mathbb{R}^d$, $\vec{y} \in \mathbb{R}^d$, and $\vec{\theta} \in \mathcal{S}$,

$$\|P_{\mathcal{S}}(\vec{y}) - \vec{\theta}\|_2 \leq \|\vec{y} - \vec{\theta}\|_2.$$

Projected Gradient Descent Analysis

Theorem – Projected GD: For convex G-Lipschitz function f, and convex set $\mathcal{S}$, Projected GD run with $t \geq \frac{R^2G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon = \min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}) + \epsilon$$

Recall: $\vec{\theta}_{i+1}^{(out)} = \vec{\theta}_i - \eta \cdot \vec{\nabla} f(\vec{\theta}_i)$ and $\vec{\theta}_{i+1} = P_{\mathcal{S}}(\vec{\theta}_{i+1}^{(out)})$.

Step 1: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \theta_*\|_2^2 - \|\vec{\theta}_{i+1}^{(out)} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$.

Step 1.a: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \vec{\theta}_*\|_2^2 - \|\vec{\theta}_{i+1} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$.

Step 2: $\frac{1}{t}\sum_{i=1}^{t} f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{R^2}{2\eta \cdot t} + \frac{\eta G^2}{2}$ $\implies$ Theorem.

Typical Optimization Problem in Machine Learning: Given data points $\vec{x}_1, \ldots, \vec{x}_n$ and labels/observations $y_1, \ldots, y_n$ solve:

$$\vec{\theta}^* = \arg\min_{\vec{\theta} \in \mathbb{R}^d} L(\vec{\theta}, \mathbf{X}, y) = \sum_{j=1}^{n} \ell(M_{\vec{\theta}}(\vec{x}_j), y_j).$$

The gradient of $L(\vec{\theta}, \mathbf{X})$ has one component per data point:

$$\vec{\nabla} L(\vec{\theta}, \mathbf{X}) = \sum_{j=1}^{n} \vec{\nabla} \ell(M_{\vec{\theta}}(\vec{x}_j), y_j).$$

When n is large this is very expensive to compute!

Training a neural network on ImageNet would require $n = 14$ million back propagations! ... per iteration of GD.

Gradient Descent At Scale

Solution: Update using just a single data point, or a small batch of data points per iteration.

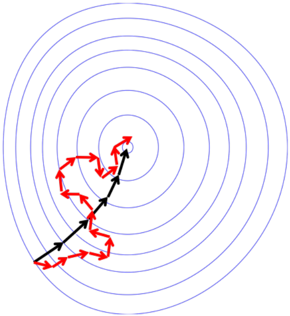
- Looking at a single data point gives you a coarse, but still useful cue on how to improve your model.
- If the data point is chosen uniformly at random, the sampled gradient is correct in expectation.

$$\vec{\nabla} L(\vec{\theta}, \mathbf{X}) = \sum_{i=j}^{n} \vec{\nabla} \ell(M_{\vec{\theta}}(\vec{x}_j), y_j) \rightarrow \mathbb{E}_{j \sim [n]}[\vec{\nabla} \ell(M_{\vec{\theta}}(\vec{x}_j), y_j)] = \frac{1}{n} \cdot \vec{\nabla} L(\vec{\theta}, \mathbf{X}).$$

- The key idea behind **stochastic gradient descent** (SGD).

Stochastic Gradient Descent

Stochastic gradient descent takes more, but much cheaper steps than gradient descent.



$\vec{\theta}^{(i+1)} = \vec{\theta}^{(i)} - \eta \cdot \vec{\nabla} L(\vec{\theta}^{(i)}, \mathsf{X})$ vs. $\vec{\theta}^{(i+1)} = \vec{\theta}^{(i)} - \eta \cdot \vec{\nabla} \ell(M_{\vec{\theta}^{(i)}}(\vec{x}_j), y_j)$

Online Gradient Descent

SGD is closely related to online gradient descent.

In reality many learning problems are online.

- Websites optimize ads or recommendations to show users, given continuous feedback from these users.
- Spam filters are incrementally updated and adapt as they see more examples of spam over time.
- Face recognition systems, other classification systems, learn from mistakes over time.

Want to minimize some global loss $L(\vec{\theta}, \mathbf{X})$, when data points are presented in an online fashion $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ (like in streaming algorithms)

Will view SGD as a special case: when data points are presented (by design) in a random order.