COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

University of Massachusetts Amherst. Spring 2020.

Lecture 22

LOGISTICS

- Problem Set 4 on Spectral Methods/Optimization due Wednesday 4/29. Can submit until Sunday 5/3 at 8pm.
- Shorter than the first 3. I may assign some additional extra credit, depending on what we cover in the next few classes.

Last Class:

- Finish up power method – Krylov methods and connection to random walks.
- Start on continuous optimization.

This Class:

- Gradient descent.
- Motivation as a greedy method
- Start on analysis for convex functions.

Given some function $f : \mathbb{R}^d \to \mathbb{R}$, find $\vec{\theta}_\star$ with:

$$f(\vec{\theta}_\star) = \min_{\vec{\theta} \in \mathbb{R}^d} f(\vec{\theta}).$$

- Typically up to some small approximation factor: i.e., find $\hat{\theta}$ with $f(\hat{\theta}) = \min_{\vec{\theta} \in \mathbb{R}^d} f(\vec{\theta}) + \epsilon$
- Often under some constraints:
  - $\|\vec{\theta}\|_2 \leq 1, \quad \|\vec{\theta}\|_1 \leq 1.$
  - $A\vec{\theta} \leq \vec{b}, \quad \vec{\theta}^T A \vec{\theta} \geq 0.$
  - $\vec{1}^T \vec{\theta} = \sum_{i=1}^d \vec{\theta}(i) \leq c.$

Let $\vec{e}_i \in \mathbb{R}^d$ denote the i^{th} standard basis vector, $\vec{e}_i = \underbrace{[0, 0, 1, 0, 0, \ldots, 0]}_{\text{1 at position } i}$.

Partial Derivative:

$$\frac{\partial f}{\partial \vec{\theta}(i)} = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon \cdot \vec{e}_i) - f(\vec{\theta})}{\epsilon}.$$

Directional Derivative:

$$D_{\vec{v}}\, f(\vec{\theta}) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon \vec{v}) - f(\vec{\theta})}{\epsilon}.$$

Gradient: Just a 'list' of the partial derivatives.

$$\vec{\nabla} f(\vec{\theta}) = \begin{bmatrix} \frac{\partial f}{\partial \vec{\theta}(1)} \\ \frac{\partial f}{\partial \vec{\theta}(2)} \\ \vdots \\ \frac{\partial f}{\partial \vec{\theta}(d)} \end{bmatrix}$$

Directional Derivative in Terms of the Gradient:

$$\begin{aligned} D_{\vec{v}} f(\vec{\theta}) &= \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon \vec{v}(\vec{e}_1 \cdot \vec{v}(1) + \vec{e}_2 \cdot \vec{v}(2) + \ldots + \vec{e}_d \cdot \vec{v}(d)) - f(\vec{\theta})}{\epsilon} \\ &\approx \vec{v}(1) \cdot \frac{\partial f}{\partial \vec{\theta}(1)} + \vec{v}(2) \cdot \frac{\partial f}{\partial \vec{\theta}(2)} + \ldots + \vec{v}(d) \cdot \frac{\partial f}{\partial \vec{\theta}(d)} \\ &= \langle \vec{v}, \vec{\nabla} f(\vec{\theta}) \rangle. \end{aligned}$$

Often the functions we are trying to optimize are very complex (e.g., a neural network). We will assume access to:

Function Evaluation: Can compute $f(\vec{\theta})$ for any $\vec{\theta}$.

Gradient Evaluation: Can compute $\vec{\nabla} f(\vec{\theta})$ for any $\vec{\theta}$.

In neural networks:

- Function evaluation is called a forward pass (propogate an input through the network).
- Gradient evaluation is called a backward pass (compute the gradient via chain rule, using backpropagation).

Running Example: Least squares regression.

Given input points $\vec{x}_1, \ldots \vec{x}_n$ (the rows of data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$) and labels $y_1, \ldots, y_n$ (the entries of $\vec{y} \in \mathbb{R}^n$) , find $\vec{\theta}_*$ minimizing:

$$L_{\mathbf{X},\vec{y}}(\vec{\theta}) = \sum_{i=1}^{n} \left(\vec{\theta}^T \vec{x}_i - y_i\right)^2 = \|\mathbf{X}\vec{\theta} - \vec{y}\|_2^2.$$

By Chain rule:

$$\frac{\partial L_{\mathbf{X},\vec{y}}(\vec{\theta})}{\partial \vec{\theta}(j)} = \sum_{i=1}^{n} 2 \cdot \left(\vec{\theta}^T \vec{x}_i - y_i\right) \cdot \frac{\partial \left(\vec{\theta}^T \vec{x}_i - y_i\right)}{\partial \vec{\theta}(j)}$$

$$= \sum_{i=1}^{n} 2 \cdot \left(\vec{\theta}^T \vec{x}_i - y_i\right) \vec{x}_i(j)$$

$$\frac{\partial \left(\vec{\theta}^T \vec{x}_i - y_i\right)}{\partial \vec{\theta}(j)} = \frac{\partial (\theta^T \vec{x}_i)}{\partial \vec{\theta}(j)} = \lim_{\epsilon \to 0} \frac{(\theta + \epsilon \vec{e}_j)^T \vec{x}_i - \theta^T \vec{x}_i}{\epsilon} = \lim_{\epsilon \to 0} \frac{\epsilon \vec{e}_j^T \vec{x}_i}{\epsilon} = \vec{x}_i(j).$$

Partial derivative for least squares regression:

$$\frac{\partial L_{\mathbf{X},\vec{y}}(\vec{\theta})}{\partial \vec{\theta}(j)} = \sum_{i=1}^{n} 2 \cdot \left(\vec{\theta}^T \vec{x}_i - y_i\right) \vec{x}_i(j).$$

$$\begin{aligned}\vec{\nabla} L_{\mathbf{X},\vec{y}}(\vec{\theta}) &= \sum_{i=1}^{n} 2 \cdot \left(\vec{\theta}^T \vec{x}_i - y_i\right) \vec{x}_i \\ &= 2\mathbf{X}^T(\mathbf{X}\vec{\theta} - \vec{y}).\end{aligned}$$

Gradient for least squares regression via linear algebraic approach:

$$\nabla L_{\mathbf{X},\vec{y}}(\vec{\theta}) = \nabla \|\mathbf{X}\vec{\theta} - \vec{y}\|_2^2$$

Gradient descent is a greedy iterative optimization algorithm: Starting at $\vec{\theta}_1$, in each iteration let $\vec{\theta}_{i+1} = \vec{\theta}_i + \eta\vec{v}$, where η is a (small) 'step size' and $\vec{v}$ is a direction chosen to minimize $f(\vec{\theta}_i + \eta\vec{v})$.

$$D_{\vec{v}}\, f(\vec{\theta}) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon\vec{v}) - f(\vec{\theta})}{\epsilon}. D_{\vec{v}}\, f(\vec{\theta}_i) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta}_i + \epsilon\vec{v}) - f(\vec{\theta}_i)}{\epsilon}.$$

So for small η:

$$\begin{aligned} f(\vec{\theta}_{i+1}) - f(\vec{\theta}_i) = f(\vec{\theta}_i + \eta\vec{v}) - f(\vec{\theta}_i) &\approx \eta \cdot D_{\vec{v}} f(\vec{\theta}_i) \\ &= \eta \cdot \langle \vec{v}, \vec{\nabla} f(\vec{\theta}_i) \rangle. \end{aligned}$$

We want to choose $\vec{v}$ minimizing $\langle \vec{v}, \vec{\nabla} f(\vec{\theta}_i) \rangle$ – i.e., pointing in the direction of $\vec{\nabla} f(\vec{\theta}_i)$ but with the opposite sign.

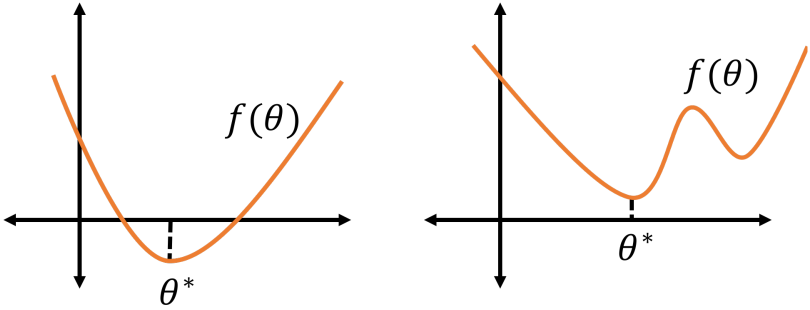
Gradient Descent

- Choose some initialization $\vec{\theta}_1$.
- For $i = 1, \ldots, t-1$
  - $\vec{\theta}_{i+1} = \vec{\theta}_i - \eta \nabla f(\vec{\theta}_i)$
- Return $\hat{\theta} = \arg\min_{\vec{\theta}_i} f(\vec{\theta}_i)$, as an approximate minimizer.

Step size η is chosen ahead of time or adapted during the algorithm (details to come.)

- For now assume η stays the same in each iteration.

$\theta \in \mathbb{R} \quad \nabla f(\theta) \in \mathbb{R}$



Gradient Descent Update: $\vec{\theta}_{i+1} = \vec{\theta}_i - \eta \nabla f(\vec{\theta}_i)$

Convex Functions: After sufficient iterations, gradient descent will converge to a approximate minimizer $\hat{\theta}$ with:

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

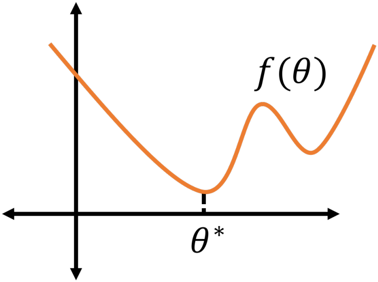
Examples: least squares regression, logistic regression, sparse regression (lasso), regularized regression, SVMS,...

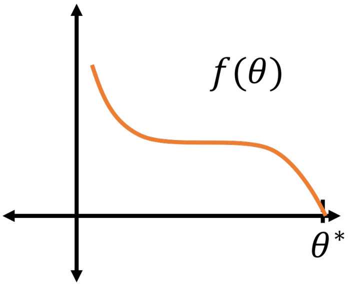
Non-Convex Functions: After sufficient iterations, gradient descent will converge to a approximate stationary point $\hat{\theta}$ with:

$$\|\nabla f(\hat{\theta})\|_2 \leq \epsilon.$$

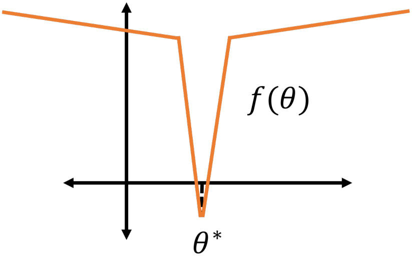
Examples: neural networks, clustering, mixture models.

Why for non-convex functions do we only guarantee convergence to a approximate stationary point rather than an approximate local minimum?





$\theta \in \mathbb{R} \quad \nabla f(\theta) \in \mathbb{R}$



Gradient Descent Update: $\vec{\theta}_{i+1} = \vec{\theta}_i - \eta \nabla f(\vec{\theta}_i)$

Both Convex and Non-convex: Need to assume the function is well-behaved in some way.

- Lipschitz (size of gradient is bounded): There is some G s.t.:

$$\forall \vec{\theta}: \quad \|\vec{\nabla} f(\vec{\theta})\|_2 \leq G \Leftrightarrow \forall \vec{\theta}_1, \vec{\theta}_2: \quad |f(\vec{\theta}_1) - f(\vec{\theta}_2)| \leq G \cdot \|\vec{\theta}_1 - \vec{\theta}_2\|_2$$

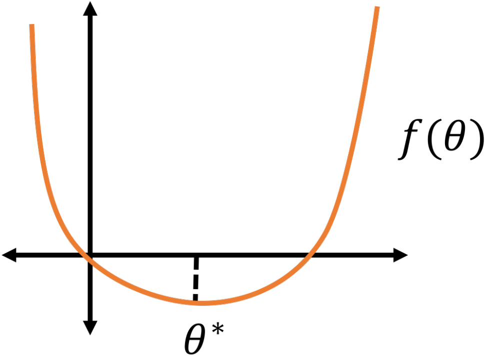
- Smooth/Lipschitz gradient (direction/size of gradient is not changing too quickly): There is some β s.t.:

$$\forall \vec{\theta}_1, \vec{\theta}_2: \quad \|\vec{\nabla} f(\vec{\theta}_1) - \vec{\nabla} f(\vec{\theta}_2)\|_2 \leq \beta \cdot \|\vec{\theta}_1 - \vec{\theta}_2\|_2.$$

Gradient Descent analysis for convex functions.

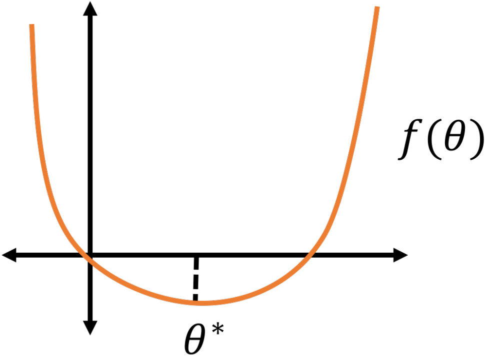
Definition – Convex Function: A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if, for any $\vec{\theta}_1, \vec{\theta}_2 \in \mathbb{R}^d$ and $\lambda \in [0, 1]$:

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



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 $\lambda \in [0, 1]$:

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



Assume that:

- f is convex.
- f is G Lipschitz ($\|\vec{\nabla} f(\vec{\theta})\|_2 \leq G$ for all $\vec{\theta}$).
- $\|\vec{\theta}_1 - \vec{\theta}_*\|_2 \leq R$ 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 $i = 1, \ldots, t-1$
  - $\vec{\theta}_{i+1} = \vec{\theta}_i - \eta \nabla f(\vec{\theta}_i)$
- 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 $t \geq \frac{R^2G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of θ_*, outputs $\hat{\theta}$ satisfying:

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

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

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 θ_*, outputs $\hat{\theta}$ satisfying:

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

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

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 θ_*, outputs $\hat{\theta}$ satisfying:

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

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

Step 1.1: $\nabla f(\theta_i)^T(\theta_i - \theta_*) \leq \frac{\|\theta_i - \theta_*\|_2^2 - \|\theta_{i+1} - \theta_*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \implies$ **Step 1.**

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 θ_*, outputs $\hat{\theta}$ satisfying:

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

Step 1: For all i, $f(\theta_i) - f(\theta_*) \leq \frac{\|\theta_i - \theta_*\|_2^2 - \|\theta_{i+1} - \theta_*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \Longrightarrow$

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

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 θ_*, outputs $\hat{\theta}$ satisfying:

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

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

Questions on Gradient Descent?

Often want to perform convex optimization with convex constraints.

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

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

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 $i = 1, \ldots, t-1$
  - $\vec{\theta}_{i+1}^{(out)} = \vec{\theta}_i - \eta \cdot \nabla f(\vec{\theta}_i)$
  - $\vec{\theta}_{i+1} = P_{\mathcal{S}}(\vec{\theta}_{i+1}^{(out)})$.
- Return $\hat{\theta} = \arg\min_{\vec{\theta}_i} f(\vec{\theta}_i)$.

Visually:

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 $\vec{\theta} \in \mathcal{S}$,

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

Theorem – Projected GD: For convex G-Lipschitz function f, and convex set $\mathcal{S}$, Projected 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 θ_*, outputs $\hat{\theta}$ satisfying:

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

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

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

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

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