COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco
University of Massachusetts Amherst. Spring 2020.
Lecture 20
- 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.
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.
CONTINUOUS OPTIMIZATION

Given some function \( f: \mathbb{R}^d \rightarrow \mathbb{R} \), find \( \hat{\theta}_* \) with:

\[
f(\hat{\theta}_*) = \min_{\theta \in \mathbb{R}^d} f(\theta).
\]

- Typically up to some small approximation factor: i.e., find \( \hat{\theta} \in \mathbb{R}^d \) with \( f(\hat{\theta}) = \min_{\theta \in \mathbb{R}^d} f(\theta) + \epsilon \)
- Often under some constraints:
  - \( \|\hat{\theta}\|_2 \leq 1, \|\hat{\theta}\|_1 \leq 1. \)
  - \( A\hat{\theta} \leq \bar{b}, \quad \hat{\theta}^T A\hat{\theta} \geq 0. \)
  - \( \bar{1}^T \hat{\theta} = \sum_{i=1}^d \hat{\theta}(i) \leq c. \)
Let \( \vec{e}_i \in \mathbb{R}^d \) denote the \( i^{th} \) standard basis vector,
\[
\vec{e}_i = [0, 0, 1, 0, 0, \ldots, 0].
\]
1 at position \( i \)
Let $\vec{e}_i \in \mathbb{R}^d$ denote the $i^{th}$ standard basis vector, $\vec{e}_i = [0, 0, 1, 0, 0, \ldots, 0]$. 1 at position $i$

Partial Derivative:

$$\frac{\partial f}{\partial \vec{\theta}(i)} = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon \cdot \vec{e}_i) - f(\vec{\theta})}{\epsilon}.$$
Let \( \vec{e}_i \in \mathbb{R}^d \) denote the \( i^{th} \) standard basis vector, 
\[
\vec{e}_i = [0, 0, 1, 0, 0, \ldots, 0].
\]

1 at position \( i \)

**Partial Derivative:**

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

**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.

\[
\nabla f(\vec{\theta}) = \begin{bmatrix}
\frac{\partial f}{\partial \theta(1)} \\
\frac{\partial f}{\partial \theta(2)} \\
\vdots \\
\frac{\partial f}{\partial \theta(d)}
\end{bmatrix}
\]
Gradient: Just a ‘list’ of the partial derivatives.

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

Directional Derivative in Terms of the Gradient:

\[ D_{\vec{v}} f(\vec{\theta}) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon \vec{v}) - f(\vec{\theta})}{\epsilon} \]
Gradient: Just a ‘list’ of the partial derivatives.

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

Directional Derivative in Terms of the Gradient:

\[
D_{\vec{v}} f(\vec{\theta}) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon (\vec{e}_1 \cdot \vec{v}(1) + \vec{e}_2 \cdot \vec{v}(2) + \ldots + \vec{e}_d \cdot \vec{v}(d))) - f(\vec{\theta})}{\epsilon}
\]
**Gradient:** Just a ‘list’ of the partial derivatives.

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

**Directional Derivative in Terms of the Gradient:**

\[
D_{\vec{v}} f(\vec{\theta}) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon (\vec{e}_1 \cdot \vec{v}(1) + \vec{e}_2 \cdot \vec{v}(2) + \ldots + \vec{e}_d \cdot \vec{v}(d)) - f(\vec{\theta})}{\epsilon}
\]
Gradient: Just a ‘list’ of the partial derivatives.

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

Directional Derivative in Terms of the Gradient:

\[
D_{\vec{v}} f(\vec{\theta}) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon (\vec{e}_1 \cdot \vec{v}(1) + \vec{e}_2 \cdot \vec{v}(2) + \ldots + \vec{e}_d \cdot \vec{v}(d)) - f(\vec{\theta})}{\epsilon}
\]

\[
\approx \vec{v}(1) \cdot \frac{\partial f}{\partial \theta(1)}
\]
Gradient: Just a ‘list’ of the partial derivatives.

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

Directional Derivative in Terms of the Gradient:

\[ D_{\vec{v}} f(\vec{\theta}) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon (\vec{e}_1 \cdot \vec{v}(1) + \vec{e}_2 \cdot \vec{v}(2) + \ldots + \vec{e}_d \cdot \vec{v}(d))) - f(\vec{\theta})}{\epsilon} \]

\[ \approx \vec{v}(1) \cdot \frac{\partial f}{\partial \theta(1)} + \vec{v}(2) \cdot \frac{\partial f}{\partial \theta(2)} + \ldots + \vec{v}(d) \cdot \frac{\partial f}{\partial \theta(d)} \]
Gradient: Just a ‘list’ of the partial derivatives.

\[ \nabla = \text{update direction} \]

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

Directional Derivative in Terms of the Gradient:

\[ D_{\vec{v}} f(\vec{\theta}) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon (\vec{e}_1 \cdot \nabla (1) + \vec{e}_2 \cdot \nabla (2) + \ldots + \vec{e}_d \cdot \nabla (d))) - f(\vec{\theta})}{\epsilon} \]

\[ = \langle \vec{v}, \hat{\nabla} f(\vec{\theta}) \rangle. \]

\[ \nabla (1) \ldots \nabla (d) \]
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 $f(\theta)$ for any $\theta$.

**Gradient Evaluation**: Can compute $\nabla f(\theta)$ for any $\theta$. 
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 $f(\theta)$ for any $\theta$.

**Gradient Evaluation:** Can compute $\nabla f(\theta)$ for any $\theta$.

In neural networks:

- Function evaluation is called a **forward pass** (propagate 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 $X \in \mathbb{R}^{n \times d}$) and labels $y_1, \ldots, y_n$ (the entries of $\vec{y} \in \mathbb{R}^n$), find $\vec{\theta}_*$ minimizing:

$$L_{X,\vec{y}}(\vec{\theta}) = \sum_{i=1}^{n} \left( \vec{\theta}^T \vec{x}_i - y_i \right)^2$$
Running Example: Least squares regression.

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

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

By Chain rule:

$$\frac{\partial L_{X,\tilde{y}}(\tilde{\theta})}{\partial \tilde{\theta}(j)} = \sum_{i=1}^{n} 2 \cdot \left( \tilde{\theta}^T \tilde{x}_i - y_i \right) \cdot \frac{\partial \left( \tilde{\theta}^T \tilde{x}_i - y_i \right)}{\partial \tilde{\theta}(j)}$$
Running Example: Least squares regression.

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

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

By Chain rule:

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

$$\frac{\partial \left( \tilde{\theta}^T \tilde{x}_i - y_i \right)}{\partial \tilde{\theta}(j)} = \frac{\partial (\tilde{\theta}^T \tilde{x}_i)}{\partial \tilde{\theta}(j)} = \hat{\theta}_j$$
GRADIENT EXAMPLE

Running Example: Least squares regression.

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

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

By Chain rule:

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

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

$$= e_j^T \tilde{x}_i = x_i(j)$$
**Running Example:** Least squares regression.

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

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

By Chain rule:

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

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

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

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

By Chain rule:

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

\[
\frac{\partial \left(\vec{\theta}^T \vec{x}_i - y_i\right)}{\partial \vec{\theta}(j)} = \frac{\partial (\vec{\theta}^T \vec{x}_i)}{\partial \vec{\theta}(j)} = \lim_{\epsilon \to 0} \frac{(\theta + \epsilon \vec{e}_j)^T \vec{x}_i - \theta^T \vec{x}_i}{\epsilon} = \lim_{\epsilon \to 0} \frac{\epsilon \vec{e}_j^T \vec{x}_i}{\epsilon} = \vec{x}_i(j).
\]
Running Example: Least squares regression.

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

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

By Chain rule:

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

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

$$\frac{\partial \left( \mathbf{\theta}^T \mathbf{x}_i - y_i \right)}{\partial \mathbf{\theta}(j)} = \frac{\partial (\mathbf{\theta}^T \mathbf{x}_i)}{\partial \mathbf{\theta}(j)} = \lim_{\epsilon \to 0} \frac{(\mathbf{\theta} + \epsilon \mathbf{e}_j)^T \mathbf{x}_i - \mathbf{\theta}^T \mathbf{x}_i}{\epsilon} = \lim_{\epsilon \to 0} \frac{\epsilon \mathbf{e}_j^T \mathbf{x}_i}{\epsilon} = \mathbf{x}_i(j).$$
Partial derivative for least squares regression:

\[
\frac{\partial \mathcal{L}_{X,Y}(\theta)}{\partial \theta(j)} = \sum_{i=1}^{n} 2 \cdot (\theta^T x_i - y_i) x_i(j).
\]
Partial derivative for least squares regression:

\[
\frac{\partial L_{x,y}(\theta)}{\partial \theta(j)} = \sum_{i=1}^{n} 2 \cdot (\theta^T \bar{x}_i - y_i) \bar{x}_i(j).
\]

\[
\nabla L_{x,y}(\theta) = \sum_{i=1}^{n} 2 \cdot (\theta^T \bar{x}_i - y_i) \bar{x}_i.
\]
Partial derivative for least squares regression:

\[
\frac{\partial L_{X,\tilde{Y}}(\tilde{\theta})}{\partial \tilde{\theta}(j)} = \sum_{i=1}^{n} 2 \cdot (\tilde{\theta}^T \tilde{x}_i - y_i) \tilde{x}_i(j).
\]

\[
\nabla L_{X,\tilde{Y}}(\tilde{\theta}) = \sum_{i=1}^{n} 2 \cdot (\tilde{\theta}^T \tilde{x}_i - y_i) \tilde{x}_i = \begin{bmatrix} x_1^T \cdots x_n^T \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \cdots \tilde{x}_n \end{bmatrix}^T \cdot \tilde{x}_i - \tilde{y}.
\]

\[
= 2X^T (X\tilde{\theta} - \tilde{y}).
\]

\[
\theta \in \mathbb{R}^d
\]

\[
X \in \mathbb{R}^{n \times d}
\]
Gradient for least squares regression via linear algebraic approach:

\[
\nabla L_{x,y}(\theta) = \nabla \|X\theta - \bar{y}\|^2 \\
\sum_{i=1}^{n} (\theta^T x_i - y_i)^2 \\
\n\nabla \left[(x\theta - y)^T (x\theta - y)\right] = \nabla \left[\theta^T X^T x \theta - 2\theta^T X y + y^T y\right] \\
= 2X^T x \theta - 2X^T y = 2X^T (x\theta - y)
\]
Gradient descent is a greedy iterative optimization algorithm:
Starting at $\theta_1$, in each iteration let $\theta_{i+1} = \theta_i + \eta \vec{v}$, where $\eta$ is a (small) ‘step size’ and $\vec{v}$ is a direction chosen to minimize $f(\theta_i + \eta \vec{v})$. 
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 $\eta$ is a (small) ‘step size’ and $\vec{v}$ is a direction chosen to minimize $f(\vec{\theta}_i + \eta \vec{v})$.

$$D_{\vec{v}} f(\vec{\theta}) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta} + \epsilon \vec{v}) - f(\vec{\theta})}{\epsilon}.$$
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 $\eta$ is a (small) ‘step size’ and $\vec{v}$ is a direction chosen to minimize $f(\vec{\theta}_i + \eta \vec{v})$.

$$D_{\vec{v}} f(\vec{\theta}_i) = \lim_{\epsilon \to 0} \frac{f(\vec{\theta}_i + \epsilon \vec{v}) - f(\vec{\theta}_i)}{\epsilon}.$$
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 \( \eta \) is a (small) ‘step size’ and \( \vec{v} \) is a direction chosen to minimize \( f(\vec{\theta}_i + \eta \vec{v}) \).

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

So for small \( \eta \):

\[
f(\vec{\theta}_{i+1}) - f(\vec{\theta}_i) = f(\vec{\theta}_i + \eta \vec{v}) - f(\vec{\theta}_i)
\]
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 $\eta$ is a (small) ‘step size’ and $\vec{v}$ is a direction chosen to minimize $f(\vec{\theta}_i + \eta \vec{v})$.

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

So for small $\eta$:

\[
f(\vec{\theta}_{i+1}) - f(\vec{\theta}_i) = f(\vec{\theta}_i + \eta \vec{v}) - f(\vec{\theta}_i) \approx \eta \cdot D_{\vec{v}} f(\vec{\theta}_i)
\]
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 \( \eta \) is a (small) ‘step size’ and \( \vec{v} \) is a direction chosen to minimize \( f(\vec{\theta}_i + \eta \vec{v}) \).

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

So for small \( \eta \):

\[
f(\vec{\theta}_{i+1}) - f(\vec{\theta}_i) = f(\vec{\theta}_i + \eta \vec{v}) - f(\vec{\theta}_i) \approx \eta \cdot D_{\vec{v}} f(\vec{\theta}_i) = \eta \cdot \langle \vec{v}, \nabla f(\vec{\theta}_i) \rangle.
\]

\[
f(\vec{\theta}_{i+1}) - f(\vec{\theta}_i) = 0
\]
Gradient descent is a greedy iterative optimization algorithm: Starting at \( \theta_1 \), in each iteration let \( \theta_{i+1} = \theta_i + \eta \vec{v} \), where \( \eta \) is a (small) ‘step size’ and \( \vec{v} \) is a direction chosen to minimize \( f(\theta_i + \eta \vec{v}) \).

\[
D_{\vec{v}} f(\theta_i) = \lim_{\epsilon \to 0} \frac{f(\theta_i + \epsilon \vec{v}) - f(\theta_i)}{\epsilon}.
\]

So for small \( \eta \):

\[
\min f(\theta_{i+1}) - f(\theta_i) = f(\theta_i + \eta \vec{v}) - f(\theta_i) \approx \eta \cdot D_{\vec{v}} f(\theta_i) = \eta \cdot \langle \vec{v}, \nabla f(\theta_i) \rangle.
\]

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

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

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

- For now assume $\eta$ stays the same in each iteration.
Gradient Descent Update: $\tilde{\theta}_{i+1} = \tilde{\theta}_i - \eta \nabla f(\tilde{\theta}_i)$
**Convex Functions:** After sufficient iterations, gradient descent will converge to a *approximate minimizer* \( \hat{\theta} \) with:

\[
f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon\]

Examples: least squares regression, logistic regression, sparse regression (lasso), regularized regression, SVMS,...
**Conditions for Gradient Descent Convergence**

**Convex Functions:** After sufficient iterations, gradient descent will converge to an **approximate minimizer** \( \hat{\theta} \) with:

\[
f(\hat{\theta}) \leq f(\bar{\theta}_*) + \epsilon = \min_{\bar{\theta}} f(\bar{\theta}) + \epsilon.
\]

Examples: least squares regression, logistic regression, sparse regression (lasso), regularized regression, SVMS,...
Convex Functions: After sufficient iterations, gradient descent will converge to a approximate minimizer $\hat{\theta}$ with:

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

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

![Diagram showing stationary points and local minimizers](image-url)
Gradient Descent Update: \( \tilde{\theta}_{i+1} = \tilde{\theta}_i - \eta \nabla f(\tilde{\theta}_i) \)
Both Convex and Non-convex: Need to assume the function is well-behaved in some way.
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.:
  \[
  \forall \theta : \quad \| \nabla f(\theta) \|_2 \leq G \iff \forall \theta_1, \theta_2 : \quad |f(\theta_1) - f(\theta_2)| \leq G \cdot \| \theta_1 - \theta_2 \|_2
  \]

  \[f(\theta) = \| \theta \|_{\infty}, \quad f'(\theta) = \text{sign}(\theta) \quad \theta > 0\]

  \[g = 1 \quad \| \theta_1 - \theta_2 \| \leq | \theta_1 - \theta_2 |\]

- Smooth/Lipschitz gradient (direction/size of gradient is not changing too quickly): There is some $\beta$ s.t.:
  \[
  \forall \theta_1, \theta_2 : \quad \| \nabla f(\theta_1) - \nabla f(\theta_2) \|_2 \leq \beta \cdot \| \theta_1 - \theta_2 \|_2.
  \]

  \[\theta_1 = \epsilon, \quad \theta_2 = -\epsilon \quad \| 1 - (-1) \| = 2 \quad | \epsilon - (-\epsilon) | = 2\epsilon\]
Gradient Descent analysis for convex functions.
Definition – Convex Function: A function \( f : \mathbb{R}^d \rightarrow \mathbb{R} \) is convex if and only if, for any \( \theta_1, \theta_2 \in \mathbb{R}^d \) and \( \lambda \in [0, 1] \):

\[
(1 - \lambda) \cdot f(\theta_1) + \lambda \cdot f(\theta_2) \geq f \left( (1 - \lambda) \cdot \theta_1 + \lambda \cdot \theta_2 \right)
\]
Corollary – Convex Function: A function \( f : \mathbb{R}^d \rightarrow \mathbb{R} \) is convex if and only if, for any \( \vec{\theta}_1, \vec{\theta}_2 \in \mathbb{R}^d \) and \( \lambda \in [0, 1] \):

\[
\begin{align*}
\forall \vec{\theta}_1, \vec{\theta}_2 \in \mathbb{R}^d \quad \lambda \in [0, 1] : \\
f(\vec{\theta}_2) - f(\vec{\theta}_1) &\geq \nabla f(\vec{\theta}_1)^T (\vec{\theta}_2 - \vec{\theta}_1) \\
f'(\vec{\theta}_1) &\leq \frac{f(\vec{\theta}_2) - f(\vec{\theta}_1)}{\vec{\theta}_2 - \vec{\theta}_1} \\
f'(\vec{\theta}_1) \cdot (\vec{\theta}_2 - \vec{\theta}_1) &\leq f(\vec{\theta}_2) - f(\vec{\theta}_1)
\end{align*}
\]
Assume that:

- $f$ is convex.
- $f$ is $G$ Lipschitz ($\|\nabla f(\theta)\|_2 \leq G$ for all $\theta$).
- $\|\theta_1 - \theta_*\|_2 \leq R$ where $\theta_1$ is the initialization point.

**Gradient Descent**

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

$$f(\hat{\theta}) \leq f(\theta_*) + \epsilon.$$
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 $\theta_*$, 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 $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius $R$ of $\theta_*$, outputs $\hat{\theta}$ satisfying:

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

Step 1: For all $i$, $f(\theta_i) - f(\theta_*) \leq \frac{||\theta_i - \theta_*||^2}{2\eta} + \frac{\eta G^2}{2}$. 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 $\theta_*$, 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: $\nabla f(\theta_i)^T(\theta_i - \theta_*) \leq \frac{\|\theta_i - \theta_*\|_2^2 - \|\theta_{i+1} - \theta_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$.
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 $\theta_*$, 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: $\nabla f(\theta_i)^T(\theta_i - \theta_*) \leq \frac{||\theta_i - \theta_*||_2^2 - ||\theta_{i+1} - \theta_*||_2^2}{2\eta} + \frac{\eta G^2}{2} \implies \text{Step 1.}$
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 $\theta_*$, outputs $\hat{\theta}$ satisfying:

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

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

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

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

**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 $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G \sqrt{t}}$, and starting point within radius $R$ of $\theta_*$, 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.

\[ \theta^* = \arg \min_{\theta \in S} f(\theta), \]

where \( S \) is a convex set.
Often want to perform convex optimization with convex constraints.

\[ \theta^* = \arg \min_{\theta \in S} f(\theta), \]

where \( S \) is a convex set.

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

\[ (1 - \lambda)\vec{\theta}_1 + \lambda \cdot \vec{\theta}_2 \in S \]
Often want to perform convex optimization with convex constraints.

\[ \theta^* = \arg \min_{\theta \in S} f(\theta), \]

where \( S \) is a convex set.

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

\[ (1 - \lambda)\bar{\theta}_1 + \lambda \cdot \bar{\theta}_2 \in S \]

E.g. \( S = \{ \bar{\theta} \in \mathbb{R}^d : \|\bar{\theta}\|_2 \leq 1 \} \).
For any convex set let $P_S(\cdot)$ denote the projection function onto $S$.

\[ P_S(\bar{y}) = \operatorname{arg\,min}_{\hat{\theta} \in S} \| \hat{\theta} - \bar{y} \|_2. \]
For any convex set let $P_S(\cdot)$ denote the projection function onto $S$.

- $P_S(\bar{y}) = \arg \min_{\tilde{\theta} \in S} \| \tilde{\theta} - \bar{y} \|_2$.
- For $S = \{ \tilde{\theta} \in \mathbb{R}^d : \| \tilde{\theta} \|_2 \leq 1 \}$ what is $P_S(\bar{y})$?
For any convex set let $P_S(\cdot)$ denote the projection function onto $S$.

- $P_S(\bar{y}) = \arg\min_{\tilde{\theta} \in S} \|\tilde{\theta} - \bar{y}\|_2$.
- For $S = \{\tilde{\theta} \in \mathbb{R}^d : \|\tilde{\theta}\|_2 \leq 1\}$ what is $P_S(\bar{y})$?
- For $S$ being a $k$ dimensional subspace of $\mathbb{R}^d$, what is $P_S(\bar{y})$?
For any convex set let $P_S(\cdot)$ denote the projection function onto $S$.

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

Projected Gradient Descent

- Choose some initialization $\bar{\theta}_1$ and set $\eta = \frac{R}{G \sqrt{t}}$.
- For $i = 1, \ldots, t - 1$
  - $\bar{\theta}^{(\text{out})}_{i+1} = \bar{\theta}_i - \eta \cdot \nabla f(\bar{\theta}_i)$
  - $\bar{\theta}_{i+1} = P_S(\bar{\theta}^{(\text{out})}_{i+1})$.
- Return $\hat{\theta} = \arg \min_{\bar{\theta}_i} f(\bar{\theta}_i)$. 
Visually:
Projected gradient descent can be analyzed identically to gradient descent!
Projected gradient descent can be analyzed identically to gradient descent!

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

$$\|P_S(\vec{y}) - \vec{\theta}\|_2 \leq \|\vec{y} - \vec{\theta}\|_2.$$
Theorem – Projected GD: For convex G-Lipschitz function $f$, and convex set $S$, 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 $\theta_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\theta_*) + \epsilon = \min_{\theta \in S} f(\theta) + \epsilon$$
Theorem – Projected GD: For convex $G$-Lipschitz function $f$, and convex set $S$, 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 $\theta_*$, outputs $\hat{\theta}$ satisfying:

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

Recall: $\theta_{i+1}^{(out)} = \theta_i - \eta \cdot \nabla f(\theta_i)$ and $\theta_{i+1} = P_S(\theta_{i+1}^{(out)})$. 
Theorem – Projected GD: For convex G-Lipschitz function $f$, and convex set $S$, 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 $\theta_*$, outputs $\hat{\theta}$ satisfying:

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

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

Step 1: For all $i$, $f(\theta_i) - f(\theta_*) \leq \frac{\|\theta_i - \theta_*\|^2 - \|\theta_{i+1}^{(out)} - \theta_*\|^2}{2\eta} + \frac{\eta G^2}{2}$. 
Theorem – Projected GD: For convex $G$-Lipschitz function $f$, and convex set $S$, 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 $\theta_*$, outputs $\hat{\theta}$ satisfying:

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

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

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

Step 1.a: For all $i$, $f(\theta_i) - f(\theta_*) \leq \frac{\|\theta_i - \theta_*\|^2 - \|\theta_{i+1} - \theta_*\|^2}{2\eta} + \frac{\eta G^2}{2}$.
Theorem – Projected GD: For convex G-Lipschitz function $f$, and convex set $S$, 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 $\theta_*$, outputs $\hat{\theta}$ satisfying:

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

Recall: $\theta^{(out)}_{i+1} = \theta_i - \eta \cdot \nabla f(\theta_i)$ and $\theta_{i+1} = P_S(\theta^{(out)}_{i+1})$.

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

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

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