COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Andrew McGregor
Lecture 24
Last Class:

- Analysis of gradient descent for optimizing convex functions.
- Introduction to convex sets and projection functions.
- (The same) analysis of projected gradient descent for optimizing under convex functions under (convex) constraints.

This Class:

- Online learning, regret, and online gradient descent.
- Application to stochastic 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.
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 $L(\vec{\theta}, \vec{X}) = \sum_{i=1}^{n} \ell(\vec{\theta}, \vec{x}_i)$, when data points are presented in an online fashion $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ (similar to streaming algorithms)
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 $L(\vec{\theta}, \vec{X}) = \sum_{i=1}^{n} \ell(\vec{\theta}, \vec{x}_i)$, when data points are presented in an online fashion $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ (similar to streaming algorithms)

Stochastic gradient descent is a special case: when data points are considered a random order for computational reasons.
Online Optimization: In place of a single function $f$, we see a different objective function at each step:

$$f_1, f_2, \ldots, f_t : \mathbb{R}^d \to \mathbb{R}$$
Online Optimization: In place of a single function $f$, we see a different objective function at each step:

$$f_1, f_2, \ldots, f_t : \mathbb{R}^d \rightarrow \mathbb{R}$$

- At each step, first pick (play) a parameter vector $\vec{\theta}^{(i)}$.
- Then are told $f_i$ and incur cost $f_i(\vec{\theta}^{(i)})$.
- **Goal:** Minimize total cost $\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)})$. 
Online Optimization: In place of a single function $f$, we see a different objective function at each step:

$$f_1, f_2, \ldots, f_t : \mathbb{R}^d \to \mathbb{R}$$

- At each step, first pick (play) a parameter vector $\theta^{(i)}$.
- Then are told $f_i$ and incur cost $f_i(\theta^{(i)})$.
- **Goal:** Minimize total cost $\sum_{i=1}^{t} f_i(\theta^{(i)})$.

Our analysis will make no assumptions on how $f_1, \ldots, f_t$ are related to each other!
Home pricing tools.

- Parameter vector $\vec{\theta}^{(i)}$: coefficients of linear model at step $i$.
- Functions $f_1, \ldots, f_t$: $f_i(\vec{\theta}^{(i)}) = (\langle \vec{x}, \vec{\theta}^{(i)} \rangle - \text{price}_i)^2$ revealed when home$_i$ is listed or sold.
- Want to minimize total squared error $\sum_{i=1}^t f_i(\vec{\theta}^{(i)})$ (same as classic least squares regression).
ONLINE OPTIMIZATION EXAMPLE

UI design via online optimization.

- Parameter vector $\vec{\theta}^{(i)}$: some encoding of the layout at step $i$.
- Functions $f_1, \ldots, f_t$: $f_i(\vec{\theta}^{(i)}) = 1$ if user does not click ‘add to cart’ and $f_i(\vec{\theta}^{(i)}) = 0$ if they do click.
- Want to maximize number of purchases, i.e., minimize $\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)})$. 
In normal optimization, we seek $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq \min_{\bar{\theta}} f(\bar{\theta}) + \epsilon.$$
In normal optimization, we seek $\hat{\theta}$ satisfying:

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

In online optimization we will ask for the same.

$$\sum_{i=1}^{t} f_i(\hat{\theta}^{(i)}) \leq \sum_{i=1}^{t} f_i(\bar{\theta}^{off}) + \epsilon$$

where $\bar{\theta}^{off} = \arg\min_{\bar{\theta}} \sum_{i=1}^{t} f_i(\bar{\theta})$ and $\epsilon$ is called the regret and $\epsilon/t$ is the average regret.
In normal optimization, we seek $\hat{\theta}$ satisfying:

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

In online optimization we will ask for the same.

$$\sum_{i=1}^{t} f_i(\bar{\theta}^{(i)}) \leq \sum_{i=1}^{t} f_i(\bar{\theta}^{\text{off}}) + \epsilon$$

where $\bar{\theta}^{\text{off}} = \arg\min_{\bar{\theta}} \sum_{i=1}^{t} f_i(\bar{\theta})$ and $\epsilon$ is called the regret and $\epsilon/t$ is the average regret.

- This error metric is a bit unusual: Comparing online solution to best fixed “online” solution in hindsight. $\epsilon$ can be negative!
What if for $i = 1, \ldots, t$, $f_i(\theta) = |\theta - 1000|$ or $f_i(\theta) = |\theta + 1000|$ in an alternating pattern?

How small can the regret $\epsilon$ be? $\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)}) \leq \sum_{i=1}^{t} f_i(\vec{\theta}^{\text{off}}) + \epsilon$. 
What if for $i = 1, \ldots, t$, $f_i(\theta) = |\theta - 1000|$ or $f_i(\theta) = |\theta + 1000|$ in an alternating pattern?

How small can the regret $\epsilon$ be? $\sum_{i=1}^{t} f_i(\vec{\theta}(i)) \leq \sum_{i=1}^{t} f_i(\vec{\theta}^{off}) + \epsilon$.

What if for $i = 1, \ldots, t$, $f_i(\theta) = |\theta - 1000|$ or $f_i(\theta) = |\theta + 1000|$ in no particular pattern? How can any online learning algorithm hope to achieve small regret?
Online Gradient Descent

Assume that:

• $f_1, \ldots, f_t$ are all convex.
• Each $f_i$ is $G$-Lipschitz, i.e., $\|\nabla f_i(\vec{\theta})\|_2 \leq G$ for all $\vec{\theta}$.
• $\|\vec{\theta}^{(1)} - \vec{\theta}^{\text{off}}\|_2 \leq R$ where $\theta^{(1)}$ is the first vector chosen.
Assume that:

- \( f_1, \ldots, f_t \) are all convex.
- Each \( f_i \) is \( G \)-Lipschitz, i.e., \( \| \nabla f_i(\tilde{\theta}) \|_2 \leq G \) for all \( \tilde{\theta} \).
- \( \| \tilde{\theta}^{(1)} - \tilde{\theta}^{\text{off}} \|_2 \leq R \) where \( \tilde{\theta}^{(1)} \) is the first vector chosen.

**Online Gradient Descent**

- Pick some initial \( \tilde{\theta}^{(1)} \).
- Set step size \( \eta = \frac{R}{G \sqrt{t}} \).
- For \( i = 1, \ldots, t \)
  - Play \( \tilde{\theta}^{(i)} \) and incur cost \( f_i(\tilde{\theta}^{(i)}) \).
  - \( \tilde{\theta}^{(i+1)} = \tilde{\theta}^{(i)} - \eta \cdot \nabla f_i(\tilde{\theta}^{(i)}) \)
Theorem: For convex $G$-Lipschitz $f_1, \ldots, f_t$, OGD initialized with starting point $\theta^{(1)}$ within radius $R$ of $\theta^{\text{off}}$, using step size $\eta = \frac{R}{G\sqrt{t}}$, has regret bounded by:

$$\left[ \sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{\text{off}}) \right] \leq RG\sqrt{t}$$
Theorem: For convex $G$-Lipschitz $f_1, \ldots, f_t$, OGD initialized with starting point $\theta^{(1)}$ within radius $R$ of $\theta^{\text{off}}$, using step size $\eta = \frac{R}{G\sqrt{t}}$, has regret bounded by:

\[
\left[ \sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{\text{off}}) \right] \leq RG\sqrt{t}
\]

Upper bound on average regret goes to 0 and $t \to \infty$. 
Theorem: For convex $G$-Lipschitz $f_1, \ldots, f_t$, OGD initialized with starting point $\theta^{(1)}$ within radius $R$ of $\theta^{\text{off}}$, using step size $\eta = \frac{R}{G\sqrt{t}}$, has regret bounded by:

$$\left[\sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{\text{off}})\right] \leq RG\sqrt{t}$$

Upper bound on average regret goes to 0 and $t \to \infty$. No assumptions on $f_1, \ldots, f_t$!
**Theorem:** For convex $G$-Lipschitz $f_1, \ldots, f_t$, OGD initialized with starting point $\theta^{(1)}$ within radius $R$ of $\theta^{\text{off}}$, using step size $\eta = \frac{R}{G \sqrt{t}}$, has regret bounded by:

$$\left[ \sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{\text{off}}) \right] \leq RG \sqrt{t}$$

Upper bound on average regret goes to 0 and $t \to \infty$. No assumptions on $f_1, \ldots, f_t$!

**Step 1.1:** For all $i$, $\nabla f_i(\theta^{(i)})^T (\theta^{(i)} - \theta^{\text{off}}) \leq \frac{\|\theta^{(i)} - \theta^{\text{off}}\|^2}{2\eta} - \frac{\|\theta^{(i+1)} - \theta^{\text{off}}\|^2}{2} + \frac{\eta G^2}{2}$. 


Theorem: For convex $G$-Lipschitz $f_1, \ldots, f_t$, OGD initialized with starting point $\theta^{(1)}$ within radius $R$ of $\theta^{\text{off}}$, using step size $\eta = \frac{R}{G\sqrt{t}}$, has regret bounded by:

$$ \left[ \sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{\text{off}}) \right] \leq RG \sqrt{t} $$

Upper bound on average regret goes to $0$ and $t \to \infty$. No assumptions on $f_1, \ldots, f_t$!

**Step 1.1:** For all $i$, \n\n$$ \nabla f_i(\theta^{(i)})^T (\theta^{(i)} - \theta^{\text{off}}) \leq \frac{\|\theta^{(i)} - \theta^{\text{off}}\|_2^2 - \|\theta^{(i+1)} - \theta^{\text{off}}\|_2^2}{2\eta} + \frac{\eta G^2}{2} \n$$

Convexity $\implies$ **Step 1:** For all $i$,

$$ f_i(\theta^{(i)}) - f_i(\theta^{\text{off}}) \leq \frac{\|\theta^{(i)} - \theta^{\text{off}}\|_2^2 - \|\theta^{(i+1)} - \theta^{\text{off}}\|_2^2}{2\eta} + \frac{\eta G^2}{2}. $$
**Theorem:** For convex $G$-Lipschitz $f_1, \ldots, f_t$, OGD initialized with starting point $\theta^{(1)}$ within radius $R$ of $\theta^{\text{off}}$, using step size $\eta = \frac{R}{G\sqrt{t}}$, has regret bounded by:

$$
\left[ \sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{\text{off}}) \right] \leq RG \sqrt{t}
$$

**Step 1:** For all $i$, $f_i(\theta^{(i)}) - f_i(\theta^{\text{off}}) \leq \frac{\|\theta^{(i)} - \theta^{\text{off}}\|_2^2 - \|\theta^{(i+1)} - \theta^{\text{off}}\|_2^2}{2\eta} + \frac{\eta G^2}{2}$
Theorem: For convex $G$-Lipschitz $f_1, \ldots, f_t$, OGD initialized with starting point $\theta^{(1)}$ within radius $R$ of $\theta^{\text{off}}$, using step size $\eta = \frac{R}{G\sqrt{t}}$, has regret bounded by:

$$\left[ \sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{\text{off}}) \right] \leq RG\sqrt{t}$$

Step 1: For all $i$, $f_i(\theta^{(i)}) - f_i(\theta^{\text{off}}) \leq \frac{\|\theta^{(i)} - \theta^{\text{off}}\|_2^2 - \|\theta^{(i+1)} - \theta^{\text{off}}\|_2^2}{2\eta} + \frac{\eta G^2}{2}$

$$\sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{\text{off}}) \leq \sum_{i=1}^{t} \frac{\|\theta^{(i)} - \theta^{\text{off}}\|_2^2 - \|\theta^{(i+1)} - \theta^{\text{off}}\|_2^2}{2\eta} + \frac{t \cdot \eta G^2}{2}$$

$$= \frac{\|\theta^{(1)} - \theta^{\text{off}}\|_2^2 - \|\theta^{(t+1)} - \theta^{\text{off}}\|_2^2}{2\eta} + \frac{t \cdot \eta G^2}{2}$$

$$\leq \frac{R^2}{2\eta} + \frac{t \eta G^2}{2} = RG\sqrt{t}$$
Stochastic gradient descent is an efficient offline optimization method, seeking $\hat{\theta}$ with

$$f(\hat{\theta}) \leq \min_{\bar{\theta}} f(\bar{\theta}) + \epsilon$$
Stochastic gradient descent is an efficient offline optimization method, seeking $\hat{\theta}$ with

$$f(\hat{\theta}) \leq \min_{\tilde{\theta}} f(\tilde{\theta}) + \epsilon$$

- The most popular optimization method in modern machine learning. Easily analyzed as a special case of online gradient descent!
Stochastic gradient descent is an efficient offline optimization method, seeking $\hat{\theta}$ with

$$f(\hat{\theta}) \leq \min_{\tilde{\theta}} f(\tilde{\theta}) + \epsilon$$

- The most popular optimization method in modern machine learning. Easily analyzed as a special case of online gradient descent!
- **Basic Idea:** In gradient descent, we set $\tilde{\theta}_{i+1} = \tilde{\theta}_i - \eta \cdot \nabla f(\tilde{\theta}_i)$. 

---

**STOCHASTIC GRADIENT DESCENT**
Stochastic gradient descent is an efficient offline optimization method, seeking $\hat{\theta}$ with

$$f(\hat{\theta}) \leq \min_{\tilde{\theta}} f(\tilde{\theta}) + \epsilon$$

- The most popular optimization method in modern machine learning. Easily analyzed as a special case of online gradient descent!
- **Basic Idea:** In gradient descent, we set $\tilde{\theta}_{i+1} = \tilde{\theta}_i - \eta \cdot \tilde{\nabla} f(\tilde{\theta}_i)$. In stochastic gradient descent we don’t compute $\tilde{\nabla} f(\tilde{\theta}_i)$ exactly but instead do something random that is correct in expectation. This saves time per step but might increase the number of steps.
Stochastic Gradient Descent

Assume that:

- \( f \) is convex and decomposable as \( f(\vec{\theta}) = \sum_{j=1}^{n} f_j(\vec{\theta}) \).

$\sum_{j=1}^{n}$
Assume that:

- $f$ is convex and decomposable as $f(\vec{\theta}) = \sum_{j=1}^{n} f_j(\vec{\theta})$.
  - For example, trying to minimize a loss function over a data set $X$,
    $L(\vec{\theta}, X) = \sum_{j=1}^{n} \ell(\vec{\theta}, \vec{x}_j)$ that is a sum of losses of element in data set.
- Each $f_j$ is $\frac{G}{n}$-Lipschitz:
STOCHASTIC GRADIENT DESCENT

Assume that:

• $f$ is convex and decomposable as $f(\vec{\theta}) = \sum_{j=1}^{n} f_j(\vec{\theta})$.
  • For example, trying to minimize a loss function over a data set $X$,
    $$L(\vec{\theta}, X) = \sum_{j=1}^{n} \ell(\vec{\theta}, \vec{x}_j)$$
    that is a sum of losses of element in data set.
  • Each $f_j$ is $\frac{G}{n}$-Lipschitz:
    $$\|\nabla f(\vec{\theta})\|_2 \leq \|\sum_{j=1}^{n} \nabla f_j(\vec{\theta})\|_2 \leq \sum_{j=1}^{n} \|\nabla f_j(\vec{\theta})\|_2 \leq G .$$

• Initialize with $\theta^{(1)}$ satisfying $\|\vec{\theta}^{(1)} - \vec{\theta}^*\|_2 \leq R$. 

Assume that:

- \( f \) is convex and decomposable as \( f(\vec{\theta}) = \sum_{j=1}^{n} f_j(\vec{\theta}) \).
  - For example, trying to minimize a loss function over a data set \( X \),
    \( L(\vec{\theta}, X) = \sum_{j=1}^{n} \ell(\vec{\theta}, \vec{x}_j) \) that is a sum of losses of element in data set.
- Each \( f_j \) is \( \frac{G}{n} \)-Lipschitz:
  \[
  \|\nabla f(\vec{\theta})\|_2 \leq \|\sum_{j=1}^{n} \nabla f_j(\vec{\theta})\|_2 \leq \sum_{j=1}^{n} \|\nabla f_j(\vec{\theta})\|_2 \leq G.
  \]
- Initialize with \( \theta^{(1)} \) satisfying \( \|\vec{\theta}^{(1)} - \vec{\theta}^*\|_2 \leq R \).

**Stochastic Gradient Descent**

- Pick some initial \( \vec{\theta}^{(1)} \).
- Set step size \( \eta = \frac{R}{G\sqrt{t}} \).
- For \( i = 1, \ldots, t \)
  - Pick random \( j_i \in 1, \ldots, n \).
  - \( \vec{\theta}^{(i+1)} = \vec{\theta}^{(i)} - \eta \cdot \nabla f_{j_i}(\vec{\theta}^{(i)}) \)
- Return \( \hat{\theta} = \frac{1}{t} \sum_{i=1}^{t} \vec{\theta}^{(i)} \).
STOCHASTIC GRADIENT DESCENT

$$\vec{\theta}^{(i+1)} = \vec{\theta}^{(i)} - \eta \cdot \hat{\nabla} f_i(\vec{\theta}^{(i)}) \quad \text{vs.} \quad \vec{\theta}^{(i+1)} = \vec{\theta}^{(i)} - \eta \cdot \hat{\nabla} f(\vec{\theta}^{(i)})$$

**Note that:** $\mathbb{E}[\hat{\nabla} f_i(\vec{\theta}^{(i)})] = \frac{1}{n} \hat{\nabla} f(\vec{\theta}^{(i)})$.

Analysis extends to any algorithm that takes the gradient step in expectation (minibatch SGD, randomly quantized, measurement noise, differentially private, etc.)
Theorem – SGD on Convex Lipschitz Functions: SGD 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: $E[f(\hat{\theta})] \leq f(\theta^*) + \epsilon$. 
Theorem – SGD on Convex Lipschitz Functions: SGD 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: $\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon$.

Step 1: $f(\hat{\theta}) - f(\theta^*) \leq \frac{1}{t} \sum_{i=1}^{t} [f(\theta^{(i)}) - f(\theta^*)]$
Theorem – SGD on Convex Lipschitz Functions: SGD 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: \( \mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon. \)

Step 1: \( f(\hat{\theta}) - f(\theta^*) \leq \frac{1}{t} \sum_{i=1}^{t} [f(\theta^{(i)}) - f(\theta^*)] \) since

\[
f(\hat{\theta}) = f\left(\frac{\sum_{i=1}^{t} \theta^{(i)}}{t}\right) \leq \frac{1}{t} \sum_{i=1}^{t} f(\theta^{(i)}) \text{ by convexity (see homework)}
\]
Theorem – SGD on Convex Lipschitz Functions: SGD 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: $\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon$.

Step 1: $f(\hat{\theta}) - f(\theta^*) \leq \frac{1}{t} \sum_{i=1}^{t}[f(\theta^{(i)}) - f(\theta^*)]$ since

$$f(\hat{\theta}) = f\left(\frac{\sum_{i=1}^{t} \theta^{(i)}}{t}\right) \leq \frac{1}{t} \sum_{i=1}^{t} f(\theta^{(i)})$$

by convexity (see homework)

Step 2: $\mathbb{E}[f(\hat{\theta}) - f(\theta^*)] \leq \frac{n}{t} \cdot \mathbb{E} \left[\sum_{i=1}^{t}[f_{ji}(\theta^{(i)}) - f_{ji}(\theta^*)]\right]$
**Theorem – SGD on Convex Lipschitz Functions:** SGD 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: $\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon$.

**Step 1:** $f(\hat{\theta}) - f(\theta^*) \leq \frac{1}{t} \sum_{i=1}^{t} [f(\theta^{(i)}) - f(\theta^*)]$ since

$$f(\hat{\theta}) = f\left(\frac{\sum_{i=1}^{t} \theta^{(i)}}{t}\right) \leq \frac{1}{t} \sum_{i=1}^{t} f(\theta^{(i)})$$ by convexity (see homework)

**Step 2:** $\mathbb{E}[f(\hat{\theta}) - f(\theta^*)] \leq \frac{n}{t} \cdot \mathbb{E}\left[\sum_{i=1}^{t} [f_{ji}(\theta^{(i)}) - f_{ji}(\theta^*)]\right]$ since

$$\mathbb{E}[f_{ji}(\hat{\theta})] = \frac{1}{n} f(\hat{\theta})$$ since $f(\hat{\theta}) = \sum_{j=1}^{n} f_j(\hat{\theta})$
Theorem – SGD on Convex Lipschitz Functions: SGD 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: $\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon$.

Step 1: $f(\hat{\theta}) - f(\theta^*) \leq \frac{1}{t} \sum_{i=1}^{t} [f(\theta(i)) - f(\theta^*)]$ since

$$f(\hat{\theta}) = f\left(\frac{\sum_{i=1}^{t} \theta(i)}{t}\right) \leq \frac{1}{t} \sum_{i=1}^{t} f(\theta(i))$$ by convexity (see homework)

Step 2: $\mathbb{E}[f(\hat{\theta}) - f(\theta^*)] \leq \frac{n}{t} \cdot \mathbb{E} \left[ \sum_{i=1}^{t} [f_{j_i}(\theta(i)) - f_{j_i}(\theta^*)] \right]$ since

$$\mathbb{E}[f_{j_i}(\tilde{\theta})] = \frac{1}{n} f(\tilde{\theta}) \text{ since } f(\tilde{\theta}) = \sum_{j=1}^{n} f_j(\tilde{\theta})$$

Step 3: $\mathbb{E}[f(\hat{\theta}) - f(\theta^*)] \leq \frac{n}{t} \cdot \mathbb{E} \left[ \sum_{i=1}^{t} [f_{j_i}(\theta(i)) - f_{j_i}(\theta^{off})] \right]$. 

14
Stochastic Gradient Descent Analysis

Theorem – SGD on Convex Lipschitz Functions: SGD 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: \( \mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon. \)

Step 1: \( f(\hat{\theta}) - f(\theta^*) \leq \frac{1}{t} \sum_{i=1}^{t} [f(\theta^{(i)}) - f(\theta^*)] \) since
\[
f(\hat{\theta}) = f\left(\frac{\sum_{i=1}^{t} \theta^{(i)}}{t}\right) \leq \frac{1}{t} \sum_{i=1}^{t} f(\theta^{(i)}) \quad \text{by convexity (see homework)}
\]

Step 2: \( \mathbb{E}[f(\hat{\theta}) - f(\theta^*)] \leq \frac{n}{t} \cdot \mathbb{E} \left[ \sum_{i=1}^{t} [f_{ji}(\theta^{(i)}) - f_{ji}(\theta^*)] \right] \) since
\[
\mathbb{E}[f_{ji}(\hat{\theta})] = \frac{1}{n} f(\hat{\theta}) \quad \text{since} \quad f(\hat{\theta}) = \sum_{j=1}^{n} f_{j}(\hat{\theta})
\]

Step 3: \( \mathbb{E}[f(\hat{\theta}) - f(\theta^*)] \leq \frac{n}{t} \cdot \mathbb{E} \left[ \sum_{i=1}^{t} [f_{ji}(\theta^{(i)}) - f_{ji}(\theta^{\text{off}})] \right]. \)

Step 4: \( \mathbb{E}[f(\hat{\theta}) - f(\theta^*)] \leq \frac{n}{t} \cdot \frac{G}{n} \cdot \sqrt{t} = \frac{RG}{\sqrt{t}}. \)

\( \boxed{\text{OGD bound}} \)
Stochastic gradient descent generally makes more iterations than gradient descent.

Each iteration is much cheaper (by a factor of $n$).

$$\vec{\nabla} \sum_{j=1}^{n} f_j(\vec{\theta}) \text{ vs. } \vec{\nabla} f_j(\vec{\theta})$$
When \( f(\theta) = \sum_{j=1}^{n} f_j(\theta) \) and \( \| \nabla f_j(\theta) \|_2 \leq \frac{G}{n} \):

**Theorem – SGD:** After \( t \geq \frac{R^2 G^2}{\epsilon^2} \) iterations outputs \( \hat{\theta} \) satisfying:
\[
\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon.
\]

When \( \| \nabla f(\theta) \|_2 \leq \tilde{G} \):

**Theorem – GD:** After \( t \geq \frac{R^2 \tilde{G}^2}{\epsilon^2} \) iterations outputs \( \hat{\theta} \) satisfying:
\[
f(\hat{\theta}) \leq f(\theta^*) + \epsilon.
\]
SGD VS. GD: ITERATIONS

When $f(\theta) = \sum_{j=1}^{n} f_j(\theta)$ and $\|\nabla f_j(\theta)\|_2 \leq \frac{G}{n}$:

**Theorem – SGD:** After $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations outputs $\hat{\theta}$ satisfying:

$$\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon.$$  

When $\|\nabla f(\theta)\|_2 \leq \bar{G}$:

**Theorem – GD:** After $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations outputs $\hat{\theta}$ satisfying:

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

$$\|\nabla f(\theta)\|_2 = \|\nabla f_1(\theta) + \ldots + \nabla f_n(\theta)\|_2 \leq \sum_{j=1}^{n} \|\nabla f_j(\theta)\|_2 \leq n \cdot \frac{G}{n} \leq G$$ but $\bar{G}$ could be quite a bit smaller than $G$. 

CONTINUOUS OPTIMIZATION

- Foundational concepts like convexity (line between any two points on curve is above the curve), convex sets (line between any two points in set in the set), directional derivative (slope of curve if we move in particular direction), and Lipschitzness (slope is bounded).
• Foundational concepts like convexity (line between any two points on curve is above the curve), convex sets (line between any two points in set in the set), directional derivative (slope of curve if we move in particular direction), and Lipschitzness (slope is bounded).

• Gradient descent greedily tries to find the min value of function $f : \mathbb{R}^d \to \mathbb{R}$ by maintaining a vector $\tilde{\theta} \in \mathbb{R}^d$ and at each step moving $\tilde{\theta}$ “downhill”, i.e., in the direction that minimizes directional derivative
• Foundational concepts like convexity (line between any two points on curve is above the curve), convex sets (line between any two points in set in the set), directional derivative (slope of curve if we move in particular direction), and Lipschitzness (slope is bounded).

• Gradient descent greedily tries to find the min value of function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ by maintaining a vector $\vec{\theta} \in \mathbb{R}^d$ and at each step moving $\vec{\theta}$ “downhill”, i.e., in the direction that minimizes directional derivative.

• Bounded the number of steps required if $f$ is convex and Lipschitz.
• Foundational concepts like convexity (line between any two points on curve is above the curve), convex sets (line between any two points in set in the set), directional derivative (slope of curve if we move in particular direction), and Lipschitzness (slope is bounded).
• Gradient descent greedily tries to find the min value of function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ by maintaining a vector $\vec{\theta} \in \mathbb{R}^d$ and at each step moving $\vec{\theta}$ “downhill”, i.e., in the direction that minimizes directional derivative
• Bounded the number of steps required if $f$ is convex and Lipschitz.
• Simple extensions for optimization over a convex constraint set or online optimization.
• Foundational concepts like convexity (line between any two points on curve is above the curve), convex sets (line between any two points in set in the set), directional derivative (slope of curve if we move in particular direction), and Lipschitzness (slope is bounded).

• Gradient descent greedily tries to find the min value of function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ by maintaining a vector $\vec{\theta} \in \mathbb{R}^d$ and at each step moving $\vec{\theta}$ “downhill”, i.e., in the direction that minimizes directional derivative.

• Bounded the number of steps required if $f$ is convex and Lipschitz.

• Simple extensions for optimization over a convex constraint set or online optimization.

• Can typically speed up offline optimization via stochastic gradient descent: requires more iterations but each iteration is faster.
Foundational concepts like convexity (line between any two points on curve is above the curve), convex sets (line between any two points in set in the set), directional derivative (slope of curve if we move in particular direction), and Lipschitzness (slope is bounded).

Gradient descent greedily tries to find the min value of function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ by maintaining a vector $\vec{\theta} \in \mathbb{R}^d$ and at each step moving $\vec{\theta}$ “downhill”, i.e., in the direction that minimizes directional derivative.

Bounded the number of steps required if $f$ is convex and Lipschitz.

Simple extensions for optimization over a convex constraint set or online optimization.

Can typically speed up offline optimization via stochastic gradient descent: requires more iterations but each iteration is faster.

Lots that we didn’t cover: accelerated methods, adaptive methods, second order methods (quasi-Newton methods). Gave mathematical tools to understand these methods. See CS 690OP for more!