

COMPSCI 514: Algorithms for Data Science

Cameron Musco

University of Massachusetts Amherst. Spring 2026.

Lecture 22

- Problem Set 4 due Friday at 11:59pm.
- Final exam is next Tuesday 5/12, 1-3pm in the lecture hall.
- Study guide and review material has been posted.
- Thursday's class will be dedicated to exam review.
- I will hold additional exam review office hours **Monday 5/11, 1:30-2:30pm.**
- Please fill out SRTIs for this class!

Summary

Last Class:



- Analysis of gradient descent for **convex** and **Lipschitz** functions.

This Class:

- Direct extension to gradient descent to constrained optimization via **projected gradient descent**.
- Analysis for **convex functions** and **convex constraint sets**.
- Online optimization and online gradient descent.
- Application to analysis of stochastic gradient descent, for gradient descent at scale.

Constrained Convex Optimization

Often want to perform **convex optimization with convex constraints**.

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

$$\theta^* = \arg \min f(\theta)$$

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

$$\mathcal{S}: \{ \theta : \|\theta\|_2 \leq 1 \}$$

$$\mathcal{S}: \{ \theta : \|\theta\|_0 \leq k \}$$

$$\mathcal{S}: \{ \theta : \theta^\top \mathbf{1} = 0, \|\theta\|_2 = 1 \}$$

Constrained Convex Optimization

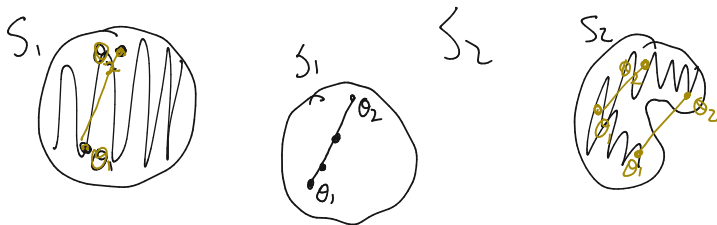
Often want to perform **convex optimization with convex constraints**.

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

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

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



Constrained Convex Optimization

Often want to perform **convex optimization with convex constraints**.

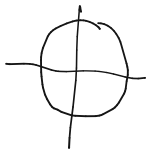
$$\frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} -2 \\ 3 \\ 0 \end{bmatrix} = \begin{bmatrix} -.5 \\ -1.5 \\ .5 \end{bmatrix} \notin \mathcal{S} \quad \vec{\theta}^* = \arg \min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}),$$

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



$$\|\theta_1\|_2 \leq 1$$

$$\|\theta_2\|_2 \leq 1$$

$$\downarrow$$
$$\|\lambda\theta_1 + (1-\lambda)\theta_2\|_2 \leq 1$$

$$\leq \|\lambda\theta_1\|_2 + \|(1-\lambda)\theta_2\|_2 \leq \lambda + (1-\lambda) = 1$$

$$\mathcal{S}: \{1, 2, 3, \dots\}$$

$$\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 2 \notin \mathcal{S}$$

$\exists \vec{\theta} \in \mathbb{R}^d : \|\theta\|_0 \leq 1$
not convex. 4

Projection Function

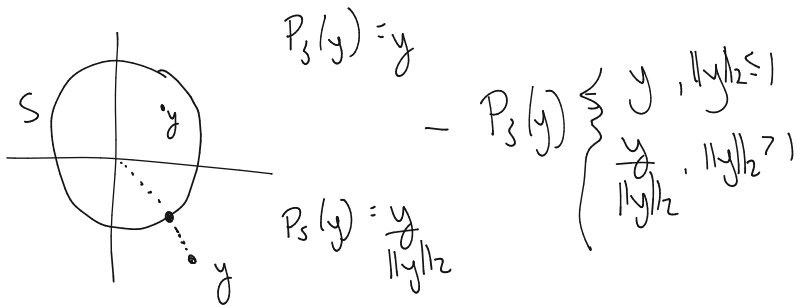
For any convex set let $P_{\mathcal{S}}(\cdot)$ denote the projection function onto \mathcal{S} .

- $P_{\mathcal{S}}(\vec{y}) = \arg \min_{\vec{\theta} \in \mathcal{S}} \|\vec{\theta} - \vec{y}\|_2.$

Projection Function

For any convex set let $P_S(\cdot)$ denote the projection function onto S .

- $P_S(\vec{y}) = \arg \min_{\vec{\theta} \in S} \|\vec{\theta} - \vec{y}\|_2$.
- For $S = \{\vec{\theta} \in \mathbb{R}^d : \|\vec{\theta}\|_2 \leq 1\}$ what is $P_S(\vec{y})$?



Projection Function

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

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

$$P_S(\vec{y}) = VV^T \vec{y} \quad \text{where } V \in \mathbb{R}^{d \times k} \text{ orthonormal} \\ \text{basis for subspace}$$

Convex Projections

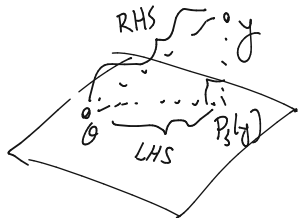
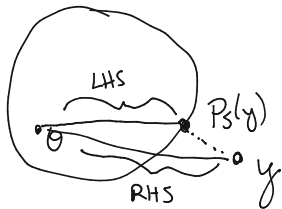
Projected gradient descent can be analyzed identically to gradient descent!

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 $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

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

Projected Gradient Descent Analysis

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

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

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

Projected Gradient Descent Analysis

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

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

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

error at Step i \leq progress towards θ^* at step i

(last class)

Projected Gradient Descent Analysis

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

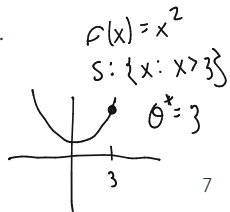
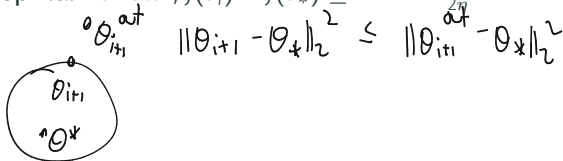
$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon = \min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}) + \epsilon$$

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

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

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



Projected Gradient Descent Analysis

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

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

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

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.

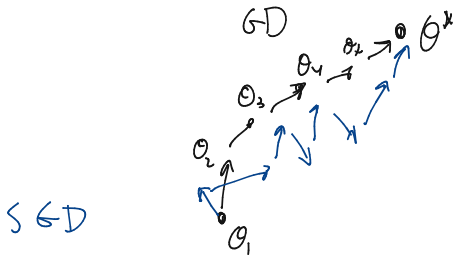
Gradient Descent at Scale: Online and Stochastic GD

Gradient Descent At Scale

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

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

The gradient of $L(\vec{\theta})$ has one component per data point so can be very expensive to compute.



Gradient Descent At Scale

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

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

The gradient of $L(\vec{\theta}, \mathbf{X})$ has one component per data point so can be very expensive to compute.

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

- If the data point is chosen uniformly at random, the sampled gradient is **correct in expectation**.

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

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

Online Gradient Descent

SGD is closely related to **online gradient descent**.

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.

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 $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ (like in streaming algorithms)

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 $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ (like in streaming algorithms)

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

Online Optimization Formal Setup

Online Optimization: In place of a single function f , we see a different objective function at each step:

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

$$f_i = \mathcal{L}(m_{\theta}(x_i) - y_i)$$

Online Optimization Formal Setup

Online Optimization: In place of a single function f , we see a different objective function at each step:

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

- 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 $\sum_{i=1}^t f_i(\vec{\theta}^{(i)})$.

No assumptions on how f_1, \dots, f_t are related to each other!

Online Optimization Example

UI design via online optimization.



- Parameter vector $\vec{\theta}^{(i)}$: some encoding of the layout at step i .
- Functions f_1, \dots, 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 $\sum_{i=1}^t f_i(\vec{\theta}^{(i)})$

Online Optimization Example

Home pricing tools.



linear model

$$\langle \vec{x}, \vec{\theta} \rangle$$



\$275,000

$$\vec{x} = [\#baths, \#beds, \#floors \dots]$$

- Parameter vector $\vec{\theta}^{(i)}$: coefficients of linear model at step i .
- Functions f_1, \dots, f_t : $f_i(\vec{\theta}^{(i)}) = (\vec{\theta}^{(i)} - price_i)^2$ 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).

In normal optimization, we seek $\hat{\theta}$ satisfying:

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

Regret

In normal optimization, we seek $\hat{\theta}$ satisfying:

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

In online optimization we will ask for the same.

$$\sum_{i=1}^t f_i(\vec{\theta}^{(i)}) \leq \min_{\vec{\theta}} \sum_{i=1}^t f_i(\vec{\theta}) + \epsilon = \sum_{i=1}^t f_i(\underbrace{\vec{\theta}^{off}}) + \epsilon$$

ϵ is called the **regret**.

Regret

In normal optimization, we seek $\hat{\theta}$ satisfying:

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

In online optimization we will ask for the same.

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

ϵ is called the **regret**. $\sum f_i(\theta^i) < \sum f_i(\theta^{\text{off}})$

- This error metric is a bit 'unfair'. **Why?**

Regret can be negative

Regret

In normal optimization, we seek $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq \min_{\vec{\theta}} f(\vec{\theta}) + \epsilon. \quad f(\hat{\theta}) \geq \min_{\theta \in \mathcal{M}} f(\theta)$$

In online optimization we will ask for the same.

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

ϵ is called the **regret**. $\sum f_i(\theta^i) \leq \min_{\theta_1, \dots, \theta_t} \sum f_i(\theta^i)$

- This error metric is a bit 'unfair'. **Why?**
- Comparing online solution to best fixed solution in hindsight. ϵ can be negative!

Assume that:

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

Online Gradient Descent

Assume that:

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

Online Gradient Descent

- Set step size $\eta = \frac{R}{G\sqrt{t}}$.
- For $i = 1, \dots, t$
 - Play $\vec{\theta}^{(i)}$ and incur cost $f_i(\vec{\theta}^{(i)})$.
 - $\vec{\theta}^{(i+1)} = \vec{\theta}^{(i)} - \eta \cdot \vec{\nabla} f_i(\vec{\theta}^{(i)})$

Online Gradient Descent Analysis

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

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

loss on training point i

$$f(\theta) = \sum f_i(\theta)$$

total training loss

$$= \sum \ell(\theta, x_i, y_i)$$

Online Gradient Descent Analysis

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

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

Average regret goes to 0 and $t \rightarrow \infty$. 'Sublinear regret' or 'no regret' algorithm.

$$\frac{1}{t} \left(\sum_{i=1}^t f_i(\theta^{(i)}) - \sum_{i=1}^t f_i(\theta^*) \right) \leq \frac{RG}{\sqrt{t}}$$

Online Gradient Descent Analysis

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

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

Average regret goes to 0 and $t \rightarrow \infty$. 'Sublinear regret' or 'no regret' algorithm. No assumptions on how f_1, \dots, f_t are related to each other!

Online Gradient Descent Analysis

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

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

Average regret goes to 0 and $t \rightarrow \infty$. ‘Sublinear regret’ or ‘no regret’ algorithm. No assumptions on how f_1, \dots, f_t are related to each other!

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

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

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

Online Gradient Descent Analysis

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

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

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

Online Gradient Descent Analysis

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

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

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

$$\left[\sum_{i=1}^t f_i(\theta^{(i)}) - \sum_{i=1}^t f_i(\theta^{off}) \right] \leq \sum_{i=1}^t \left(\frac{\|\theta^{(i)} - \theta^{off}\|_2^2 - \|\theta^{(i+1)} - \theta^{off}\|_2^2}{2\eta} + \frac{\eta G^2}{2} \right).$$

Recall: Stochastic gradient descent is an efficient offline optimization method, seeking $\hat{\theta}$ with

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

Stochastic Gradient Descent

Recall: Stochastic gradient descent is an efficient offline optimization method, seeking $\hat{\theta}$ with

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

Easily analyzed as a special case of online gradient descent!

Stochastic Gradient Descent

Assume that:

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

Stochastic Gradient Descent

Assume that:

- f is convex and decomposable as $f(\vec{\theta}) = \sum_{j=1}^n f_j(\vec{\theta})$.
- E.g., $L(\vec{\theta}, \mathbf{X}) = \sum_{j=1}^n \ell(M_{\vec{\theta}}(\vec{x}_j), y_j)$.

Stochastic Gradient Descent

Assume that:

- f is convex and decomposable as $f(\vec{\theta}) = \sum_{j=1}^n f_j(\vec{\theta})$.
 - E.g., $L(\vec{\theta}, \mathbf{X}) = \sum_{j=1}^n \ell(M_{\vec{\theta}}(\vec{x}_j), y_j)$.
- Each f_j is $\frac{G}{n}$ -Lipschitz (i.e., $\|\vec{\nabla} f_j(\vec{\theta})\|_2 \leq \frac{G}{n}$ for all $\vec{\theta}$.)

Stochastic Gradient Descent

Assume that:

- f is convex and decomposable as $f(\vec{\theta}) = \sum_{j=1}^n f_j(\vec{\theta})$.
 - E.g., $L(\vec{\theta}, \mathbf{X}) = \sum_{j=1}^n \ell(M_{\vec{\theta}}(\vec{x}_j), y_j)$.
- Each f_j is $\frac{G}{n}$ -Lipschitz (i.e., $\|\vec{\nabla} f_j(\vec{\theta})\|_2 \leq \frac{G}{n}$ for all $\vec{\theta}$)
 - What does this imply about how Lipschitz f is?

Stochastic Gradient Descent

Assume that:

- f is convex and decomposable as $f(\vec{\theta}) = \sum_{j=1}^n f_j(\vec{\theta})$.
 - E.g., $L(\vec{\theta}, \mathbf{X}) = \sum_{j=1}^n \ell(M_{\vec{\theta}}(\vec{x}_j), y_j)$.
- Each f_j is $\frac{G}{n}$ -Lipschitz (i.e., $\|\vec{\nabla} f_j(\vec{\theta})\|_2 \leq \frac{G}{n}$ for all $\vec{\theta}$)
 - What does this imply about how Lipschitz f is?
- Initialize with $\theta^{(1)}$ satisfying $\|\vec{\theta}^{(1)} - \vec{\theta}^*\|_2 \leq R$.

Stochastic Gradient Descent

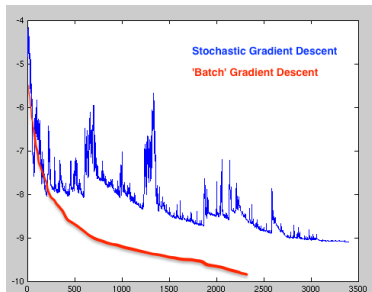
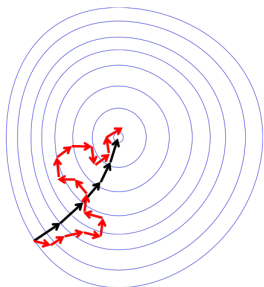
Assume that:

- f is convex and decomposable as $f(\vec{\theta}) = \sum_{j=1}^n f_j(\vec{\theta})$.
 - E.g., $L(\vec{\theta}, \mathbf{X}) = \sum_{j=1}^n \ell(M_{\vec{\theta}}(\vec{x}_j), y_j)$.
- Each f_j is $\frac{G}{n}$ -Lipschitz (i.e., $\|\vec{\nabla} f_j(\vec{\theta})\|_2 \leq \frac{G}{n}$ for all $\vec{\theta}$)
 - What does this imply about how Lipschitz f is?
- Initialize with $\theta^{(1)}$ satisfying $\|\vec{\theta}^{(1)} - \vec{\theta}^*\|_2 \leq R$.

Stochastic Gradient Descent

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

Stochastic Gradient Descent



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

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

Analysis extends to **any** algorithm that takes the gradient step **in expectation** (batch GD, randomly quantized, measurement noise, differentially private GD, etc.)

Stochastic Gradient Descent Analysis

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 θ^* , outputs $\hat{\theta}$ satisfying: $\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon$.

Stochastic Gradient Descent Analysis

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 θ^* , outputs $\hat{\theta}$ satisfying: $\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon$.

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

Stochastic Gradient Descent Analysis

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 θ^* , outputs $\hat{\theta}$ satisfying: $\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon$.

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

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

Stochastic Gradient Descent Analysis

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 θ^* , outputs $\hat{\theta}$ satisfying: $\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon$.

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

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

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

Stochastic Gradient Descent Analysis

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 θ^* , outputs $\hat{\theta}$ satisfying: $\mathbb{E}[f(\hat{\theta})] \leq f(\theta^*) + \epsilon$.

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

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

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

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

Stochastic gradient descent generally makes more iterations than gradient descent.

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

$$\vec{\nabla} \sum_{j=1}^n f_j(\vec{\theta}) \text{ vs. } \vec{\nabla} f_j(\vec{\theta})$$

SGD vs. GD

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

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

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

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

SGD vs. GD

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

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

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

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

$$\|\vec{\nabla} f(\vec{\theta})\|_2 = \|\vec{\nabla} f_1(\vec{\theta}) + \dots + \vec{\nabla} f_n(\vec{\theta})\|_2 \leq \sum_{j=1}^n \|\vec{\nabla} f_j(\vec{\theta})\|_2 \leq n \cdot \frac{G}{n} \leq G.$$

SGD vs. GD

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

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

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

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

$$\|\vec{\nabla} f(\vec{\theta})\|_2 = \|\vec{\nabla} f_1(\vec{\theta}) + \dots + \vec{\nabla} f_n(\vec{\theta})\|_2 \leq \sum_{j=1}^n \|\vec{\nabla} f_j(\vec{\theta})\|_2 \leq n \cdot \frac{G}{n} \leq G.$$

When would this bound be tight?

Questions?