COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

University of Massachusetts Amherst. Fall 2021.

Lecture 25 (Final Lecture!)

LOGISTICS

- Problem Set 5 is due Dec 13. Can be used to replace your lowest problem set grade.
- Problem Set 4 solutions are posted.
- Exam is next Thursday Dec 16, from 10:30am-12:30pm in class.
- See course website/Moodle/Piazza for exam review guide, practice exam, additional office hours schedule.
- It would be really helpful if you could fill out SRTIs for this class (they close Dec 18).
- `http://owl.umass.edu/partners/courseEvalSurvey/uma/`.

Question 6: was on a topic we will cover today (convex sets). It will count only as bonus.

Question 6: was on a topic we will cover today (convex sets). It will count only as bonus.

$$\vec{\theta} = \begin{bmatrix} \theta(1) \\ \theta(2) \\ \theta(3) \end{bmatrix}$$

Question 5:

$$f(\vec{\theta}) = \theta(1) + 2\theta(2) - 2\theta(3)$$

Consider the function $f(\vec{\theta}) = \vec{x}^T\vec{\theta}$ for $x = [1, 2, -2]$. Give the minimum value of G such that $f(\vec{\theta})$ is G-Lipschitz. Give you answer to 2 decimal places.

$$[x^T]\begin{bmatrix}\theta\end{bmatrix} = \langle x, \theta \rangle$$

$$G = \|x\|_2 = \sqrt{4+4+1} = 3$$

$$\max_{\vec{\theta}} \|\nabla f(\vec{\theta})\|_2$$

$$\nabla f(\vec{\theta}) = \begin{bmatrix} \vec{x} \end{bmatrix}$$

$$\frac{\partial f}{\partial \theta(1)} = 1 \qquad \vec{x} = \begin{bmatrix} 1 \\ 2 \\ -2 \end{bmatrix}$$

$$\frac{\partial f}{\partial \theta(2)} = 2$$

Last Class:

- Analysis of gradient descent for convex and Lipschitz functions.

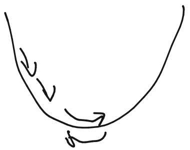
This Class:

- Extend gradient descent to constrained optimization via projected gradient descent.
- Course wrap up and review.

GD ANALYSIS PROOF

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

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



GD ANALYSIS PROOF

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

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

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

Step 2: $\underbrace{\frac{1}{t}\sum_{i=1}^{t} f(\vec{\theta}_i) - f(\vec{\theta}_*)}_{\text{"average error"}} \leq \frac{R^2}{2\eta \cdot t} + \frac{\eta G^2}{2} < \epsilon$

"average error"

$\geq$ min error

Often want to perform convex optimization with convex constraints.

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

$f : \mathbb{R}^d \to \mathbb{R}$

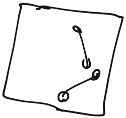
where $\mathcal{S}$ is a convex set.

Often want to perform convex optimization with convex constraints.

convex function

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

convex set

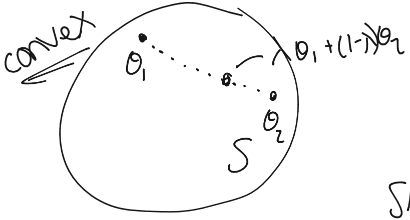


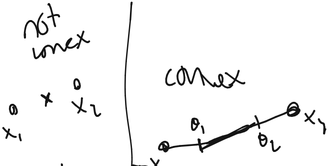
where $\mathcal{S}$ is a convex set.

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

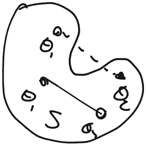
$S \subseteq \mathbb{R}^2$

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



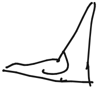


$S = \{\vec{x}_1, \vec{x}_2\}$



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

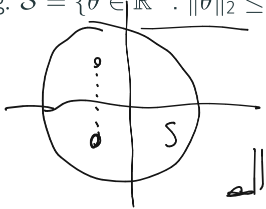
$$(1-\lambda)\vec{\theta}_1 + \lambda \cdot \vec{\theta}_2 \in \mathcal{S}$$

$S' = \{\vec{\theta} \in \mathbb{R}^d : \|\theta\|_* \leq 1\}$

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

$\theta_1 \in S \Rightarrow \|\theta_1\| \leq 1$

$\theta_2 \in S \Rightarrow \|\theta_2\| \leq 1$



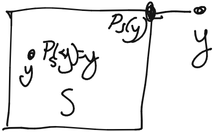
for any λ

$$\|(1-\lambda)\theta_1 + \lambda\theta_2\|_2 \leq \underbrace{\|(1-\lambda)\theta_1\|_2 + \|\lambda\theta_2\|}_{\leq 1-\lambda + \lambda \leq 1} = (1-\lambda)\|\theta_1\|_2 + \lambda\|\theta_2\| \leq 1$$

PROJECTED GRADIENT DESCENT

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

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

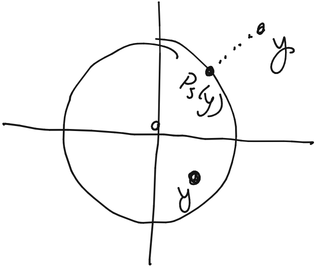


For any convex set let $P_{\mathcal{S}}(\cdot)$ denote the projection function onto $\mathcal{S}$.

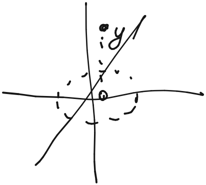
- $P_{\mathcal{S}}(\vec{y}) = \arg\min_{\vec{\theta} \in \mathcal{S}} \|\vec{\theta} - \vec{y}\|_2$.
- For $\mathcal{S} = \{\vec{\theta} \in \mathbb{R}^d : \|\vec{\theta}\|_2 \leq 1\}$ what is $P_{\mathcal{S}}(\vec{y})$?

$P_S : \mathbb{R}^d \to \mathbb{R}^d$

$\mathbb{R}^d \to S$



$$P_S(\vec{y}) = \begin{cases} \text{if } \|y\| \geq 1, & \frac{y}{\|y\|} \\ \text{if } \|y\| \leq 1, & y \end{cases}$$



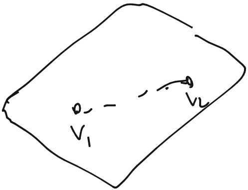
$\checkmark\ S_1 = \{x : \|x\| \leq 10\}$ convex $\quad S_2 = \{x : \|x\| = 10\}$

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

- $P_{\mathcal{S}}(\vec{y}) = \arg\min_{\vec{\theta} \in \mathcal{S}} \|\vec{\theta} - \vec{y}\|_2$.
- 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 $P_{\mathcal{S}}(\vec{y})$? $= W^T y$ (d×d)

is S convex?

let $V \in \mathbb{R}^{d \times k}$ be a basis for the subspace



$\theta_1, \theta_2 \in S$

$\theta_1 = Vc_1 \quad \theta_2 = Vc_2$

$\lambda\theta_1 + (1-\lambda)\theta_2 = V(\lambda c_1 + (1-\lambda)c_2) \in S$

PROJECTED GRADIENT DESCENT

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

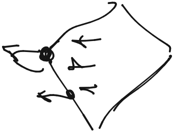
- $P_{\mathcal{S}}(\vec{y}) = \arg\min_{\vec{\theta} \in \mathcal{S}} \|\vec{\theta} - \vec{y}\|_2$.
- For $\mathcal{S} = \{\vec{\theta} \in \mathbb{R}^d : \|\vec{\theta}\|_2 \leq 1\}$ what is $P_{\mathcal{S}}(\vec{y})$?
- For $\mathcal{S}$ being a k dimensional subspace of $\mathbb{R}^d$, what is $P_{\mathcal{S}}(\vec{y})$?

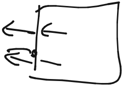
$S = \{x_1, x_2, x_3\}$

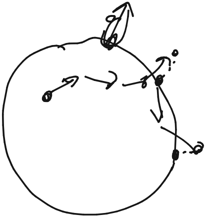
$f(x_1) = 10 \quad f(x)_2 = 5 \quad f(x_3) = 2$

Projected Gradient Descent

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





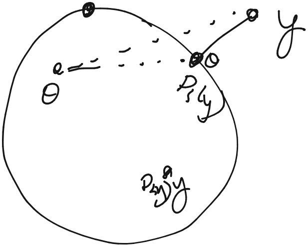


Projected gradient descent can be analyzed identically to gradient descent!

Projected gradient descent can be analyzed identically to gradient descent!

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

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



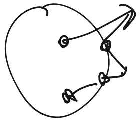
Theorem – Projected GD: For convex G-Lipschitz function f, and convex set $\mathcal{S}$, Projected GD run with $t \geq \frac{R^2G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

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

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

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

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



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

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

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

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

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

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

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

$\|\theta_{i+1}^{out} - \theta_*\| \geq$

$\|\theta_{i+1} - \theta^*\|$

by convexity

$\theta^* \in S$

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

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

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

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

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

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

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

Randomization as a computational resource for massive datasets.

Randomization as a computational resource for massive datasets.

- Focus on problems that are easy on small datasets but hard at massive scale – set size estimation, load balancing, distinct elements counting (MinHash), checking set membership (Bloom Filters), frequent items counting (Count-min sketch), near neighbor search (locality sensitive hashing).

Randomization as a computational resource for massive datasets.

- Focus on problems that are easy on small datasets but hard at massive scale – set size estimation, load balancing, distinct elements counting (MinHash), checking set membership (Bloom Filters), frequent items counting (Count-min sketch), near neighbor search (locality sensitive hashing).
- Just the tip of the iceberg on randomized streaming/sketching/hashing algorithms. Check out 690RA if you want to learn more.

Randomization as a computational resource for massive datasets.

- Focus on problems that are easy on small datasets but hard at massive scale – set size estimation, load balancing, distinct elements counting (MinHash), checking set membership (Bloom Filters), frequent items counting (Count-min sketch), near neighbor search (locality sensitive hashing).
- Just the tip of the iceberg on randomized streaming/sketching/hashing algorithms. Check out 690RA if you want to learn more.
- In the process covered probability/statistics tools that are very useful beyond algorithm design: concentration inequalities, higher moment bounds, law of large numbers, central limit theorem, linearity of expectation and variance, union bound, median as a robust estimator.

Methods for working with (compressing) high-dimensional data

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.
- Connections to the weird geometry of high-dimensional space.

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.
- Connections to the weird geometry of high-dimensional space. final
- Dimensionality reduction via low-rank approximation and optimal solution with PCA/eigendecomposition/SVD.

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.
- Connections to the weird geometry of high-dimensional space.
- Dimensionality reduction via low-rank approximation and optimal solution with PCA/eigendecomposition/SVD.
- Low-rank approximation of similarity matrices and entity embeddings (e.g., LSA, word2vec, DeepWalk).

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.
- Connections to the weird geometry of high-dimensional space.
- Dimensionality reduction via low-rank approximation and optimal solution with PCA/eigendecomposition/SVD.
- Low-rank approximation of similarity matrices and entity embeddings (e.g., LSA, word2vec, DeepWalk).
- Spectral graph theory – nonlinear dimension reduction and spectral clustering for community detection.

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.
- Connections to the weird geometry of high-dimensional space.
- Dimensionality reduction via low-rank approximation and optimal solution with PCA/eigendecomposition/SVD.
- Low-rank approximation of similarity matrices and entity embeddings (e.g., LSA, word2vec, DeepWalk).
- Spectral graph theory – nonlinear dimension reduction and spectral clustering for community detection.
- In the process covered linear algebraic tools that are very broadly useful in ML and data science: eigendecomposition, singular value decomposition, projection, norm transformations.

Foundations of continuous optimization and gradient descent.

Foundations of continuous optimization and gradient descent.

- Foundational concepts like convexity, convex sets, Lipschitzness, directional derivative/gradient.

Foundations of continuous optimization and gradient descent.

- Foundational concepts like convexity, convex sets, Lipschitzness, directional derivative/gradient.
- How to analyze gradient descent in a simple setting (convex Lipschitz functions).

Foundations of continuous optimization and gradient descent.

- Foundational concepts like convexity, convex sets, Lipschitzness, directional derivative/gradient.
- How to analyze gradient descent in a simple setting (convex Lipschitz functions).
- Simple extension to projected gradient descent for optimization over a convex constraint set.

Foundations of continuous optimization and gradient descent.

- Foundational concepts like convexity, convex sets, Lipschitzness, directional derivative/gradient.
- How to analyze gradient descent in a simple setting (convex Lipschitz functions).
- Simple extension to projected gradient descent for optimization over a convex constraint set.
- Lots that we didn't cover: online and stochastic gradient descent, accelerated methods, adaptive methods, second order methods (quasi-Newton methods), practical considerations. Gave mathematical tools to understand these methods.

Thanks for a great semester!

It felt really good to be back teaching in person, especially with all the participation in this class.

FINAL EXAM QUESTIONS/REVIEW