COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

University of Massachusetts Amherst. Fall 2020.

Lecture 25 (Final Lecture!)

LOGISTICS

- Problem Set 5 was posted this morning, due 11/30.
- Problem Set 4 solutions were also posted.
- Exam will span December 3-4. Any two hour period.
- Exam review guide, practice problems, logistical details have been posted under the schedule tab on the course page.
- I am holding an optional SRTI (course reviews) for this class and would really appreciate your feedback (closes Dec 6).
- `http://owl.umass.edu/partners/courseEvalSurvey/uma/`.
- We will post our exam review office hour schedules in the next day or two.

Last Class:

- Introduction to online learning and regret.
- Online gradient descent and its guarantees.

This Class:

- Finish online gradient descent analysis.
- Application to stochastic gradient descent.
- Course wrap up.

ONLINE OPTIMIZATION FORMAL SETUP

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

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

Will make no assumptions on how $f_1, \ldots, f_t$ are related to each other.

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

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

$\theta_1 \quad \theta_2 \quad \theta_t$

- 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 $\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)})$.
- **Metric:** Regret $= \sum_{i=1}^{t} f_i(\vec{\theta}^{(i)}) - \min_{\vec{\theta}} \sum_{i=1}^{t} f_i(\vec{\theta})$.

$f(\theta^{OFF})$

Will make no assumptions on how $f_1, \ldots, f_t$ are related to each other.

$f_i = |1000 \pm x|$

Assume that:

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

Online Gradient Descent

$\min_{\theta} \sum f_i(\theta)$

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

$\underline{f_{i+1}}$ $\underline{f_i}$

Theorem – OGD on Convex Lipschitz Functions: For convex G-Lipschitz $f_1, \ldots, f_t$, 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:

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

Upper bound on average regret goes to 0 and $t \to \infty$.

ONLINE GRADIENT DESCENT ANALYSIS

Theorem – OGD on Convex Lipschitz Functions: For convex G-Lipschitz $f_1, \ldots, f_t$, 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:

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

Upper bound on average regret goes to 0 and $t \to \infty$.

"distance from opt", "overshoot"

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

$\theta^{i+1} = \theta^{i} - \eta \nabla f_i(\theta^{i})$

Theorem – OGD on Convex Lipschitz Functions: For convex G-Lipschitz $f_1, \ldots, f_t$, 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:

f

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

Upper bound on average regret goes to 0 and $t \to \infty$.

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

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

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

Theorem – OGD on Convex Lipschitz Functions: For convex G-Lipschitz $f_1, \ldots, f_t$, 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, $f_i(\theta^{(i)}) - f_i(\theta^{off}) \leq \frac{\|\theta^{(i)} - \theta^{off}\|_2^2 - \|\theta^{(i+1)} - \theta^{off}\|_2^2}{2\eta} + \frac{\eta G^2}{2}$

Theorem – OGD on Convex Lipschitz Functions: For convex G-Lipschitz $f_1, \ldots, f_t$, 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:

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

"telescoping sum

$\|\theta^1 - \theta^{off}\| + \|\theta^2 - \theta^{off}\| + \|\theta^2 - \theta^{off}\| - \|\theta^3 - \theta^{off}\|$

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

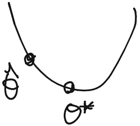
$$\left[\sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{off})\right] \leq \sum_{i=1}^{t} \frac{\|\theta^{(i)} - \theta^{off}\|_2^2 - \|\theta^{(i+1)} - \theta^{off}\|_2^2}{2\eta} + \frac{t \cdot \eta G^2}{2}.$$

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

$\eta = \frac{R}{G\sqrt{t}}$

$$\leq \frac{R^2}{2\eta} + \frac{t\eta G^2}{2} = \frac{RG\sqrt{t}}{2} + \frac{RG\sqrt{t}}{2} = \boxed{RG\sqrt{t}}$$

STOCHASTIC GRADIENT DESCENT



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

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

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

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

- The most popular optimization method in modern machine learning.
- Easily analyzed as a special case of online gradient descent!

finite sum problem

Assume that:

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

training set

loss at point x_j

Assume that:

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

$$\|\nabla f(\theta)\| = \|\nabla f_1(\theta) + \nabla f_2(\theta) + \dots \nabla f_n(\theta)\|$$

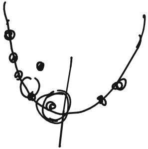
$$\underset{\Delta\text{-ineq}}{\leq} \sum_{i=1}^{n} \|\nabla f_i(\theta)\|_2 \leq n \cdot \frac{G}{n} = G$$

Assume that:

- f is convex and decomposable as $f(\vec{\theta}) = \sum_{j=1}^{n} f_j(\vec{\theta})$.
  - E.g., $L(\vec{\theta}, \mathbf{X}) = \sum_{j=1}^{n} \ell(\vec{\theta}, \vec{x}_j)$.
- 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 $\|\vec{\theta}^{(1)} - \vec{\theta}^{*}\|_2 \leq R$.

Assume that: $\nabla f(\theta)$ — n backward passes

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

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

backward pass

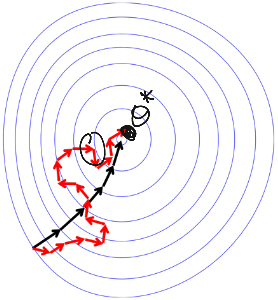
x_{j_i}

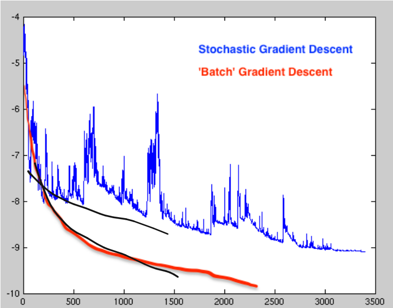
$\hat{\theta} = \operatorname{argmin}_{\theta^i} f(\theta^i)$

why?

when n is large each step is much faster than a full gradient.

f: total loss on training set





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

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

$\sum_{j=1}^{n} \frac{1}{n} \cdot \nabla f_j(\theta^i) = \frac{1}{n} \sum_{j=1}^{n} \nabla f_j(\theta^i)$

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

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

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

$\frac{1}{t}\sum_{i=1}^{n}\theta^i$

why does this hold?

Convexity

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

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

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

$$\mathbb{E} f_{j_i}(\theta^i) = \sum_{j=1}^{n} \frac{1}{n} f_j(\theta^i) = \frac{1}{n} f(\theta^i)$$

$$n \cdot \mathbb{E} f_{j_i}(\theta^i) = f(\theta^i)$$

$$\mathbb{E}[f(\hat{\theta}) - f(\theta^*)] \leq \frac{1}{t}\mathbb{E}\left[\sum_{i=1}^{t} f(\theta^i) - f(\theta^*)\right]$$

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

$$\leq \frac{n}{t}\mathbb{E}\left[\sum_{i=1}^{t} f_{j_i}(\theta^i) - f_{j_i}(\theta^*)\right]$$

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

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

optimum of $\sum_{i=1}^{t} f_{j_i}(\theta) \approx f(\theta)$

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

regret.

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

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

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

$t = \frac{R^2G^2}{\epsilon^2}$

Stochastic gradient descent generally makes more iterations than gradient descent.

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

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

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:

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

How much smaller is $\bar{G}$ than G

$\bar{G} \leq G$

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:

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

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

$\bar{G} < G$

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

$\bar{G} << G$

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

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

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

When would this bound be tight?

$\|\vec{\nabla} f(\vec{\theta})\|_2 = n \cdot \frac{G}{n} = G$

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.

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.
- 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.
- Online optimization and online gradient descent → stochastic gradient descent.

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.
- Online optimization and online gradient descent $\rightarrow$ stochastic gradient descent.
- Lots that we didn't cover: accelerated methods, adaptive methods, second order methods (quasi-Newton methods), practical considerations. Gave mathematical tools to understand these methods.

$$\hat{\theta} = \sum_{i=1}^{t} \theta^i$$

$$\hat{\theta} = \underset{\theta_1, \dots, \theta_t}{\operatorname{argmin}} \ f(\theta^i)$$

$$\hat{\theta} = \hat{\theta} + \theta^i$$

$$\hat{\theta} / t$$

Thanks for a great semester!

loss over my training set

10 mil points

$$\nabla f(\theta)$$

$$f(\theta)$$