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/](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: 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 $\bar{\theta}^{(i)}$.
- Then are told $f_i$ and incur cost $f_i(\bar{\theta}^{(i)})$.
- **Goal:** Minimize total cost $\sum_{i=1}^{t} f_i(\bar{\theta}^{(i)})$.
- **Metric:** Regret $= \sum_{i=1}^{t} f_i(\bar{\theta}^{(i)}) - \min_{\bar{\theta}} \sum_{i=1}^{t} f_i(\bar{\theta})$.

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

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

Online Gradient Descent

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

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

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

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

$$\left[ \sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{\text{off}}) \right] \leq \sum_{i=1}^{t} \frac{\|\theta^{(i)} - \theta^{\text{off}}\|_2^2 - \|\theta^{(i+1)} - \theta^{\text{off}}\|_2^2}{2\eta} + \frac{t \cdot \eta G^2}{2}.$$
Stochastic gradient descent is an efficient **offline optimization method**, seeking $\hat{\theta}$ with

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

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

Assume that:

• $f$ is convex and decomposable as $f(\vec{\theta}) = \sum_{j=1}^{n} f_j(\vec{\theta})$.
  • E.g., $L(\vec{\theta}, X) = \sum_{j=1}^{n} \ell(\vec{\theta}, \vec{x}_j)$.
  • Each $f_j$ is $\frac{G}{n}$-Lipschitz (i.e., $\|\nabla f_j(\vec{\theta})\|_2 \leq \frac{G}{n}$ 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 $j_i \in 1, \ldots, n$.
  • $\vec{\theta}^{(i+1)} = \vec{\theta}^{(i)} - \eta \cdot \nabla f_{j_i}(\vec{\theta}^{(i)})$.
• Return $\hat{\vec{\theta}} = \frac{1}{t} \sum_{i=1}^{t} \vec{\theta}^{(i)}$. 
\[ \theta^{(i+1)} = \theta^{(i)} - \eta \cdot \nabla f_j(\theta^{(i)}) \] vs. \[ \theta^{(i+1)} = \theta^{(i)} - \eta \cdot \nabla f^{(i)} \]

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

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

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

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

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

**Step 4:** \( \mathbb{E}[f(\hat{\theta}) - f(\theta^*)] \leq \frac{n}{t} \cdot R \cdot \frac{G}{n} \cdot \sqrt{t} = \frac{RG}{\sqrt{t}}. \) 
\( \text{OGD bound} \)
Stochastic gradient descent generally makes more iterations than gradient descent.

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

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

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

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

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

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

When would this bound be tight?
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

- 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.

- 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.
Thanks for a great semester!