CS-GY 6763: Lecture 6
Gradient Descent and Projected Gradient Descent

NYU Tandon School of Engineering, Prof. Christopher Musco

ADMINISTRATIVE

- Homework 2 due next Tuesday evening.
- First reading group meeting, **Monday 3-4pm**. Meet on 11th floor of 370 Jay St. Thank you **Roman and Marc** for volunteering to present! Please review their chosen paper before the meeting.
- Midterm **next Friday**. 1 hour test, followed by break, then lecture by our TA Feyza on fine-grained complexity. I will post midterm review document and practice questions.
    - Nothing from today will be covered on the midterm.

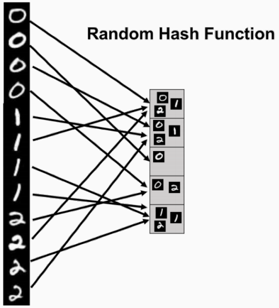
FINISH UP LSH + NEAR NEIGHBOR SEARCH

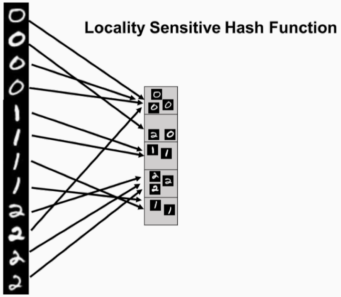
LOCALITY SENSITIVE HASH FUNCTIONS

Let $h : \mathbb{R}^d \rightarrow \{1, \ldots, m\}$ be a random hash function.

We call h locality sensitive for similarity function $s(\mathbf{q}, \mathbf{y})$ if $\Pr[h(\mathbf{q}) == h(\mathbf{y})]$ is:

- Higher when $\mathbf{q}$ and $\mathbf{y}$ are more similar, i.e. $s(\mathbf{q}, \mathbf{y})$ is higher.
- Lower when $\mathbf{q}$ and $\mathbf{y}$ are more dissimilar, i.e. $s(\mathbf{q}, \mathbf{y})$ is lower.

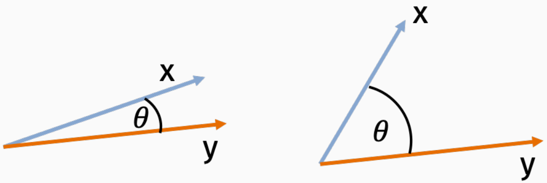




OTHER LSH FUNCTIONS

We saw how MinHash gives an LSH function for Jaccard similarity. Good locality sensitive hash functions exists for other similarity measures.

Cosine similarity $\cos(\theta(\mathbf{x}, \mathbf{y})) = \frac{\langle \mathbf{x},\mathbf{y}\rangle}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2}$:



$$-1 \leq \cos(\theta(\mathbf{x}, \mathbf{y})) \leq 1.$$

COSINE SIMILARITY

Cosine similarity is natural "inverse" for Euclidean distance.

Euclidean distance $\|\mathbf{x} - \mathbf{y}\|_2^2$:

- Suppose for simplicity that $\|\mathbf{x}\|_2^2 = \|\mathbf{y}\|_2^2 = 1$.

SIMHASH

Locality sensitive hash for **cosine similarity**:

- Let $\mathbf{g} \in \mathbb{R}^d$ be randomly chosen with each entry $\mathcal{N}(0,1)$.
- Let $f : \{-1, 1\} \to \{1, \ldots, m\}$ be a uniformly random hash function.
- $h : \mathbb{R}^d \to \{1, \ldots, m\}$ is definied $h(\mathbf{x}) = f(\text{sign}(\langle \mathbf{g}, \mathbf{x} \rangle))$.

Claim: If $\cos(\theta(\mathbf{x}, \mathbf{y})) = v$, then

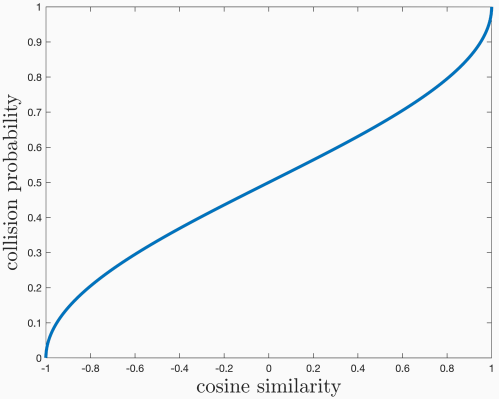
$$\Pr[h(\mathbf{x}) == h(\mathbf{y})] = 1 - \frac{\theta}{\pi} + \frac{1 - v}{m}$$

.

SIMHASH ANALYSIS IN 2D

Lemma to prove: If $\cos(\theta(\mathbf{x},\mathbf{y})) = v$, then

$$\Pr[g(\mathbf{x}) == g(\mathbf{y})] = 1 - \frac{\theta}{\pi} = 1 - \frac{\cos^{-1}(v)}{\pi}$$



$$\Pr[h(\mathbf{x}) == h(\mathbf{y})] = \Pr[g(\mathbf{x}) == g(\mathbf{y})] + \tfrac{1-v}{m}.$$

SIMHASH

SimHash can be tuned, just like MinHash-based LSH function for Jaccard similarity. Version of hash function with r bands:

- Let $\mathbf{g}_1, \ldots, \mathbf{g}_r \in \mathbb{R}^d$ be chosen with each entry $\mathcal{N}(0, 1)$.
- Let $f : \{-1, 1\}^r \rightarrow \{1, \ldots, m\}$ be a uniformly random hash function.
- $h : \mathbb{R}^d \rightarrow \{1, \ldots, m\}$ is defined

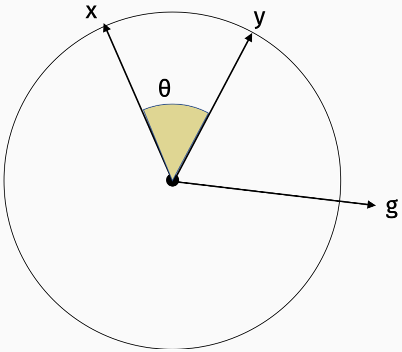
$$h(\mathbf{x}) = f([\text{sign}(\langle \mathbf{g}_1, \mathbf{x} \rangle), \ldots, \text{sign}(\langle \mathbf{g}_r, \mathbf{x} \rangle)])$$

.

$$\Pr[h(\mathbf{x}) == h(\mathbf{y})] = \left(1 - \frac{\theta}{\Pi}\right)^r$$

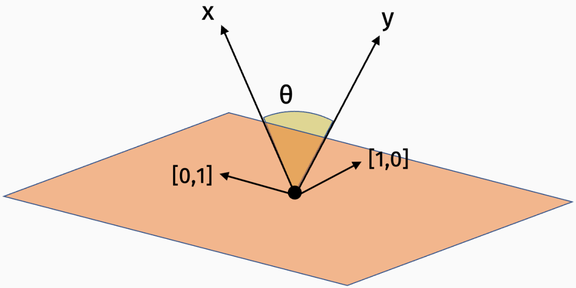
SimHash analysis 2D

To prove: $\Pr[g(\mathbf{x}) == g(\mathbf{y})] = 1 - \frac{\theta}{\pi}$, where $g(\mathbf{x}) = \text{sign}(\langle \mathbf{g}, \mathbf{x} \rangle)$.



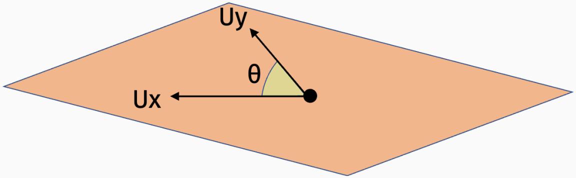
$\Pr[g(\mathbf{x}) == g(\mathbf{y})] = \Pr[\text{sign}(\langle \mathbf{g}, \mathbf{x} \rangle) == \text{sign}(\langle \mathbf{g}, \mathbf{y} \rangle)] =$ probability $\mathbf{x}$ and $\mathbf{y}$ are on the same side of hyperplane orthogonal to $\mathbf{g}$.

SIMHASH ANALYSIS HIGHER DIMENSIONS



There is always some rotation matrix $\mathbf{U}$ such that $\mathbf{U}\mathbf{x}, \mathbf{U}\mathbf{y}$ are spanned by the first two-standard basis vectors and have the same cosine similarity as $\mathbf{x}$ and $\mathbf{y}$.

SIMHASH ANALYSIS HIGHER DIMENSIONS



There is always some rotation matrix $\mathbf{U}$ such that $\mathbf{x}, \mathbf{y}$ are spanned by the first two-standard basis vectors.

Note: A rotation matrix $\mathbf{U}$ has the property that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. I.e., $\mathbf{U}^T$ is a rotation matrix itself, which reverses the rotation of $\mathbf{U}$.

SIMHASH ANALYSIS HIGHER DIMENSIONS

Claim:

$$\begin{aligned}
\Pr[\mathrm{sign}(\langle g, x\rangle) == \mathrm{sign}(\langle g, y\rangle)] \\
&= \Pr[\mathrm{sign}(\langle g, Ux\rangle) == \mathrm{sign}(\langle g, Uy\rangle)] \\
&= \Pr[\mathrm{sign}(\langle g[1,2], (Ux)[1,2]\rangle) == \mathrm{sign}(\langle g[1,2], (Uy[1,2]\rangle)] \\
&= 1 - \frac{\theta}{\pi}
\end{aligned}$$

WORST CASE THEORETICAL RESULT

Last class and on the homework, we show how to build LSH data structures for specific point sets that achieves $o(n)$ search time by using $\Omega(n)$ space. However, we did't prove any "worst-case" theoretical guarantees.

Such guarantees can be proven, and were actually a major driving force in the development of LSH methods.

WORST CASE THEORETICAL RESULT

Near Neighbor Search Problem.

Theorem (Indyk, Motwani, 1998)

If there exists some q with $\|\mathbf{q} - \mathbf{y}\|_0 \leq R$, return a vector $\tilde{\mathbf{q}}$ with $\|\tilde{\mathbf{q}} - \mathbf{y}\|_0 \leq C \cdot R$ in:

- *Time: $O\left(n^{1/C}\right)$.*
- *Space: $O\left(n^{1+1/C}\right)$.*

$\|\mathbf{q} - \mathbf{y}\|_0 =$ "hamming distance" = number of elements that differ between $\mathbf{q}$ and $\mathbf{y}$.

R is a fixed parameter given as part of the input.

APPROXIMATE NEAREST NEIGHBOR SEARCH

Exponential search over values of R easily yields:

Theorem (Indyk, Motwani, 1998)

Let q be the closest database vector to $\mathbf{y}$. Return a vector $\tilde{\mathbf{q}}$ with $\|\tilde{\mathbf{q}} - \mathbf{y}\|_0 \leq C \cdot \|\mathbf{q} - \mathbf{y}\|_0$ in:

- *Time:* $\tilde{O}\left(n^{1/C}\right)$.
- *Space:* $\tilde{O}\left(n^{1+1/C}\right)$.

OPTIMIZATION

CONTINUOUS OPTIMIZATION

Given function $f : \mathbb{R}^d \rightarrow \mathbb{R}$. Find $\hat{\mathbf{x}}$ such that:

$$f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x}} f(\mathbf{x}) + \epsilon.$$

NEXT UNIT: CONTINUOUS OPTIMIZATION

Have some function $f : \mathbb{R}^d \to \mathbb{R}$. Want to find $\mathbf{x}^*$ such that:

$$f(\mathbf{x}^*) = \min_{\mathbf{x}} f(\mathbf{x}).$$

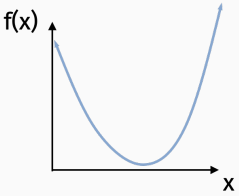
Or at least $\hat{\mathbf{x}}$ which is close to a minimum. E.g. $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x}} f(\mathbf{x}) + \epsilon$

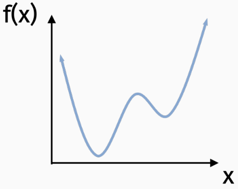
Often we have some additional constraints:

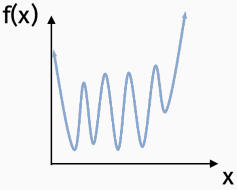
- $\mathbf{x} > 0$.
- $\|\mathbf{x}\|_2 \leq R$, $\|\mathbf{x}\|_1 \leq R$.
- $\mathbf{a}^T\mathbf{x} > c$.

CONTINUOUS OPTIMIZATION

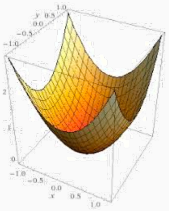
Dimension $d = 1$:

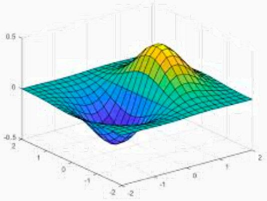


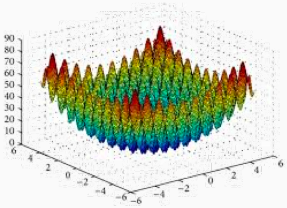




Dimension $d = 2$:







OPTIMIZATION IN MACHINE LEARNING

Continuouos optimization is the foundation of modern machine learning.

Supervised learning: Want to learn a model that maps inputs

- numerical data vectors
- images, video
- text documents

to predictions

- numerical value (probability stock price increases)
- label (is the image a cat? does the image contain a car?)
- decision (turn car left, rotate robotic arm)

MACHINE LEARNING MODEL

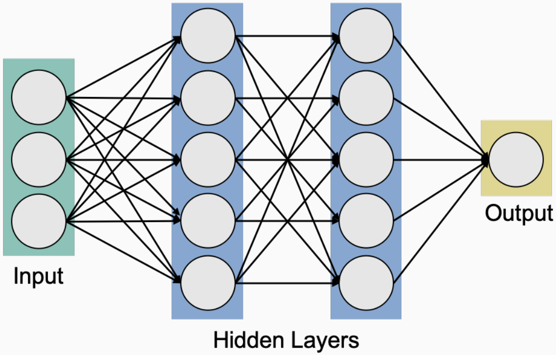
Let $M_{\mathbf{x}}$ be a model with parameters $\mathbf{x} = \{x_1, \ldots, x_k\}$, which takes as input a data vector $\mathbf{a}$ and outputs a prediction.

Example:

$$M_{\mathbf{x}}(\mathbf{a}) = \mathrm{sign}(\mathbf{a}^T\mathbf{x})$$

MACHINE LEARNING MODEL

Example:



$x \in \mathbb{R}^{(\text{\# of connections})}$ is the parameter vector containing all the network weights.

SUPERVISED LEARNING

Classic approach in <u>supervised learning</u>: Find a model that works well on data that you already have the answer for (labels, values, classes, etc.).

- Model $M_{\mathbf{x}}$ parameterized by a vector of numbers $\mathbf{x}$.
- Dataset $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(n)}$ with outputs $y^{(1)}, \ldots, y^{(n)}$.

Want to find $\hat{\mathbf{x}}$ so that $M_{\hat{\mathbf{x}}}(\mathbf{a}^{(i)}) \approx y^{(i)}$ for $i \in 1, \ldots, n$.

How do we turn this into a function minimization problem?

LOSS FUNCTION

Loss function $L\left(M_{\mathsf{x}}(\mathsf{a}), y\right)$: Some measure of distance between prediction $M_{\mathsf{x}}(\mathsf{a})$ and target output y. Increases if they are further apart.

- Squared (ℓ_2) loss: $|M_{\mathsf{x}}(\mathsf{a}) - y|^2$
- Absolute deviation (ℓ_1) loss: $|M_{\mathsf{x}}(\mathsf{a}) - y|$
- Hinge loss: 1 - $y \cdot M_{\mathsf{x}}(\mathsf{a})$
- Cross-entropy loss (log loss).
- Etc.

EMPIRICAL RISK MINIMIZATION

Empirical risk minimization:

$$f(\mathbf{x}) = \sum_{i=1}^{n} L\left(M_{\mathbf{x}}(\mathbf{a}^{(i)}), y^{(i)}\right)$$

Solve the optimization problem $\min_{\mathbf{x}} f(\mathbf{x})$.

EXAMPLE: LEAST SQUARES REGRESSION

- $M_{\mathbf{x}}(\mathbf{a}) = \mathbf{x}^T\mathbf{a}$. $\mathbf{x}$ contains the regression coefficients.
- $L(z, y) = |z - y|^2$.
- $f(\mathbf{x}) = \sum_{i=1}^{n} |\mathbf{x}^T\mathbf{a}^{(i)} - y^{(i)}|^2$

$$f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$$

where $\mathbf{A}$ is a matrix with $\mathbf{a}^{(i)}$ as its i^{th} row and $\mathbf{y}$ is a vector with $y^{(i)}$ as its i^{th} entry.

ALGORITHMS FOR CONTINUOUS OPTIMIZATION

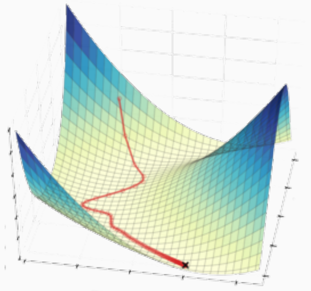
The choice of algorithm to minimize $f(\mathbf{x})$ will depend on:

- The form of $f(\mathbf{x})$ (is it linear, is it quadratic, does it have finite sum structure, etc.)
- If there are any additional constraints imposed on $\mathbf{x}$. E.g. $\|\mathbf{x}\|_2 \leq c$.

What are some example algorithms for continuous optimization?

FIRST TOPIC: GRADIENT DESCENT + VARIANTS

Gradient descent: A greedy algorithm for minimizing functions of multiple variables that often works amazingly well.



Runtime generally scales <u>linearly</u> with the dimension of $\mathbf{x}$ (although this is a bit of an over-simplification).

SECOND TOPIC: METHODS SUITABLE FOR LOWER DIMENSION

- Cutting plane methods (e.g. center-of-gravity, ellipsoid)
- Interior point methods

Fast and more accurate in low-dimensions, slower in very high dimensions. Generally runtime scales polynomially with the dimension of $\mathbf{x}$.

CALCULUS REVIEW

For $i = 1, \ldots, d$, let x_i be the i^{th} entry of $\mathbf{x}$. Let $\mathbf{e}^{(i)}$ be the i^{th} standard basis vector.

Partial derivative:

$$\frac{\partial f}{\partial x_i}(\mathbf{x}) = \lim_{t \to 0} \frac{f(\mathbf{x} + t\mathbf{e}^{(i)}) - f(\mathbf{x})}{t}$$

Directional derivative:

$$D_{\mathbf{v}} f(\mathbf{x}) = \lim_{t \to 0} \frac{f(\mathbf{x} + t\mathbf{v}) - f(\mathbf{x})}{t}$$

CALCULUS REVIEW

Gradient:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \frac{\partial f}{\partial x_2}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial x_d}(\mathbf{x}) \end{bmatrix}$$

Directional derivative:

$$D_{\mathbf{v}} f(\mathbf{x}) = \lim_{t \to 0} \frac{f(\mathbf{x} + t\mathbf{v}) - f(\mathbf{x})}{t} = \nabla f(\mathbf{x})^T \mathbf{v}.$$

FIRST ORDER OPTIMIZATION

Given a function f to minimize, assume we have:

- **Function oracle**: Evaluate $f(\mathbf{x})$ for any $\mathbf{x}$.
- **Gradient oracle**: Evaluate $\nabla f(\mathbf{x})$ for any $\mathbf{x}$.

We view the implementation of these oracles as black-boxes, but they can often require a fair bit of computation.

EXAMPLE GRADIENT EVALUATION

Linear least-squares regression:

- Given $\mathbf{a}^{(1)}, \ldots \mathbf{a}^{(n)} \in \mathbb{R}^d$, $y^{(1)}, \ldots y^{(n)} \in \mathbb{R}$.
- Want to minimize:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \left(\mathbf{x}^T \mathbf{a}^{(i)} - y^{(i)} \right)^2 = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2.$$

What is the time complexity to implement a function oracle for $f(\mathbf{x})$?

EXAMPLE GRADIENT EVALUATION

Linear least-squares regression:

- Want to minimize:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \left(\mathbf{x}^T \mathbf{a}^{(i)} - y^{(i)} \right)^2 = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2.$$

$$\frac{\partial f}{\partial x_j} = \sum_{i=1}^{n} 2 \left(\mathbf{x}^T \mathbf{a}^{(i)} - y^{(i)} \right) \cdot a_j^{(i)} = 2\boldsymbol{\alpha}^{(j)T}(\mathbf{A}\mathbf{x} - \mathbf{y})$$

where $\boldsymbol{\alpha}^{(j)}$ is the jth <u>column</u> of $\mathbf{A}$.

EXAMPLE GRADIENT EVALUATION

Linear least-squares regression:

$$\frac{\partial f}{\partial x_j} = \sum_{i=1}^{n} 2\left(\mathbf{x}^T \mathbf{a}^{(i)} - y^{(i)}\right) \cdot a_j^{(i)} = 2\boldsymbol{\alpha}^{(j)^T}(\mathbf{A}\mathbf{x} - \mathbf{y})$$

where $\boldsymbol{\alpha}^{(j)}$ is the jth column of $\mathbf{A}$.

$$\nabla f(\mathbf{x}) = 2\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{y})$$

What is the time complexity of a gradient oracle for $\nabla f(\mathbf{x})$?

DECENT METHODS

Greedy approach: Given a starting point $\mathbf{x}$, make a small adjustment that decreases $f(\mathbf{x})$. In particular, $\mathbf{x} \leftarrow \mathbf{x} + \eta \mathbf{v}$.

What property do I want in $\mathbf{v}$?

Leading question: When η is small, what's an approximation for $f(\mathbf{x} + \eta \mathbf{v}) - f(\mathbf{x})$?

$$f(\mathbf{x} + \eta \mathbf{v}) - f(\mathbf{x}) \approx$$

DIRECTIONAL DERIVATIVES

$$D_{\mathbf{v}} f(\mathbf{x}) = \lim_{t \to 0} \frac{f(\mathbf{x} + t\mathbf{v}) - f(\mathbf{x})}{t} = \nabla f(\mathbf{x})^T \mathbf{v}.$$

So:

$$f(\mathbf{x} + \eta \mathbf{v}) - f(\mathbf{x}) \approx \eta \cdot \nabla f(\mathbf{x})^T \mathbf{v}.$$

How should we choose $\mathbf{v}$ so that $f(\mathbf{x} + \eta \mathbf{v}) < f(\mathbf{x})$?

GRADIENT DESCENT

Prototype algorithm:

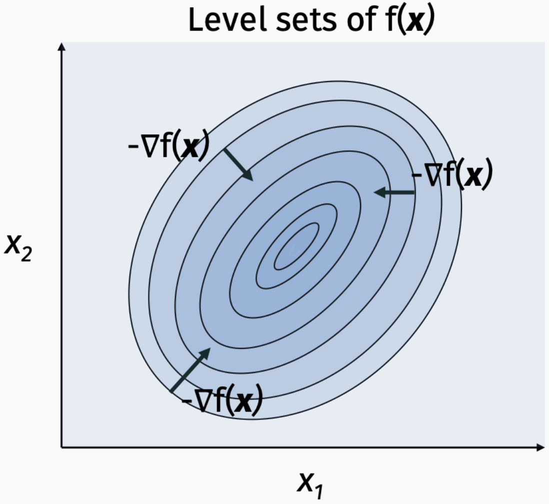
- Choose starting point $\mathbf{x}^{(0)}$.
- For $i = 0, \ldots, T$:
    - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f(\mathbf{x}^{(i)})$
- Return $\mathbf{x}^{(T)}$.

η is a step-size parameter, which is often adapted on the go. For now, assume it is fixed ahead of time.

GRADIENT DESCENT INTUITION

1 dimensional example:

2 dimensional example:



KEY RESULTS

For a convex function $f(\mathbf{x})$: For sufficiently small η and a sufficiently large number of iterations T, gradient descent will converge to a **near global minimum**:

$$f(\mathbf{x}^{(T)}) \leq f(\mathbf{x}^*) + \epsilon.$$

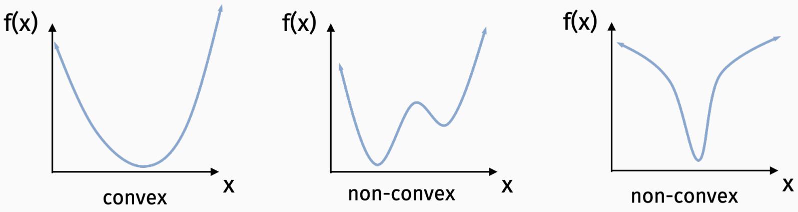
Examples: least squares regression, logistic regression, kernel regression, SVMs.

For a non-convex function $f(\mathbf{x})$: For sufficiently small η and a sufficiently large number of iterations T, gradient descent will converge to a **near stationary point**:

$$\|\nabla f(\mathbf{x}^{(T)})\|_2 \leq \epsilon.$$

Examples: neural networks, matrix completion problems, mixture models.

CONVEX VS. NON-CONVEX



One issue with non-convex functions is that they can have local minima. Even when they don't, convergence analysis requires different assumptions than convex functions.

APPROACH FOR THIS UNIT

We care about how fast gradient descent and related methods converge, not just that they do converge.

- Bounding iteration complexity requires placing some assumptions on $f(\mathbf{x})$.
- Stronger assumptions lead to better bounds on the convergence.

Understanding these assumptions can help us design faster variants of gradient descent (there are many!).

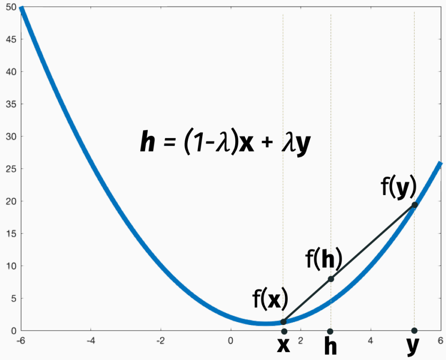
Today, we will start with **convex** functions.

CONVEXITY

Definition (Convex)

A function f is convex iff for any $\mathbf{x}, \mathbf{y}, \lambda \in [0,1]$:

$$(1-\lambda)\cdot f(\mathbf{x}) + \lambda \cdot f(\mathbf{y}) \geq f\left((1-\lambda)\cdot \mathbf{x} + \lambda \cdot \mathbf{y}\right)$$



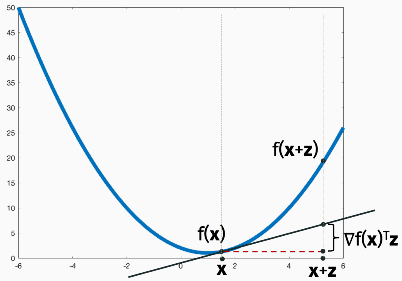
Definition (Convex)

A function f is convex if and only if for any $\mathbf{x}, \mathbf{y}$:

$$f(\mathbf{x}+\mathbf{z}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{z}$$

Equivalently:

$$f(\mathbf{x}) - f(\mathbf{y}) \leq \nabla f(\mathbf{x})^T (\mathbf{x}-\mathbf{y})$$



DEFINITIONS OF CONVEXITY

It is easy but not obvious how to prove the equivalence between these definitions. A short proof can be found in Karthik Sridharan's lecture notes here:

http://www.cs.cornell.edu/courses/cs6783/2018fa/lec16-supplement.pdf

GRADIENT DESCENT ANALYSIS

Assume:

- f is convex.
- Lipschitz function: for all $\mathbf{x}$, $\|\nabla f(\mathbf{x})\|_2 \leq G$.
- Starting radius: $\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2 \leq R$.

Gradient descent:

- Choose number of steps T.
- Starting point $\mathbf{x}^{(0)}$. E.g. $\mathbf{x}^{(0)} = \vec{0}$.
- $\eta = \frac{R}{G\sqrt{T}}$
- For $i = 0, \ldots, T$:
    - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f(\mathbf{x}^{(i)})$
- Return $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}^{(i)}} f(\mathbf{x}^{(i)})$.

GRADIENT DESCENT ANALYSIS

Claim (GD Convergence Bound)

If we run GD for $T \geq \frac{R^2 G^2}{\epsilon^2}$ iterations then $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

Proof is made tricky by the fact that $f(\mathbf{x}^{(i)})$ does not improve monotonically. We can "overshoot" the minimum.

GRADIENT DESCENT ANALYSIS

Claim (GD Convergence Bound)

If we run GD for $T \geq \frac{R^2G^2}{\epsilon^2}$ iterations with step-size $\eta = \frac{R}{G\sqrt{T}}$, then $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

Proof is made tricky by the fact that $f(\mathbf{x}^{(i)})$ does not improve monotonically. We can "overshoot" the minimum.

We will prove that the average solution value is low after $T = \frac{R^2G^2}{\epsilon^2}$ iterations. I.e. that:

$$\frac{1}{T}\sum_{i=0}^{T-1}\left[f(\mathbf{x}^{(i)}) - f(\mathbf{x}^*)\right] \leq \epsilon.$$

Of course the best solution found, $\hat{\mathbf{x}}$ is only better than the average.

GRADIENT DESCENT ANALYSIS

Claim (GD Convergence Bound)

If we run GD for $T \geq \frac{R^2G^2}{\epsilon^2}$ iterations with step-size $\eta = \frac{R}{G\sqrt{T}}$, then $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

Claim 1: For all $i = 0, \ldots, T$,

$$f(\mathbf{x}^{(i)}) - f(\mathbf{x}^*) \leq \frac{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$$

Claim 1(a): For all $i = 0, \ldots, T$,

$$\nabla f(\mathbf{x}^{(i)})^T(\mathbf{x}^{(i)} - \mathbf{x}^*) \leq \frac{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$$

Claim 1 follows from Claim 1(a) by definition of convexity.

GRADIENT DESCENT ANALYSIS

Claim (GD Convergence Bound)

If we run GD for $T \geq \frac{R^2G^2}{\epsilon^2}$ *iterations with step size* $\eta = \frac{R}{G\sqrt{T}}$*, then* $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

Claim 1(a): For all $i = 0, \ldots, T$,

$$\frac{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \geq \nabla f(\mathbf{x}^{(i)})^T(\mathbf{x}^{(i)} - \mathbf{x}^*)$$

GRADIENT DESCENT ANALYSIS

Claim (GD Convergence Bound)

If $T \geq \frac{R^2G^2}{\epsilon^2}$ and $\eta = \frac{R}{G\sqrt{T}}$, then $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

Claim 1: For all $i = 0, \ldots, T$,

$$f(\mathbf{x}^{(i)}) - f(\mathbf{x}^*) \leq \frac{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$$

Telescoping sum:

$$\begin{aligned}
\sum_{i=0}^{T-1} \left[f(\mathbf{x}^{(i)}) - f(\mathbf{x}^*)\right] \leq\ & \frac{\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \\
& + \frac{\|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(2)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \\
& + \frac{\|\mathbf{x}^{(2)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(3)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \\
& \vdots \\
& + \frac{\|\mathbf{x}^{(T-1)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2}
\end{aligned}$$

Claim (GD Convergence Bound)

If $T \geq \frac{R^2G^2}{\epsilon^2}$ and $\eta = \frac{R}{G\sqrt{T}}$, then $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

Telescoping sum:

$$\sum_{i=0}^{T-1}\left[f(\mathbf{x}^{(i)}) - f(\mathbf{x}^*)\right] \leq \frac{\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{T\eta G^2}{2}$$

$$\frac{1}{T}\sum_{i=0}^{T-1}\left[f(\mathbf{x}^{(i)}) - f(\mathbf{x}^*)\right] \leq \frac{R^2}{2T\eta} + \frac{\eta G^2}{2}$$

GRADIENT DESCENT ANALYSIS

Claim (GD Convergence Bound)

If $T \geq \frac{R^2G^2}{\epsilon^2}$ *and* $\eta = \frac{R}{G\sqrt{T}}$, *then* $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

Final step:

$$\frac{1}{T}\sum_{i=0}^{T-1}\left[f(\mathbf{x}^{(i)}) - f(\mathbf{x}^*)\right] \leq \epsilon$$

$$\left[\frac{1}{T}\sum_{i=0}^{T-1} f(\mathbf{x}^{(i)})\right] - f(\mathbf{x}^*) \leq \epsilon$$

We always have that $f(\hat{\mathbf{x}}) = \min_i f(\mathbf{x}^{(i)}) \leq \frac{1}{T}\sum_{i=0}^{T-1} f(\mathbf{x}^{(i)})$, which gives the final bound:

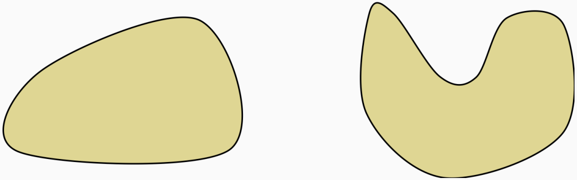
$$f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon.$$

CONSTRAINED CONVEX OPTIMIZATION

Typical goal: Solve a convex minimization problem with additional convex constraints.

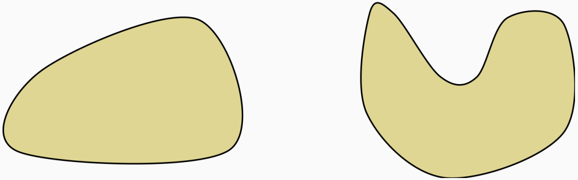
$$\min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$$

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



Which of these is convex?

CONSTRAINED CONVEX OPTIMIZATION



Definition (Convex set)

A set $\mathcal{S}$ is convex if for any $\mathbf{x}, \mathbf{y} \in \mathcal{S}, \lambda \in [0, 1]$:

$$(1 - \lambda)\mathbf{x} + \lambda\mathbf{y} \in \mathcal{S}.$$

CONSTRAINED CONVEX OPTIMIZATION

Examples:

- **Norm constraint:** minimize $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ subject to $\|\mathbf{x}\|_2 \leq \lambda$. Used e.g. for regularization, finding a sparse solution, etc.
- **Positivity constraint:** minimize $f(\mathbf{x})$ subject to $\mathbf{x} \geq 0$.
- **Linear constraint:** minimize $\mathbf{c}^T\mathbf{x}$ subject to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$. Linear program used in training support vector machines, industrial optimization, subroutine in integer programming, etc.

PROBLEM WITH GRADIENT DESCENT

Gradient descent:

- For $i = 0, \ldots, T$:
  - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f(\mathbf{x}^{(i)})$
- Return $\hat{\mathbf{x}} = \arg\min_i f(\mathbf{x}^{(i)})$.

Even if we start with $\mathbf{x}^{(0)} \in \mathcal{S}$, there is no guarantee that $\mathbf{x}^{(0)} - \eta \nabla f(\mathbf{x}^{(0)})$ will remain in our set.

Extremely simple modification: Force $\mathbf{x}^{(i)}$ to be in $\mathcal{S}$ by projecting onto the set.

CONSTRAINED FIRST ORDER OPTIMIZATION

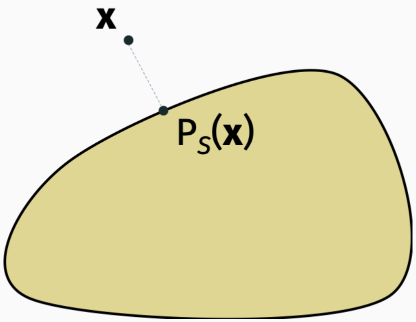
Given a function f to minimize and a convex constraint set $\mathcal{S}$, assume we have:

- **Function oracle**: Evaluate $f(\mathbf{x})$ for any $\mathbf{x}$.
- **Gradient oracle**: Evaluate $\nabla f(\mathbf{x})$ for any $\mathbf{x}$.
- **Projection oracle**: Evaluate $P_{\mathcal{S}}(\mathbf{x})$ for any $\mathbf{x}$.

$$P_{\mathcal{S}}(\mathbf{x}) = \arg\min_{\mathbf{y}\in\mathcal{S}} \|\mathbf{x}-\mathbf{y}\|_2$$

PROJECTION ORACLES

- How would you implement $P_{\mathcal{S}}$ for $\mathcal{S} = \{\mathbf{y} : \|\mathbf{y}\|_2 \leq 1\}$.
- How would you implement $P_{\mathcal{S}}$ for $\mathcal{S} = \{\mathbf{y} : \mathbf{y} = \mathbf{Q}\mathbf{z}\}$.



PROJECTED GRADIENT DESCENT

Given function $f(\mathbf{x})$ and set $\mathcal{S}$, such that $\|\nabla f(\mathbf{x})\|_2 \leq G$ for all $\mathbf{x} \in \mathcal{S}$ and starting point $\mathbf{x}^{(0)}$ with $\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2 \leq R$.

Projected gradient descent:

- Select starting point $\mathbf{x}^{(0)}$, $\eta = \frac{R}{G\sqrt{T}}$.
- For $i = 0, \ldots, T$:
  - $\mathbf{z} = \mathbf{x}^{(i)} - \eta \nabla f(\mathbf{x}^{(i)})$
  - $\mathbf{x}^{(i+1)} = P_{\mathcal{S}}(\mathbf{z})$
- Return $\hat{\mathbf{x}} = \arg\min_i f(\mathbf{x}^{(i)})$.

Claim (PGD Convergence Bound)

If $f, \mathcal{S}$ are convex and $T \geq \frac{R^2G^2}{\epsilon^2}$, then $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

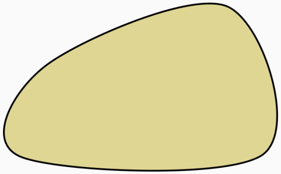
PROJECTED GRADIENT DESCENT ANALYSIS

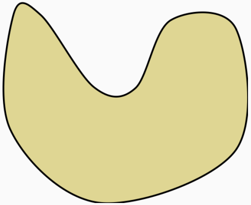
Analysis is almost identical to standard gradient descent! We just need one additional claim:

Claim (Contraction Property of Convex Projection)

If $\mathcal{S}$ is convex, then for <u>any</u> $\mathbf{y} \in \mathcal{S}$,

$$\|\mathbf{y} - P_{\mathcal{S}}(\mathbf{x})\|_2 \leq \|\mathbf{y} - \mathbf{x}\|_2.$$





GRADIENT DESCENT ANALYSIS

Claim (PGD Convergence Bound)

If $f, \mathcal{S}$ are convex and $T \geq \frac{R^2 G^2}{\epsilon^2}$, then $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

Claim 1: For all $i = 0, \ldots, T$, let $\mathbf{z}^{(i)} = \mathbf{x}^{(i)} - \eta \nabla f(\mathbf{x}^{(i)})$. Then:

$$\begin{aligned} f(\mathbf{x}^{(i)}) - f(\mathbf{x}^*) &\leq \frac{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_2^2 - \|\mathbf{z}^{(i)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \\ &\leq \frac{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \end{aligned}$$

Same telescoping sum argument:

$$\left[\frac{1}{T}\sum_{i=0}^{T-1} f(\mathbf{x}^{(i)})\right] - f(\mathbf{x}^*) \leq \frac{R^2}{2T\eta} + \frac{\eta G^2}{2}.$$

GRADIENT DESCENT

Conditions:

- **Convexity:** f is a convex function, $\mathcal{S}$ is a convex set.
- **Bounded initial distant:**

$$\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2 \leq R$$

- **Bounded gradients (Lipschitz function)**:

$$\|\nabla f(\mathbf{x})\|_2 \leq G \text{ for all } \mathbf{x} \in \mathcal{S}.$$

Theorem (GD Convergence Bound)

(Projected) Gradient Descent returns $\hat{\mathbf{x}}$ *with* $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) + \epsilon$ *after*

$$T = \frac{R^2 G^2}{\epsilon^2} \textit{ iterations.}$$

BEYOND THE BASIC BOUND

Can our convergence bound be tightened for certain functions? Can it guide us towards faster algorithms?

Goals:

- Improve ϵ dependence below $1/\epsilon^2$.
    - Ideally $1/\epsilon$ or $\log(1/\epsilon)$.
- Reduce or eliminate dependence on G and R.

Will need to take advantage of additional problem structure.

SMOOTHNESS

Definition (β-smoothness)

A function f is β smooth if, for all $\mathbf{x}, \mathbf{y}$

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq \beta \|\mathbf{x} - \mathbf{y}\|_2$$

For a scalar valued function f, equivalent to $f''(x) \leq \beta$. After

some calculus (see Lem. 3.4 in **Bubeck's book**), this implies:

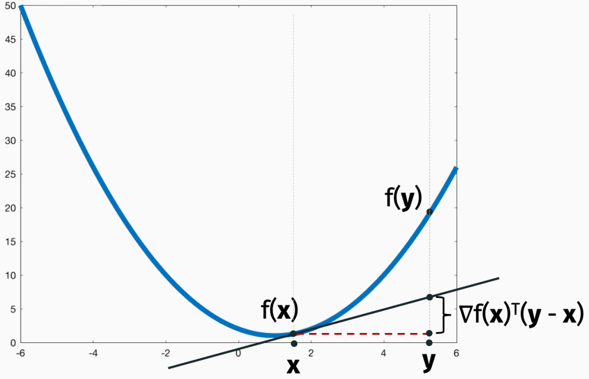
$$[f(\mathbf{y}) - f(\mathbf{x})] - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq \frac{\beta}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$$

SMOOTHNESS

Recall from convexity that $f(\mathbf{y}) - f(\mathbf{x}) \geq \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x})$.

So now we have an upper and lower bound.

$$0 \leq [f(\mathbf{y}) - f(\mathbf{x})] - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq \frac{\beta}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$$



CONVERGENCE GUARANTEE

Theorem (GD convergence for β-smooth functions.)

Let f be a β smooth convex function and assume we have $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$. If we run GD for T steps, we have:

$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \frac{2\beta R^2}{T}$$

Corollary: If $T = O\left(\frac{\beta R^2}{\epsilon}\right)$ we have $f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \epsilon$.

Compare this to $T = O\left(\frac{G^2 R^2}{\epsilon^2}\right)$ without a smoothness assumption.

GUARANTEED PROGRESS

Why do you think gradient descent might be faster when a function is β-smooth? Think about scalar case, in which case smoothness means $f''(x) \leq \beta$.

GUARANTEED PROGRESS

Previously learning rate/step size η depended on G. Now choose it based on β:

$$\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} - \frac{1}{\beta}\nabla f(\mathbf{x}^{(t)})$$

Progress per step of gradient descent:

1. $\left[f(\mathbf{x}^{(t+1)}) - f(\mathbf{x}^{(t)})\right] - \nabla f(\mathbf{x}^{(t)})^T(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}) \leq \frac{\beta}{2}\|\mathbf{x}^{(t)} - \mathbf{x}^{(t+1)}\|_2^2.$
2. $\left[f(\mathbf{x}^{(t+1)}) - f(\mathbf{x}^{(t)})\right] + \frac{1}{\beta}\|\nabla f(\mathbf{x}^{(t)})\|_2^2 \leq \frac{\beta}{2}\|\frac{1}{\beta}\nabla f(\mathbf{x}^{(t)})\|_2^2.$
3. $f(\mathbf{x}^{(t)}) - f(\mathbf{x}^{(t+1)}) \geq \frac{1}{2\beta}\|\nabla f(\mathbf{x}^{(t)})\|_2^2.$

CONVERGENCE GUARANTEE

Theorem (GD convergence for β-smooth functions.)

Let f be a β smooth convex function and assume we have $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$. If we run GD for T steps with $\eta = \frac{1}{\beta}$ we have:

$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \frac{2\beta R^2}{T}$$

Corollary: If $T = O\left(\frac{\beta R^2}{\epsilon}\right)$ we have $f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \epsilon$.

Again getting this result from the previous page is not hard, but also not obvious/direct. A concise proof can be found in Garrigos and Gower's notes.

GUARANTEED PROGRESS

Where did we use convexity in this proof?

Progress per step of gradient descent:

1. $\left[f(\mathbf{x}^{(t+1)}) - f(\mathbf{x}^{(t)})\right] - \nabla f(\mathbf{x}^{(t)})^T(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}) \leq \frac{\beta}{2}\|\mathbf{x}^{(t)} - \mathbf{x}^{(t+1)}\|_2^2.$
2. $\left[f(\mathbf{x}^{(t+1)}) - f(\mathbf{x}^{(t)})\right] + \frac{1}{\beta}\|\nabla f(\mathbf{x}^{(t)})\|_2^2 \leq \frac{\beta}{2}\|\frac{1}{\beta}\nabla f(\mathbf{x}^{(t)})\|_2^2.$
3. $f(\mathbf{x}^{(t)}) - f(\mathbf{x}^{(t+1)}) \geq \frac{1}{2\beta}\|\nabla f(\mathbf{x}^{(t)})\|_2^2.$

STATIONARY POINTS

Definition (Stationary point)

For a differentiable function f, a stationary point is any $\mathbf{x}$ with:

$$\nabla f(\mathbf{x}) = \mathbf{0}$$

local/global minima - local/global maxima - saddle points

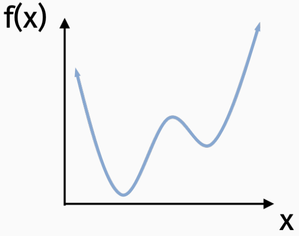
CONVERGENCE TO STATIONARY POINT

Theorem (Convergence to Stationary Point)

For any β-smooth differentiable function f (convex or not), if we run GD for T steps, we can find a point $\hat{\mathbf{x}}$ such that:

$$\|\nabla f(\hat{\mathbf{x}})\|_2^2 \leq \frac{2\beta}{T}\left(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)\right)$$

Corollary: If $T \geq \frac{2\beta}{\epsilon}$, then $\|\nabla f(\hat{\mathbf{x}})\|_2^2 \leq \epsilon\left(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)\right)$.



TELESCOPING SUM PROOF

Theorem (Convergence to Stationary Point)

For any β-smooth differentiable function f (convex or not), if we run GD for T steps, we can find a point $\hat{\mathbf{x}}$ such that:

$$\|\nabla f(\hat{\mathbf{x}})\|_2^2 \leq \frac{2\beta}{T}\left(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)\right)$$

We have that $\frac{1}{2\beta}\|\nabla f(\mathbf{x}^{(t)})\|_2^2 \leq f(\mathbf{x}^{(t)}) - f(\mathbf{x}^{(t+1)})$. So:

$$\sum_{t=0}^{T-1} \frac{1}{2\beta}\|\nabla f(\mathbf{x}^{(t)})\|_2^2 \leq f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(t)})$$

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(\mathbf{x}^{(t)})\|_2^2 \leq \frac{2\beta}{T}\left(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)\right)$$

$$\min_t \|\nabla f(\mathbf{x}^{(t)})\|_2^2 \leq \frac{2\beta}{T}\left(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)\right)$$

QUESTIONS IN NON-CONVEX OPTIMIZATION

If GD can find a stationary point, are there algorithms which find a stationary point faster using preconditioning, acceleration, stochastic methods, etc.?

QUESTIONS IN NON-CONVEX OPTIMIZATION

What if my function only has global minima and saddle points? Randomized methods (SGD, perturbed gradient methods, etc.) can provably “escape” saddle points.

Example: $\min_{\mathbf{x}} \frac{-\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$

- **Global minimum**: Top eigenvector of $\mathbf{A}^T\mathbf{A}$ (i.e., top principal component of $\mathbf{A}$).
- **Saddle points:** All other eigenvectors of $\mathbf{A}$.

Useful for lots of other matrix factorization problems beyond vanilla PCA.

BACK TO CONVEX FUNCTIONS

I said it was a bit tricky to prove that $f(\hat{x}) - f(x^*) \leq \frac{2\beta R^2}{T}$ for convex functions. But we just easily proved that $\|\nabla f(\hat{x})\|_2^2$ is small. Why doesn't this show we are close to the minimum?

STRONG CONVEXITY

Definition (α-strongly convex)

A convex function f is α-strongly convex if, for all $\mathbf{x}, \mathbf{y}$

$$[f(\mathbf{y}) - f(\mathbf{x})] - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \geq \frac{\alpha}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$$

Compare to smoothness condition.

$$[f(\mathbf{y}) - f(\mathbf{x})] - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq \frac{\beta}{2}\|\mathbf{x} - \mathbf{y}\|_2^2.$$

For a twice-differentiable scalar function f, equivalent to $f''(x) \geq \alpha$.

When f is convex, we always have that $f''(x) \geq 0$, so larger values of α correspond to a “stronger” condition.

GD FOR STRONGLY CONVEX FUNCTION

Gradient descent for strongly convex functions:

- Choose number of steps T.
- For $i = 1, \ldots, T$:
    - $\eta = \frac{2}{\alpha \cdot (i+1)}$
    - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f(\mathbf{x}^{(i)})$
- Return $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}^{(i)}} f(\mathbf{x}^{(i)})$.

CONVERGENCE GUARANTEE

Theorem (GD convergence for α-strongly convex functions.)

Let f be an α-strongly convex function and assume we have that, for all $\mathbf{x}$, $\|\nabla f(\mathbf{x})\|_2 \leq G$. If we run GD for T steps (with adaptive step sizes) we have:

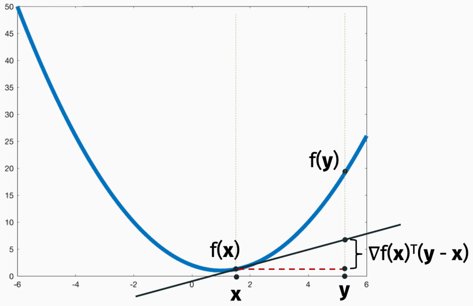
$$f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \leq \frac{2G^2}{\alpha(T-1)}$$

Corollary: If $T = O\left(\frac{G^2}{\alpha\epsilon}\right)$ we have $f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \leq \epsilon$

CONVERGENCE GUARANTEE

We could also have that f is both β-smooth and α-strongly convex.

$$\frac{\alpha}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 \leq [f(\mathbf{y})-f(\mathbf{x})]-\nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) \leq \frac{\beta}{2}\|\mathbf{x}-\mathbf{y}\|_2^2.$$



We will discuss and analyzing this setting after the midterm!