

# CS-GY 9223 D: Lecture 5

## Gradient Descent and Projected Gradient Descent

---

NYU Tandon School of Engineering, Prof. Christopher Musco

## PROJECT

- Choose your partner and email me by **next Wednesday, 10/14.**
- Topic and 1 page proposal due **11/04.**
- See project guidelines on course webpage for details.

## What techniques did we learn?

- Concentration Bounds

- Hash tables

- sketching
- near neighbor search
- load balancing

- Repetition to  
decrease variance  
+ increase success  
prob.

Johnson-Lindenstrauss  
Lemma

Sketches that  
preserve geometry

## NEW UNIT: CONTINUOUS OPTIMIZATION

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

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

Or at least  $\hat{\mathbf{x}}$  which is close to a minimum. E.g.

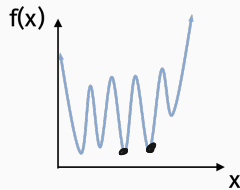
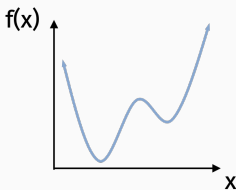
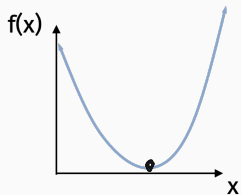
$$\underline{\underline{f(\hat{\mathbf{x}})}} \leq \underline{\min_{\mathbf{x}} f(\mathbf{x})} + \epsilon = f(\mathbf{x}^*) + \epsilon$$

Often we have some additional constraints:

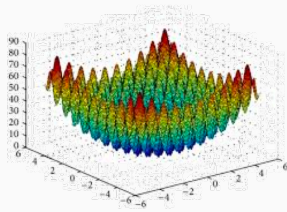
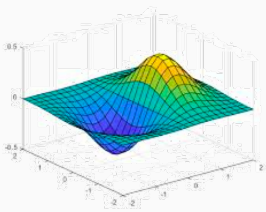
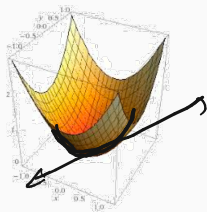
- $\mathbf{x} > 0$ .
- $\|\mathbf{x}\|_2 \leq R, \|\mathbf{x}\|_1 \leq R$ .
- $\mathbf{a}^T \mathbf{x} > c$ .

# CONTINUOUS OPTIMIZATION

Dimension  $d = 1$ :



Dimension  $d = 2$ :



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)

$\theta$

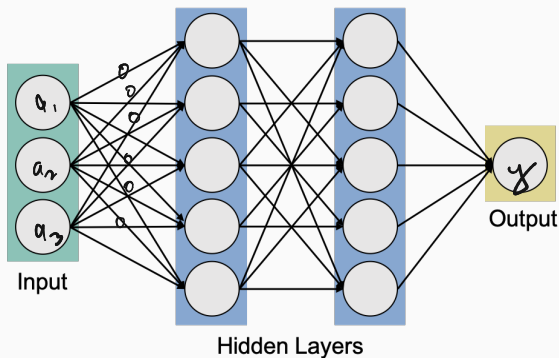
Let  $M_x$  be a model with parameters  $\underline{x} \equiv \{x_1, \dots, x_k\}$ , which takes as input a data vector  $\underline{a}$  and outputs a prediction.

Example:

$$M_x(\underline{a}) = \text{sign}(\underline{\underline{a}}^T \underline{\underline{x}})$$

# MACHINE LEARNING MODEL

Example:



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



Classic approach in supervised learning: 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  $\underline{\mathbf{a}}^{(1)}, \dots, \underline{\mathbf{a}}^{(n)}$  with outputs  $\underline{y}^{(1)}, \dots, \underline{y}^{(n)}$ .

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

**How do we turn this into a function minimization problem?**

## LOSS FUNCTION

**Loss function**  $L(M_x(\mathbf{a}), y)$ : Some measure of distance between prediction  $M_x(\mathbf{a})$  and target output  $y$ . Increases if they are further apart.

- Squared ( $\ell_2$ ) loss:  $|M_x(\mathbf{a}) - y|^2$
- Absolute deviation ( $\ell_1$ ) loss:  $|M_x(\mathbf{a}) - y|$
- Hinge loss:  $1 - y \cdot M_x(\mathbf{a})$
- Cross-entropy loss (log loss).
- Etc.

$$a^{(1)}, \dots, a^{(n)}$$

Empirical risk minimization:

$$\underline{f(x)} = \sum_{i=1}^n L(\underbrace{M_x(a^{(i)}), y^{(i)}})$$

Solve the optimization problem  $\min_x f(x)$ .

## EXAMPLE: LINEAR REGRESSION

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

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

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

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.

$$\|\mathbf{x}\|_2 \leq c.$$

Momentum + Accelerated Gradient Descent

What are some example algorithms for continuous optimization?

Newton Method

Linear Programs.

Quadratic Programming.

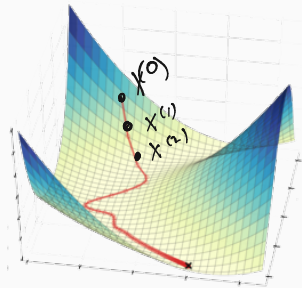
Gradient Descent.

Coordinate Descent

Stochastic Gradient Descent

# GRADIENT DESCENT

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



## CALCULUS REVIEW

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

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \underset{i}{1} & 0 & 0 & 0 \end{bmatrix} = \mathbf{e}^{(i)}$$

Partial derivative:

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

Directional derivative:

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

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:

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

$$\nabla f(\mathbf{x}) \cdot \mathbf{v}$$

$$\langle \nabla f(\mathbf{x}), \mathbf{v} \rangle$$



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:

$$A \in \mathbb{R}^{n \times d}$$

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

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

→ can compute in  $O(nd)$  time

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

$(n \times d)(d \times 1) \rightarrow n \times 1$

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

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

$$(d \times n)(n \times 1) = (d \times 1)$$

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

$O(nd)$  time.

$$2\mathbf{A}^T (\mathbf{Ax}) - \mathbf{Ay}$$

**Greedy approach:** Given a starting point  $x$ , make a small adjustment that decreases  $f(x)$ . In particular, ~~take~~  $x + \underline{\eta}v$  and  ~~$f(x)$~~   $f(x + \eta v)$ .

$$f(x + \eta v) < f(x)$$

What property do I want in  $v$ ?

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

$$\begin{aligned} \underline{f(x + \eta v) - f(x)} &\approx \eta D_v f(x) \\ &= \eta \cdot (v^\top \nabla f(x)) \end{aligned}$$

## DIRECTIONAL DERIVATIVES

$$D_v f(x) = \lim_{t \rightarrow 0} \frac{f(x + tv) - f(x)}{t} = \nabla f(x)^T v.$$

So:

$$f(x + \eta v) - f(x) \approx \eta \left( v^T \nabla f(x) \right)$$

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

$$\begin{aligned} v &= -\nabla f(x) & \underline{v^T \nabla f(x)} &= -\nabla f(x)^T \nabla f(x) \\ & & &= -\underline{\|\nabla f(x)\|_2^2} \end{aligned}$$

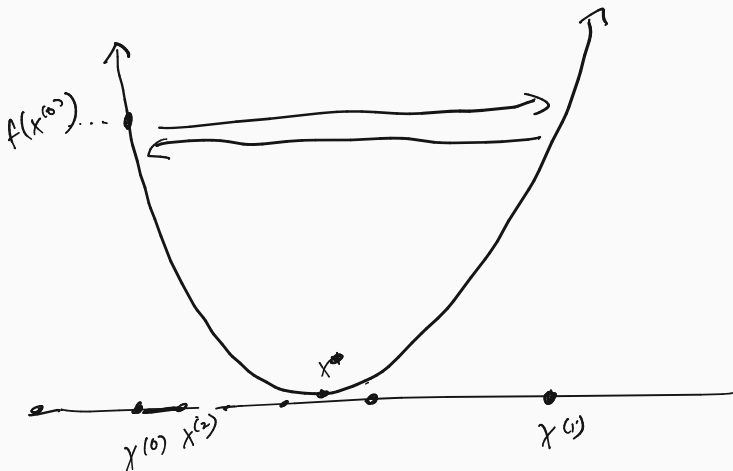
Prototype algorithm:

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

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

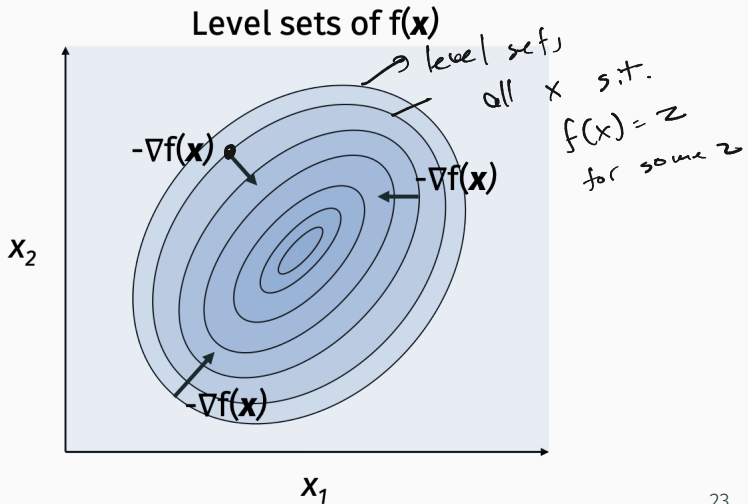
## GRADIENT DESCENT INTUITION

1 dimensional example:  $\nabla f(x^{(0)})$



# GRADIENT DESCENT INTUITION

2 dimensional example:



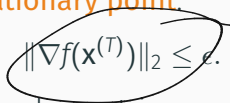
## KEY RESULTS

**For a convex function  $f(\mathbf{x})$ :** For sufficiently small  $\eta$  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.

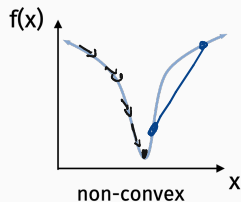
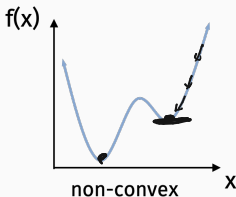
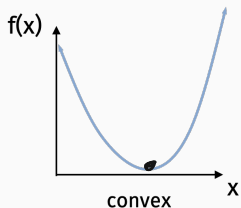
**For a non-convex function  $f(\mathbf{x})$ :** For sufficiently small  $\eta$  and a sufficiently large number of iterations  $T$ , gradient descent will converge to a **near stationary point**:


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

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!).

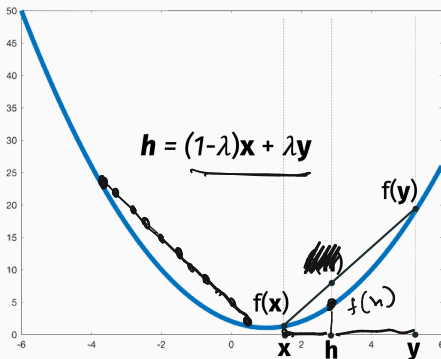
Today, we will start with **convex functions** only.

# CONVEXITY

## Definition (Convex)

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

$$(1 - \lambda) \cdot f(\underline{x}) + \lambda \cdot f(\underline{y}) \geq f(\underbrace{(1 - \lambda) \cdot \underline{x} + \lambda \cdot \underline{y}})$$



# GRADIENT DESCENT

## Definition (Convex)

A function  $f$  is convex if and only if for any  $x, y$ :

For any  $x, z$

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

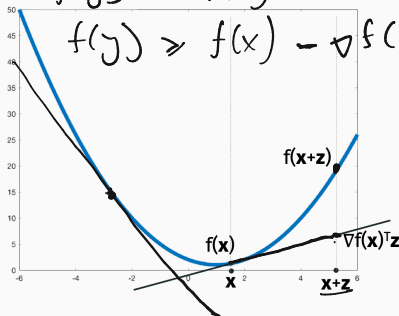
Equivalently:

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

$$z = y - x$$

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

$$f(y) \geq f(x) - \nabla f(x)^T (x - y)$$



# 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  $T$ .
- $\eta = \frac{R}{G\sqrt{T}}$
- For  $i = 0, \dots, T$ :
  - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f(\mathbf{x}^{(i)})$
- Return  $\hat{\mathbf{x}} = \arg \min_i f(\mathbf{x}^{(i)})$ .

$$\hat{\mathbf{x}} = \mathbf{x}^{(\tau)}$$

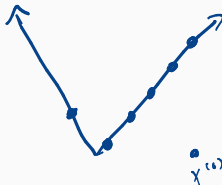
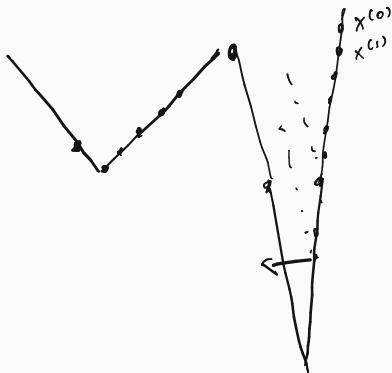
# GRADIENT DESCENT ANALYSIS

## Claim (GD Convergence Bound)

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

$$n > \frac{B}{G\sqrt{T}}$$

$$n = \frac{\epsilon}{G^2}$$



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

# GRADIENT DESCENT ANALYSIS

## Claim (GD Convergence Bound)

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

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

$$z = x^{(i)} - \eta \nabla f(x^{(i)})$$

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

# GRADIENT DESCENT ANALYSIS

## Claim (GD Convergence Bound)

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

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

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

Telescoping sum:

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

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

Handwritten annotations for the final inequality:

- $\frac{R^2}{2T\eta} = \frac{R^2}{2T \cdot \frac{R}{G\sqrt{T}}} = \frac{R G}{\sqrt{T}}$
- $\frac{\eta G^2}{2} = \frac{\frac{R}{G\sqrt{T}} G^2}{2} = \frac{R G}{2\sqrt{T}}$
- Sum:  $\frac{R G}{\sqrt{T}} + \frac{R G}{2\sqrt{T}} = \frac{3 R G}{2\sqrt{T}} \leq \epsilon$  (assuming  $\frac{3 R G}{2\sqrt{T}} \leq \epsilon$ )



## GRADIENT DESCENT ANALYSIS

### Claim (GD Convergence Bound)

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

Final step:

$$\begin{aligned} f(\hat{\mathbf{x}}) &\leq \frac{1}{T} \sum_{i=0}^{T-1} [f(\mathbf{x}^{(i)}) - f(\mathbf{x}^*)] \leq \epsilon \\ &\quad \left[ \frac{1}{T} \sum_{i=0}^{T-1} f(\mathbf{x}^{(i)}) \right] - \underline{f(\mathbf{x}^*)} \leq \epsilon \end{aligned}$$

We always have that  $\min_i f(\mathbf{x}^{(i)}) \leq \frac{1}{T} \sum_{i=0}^{T-1} f(\mathbf{x}^{(i)})$ , so this is what we return:

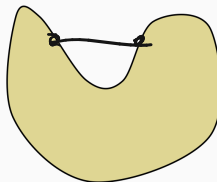
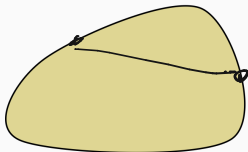
$$\begin{aligned} \frac{1}{T} \sum_{i=0}^{T-1} f(\mathbf{x}^{(i)}) &\geq \frac{1}{T} \sum_{i=0}^{T-1} (\min_{j \in \{1, \dots, T\}} f(\mathbf{x}^{(j)})) = \frac{1}{T} \sum_{i=0}^{T-1} f(\hat{\mathbf{x}}) = f(\hat{\mathbf{x}}) \end{aligned}$$

# CONSTRAINED CONVEX OPTIMIZATION

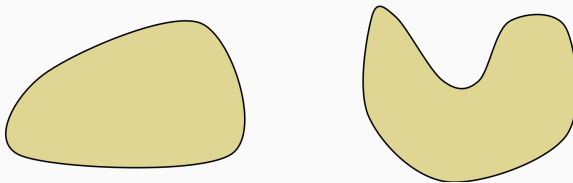
**Typical goal:** Solve a convex minimization problem with additional convex constraints.

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

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



Which of these is convex?



## Definition (Convex set)

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

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

## PROBLEM WITH GRADIENT DESCENT

Gradient descent:

- For  $i = 0, \dots, T$ :
  - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f(\mathbf{x}^{(i)})$
- 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  $\underline{\mathbf{x}^{(0)} - \eta \nabla f(\mathbf{x}^{(0)})}$  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

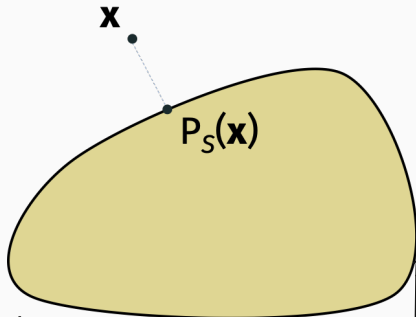
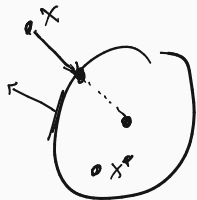
Given a function  $f$  to minimize and a convex constraint set  $\mathcal{S}$ , assume we have:

- **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  $\mathbf{x}$ .

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

# PROJECTION ORACLES

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



$$P_S(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$$

$$\frac{\mathbf{Q}(\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T\mathbf{x}}{\|\mathbf{Q}^T\mathbf{x}\|_2}$$

$$\frac{\mathbf{Q}(\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T\mathbf{x}}{\|\mathbf{Q}^T\mathbf{x}\|_2}$$

$$\mathbf{y} = \mathbf{Q}\mathbf{z}$$

## 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  $i = 0, \dots, T$ :
  - $\mathbf{z} = \mathbf{x}^{(i)} - \eta \nabla f(\mathbf{x}^{(i)})$
  - $\mathbf{x}^{(i+1)} = P_{\mathcal{S}}(\mathbf{z})$
- Return  $\hat{\mathbf{x}} = \arg \min_i f(\mathbf{x}^{(i)})$ .

**Claim (PGD Convergence Bound)**

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

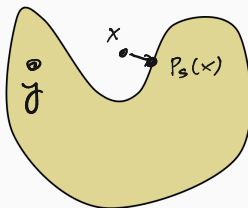
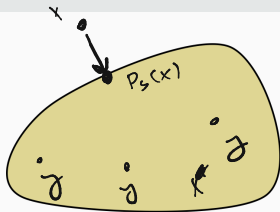
## PROJECTED GRADIENT DESCENT ANALYSIS

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

### Claim (Contraction Property of Convex Projection)

If  $S$  is convex, then for any  $y \in S$ ,

$$\|y - P_S(x)\|_2 \leq \|y - x\|_2.$$





# GRADIENT DESCENT ANALYSIS

## Claim (PGD Convergence Bound)

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

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

$$\begin{aligned} f(x^{(i)}) - f(x^*) &\leq \frac{\|x^{(i)} - x^*\|_2^2 - \|z - x^*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \quad \text{from before} \\ &\leq \frac{\|x^{(i)} - x^*\|_2^2 - \|x^{(i+1)} - x^*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \quad \text{in } \|\nabla f(x^{(i)})\| \end{aligned}$$

$$\|x^{(i+1)} - x^*\|_2^2 \leq \|z - x^*\|_2^2$$

Same telescoping sum argument:

$$\min_i f(x^{(i)}) - f(x^*) \leq \frac{1}{T} \sum_{i=0}^{T-1} f(x^{(i)}) - f(x^*) \leq \frac{R^2}{2T\eta} + \frac{\eta G^2}{2} \leq \epsilon$$