CS-GY 6763: Lecture 5
Gradient Descent and Projected Gradient Descent

NYU Tandon School of Engineering, Prof. Christopher Musco

PROJECT

- Choose your partner and email me by end of this week (deadline was originally today).
- Sign-up to present or lead discussion for 1 reading group slot. We need presenters for next week!

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

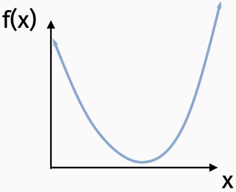
$$f(\mathbf{x}^*) = \min_{\mathbf{x}} f(\mathbf{x}).$$

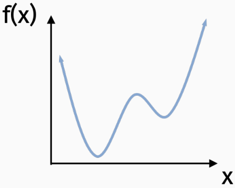
Or at least $\hat{\mathbf{x}}$ which is close to a minimum. E.g. $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x}} f(\mathbf{x}) + \epsilon$

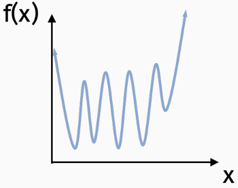
Often we have some additional constraints:

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

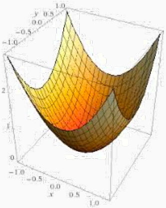
Dimension $d = 1$:

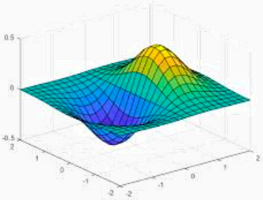


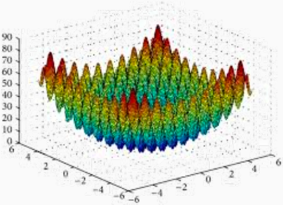




Dimension $d = 2$:







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

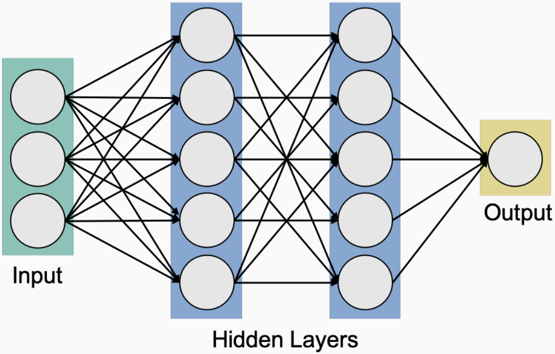
- numerical value (probability stock price increases)
- label (is the image a cat? does the image contain a car?)
- decision (turn car left, rotate robotic arm)

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:

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

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 $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(n)}$ 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 $L(M_{\mathsf{x}}(\mathsf{a}), y)$: 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:

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

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

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

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

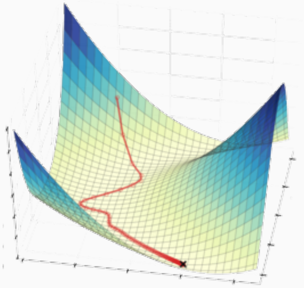
where $\mathbf{A}$ is a matrix with $\mathbf{a}^{(i)}$ as its i^{th} row and $\mathbf{y}$ is a vector with $y^{(i)}$ as its i^{th} entry.

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?

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



(and sometimes we can prove it works)

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:

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

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:

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

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.

Linear least-squares regression:

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

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

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

where $\boldsymbol{\alpha}^{(j)}$ is the j^{th} 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 $\nabla f(\mathbf{x})$?

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

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

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

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

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

So:

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

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

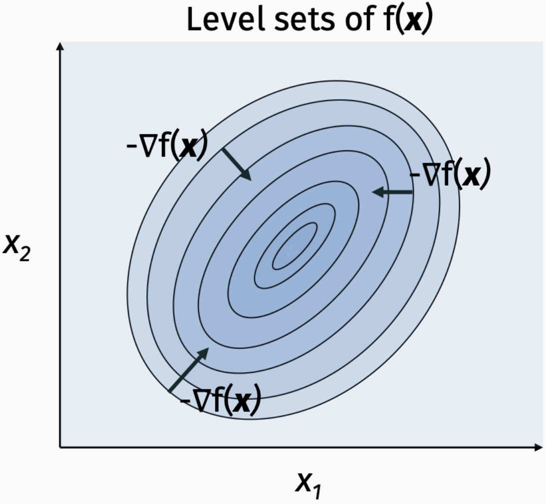
Prototype algorithm:

- 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.

1 dimensional example:

2 dimensional example:



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**:

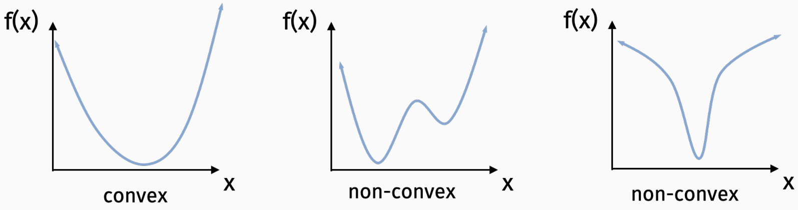
$$f(\mathbf{x}^{(T)}) \leq f(\mathbf{x}^*) + \epsilon.$$

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.



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.

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.

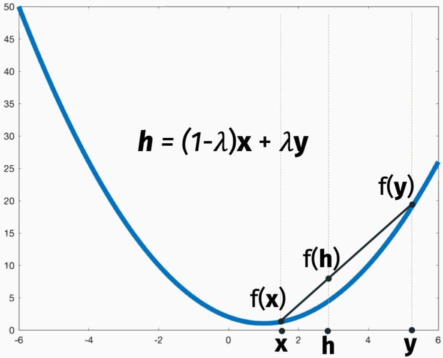
Understanding these assumptions can help us design faster variants of gradient descent (there are many!).

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

Definition (Convex)

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

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



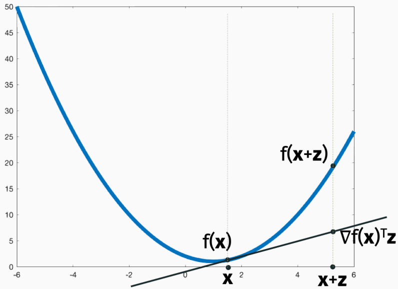
Definition (Convex)

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:

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



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 $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}^{(i)}} f(\mathbf{x}^{(i)})$.

Claim (GD Convergence Bound)

If $T \geq \frac{R^2G^2}{\epsilon^2}$, *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.

Claim (GD Convergence Bound)

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

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

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

Claim (GD Convergence Bound)

If $T \geq \frac{R^2 G^2}{\epsilon^2}$ *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:

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

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

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:

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

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

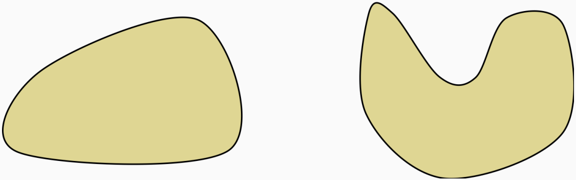
We always have that $\min_i f(\mathbf{x}^{(i)}) \leq \frac{1}{T}\sum_{i=0}^{T-1} f(\mathbf{x}^{(i)})$, so this is what we return:

$$f(\hat{\mathbf{x}}) = \min_{i \in 1,\ldots,T} f(\mathbf{x}^{(i)}) \leq f(\mathbf{x}^*) + \epsilon.$$

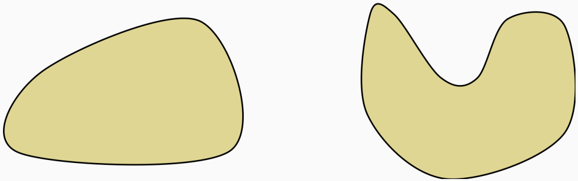
Typical goal: Solve a convex minimization problem with additional convex constraints.

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

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



Which of these is convex?



Definition (Convex set)

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

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

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.

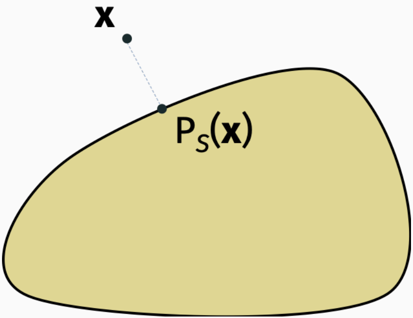
Extremely simple modification: Force $\mathbf{x}^{(i)}$ to be in $\mathcal{S}$ by **projecting** onto the set.

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 $\mathbf{x}$.

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

- 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 $\mathcal{S} = \{\mathbf{y} : \mathbf{y} = \mathbf{Q}\mathbf{z}\}$.



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)

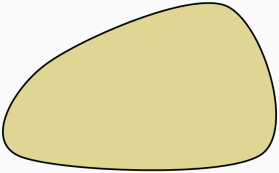
If $f, \mathcal{S}$ are convex and $T \geq \frac{R^2G^2}{\epsilon^2}$, then $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

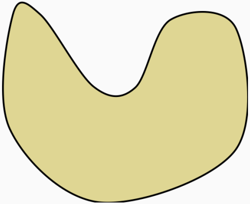
Analysis is almost identical to standard gradient descent! We just need one additional claim:

Claim (Contraction Property of Convex Projection)

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

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





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 $i = 0, \ldots, T$,

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

Same telescoping sum argument:

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

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

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

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.
- **Next class:** Take advantage of additional problem structure (e.g. repetition in features and data points in ML problems).

Definition (β-smoothness)

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

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

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

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

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

Recall from definition of 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.

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

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:

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

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

$$f(\mathbf{x}^{(t)}) - f(\mathbf{x}^{(t+1)}) \geq \frac{1}{2\beta}\|\nabla f(\mathbf{x}^{(t)})\|_2^2$$

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

Definition (α-strongly convex)

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

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

α is a parameter that will depend on our function.

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

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 $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}^{(i)}} f(\mathbf{x}^{(i)})$.

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

What if f is both β-smooth and α-strongly convex?

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

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

Theorem (GD for β-smooth, α-strongly convex.)

Let f be a β-smooth and α-strongly convex function. If we run GD for T steps (with step size $\eta = \frac{1}{\beta}$) we have:

$$\|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2 \leq e^{-(T-1)\frac{\alpha}{\beta}}\|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2^2$$

$\kappa = \frac{\beta}{\alpha}$ is called the "condition number" of f.

Is it better if κ is large or small?

Converting to more familiar form: Using that fact the $\nabla f(\mathbf{x}^*) = \mathbf{0}$ along with

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

we have:

$$\|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2^2 \leq \frac{2}{\alpha}\left[f(\mathbf{x}^{(1)}) - f(\mathbf{x}^*)\right]$$
$$\|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2 \geq \frac{2}{\beta}\left[f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*)\right]$$

Corollary (GD for β-smooth, α-strongly convex.)

Let f be a β-smooth and α-strongly convex function. If we run GD for T steps (with step size $\eta = \frac{1}{\beta}$) we have:

$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \frac{\beta}{\alpha} e^{-(T-1)\frac{\alpha}{\beta}} \cdot \left[f(\mathbf{x}^{(1)}) - f(\mathbf{x}^*)\right]$$

Corollary: If $T = O\left(\frac{\beta}{\alpha}\log(\beta/\alpha\epsilon)\right) = O(\kappa\log(\kappa/\epsilon))$ we have:

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

Alternative Corollary: If $T = O\left(\frac{\beta}{\alpha}\log(R\beta/\epsilon)\right)$ we have:

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

Let f be a twice differentiable function from $\mathbb{R}^d \to \mathbb{R}$. Let the **Hessian** $\mathbf{H} = \nabla^2 f(\mathbf{x})$ contain all of its second derivatives at a point $\mathbf{x}$. So $\mathbf{H} \in \mathbb{R}^{d \times d}$. We have:

$$\mathbf{H}_{i,j} = \left[\nabla^2 f(\mathbf{x})\right]_{i,j} = \frac{\partial^2 f}{\partial x_i x_j}.$$

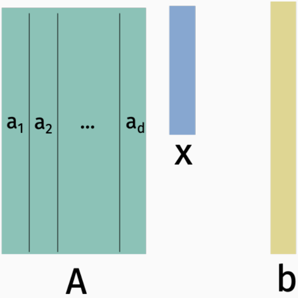
For vector $\mathbf{x}, \mathbf{v}$:

$$\nabla f(\mathbf{x} + t\mathbf{v}) \approx \nabla f(\mathbf{x}) + t\left[\nabla^2 f(\mathbf{x})\right]\mathbf{v}.$$

Let f be a twice differentiable function from $\mathbb{R}^d \to \mathbb{R}$. Let the Hessian $\mathbf{H} = \nabla^2 f(\mathbf{x})$ contain all of its second derivatives at a point $\mathbf{x}$. So $\mathbf{H} \in \mathbb{R}^{d \times d}$. We have:

$$\mathbf{H}_{i,j} = \left[\nabla^2 f(\mathbf{x})\right]_{i,j} = \frac{\partial^2 f}{\partial x_i x_j}.$$

Example: Let $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$. Recall that $\nabla f(\mathbf{x}) = 2\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$.



Claim: If f is twice differentiable, then it is convex if and only if the matrix $\mathbf{H} = \nabla^2 f(\mathbf{x})$ is positive semidefinite for all $\mathbf{x}$.

Definition (Positive Semidefinite (PSD))

A square, symmetric matrix $\mathbf{H} \in \mathbb{R}^{d\times d}$ is positive semidefinite (PSD) for any vector $\mathbf{y} \in \mathbb{R}^d$, $\mathbf{y}^T\mathbf{H}\mathbf{y} \geq 0$.

This is a natural notion of "positivity" for symmetric matrices. To denote that $\mathbf{H}$ is PSD we will typically use "Loewner order" notation (`\succeq` in LaTex):

$$\mathbf{H} \succeq 0.$$

We write $\mathbf{B} \succeq \mathbf{A}$ or equivalently $\mathbf{A} \preceq \mathbf{B}$ to denote that $(\mathbf{B} - \mathbf{A})$ is positive semidefinite. This gives a partial ordering on matrices.

Claim: If f is twice differentiable, then it is convex if and only if the matrix $\mathbf{H} = \nabla^2 f(\mathbf{x})$ is positive semidefinite for all $\mathbf{x}$.

Definition (Positive Semidefinite (PSD))

A square, symmetric matrix $\mathbf{H} \in \mathbb{R}^{d \times d}$ is positive semidefinite (PSD) for any vector $\mathbf{y} \in \mathbb{R}^d$, $\mathbf{y}^T \mathbf{H} \mathbf{y} \geq 0$.

For the least squares regression loss function: $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, $\mathbf{H} = \nabla^2 f(\mathbf{x}) = 2\mathbf{A}^T\mathbf{A}$ for all $\mathbf{x}$. Is $\mathbf{H}$ PSD?

If f is β-smooth and α-strongly convex then at any point $\mathbf{x}$, $\mathbf{H} = \nabla^2 f(\mathbf{x})$ satisfies:

$$\alpha \mathbf{I}_{d\times d} \preceq \mathbf{H} \preceq \beta \mathbf{I}_{d\times d},$$

where $\mathbf{I}_{d\times d}$ is a $d \times d$ identity matrix.

This is the natural matrix generalization of the statement for scalar valued functions:

$$\alpha \leq f''(x) \leq \beta.$$

SMOOTH AND STRONGLY CONVEX HESSIAN

$$\alpha \mathbf{I}_{d\times d} \preceq \mathbf{H} \preceq \beta \mathbf{I}_{d\times d}.$$

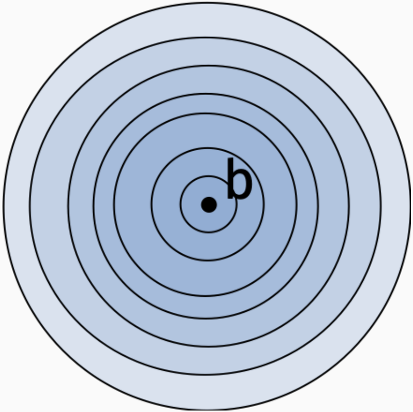
Equivalently for any $\mathbf{z}$,

$$\alpha\|\mathbf{z}\|_2^2 \leq \mathbf{z}^T\mathbf{H}\mathbf{z} \leq \beta\|\mathbf{z}\|_2^2.$$

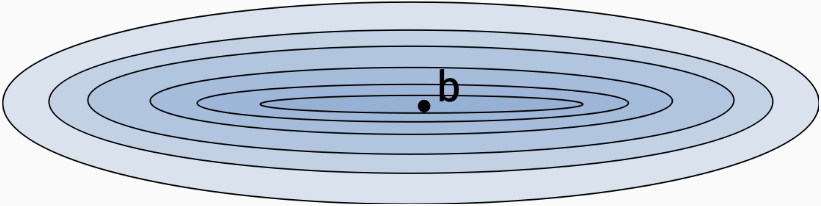
Let $f(\mathbf{x}) = \|\mathbf{D}\mathbf{x} - \mathbf{b}\|_2^2$ where $\mathbf{D}$ is a diagaonl matrix. For now imagine we're in two dimensions: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\mathbf{D} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}$.

What are α, β for this problem?

$$\alpha\|\mathbf{z}\|_2^2 \leq \mathbf{z}^T\mathbf{H}\mathbf{z} \leq \beta\|\mathbf{z}\|_2^2$$

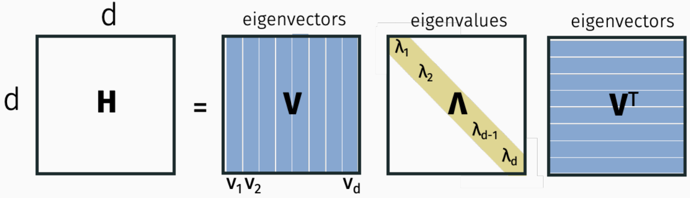


Level sets of $\|\mathbf{Dx} - \mathbf{b}\|_2^2$ when $d_1^2 = 1, d_2^2 = 1$.



Level sets of $\|\mathbf{D}\mathbf{x} - \mathbf{b}\|_2^2$ when $d_1^2 = \frac{1}{3}, d_2^2 = 2$.

Any symmetric matrix $\mathbf{H}$ has an orthogonal, real valued eigendecomposition.

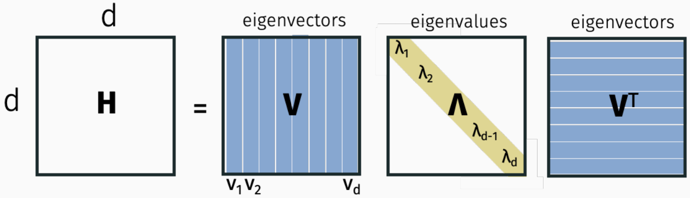


Here $\mathbf{V}$ is square and orthogonal, so $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$. And for each $\mathbf{v}_i$, we have:

$$\mathbf{H}\mathbf{v}_i = \lambda_i \mathbf{v}_i.$$

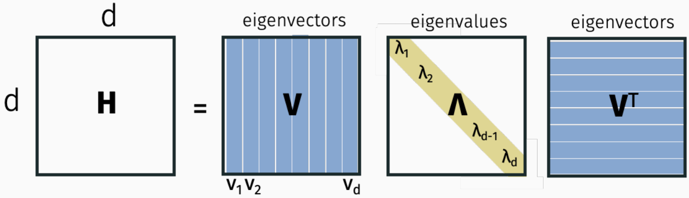
By definition, that's what makes $\mathbf{v}_1, \ldots, \mathbf{v}_d$ eigenvectors.

Recall $\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}$.



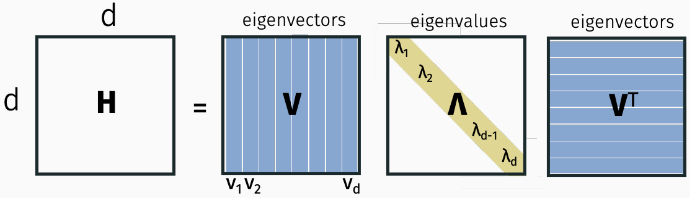
Claim: H is PSD $\Leftrightarrow \lambda_1, ..., \lambda_d \geq 0$.

Recall $\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}$.



Claim: $\alpha \mathbf{I} \preceq \mathbf{H} \preceq \beta \mathbf{I} \Leftrightarrow \alpha \leq \lambda_1, ..., \lambda_d \leq \beta$.

Recall $\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}$.



In other words, if we let $\lambda_{\max}(\mathbf{H})$ and $\lambda_{\min}(\mathbf{H})$ be the smallest and largest eigenvalues of $\mathbf{H}$, then for all $\mathbf{z}$ we have:

$$\mathbf{z}^T\mathbf{H}\mathbf{z} \leq \lambda_{\max}(\mathbf{H}) \cdot \|\mathbf{z}\|^2$$
$$\mathbf{z}^T\mathbf{H}\mathbf{z} \geq \lambda_{\min}(\mathbf{H}) \cdot \|\mathbf{z}\|^2$$

If the maximum eigenvalue of $\mathbf{H} = \nabla^2 f(\mathbf{x}) = \beta$ and the minimum eigenvalue of $\mathbf{H} = \nabla^2 f(\mathbf{x}) = \alpha$ then $f(\mathbf{x})$ is β-smooth and α-strongly convex.

$$\lambda_{\max}(\mathbf{H}) = \beta$$
$$\lambda_{\min}(\mathbf{H}) = \alpha$$

Theorem (GD for β-smooth, α-strongly convex.)

Let f be a β-smooth and α-strongly convex function. If we run GD for T steps (with step size $\eta = \frac{2}{\beta}$) we have:

$$\|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2 \leq e^{-T/\kappa}\|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2$$

Goal: Prove for $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$.

Let $\lambda_{\max} = \lambda_{\max}(\mathbf{A}^T\mathbf{A})$. Gradient descent update is:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \frac{1}{2\,\lambda_{\max}} 2\mathbf{A}^T(\mathbf{A}\mathbf{x}^{(t)} - \mathbf{b})$$

Richardson Iteration view:

$$(\mathbf{x}^{(t+1)} - \mathbf{x}^*) = \left(\mathbf{I} - \frac{1}{\lambda_{\max}}\mathbf{A}^T\mathbf{A}\right)(\mathbf{x}^{(t)} - \mathbf{x}^*)$$

What is the maximum eigenvalue of the symmetric matrix $\left(\mathbf{I} - \frac{1}{\lambda_{\max}}\mathbf{A}^T\mathbf{A}\right)$ in terms of the eigenvalues $\lambda_{\max} = \lambda_1 \geq \ldots \geq \lambda_d = \lambda_{\min}$ of $\mathbf{A}^T\mathbf{A}$?

$$(\mathbf{x}^{(T+1)} - \mathbf{x}^*) = \left(\mathbf{I} - \frac{1}{\lambda_{\max}}\mathbf{A}^T\mathbf{A}\right)^T (\mathbf{x}^{(1)} - \mathbf{x}^*)$$

What is the maximum eigenvalue of the symmetric matrix $\left(\mathbf{I} - \frac{1}{\lambda_{\max}}\mathbf{A}^T\mathbf{A}\right)^T$?

So we have $\|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2 \leq$

We now have a pretty good understanding of gradient descent.

Number of iterations for ϵ error:

	G-Lipschitz	β-smooth
R bounded start	$O\left(\frac{G^2R^2}{\epsilon^2}\right)$	$O\left(\frac{\beta R^2}{\epsilon}\right)$
α-strong convex	$O\left(\frac{G^2}{\alpha\epsilon}\right)$	$O\left(\frac{\beta}{\alpha}\log(1/\epsilon)\right)$

ACCELERATION

Nesterov's accelerated gradient descent:

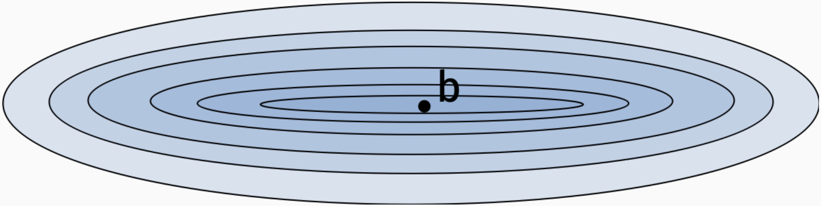
- $\mathbf{x}^{(1)} = \mathbf{y}^{(1)} = \mathbf{z}^{(1)}$
- For $t = 1, \ldots, T$
  - $\mathbf{y}^{(t+1)} = \mathbf{x}^{(t)} - \frac{1}{\beta}\nabla f(\mathbf{x}^{(t)})$
  - $\mathbf{x}^{(t+1)} = \left(1 + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)\mathbf{y}^{(t+1)} + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\left(\mathbf{y}^{(t+1)} - \mathbf{y}^{(t)}\right)$

Theorem (AGD for β-smooth, α-strongly convex.)

Let f be a β-smooth and α-strongly convex function. If we run AGD for T steps we have:

$$f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*) \leq \kappa e^{-(t-1)\sqrt{\kappa}}\left[f(\mathbf{x}^{(1)}) - f(\mathbf{x}^*)\right]$$

Corollary: If $T = O\left(\sqrt{\kappa}\log(\kappa/\epsilon)\right)$ achieve error ϵ.



Level sets of $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$.

Other terms for similar ideas:

- Momentum
- Heavy-ball methods

What if we look back beyond two iterates?