# CS-GY 9223 I: Lecture 7
# Preconditioning, acceleration, coordinate decent, etc.

NYU Tandon School of Engineering, Prof. Christopher Musco

# LOGISTICS

- Self-proctored, 2-hour midterm to be taken anytime next week.
- No Collaboration allowed at all. Or outside resources. Just use your own notes and material from the class.
- Sample problems are available on course website. We can review during office hours tomorrow or next week.
- You should have received an invite to Gradescope. Hopefully tonight/tomorrow I can upload a ”practice test” to make sure their system works.

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*

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

$\mathbf{x}^* = \min_{\mathbf{x}} \sum_{i=1}^{T} f_i(\mathbf{x}^*)$ (the offline optimum)

**Conditions:**

- $f_1, \ldots, f_T$ are all convex.
- Each is $G$-Lipschitz: for all $\mathbf{x}$, $i$, $\|\nabla f_i(\mathbf{x})\|_2 \leq G$.
- Starting radius: $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$.

**Theorem (OGD Regret Bound)**

*After T steps,* $\left[\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)})\right] - \left[\sum_{i=1}^{T} f_i(\mathbf{x}^*)\right] \leq RG\sqrt{T}$. *I.e. the average regret* $\frac{1}{T}\left[\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)})\right]$ *is* $\leq \epsilon$ *after:*

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

## Conditions:

- Finite sum structure: $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$, with $f_1, \ldots, f_n$ all convex.
- Lipschitz functions: for all $\mathbf{x}$, $j$, $\|\nabla f_j(\mathbf{x})\|_2 \leq \frac{G'}{n}$.
- Starting radius: $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$.

**Theorem (SGD Regret Bound)**

*Stochastic Gradient Descent returns $\hat{\mathbf{x}}$ with $\mathbb{E}[f(\hat{\mathbf{x}})] \leq \min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) + \epsilon$ after*

$$T = \frac{R^2 G'^2}{\epsilon^2} \text{ iterations.}$$

*We always have that $G' > G$, but iterations are typically cheaper by a factor of $n$.*

Can our convergence bounds be tightened for certain functions? Can they guide us towards faster algorithms?

**Goals:**

- Improve $\epsilon$ dependence below $1/\epsilon^2$.
    - Ideally $1/\epsilon$ or $\log(1/\epsilon)$.
- Reduce or eliminate dependence on $G$ and $R$.
- Further take advantage of structure in the data (e.g. repetition in features in addition to data points).

**Definition ($\beta$-smoothness)**

A function $f$ is $\beta$ 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 $\eta$ depended on $G$. Now choose it based on $\beta$:

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

**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 $\beta$-smooth functions.)**

*Let $f$ be a $\beta$ 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:*

$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \frac{2\beta R^2}{T-1}$$

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

**Definition ($\alpha$-strongly convex)**

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

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

$\alpha$ 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 $\alpha$-strongly convex functions.)**

*Let $f$ be an $\alpha$-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 $\beta$-smooth and $\alpha$-strongly convex?

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

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

**Theorem (GD for $\beta$-smooth, $\alpha$-strongly convex.)**

*Let $f$ be a $\beta$-smooth and $\alpha$-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 $\kappa$ 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 $\beta$-smooth, $\alpha$-strongly convex.)**

*Let $f$ be a $\beta$-smooth and $\alpha$-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:

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

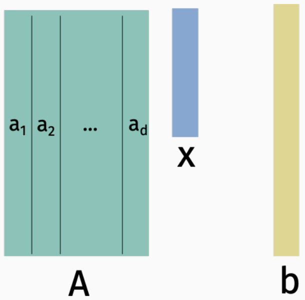
For vector $\mathbf{x}, \mathbf{y}$:

$$\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}) \approx \left[\nabla^2 f(\mathbf{x})\right](\mathbf{x} - \mathbf{y}).$$

Let $f$ be a twice differentiable function from $\mathbb{R}^d \rightarrow \mathbb{R}$. 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 $\beta$-smooth and $\alpha$-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:

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

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

Equivalently for any $\mathbf{z}$,

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

**Exercise:** Show that for $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$,

$$[f(\mathbf{x}) - f(\mathbf{y})] - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) = (\mathbf{x} - \mathbf{y})^T \left[2\mathbf{A}^T\mathbf{A}\right](\mathbf{x} - \mathbf{y}).$$

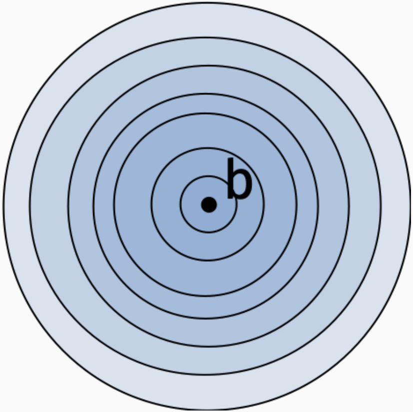
This would imply:

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

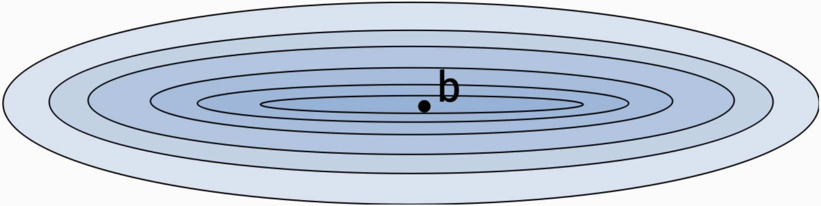
Let $f(\mathbf{x}) = \|\mathbf{D}\mathbf{x} - \mathbf{b}\|_2^2$ where $\mathbf{D}$ is a diagaonl matrix. For now imagine we're in two dimensions: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\mathbf{D} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}$.

What are $\alpha, \beta$ for this problem?

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

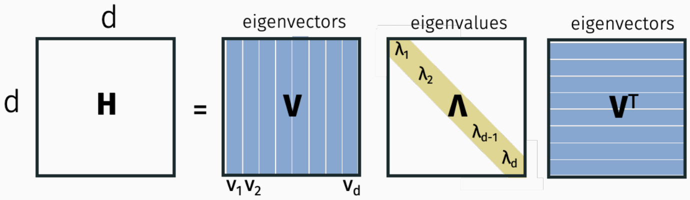



Level sets of $\|\mathbf{D}\mathbf{x} - \mathbf{b}\|_2^2$ when $d_1^2 = 1, d_2^2 = 1$.




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

Any symmetric matrix **H** has an orthogonal, real valued eigendecomposition.

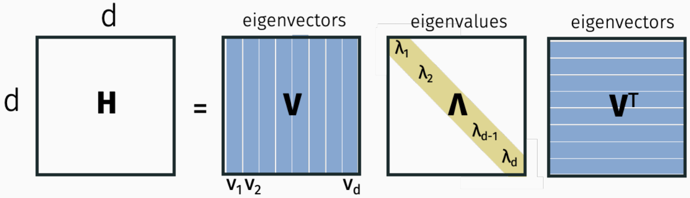



Here **V** is square and orthogonal, so $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$. And for each $\mathbf{v}_i$, we have:

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

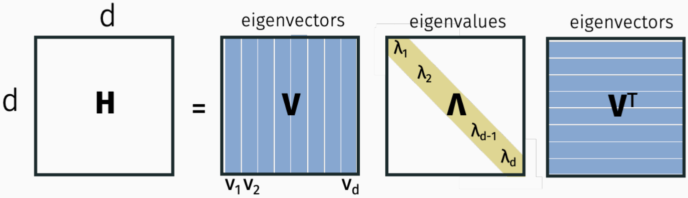
That's what makes $\mathbf{v}_1, \ldots, \mathbf{v}_d$ eigenvectors.

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



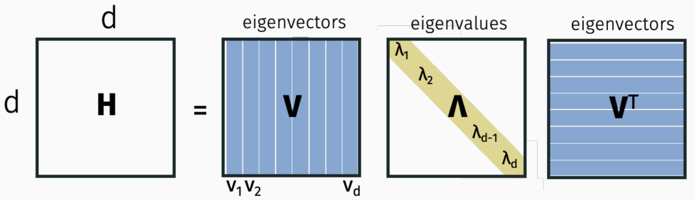

**Claim:** $\mathbf{H} \Leftrightarrow \lambda_1, ..., \lambda_d \geq 0$.

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 $f(\mathbf{x})$ is $\beta$-smooth and $\alpha$-strongly convex, then for any $\mathbf{x}$ we have the the maximum eigenvalue of $\mathbf{H} = \nabla^2 f(\mathbf{x}) = \beta$ and the minimum eigenvalue of $\mathbf{H} = \nabla^2 f(\mathbf{x}) = \alpha$.

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

**Theorem (GD for $\beta$-smooth, $\alpha$-strongly convex.)**

*Let $f$ be a $\beta$-smooth and $\alpha$-strongly convex function. If we run GD for $T$ steps (with step size $\eta = \frac{1}{2\beta}$) we have:*

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

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

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 really good understanding of gradient descent.

**Number of iterations for $\epsilon$ error:**

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

How do we use this understanding to design faster algorithms?

# ACCELERATION

Nesterov's accelerated gradient descent:

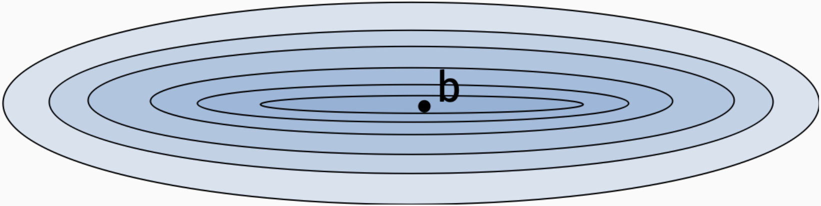
- $\mathbf{x}^{(1)} = \mathbf{y}^{(1)} = \mathbf{z}^{(1)}$
- For $t = 1, \ldots, T$
  - $\mathbf{y}^{(t+1)} = \mathbf{x}^{(t)} - \frac{1}{\beta}\nabla f(\mathbf{x}^{(t)})$
  - $\mathbf{x}^{(t+1)} = \left(1 + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)\mathbf{y}^{(t+1)} + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\left(\mathbf{y}^{(t+1)}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

**Theorem (AGD for $\beta$-smooth, $\alpha$-strongly convex.)**

*Let $f$ be a $\beta$-smooth and $\alpha$-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 $\epsilon$.




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?

# PRECONDITIONING

**Main idea:** Instead of minimizing $f(\mathbf{x})$, find another function $g(\mathbf{x})$ with the same minimum but which is better suited for first order optimization (e.g., has a smaller conditioner number).

**Claim:** Let $h(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be an invertible function. Let $g(\mathbf{x}) = f(h(\mathbf{x}))$. Then

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} g(\mathbf{x}) \quad \text{and} \quad \arg\min_{\mathbf{x}} f(\mathbf{x}) = h\left(\arg\min_{\mathbf{x}} g(\mathbf{x})\right).$$

**First Goal:** We need $g(\mathbf{x})$ to still be convex.

**Claim:** Let $\mathbf{P}$ be an invertible $d \times d$ matrix and let $g(\mathbf{x}) = f(\mathbf{P}\mathbf{x})$.

$g(\mathbf{x})$ is always convex.

**Second Goal:**

$g(\mathbf{x})$ should have better condition number $\kappa$ than $f(\mathbf{x})$.

**Example:**

- $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$. $\kappa_f = \frac{\lambda_1(\mathbf{A}^T\mathbf{A})}{\lambda_d(\mathbf{A}^T\mathbf{A})}$.
- $g(\mathbf{x}) = \|\mathbf{A}\mathbf{P}\mathbf{x} - \mathbf{b}\|_2^2$. $\kappa_g = \frac{\lambda_1(\mathbf{P}^T\mathbf{A}^T\mathbf{A}\mathbf{P})}{\lambda_d(\mathbf{P}^T\mathbf{A}^T\mathbf{A}\mathbf{P})}$.

**Ideal preconditioner:** Choose $P$ so that $\mathbf{P}^T\mathbf{A}^T\mathbf{A}\mathbf{P} = \mathbf{I}$. For example, could set $P = \sqrt{(\mathbf{A}^T\mathbf{A})^{-1}}$.

What's the problem with this choice?

**Third Goal:** $\mathbf{P}$ should be easy to compute.

Many, many problem specific preconditioners are used in practice. There design is usually a heuristic process.

**Example:** Diagonal preconditioner.

- Let $\mathbf{D} = \text{diag}(\mathbf{A}^T\mathbf{A})$
- Intuitively, we roughly have that $\mathbf{D} \approx \mathbf{A}^T\mathbf{A}$.
- Let $\mathbf{P} = \sqrt{\mathbf{D}^{-1}}$

$\mathbf{P}$ is often called a **Jacobi preconditioner**. Often works very well in practice!

# DIAGONAL PRECONDITIONER

```
A =

    -734     1     33     9111     0
     -31    -2    108     5946   -19
     232    -1    101     3502    10
     426     0    -65    12503     9
    -373     0     26     9298     0
    -236    -2    -94     2398    -1
    2024     0   -132    -6904   -25
   -2258    -1     92    -6516     6
    2229     0      0    11921   -22
     338     1     -5   -16118   -23
```

```
>> cond(A'*A)

ans =

   8.4145e+07
```

```
>> P = sqrt(inv(diag(diag(A'*A))));
>> cond(P*A'*A*P)

ans =

   10.3878
```

**Another view**: If $g(\mathbf{x}) = f(\mathbf{P}\mathbf{x})$ then $\nabla g(\mathbf{x}) = \mathbf{P}^T \nabla f(\mathbf{P}\mathbf{x})$.

$\nabla g(\mathbf{x}) = \mathbf{P}\nabla f(\mathbf{P}\mathbf{x})$ when $\mathbf{P}$ is symmetric.

**Gradient descent on $g$:**

- For $t = 1, \ldots, T$,
  - $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \mathbf{P}\left[\nabla f(\mathbf{P}\mathbf{x}^{(t)})\right]$

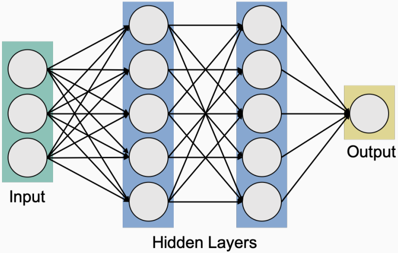
**Gradient descent on $g$:**

- For $t = 1, \ldots, T$,
  - $\mathbf{y}^{(t+1)} = \mathbf{y}^{(t)} - \eta \mathbf{P}^2\left[\nabla f(\mathbf{y}^{(t)})\right]$

When **P** is diagonal, this is just gradient descent with a different step size for each parameter!

Algorithms based on this idea:

- AdaGrad
- RMSprop
- Adam optimizer




(Pretty much all of the most widely used optimization methods for training neural networks.)

# COORDINATE DESCENT

**Main idea:** Trade slower convergence (more iterations) for cheaper iterations.

**Stochastic Gradient Descent:** When $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$, approximate $\nabla f(\mathbf{x})$ with $\nabla f_i(\mathbf{x})$ for randomly chosen $i$.

**Main idea:** Trade slower convergence (more iterations) for cheaper iterations.

**Stochastic Coordinate Descent:** Only compute a single random entry of $\nabla f(\mathbf{x})$ on each iteration:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \frac{\partial f}{\partial x_2}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial x_d}(\mathbf{x}) \end{bmatrix} \qquad \nabla_i f(\mathbf{x}) = \begin{bmatrix} 0 \\ \frac{\partial f}{\partial x_i}(\mathbf{x}) \\ \vdots \\ 0 \end{bmatrix}$$

**Update:** $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} + \eta \nabla_i f(\mathbf{x}^{(t)})$.

When $\mathbf{x}$ has $d$ parameters, computing $\nabla_i f(\mathbf{x})$ often costs just a $1/d$ fraction of what it costs to compute $\nabla f(\mathbf{x})$

**Example:** $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ for $\mathbf{A} \in \mathbb{R}^{n \times d}, \mathbf{x} \in \mathbb{R}^d, \mathbf{b} \in \mathbb{R}^n$.

- $\nabla f(\mathbf{x}) = 2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b}$.
- $\nabla_i f(\mathbf{x}) = 2\left[\mathbf{A}^T\mathbf{A}\mathbf{x}\right]_i - 2\left[\mathbf{A}^T\mathbf{b}\right]$.

Stochastic Coordinate Descent:

- Choose number of steps $T$ and step size $\eta$.
- For $i = 1, \ldots, T$:
    - Pick random $j_i \in 1, \ldots, d$.
    - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla_{j_i} f(\mathbf{x}^{(i)})$
- Return $\hat{\mathbf{x}} = \frac{1}{T} \sum_{i=1}^{T} \mathbf{x}^{(i)}$.

**Theorem (Stochastic Coordinate Descent convergence)**

*Given a G-Lipschitz function f with minimizer* $\mathbf{x}^*$ *and initial point* $\mathbf{x}^{(1)}$ *with* $\|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2 \leq R$*, SCD with step size* $\eta = \frac{1}{Rd}$ *satisfies the guarantee:*

$$\mathbb{E}[f(\hat{\mathbf{x}}) - f(\mathbf{x}^*)] \leq \frac{2GR}{\sqrt{T/d}}$$

Often it doesn't make sense to sample $i$ uniformly at random:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -.5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 10 \\ 42 \\ -11 \\ -51 \\ 34 \\ -22 \end{bmatrix}$$

Select indices $i$ proportional to $\|\mathbf{a}_i\|_2^2$:

$$\Pr[\text{select index } i \text{ to update}] = \frac{\|\mathbf{a}_i\|_2^2}{\sum_{j=1}^{d} \|\mathbf{a}_j\|_2^2} = \frac{\|\mathbf{a}_i\|_2^2}{\|\mathbf{A}\|_2^2}$$