Introduction to Machine Learning (67577)
Lecture 6

Shai Shalev-Shwartz

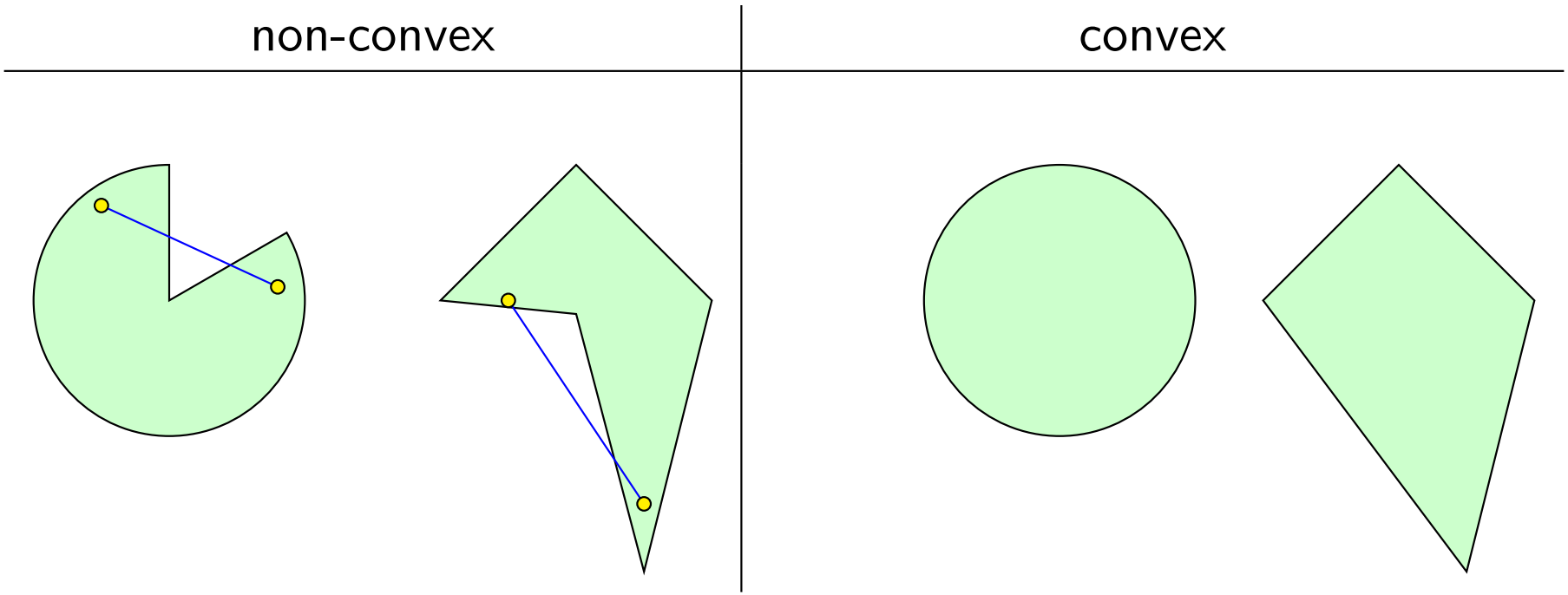
School of CS and Engineering,
The Hebrew University of Jerusalem

Convexity,Optimization,Surrogates,SGD

Outline

Definition (Convex Set)

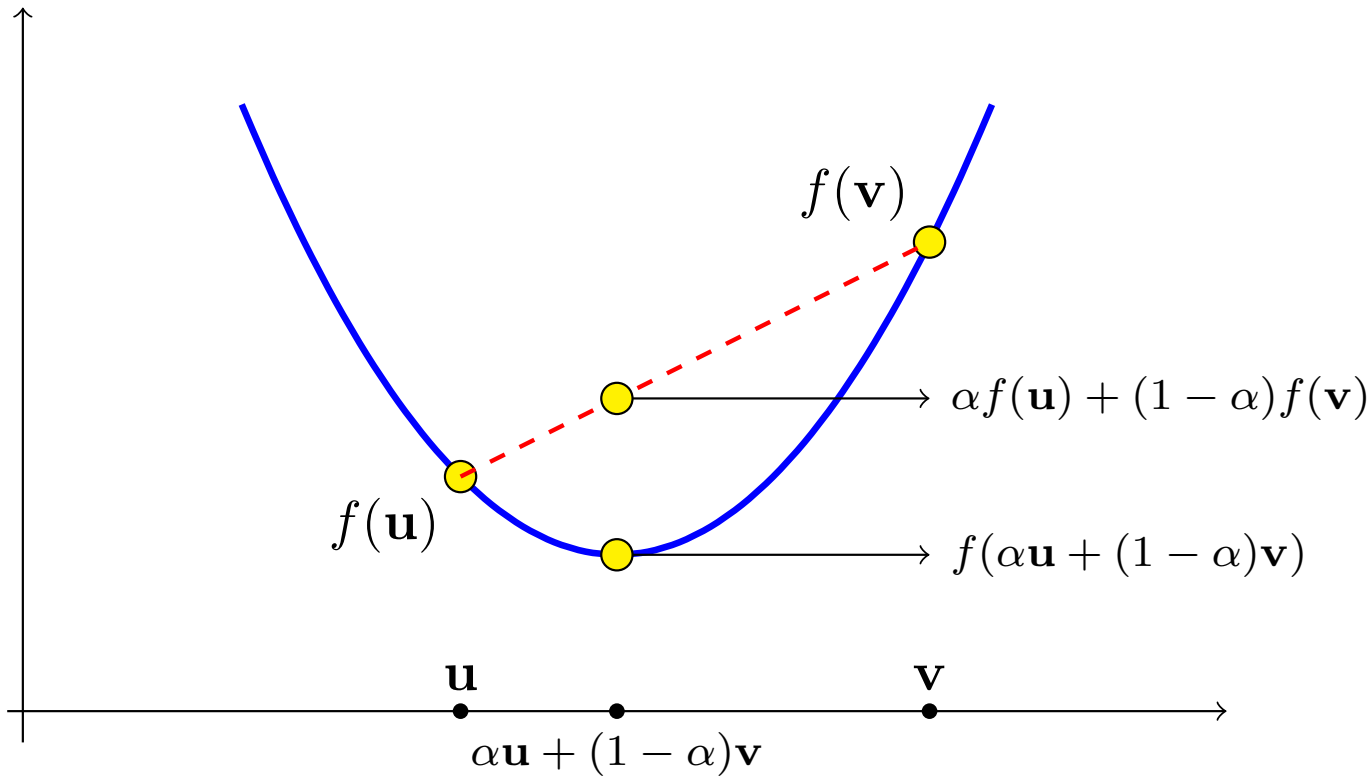
A set C in a vector space is convex if for any two vectors $\mathbf{u}, \mathbf{v}$ in C, the line segment between $\mathbf{u}$ and $\mathbf{v}$ is contained in C. That is, for any $\alpha \in [0, 1]$ we have that the convex combination $\alpha\mathbf{u} + (1 - \alpha)\mathbf{v}$ is in C.



Definition (Convex function)

Let C be a convex set. A function $f : C \to \mathbb{R}$ is convex if for every $\mathbf{u}, \mathbf{v} \in C$ and $\alpha \in [0, 1]$,

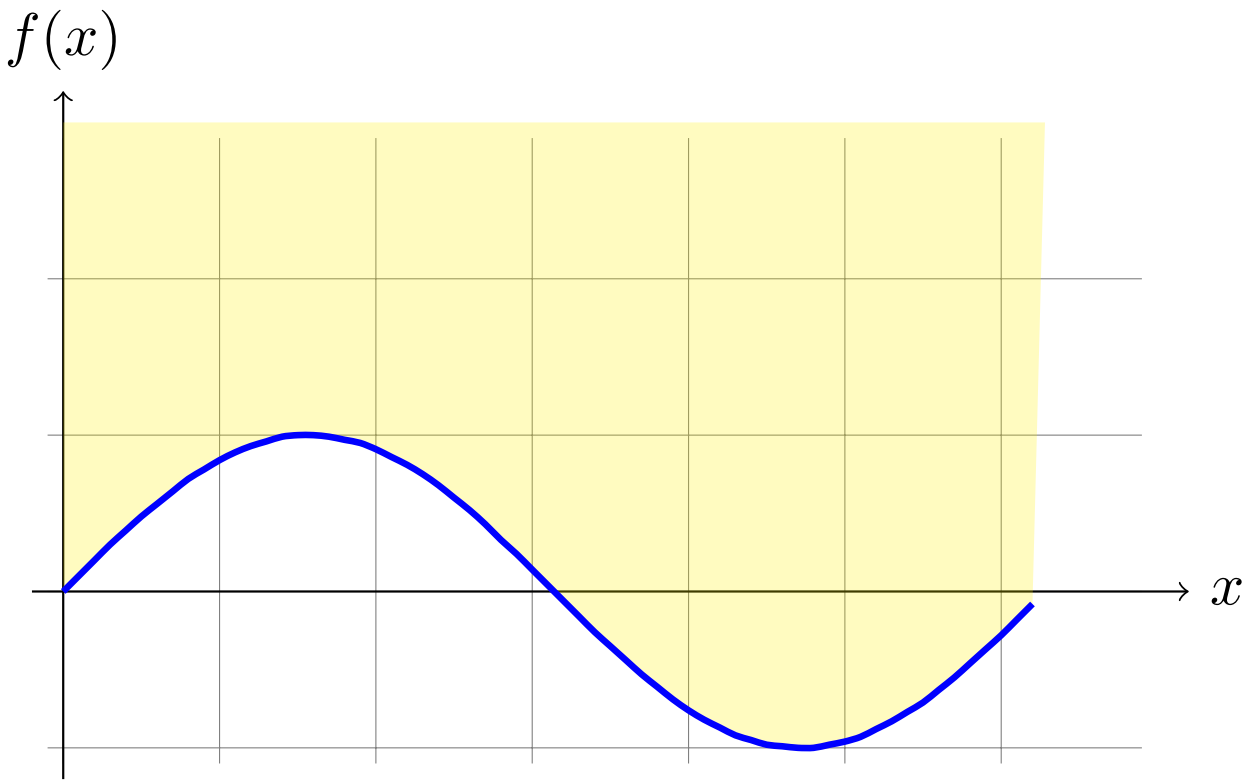
$$f(\alpha \mathbf{u} + (1 - \alpha)\mathbf{v}) \leq \alpha f(\mathbf{u}) + (1 - \alpha) f(\mathbf{v}) \ .$$



Epigraph

A function f is convex if and only if its *epigraph* is a convex set:

$$\text{epigraph(f)} = \{(\mathbf{x}, \beta) : f(\mathbf{x}) \leq \beta\} \ .$$



Property I: local minima are global

If f is convex then every local minimum of f is also a global minimum.

Property I: local minima are global

If f is convex then every local minimum of f is also a global minimum.

- let $B(\mathbf{u}, r) = \{\mathbf{v} : \|\mathbf{v} - \mathbf{u}\| \leq r\}$

Property I: local minima are global

If f is convex then every local minimum of f is also a global minimum.

- let $B(\mathbf{u}, r) = \{\mathbf{v} : \|\mathbf{v} - \mathbf{u}\| \leq r\}$
- $f(\mathbf{u})$ is a local minimum of f at $\mathbf{u}$ if $\exists r > 0$ s.t. $\forall \mathbf{v} \in B(\mathbf{u}, r)$ we have $f(\mathbf{v}) \geq f(\mathbf{u})$

Property I: local minima are global

If f is convex then every local minimum of f is also a global minimum.

- let $B(\mathbf{u}, r) = \{\mathbf{v} : \|\mathbf{v} - \mathbf{u}\| \leq r\}$
- $f(\mathbf{u})$ is a local minimum of f at $\mathbf{u}$ if $\exists r > 0$ s.t. $\forall \mathbf{v} \in B(\mathbf{u}, r)$ we have $f(\mathbf{v}) \geq f(\mathbf{u})$
- It follows that for any $\mathbf{v}$ (not necessarily in B), there is a small enough $\alpha > 0$ such that $\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u}) \in B(\mathbf{u}, r)$ and therefore

$$f(\mathbf{u}) \leq f(\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u})) \ .$$

Property I: local minima are global

If f is convex then every local minimum of f is also a global minimum.

- let $B(\mathbf{u}, r) = \{\mathbf{v} : \|\mathbf{v} - \mathbf{u}\| \leq r\}$
- $f(\mathbf{u})$ is a local minimum of f at $\mathbf{u}$ if $\exists r > 0$ s.t. $\forall \mathbf{v} \in B(\mathbf{u}, r)$ we have $f(\mathbf{v}) \geq f(\mathbf{u})$
- It follows that for any $\mathbf{v}$ (not necessarily in B), there is a small enough $\alpha > 0$ such that $\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u}) \in B(\mathbf{u}, r)$ and therefore

$$f(\mathbf{u}) \leq f(\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u})) \ .$$

- If f is convex, we also have that

$$f(\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u})) = f(\alpha\mathbf{v} + (1 - \alpha)\mathbf{u}) \leq (1 - \alpha)f(\mathbf{u}) + \alpha f(\mathbf{v}) \ .$$

Property I: local minima are global

If f is convex then every local minimum of f is also a global minimum.

- let $B(\mathbf{u}, r) = \{\mathbf{v} : \|\mathbf{v} - \mathbf{u}\| \leq r\}$
- $f(\mathbf{u})$ is a local minimum of f at $\mathbf{u}$ if $\exists r > 0$ s.t. $\forall \mathbf{v} \in B(\mathbf{u}, r)$ we have $f(\mathbf{v}) \geq f(\mathbf{u})$
- It follows that for any $\mathbf{v}$ (not necessarily in B), there is a small enough $\alpha > 0$ such that $\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u}) \in B(\mathbf{u}, r)$ and therefore

$$f(\mathbf{u}) \leq f(\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u})) \ .$$

- If f is convex, we also have that

$$f(\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u})) = f(\alpha\mathbf{v} + (1 - \alpha)\mathbf{u}) \leq (1 - \alpha)f(\mathbf{u}) + \alpha f(\mathbf{v}) \ .$$

- Combining, we obtain that $f(\mathbf{u}) \leq f(\mathbf{v})$.

Property I: local minima are global

If f is convex then every local minimum of f is also a global minimum.

- let $B(\mathbf{u}, r) = \{\mathbf{v} : \|\mathbf{v} - \mathbf{u}\| \leq r\}$
- $f(\mathbf{u})$ is a local minimum of f at $\mathbf{u}$ if $\exists r > 0$ s.t. $\forall \mathbf{v} \in B(\mathbf{u}, r)$ we have $f(\mathbf{v}) \geq f(\mathbf{u})$
- It follows that for any $\mathbf{v}$ (not necessarily in B), there is a small enough $\alpha > 0$ such that $\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u}) \in B(\mathbf{u}, r)$ and therefore

$$f(\mathbf{u}) \leq f(\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u})) \ .$$

- If f is convex, we also have that

$$f(\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u})) = f(\alpha\mathbf{v} + (1 - \alpha)\mathbf{u}) \leq (1 - \alpha)f(\mathbf{u}) + \alpha f(\mathbf{v}) \ .$$

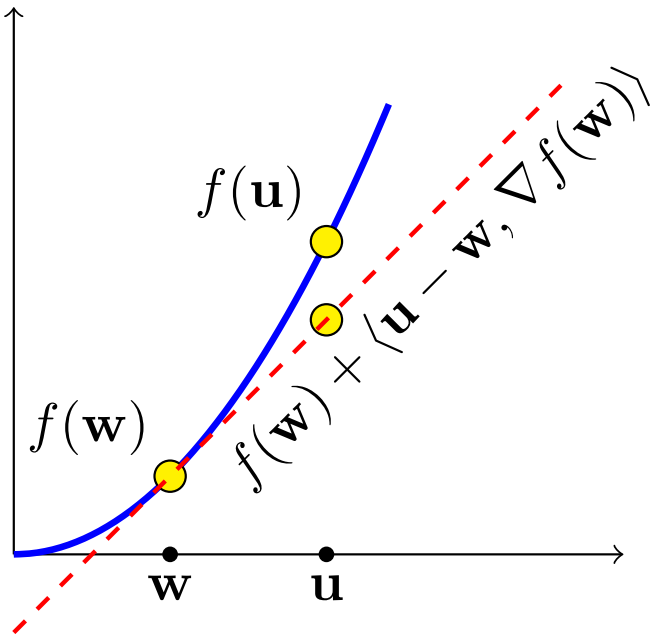
- Combining, we obtain that $f(\mathbf{u}) \leq f(\mathbf{v})$.
- This holds for every $\mathbf{v}$, hence $f(\mathbf{u})$ is also a global minimum of f.

Property II: tangents lie below f

If f is convex and differentiable, then

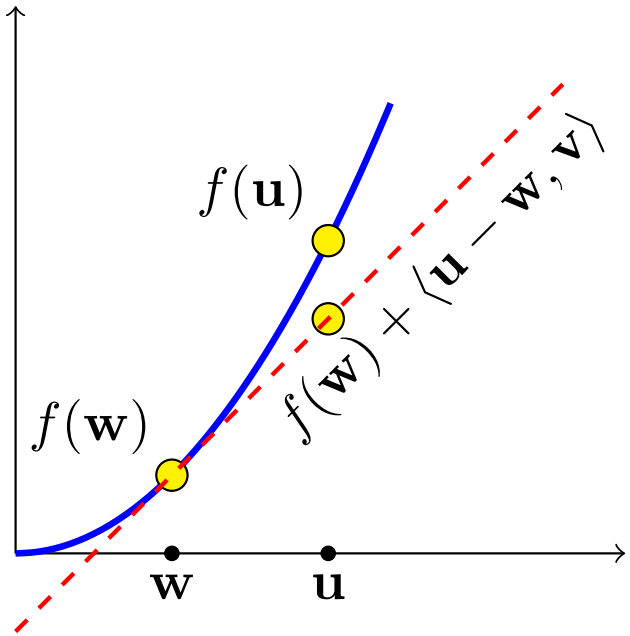
$$\forall \mathbf{u}, \quad f(\mathbf{u}) \geq f(\mathbf{w}) + \langle \nabla f(\mathbf{w}), \mathbf{u} - \mathbf{w} \rangle$$

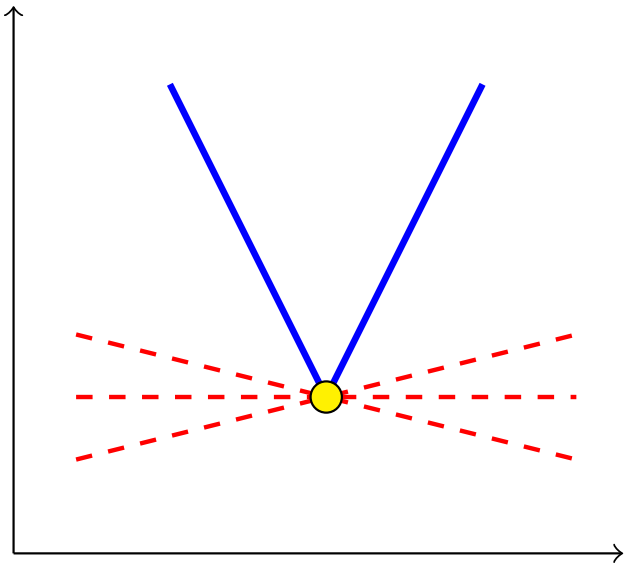
(recall, $\nabla f(\mathbf{w}) = \left(\frac{\partial f(\mathbf{w})}{\partial w_1}, \ldots, \frac{\partial f(\mathbf{w})}{\partial w_d} \right)$ is the gradient of f at $\mathbf{w}$)



Sub-gradients

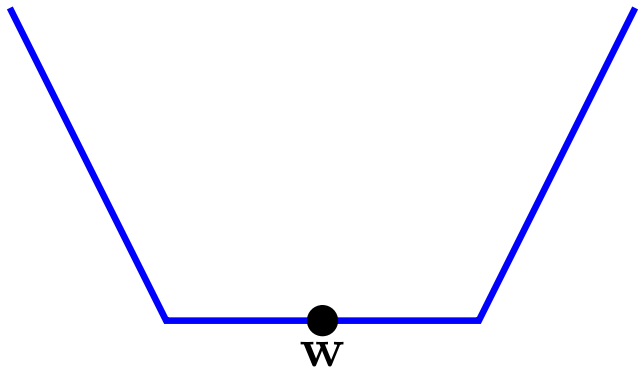
- $\mathbf{v}$ is sub-gradient of f at $\mathbf{w}$ if $\forall \mathbf{u}, \quad f(\mathbf{u}) \geq f(\mathbf{w}) + \langle \mathbf{v}, \mathbf{u} - \mathbf{w} \rangle$
- The differential set, $\partial f(\mathbf{w})$, is the set of sub-gradients of f at $\mathbf{w}$
- Lemma: f is convex iff for every $\mathbf{w}$, $\partial f(\mathbf{w}) \neq \emptyset$





Property II: tangents lie below f

f is "locally flat" around $\mathbf{w}$ (i.e. $\mathbf{0}$ is a sub-gradient) iff $\mathbf{w}$ is a global minimizer



Lipschitzness

Definition (Lipschitzness)

A function $f : C \to \mathbb{R}$ is ρ-Lipschitz if for every $\mathbf{w}_1, \mathbf{w}_2 \in C$ we have that $|f(\mathbf{w}_1) - f(\mathbf{w}_2)| \leq \rho \|\mathbf{w}_1 - \mathbf{w}_2\|$.

Lipschitzness

Definition (Lipschitzness)

A function $f : C \to \mathbb{R}$ is ρ-Lipschitz if for every $\mathbf{w}_1, \mathbf{w}_2 \in C$ we have that $|f(\mathbf{w}_1) - f(\mathbf{w}_2)| \leq \rho \|\mathbf{w}_1 - \mathbf{w}_2\|$.

Lemma

If f is convex then f is ρ-Lipschitz iff the norm of all sub-gradients of f is at most ρ

Outline

Convex optimization

Approximately solve:

$$\operatorname*{argmin}_{\mathbf{w} \in C} f(\mathbf{w})$$

where C is a convex set and f is a convex function.

Convex optimization

Approximately solve:

$$\operatorname*{argmin}_{\mathbf{w} \in C} f(\mathbf{w})$$

where C is a convex set and f is a convex function.

Special cases:

- Feasibility problem: f is a constant function

Convex optimization

Approximately solve:

$$\operatorname*{argmin}_{\mathbf{w} \in C} f(\mathbf{w})$$

where C is a convex set and f is a convex function.

Special cases:

- Feasibility problem: f is a constant function
- Unconstrained minimization: $C = \mathbb{R}^d$

Convex optimization

Approximately solve:

$$\operatorname*{argmin}_{\mathbf{w} \in C} f(\mathbf{w})$$

where C is a convex set and f is a convex function.

Special cases:

- Feasibility problem: f is a constant function
- Unconstrained minimization: $C = \mathbb{R}^d$
- Can reduce one to another:

Convex optimization

Approximately solve:

$$\operatorname*{argmin}_{\mathbf{w} \in C} f(\mathbf{w})$$

where C is a convex set and f is a convex function.

Special cases:

- Feasibility problem: f is a constant function
- Unconstrained minimization: $C = \mathbb{R}^d$
- Can reduce one to another:
  - Adding the function $I_C(\mathbf{w})$ to the objective eliminates the constraint

Convex optimization

Approximately solve:

$$\operatorname*{argmin}_{\mathbf{w} \in C} f(\mathbf{w})$$

where C is a convex set and f is a convex function.

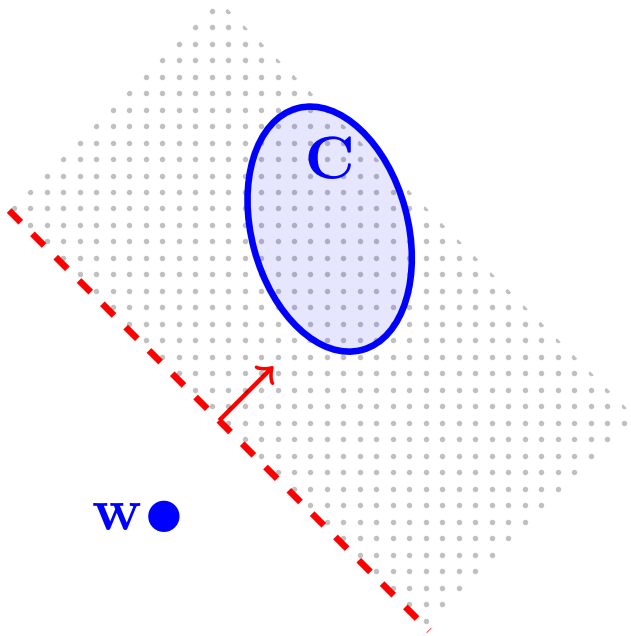
Special cases:

- Feasibility problem: f is a constant function
- Unconstrained minimization: $C = \mathbb{R}^d$
- Can reduce one to another:
  - Adding the function $I_C(\mathbf{w})$ to the objective eliminates the constraint
  - Adding the constraint $f(\mathbf{w}) \leq f^* + \epsilon$ eliminates the objective

Outline

The Ellipsoid Algorithm

- Consider a feasibility problem: find $\mathbf{w} \in C$
- Assumptions:
  - $B(\mathbf{w}^*, r) \subseteq C \subset B(0, R)$
  - Separation oracle: Given $\mathbf{w}$, the oracle tells if it's in C or not. If $\mathbf{w} \notin C$ then the oracle finds $\mathbf{v}$ s.t. for every $\mathbf{w}' \in C$ we have $\langle \mathbf{w}, \mathbf{v} \rangle < \langle \mathbf{w}', \mathbf{v} \rangle$



The Ellipsoid Algorithm

- We implicitly maintain an ellipsoid: $\mathcal{E}_t = \mathcal{E}(A_t^{1/2}, \mathbf{w}_t)$
- Start with $\mathbf{w}_1 = \mathbf{0}$, $A_1 = I$
- For $t = 1, 2, \ldots$
  - Call oracle with $\mathbf{w}_t$
  - If $\mathbf{w}_t \in C$, **break and return** $\mathbf{w}_t$
  - Otherwise, let $\mathbf{v}_t$ be the vector defining a separating hyperplane
  - Update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \frac{1}{d+1} \frac{A_t \mathbf{v}_t}{\sqrt{\mathbf{v}_t^\top A_t \mathbf{v}_t}}$$

$$A_{t+1} = \frac{d^2}{d^2-1} \left(A_t - \frac{2}{d+1} \frac{A_t \mathbf{v}_t \mathbf{v}_t^\top A_t}{\mathbf{v}_t^\top A_t \mathbf{v}_t} \right)$$

The Ellipsoid Algorithm

- We implicitly maintain an ellipsoid: $\mathcal{E}_t = \mathcal{E}(A_t^{1/2}, \mathbf{w}_t)$
- Start with $\mathbf{w}_1 = \mathbf{0}$, $A_1 = I$
- For $t = 1, 2, \ldots$
  - Call oracle with $\mathbf{w}_t$
  - If $\mathbf{w}_t \in C$, **break and return** $\mathbf{w}_t$
  - Otherwise, let $\mathbf{v}_t$ be the vector defining a separating hyperplane
  - Update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \frac{1}{d+1} \frac{A_t \mathbf{v}_t}{\sqrt{\mathbf{v}_t^\top A_t \mathbf{v}_t}}$$

$$A_{t+1} = \frac{d^2}{d^2-1} \left(A_t - \frac{2}{d+1} \frac{A_t \mathbf{v}_t \mathbf{v}_t^\top A_t}{\mathbf{v}_t^\top A_t \mathbf{v}_t} \right)$$

Theorem

The Ellipsoid converges after at most $2d(2d+2)\log(R/r)$ *iterations.*

Implementing the separation oracle using sub-gradients

- Suppose $C = \cap_{i=1}^{n} \{\mathbf{w} : f_i(\mathbf{w}) \leq 0\}$ where each f_i is a convex function.
- Given $\mathbf{w}$, we can check if $f_i(\mathbf{w}) \leq 0$ for every i
- If $f_i(\mathbf{w}) > 0$ for some i, consider $\mathbf{v} \in \partial f_i(\mathbf{w})$, then, for every $\mathbf{w}' \in C$

$$0 \geq f_i(\mathbf{w}') \geq f_i(\mathbf{w}) + \langle \mathbf{w}' - \mathbf{w}, \mathbf{v} \rangle > \langle \mathbf{w}' - \mathbf{w}, \mathbf{v} \rangle$$

- So, the oracle can return $-\mathbf{v}$

The Ellipsoid Algorithm for unconstrained minimization

- Consider $\min_{\mathbf{w}} f(\mathbf{w})$ and let $\mathbf{w}^*$ be a minimizer

The Ellipsoid Algorithm for unconstrained minimization

- Consider $\min_{\mathbf{w}} f(\mathbf{w})$ and let $\mathbf{w}^*$ be a minimizer
- Let $C = \{\mathbf{w} : f(\mathbf{w}) - f(\mathbf{w}^*) - \epsilon \leq 0\}$

The Ellipsoid Algorithm for unconstrained minimization

- Consider $\min_{\mathbf{w}} f(\mathbf{w})$ and let $\mathbf{w}^*$ be a minimizer
- Let $C = \{\mathbf{w} : f(\mathbf{w}) - f(\mathbf{w}^*) - \epsilon \leq 0\}$
- We can apply the Ellipsoid algorithm while letting $\mathbf{v}_t \in \partial f(\mathbf{w}_t)$

The Ellipsoid Algorithm for unconstrained minimization

- Consider $\min_{\mathbf{w}} f(\mathbf{w})$ and let $\mathbf{w}^*$ be a minimizer
- Let $C = \{\mathbf{w} : f(\mathbf{w}) - f(\mathbf{w}^*) - \epsilon \leq 0\}$
- We can apply the Ellipsoid algorithm while letting $\mathbf{v}_t \in \partial f(\mathbf{w}_t)$
- Analysis:

The Ellipsoid Algorithm for unconstrained minimization

- Consider $\min_{\mathbf{w}} f(\mathbf{w})$ and let $\mathbf{w}^*$ be a minimizer
- Let $C = \{\mathbf{w} : f(\mathbf{w}) - f(\mathbf{w}^*) - \epsilon \leq 0\}$
- We can apply the Ellipsoid algorithm while letting $\mathbf{v}_t \in \partial f(\mathbf{w}_t)$
- Analysis:
- Let r be s.t. $B(\mathbf{w}^*, r) \subseteq C$

The Ellipsoid Algorithm for unconstrained minimization

- Consider $\min_{\mathbf{w}} f(\mathbf{w})$ and let $\mathbf{w}^*$ be a minimizer
- Let $C = \{\mathbf{w} : f(\mathbf{w}) - f(\mathbf{w}^*) - \epsilon \leq 0\}$
- We can apply the Ellipsoid algorithm while letting $\mathbf{v}_t \in \partial f(\mathbf{w}_t)$
- Analysis:
- Let r be s.t. $B(\mathbf{w}^*, r) \subseteq C$
- For example, if f is ρ-Lipschitz we can take $r = \epsilon/\rho$

The Ellipsoid Algorithm for unconstrained minimization

- Consider $\min_{\mathbf{w}} f(\mathbf{w})$ and let $\mathbf{w}^*$ be a minimizer
- Let $C = \{\mathbf{w} : f(\mathbf{w}) - f(\mathbf{w}^*) - \epsilon \leq 0\}$
- We can apply the Ellipsoid algorithm while letting $\mathbf{v}_t \in \partial f(\mathbf{w}_t)$
- Analysis:
- Let r be s.t. $B(\mathbf{w}^*, r) \subseteq C$
- For example, if f is ρ-Lipschitz we can take $r = \epsilon/\rho$
- Let $R = \|\mathbf{w}^*\| + r$

The Ellipsoid Algorithm for unconstrained minimization

- Consider $\min_{\mathbf{w}} f(\mathbf{w})$ and let $\mathbf{w}^*$ be a minimizer
- Let $C = \{\mathbf{w} : f(\mathbf{w}) - f(\mathbf{w}^*) - \epsilon \leq 0\}$
- We can apply the Ellipsoid algorithm while letting $\mathbf{v}_t \in \partial f(\mathbf{w}_t)$
- Analysis:
- Let r be s.t. $B(\mathbf{w}^*, r) \subseteq C$
- For example, if f is ρ-Lipschitz we can take $r = \epsilon/\rho$
- Let $R = \|\mathbf{w}^*\| + r$
- Then, after $2d(2d+2)\log(R/r)$ iterations, $\mathbf{w}_t$ must be in C

The Ellipsoid Algorithm for unconstrained minimization

- Consider $\min_{\mathbf{w}} f(\mathbf{w})$ and let $\mathbf{w}^*$ be a minimizer
- Let $C = \{\mathbf{w} : f(\mathbf{w}) - f(\mathbf{w}^*) - \epsilon \leq 0\}$
- We can apply the Ellipsoid algorithm while letting $\mathbf{v}_t \in \partial f(\mathbf{w}_t)$
- Analysis:
- Let r be s.t. $B(\mathbf{w}^*, r) \subseteq C$
- For example, if f is ρ-Lipschitz we can take $r = \epsilon/\rho$
- Let $R = \|\mathbf{w}^*\| + r$
- Then, after $2d(2d+2)\log(R/r)$ iterations, $\mathbf{w}_t$ must be in C
- For f being ρ-Lipschitz, we obtain the iteration bound

$$2d(2d+2)\log\left(\frac{\rho\,\|\mathbf{w}^*\|}{\epsilon} + 1\right)$$

Outline

Gradient Descent

- Start with initial $\mathbf{w}^{(1)}$ (usually, the zero vector)

Gradient Descent

- Start with initial $\mathbf{w}^{(1)}$ (usually, the zero vector)
- At iteration t, update

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla f(\mathbf{w}^{(t)}) \ ,$$

where $\eta > 0$ is a parameter

Gradient Descent

- Start with initial $\mathbf{w}^{(1)}$ (usually, the zero vector)
- At iteration t, update

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla f(\mathbf{w}^{(t)}) \ ,$$

  where $\eta > 0$ is a parameter
- Intuition:

Gradient Descent

- Start with initial $\mathbf{w}^{(1)}$ (usually, the zero vector)
- At iteration t, update

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla f(\mathbf{w}^{(t)}) \ ,$$

  where $\eta > 0$ is a parameter
- Intuition:
  - By Taylor's approximation, if $\mathbf{w}$ close to $\mathbf{w}^{(t)}$ then $f(\mathbf{w}) \approx f(\mathbf{w}^{(t)}) + \langle \mathbf{w} - \mathbf{w}^{(t)}, \nabla f(\mathbf{w}^{(t)}) \rangle$

Gradient Descent

- Start with initial $\mathbf{w}^{(1)}$ (usually, the zero vector)
- At iteration t, update

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla f(\mathbf{w}^{(t)}) \ ,$$

  where $\eta > 0$ is a parameter
- Intuition:
  - By Taylor's approximation, if $\mathbf{w}$ close to $\mathbf{w}^{(t)}$ then $f(\mathbf{w}) \approx f(\mathbf{w}^{(t)}) + \langle \mathbf{w} - \mathbf{w}^{(t)}, \nabla f(\mathbf{w}^{(t)}) \rangle$
  - Hence, we want to minimize the approximation while staying close to $\mathbf{w}^{(t)}$:

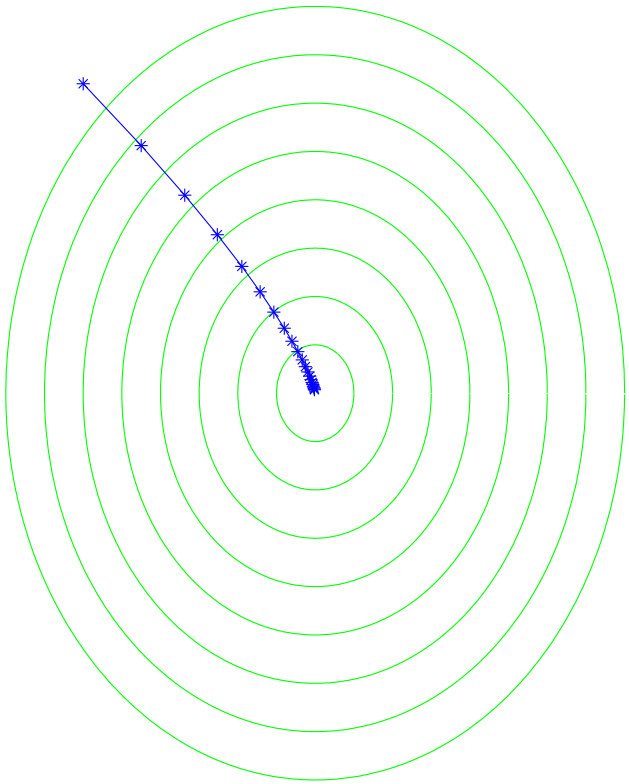
$$\mathbf{w}^{(t+1)} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}^{(t)}\|^2 + \eta \left(f(\mathbf{w}^{(t)}) + \langle \mathbf{w} - \mathbf{w}^{(t)}, \nabla f(\mathbf{w}^{(t)}) \rangle \right) \ .$$

Gradient Descent

- Initialize $\mathbf{w}^{(1)} = \mathbf{0}$
- Update

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla f(\mathbf{w}^{(t)})$$

- Output $\bar{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}^{(t)}$



Sub-Gradient Descent

Replace gradients with sub-gradients:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t \ ,$$

where $\mathbf{v}_t \in \partial f(\mathbf{w}^{(t)})$

Analyzing sub-gradient descent

Lemma

$$\sum_{t=1}^{T}(f(\mathbf{w}^{(t)}) - f(\mathbf{w}^\star)) \leq \sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle$$

$$= \frac{\|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 \ .$$

Proof:

- The inequality is by the definition of sub-gradients
- The equality follows from the definition of the update using algebraic manipulations

Analyzing sub-gradient descent for Lipschitz functions

- Since f is convex and ρ-Lipschitz, $\|\mathbf{v}_t\| \leq \rho$ for every t

Analyzing sub-gradient descent for Lipschitz functions

- Since f is convex and ρ-Lipschitz, $\|\mathbf{v}_t\| \leq \rho$ for every t
- Therefore,

$$\frac{1}{T}\sum_{t=1}^{T}(f(\mathbf{w}_t) - f(\mathbf{w}^\star)) \;\leq\; \frac{\|\mathbf{w}^\star\|^2}{2\eta T} + \frac{\eta\rho^2}{2}$$

Analyzing sub-gradient descent for Lipschitz functions

- Since f is convex and ρ-Lipschitz, $\|\mathbf{v}_t\| \leq \rho$ for every t
- Therefore,

$$\frac{1}{T}\sum_{t=1}^{T}(f(\mathbf{w}_t) - f(\mathbf{w}^\star)) \;\leq\; \frac{\|\mathbf{w}^\star\|^2}{2\eta T} + \frac{\eta\rho^2}{2}$$

- For every $\mathbf{w}^\star$, if $T \geq \frac{\|\mathbf{w}^*\|^2\rho^2}{\epsilon^2}$, and $\eta = \sqrt{\frac{\|\mathbf{w}^*\|^2}{\rho^2\, T}}$, then the right-hand side is at most ϵ

Analyzing sub-gradient descent for Lipschitz functions

- Since f is convex and ρ-Lipschitz, $\|\mathbf{v}_t\| \leq \rho$ for every t
- Therefore,

$$\frac{1}{T}\sum_{t=1}^{T}(f(\mathbf{w}_t) - f(\mathbf{w}^\star)) \;\leq\; \frac{\|\mathbf{w}^\star\|^2}{2\eta T} + \frac{\eta\rho^2}{2}$$

- For every $\mathbf{w}^\star$, if $T \geq \frac{\|\mathbf{w}^*\|^2\rho^2}{\epsilon^2}$, and $\eta = \sqrt{\frac{\|\mathbf{w}^*\|^2}{\rho^2\, T}}$, then the right-hand side is at most ϵ
- By convexity, $f(\bar{\mathbf{w}}) \leq \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{w}_t)$, hence $f(\bar{\mathbf{w}}) - f(\mathbf{w}^\star) \leq \epsilon$

Analyzing sub-gradient descent for Lipschitz functions

- Since f is convex and ρ-Lipschitz, $\|\mathbf{v}_t\| \leq \rho$ for every t
- Therefore,

$$\frac{1}{T}\sum_{t=1}^{T}(f(\mathbf{w}_t) - f(\mathbf{w}^\star)) \;\leq\; \frac{\|\mathbf{w}^\star\|^2}{2\eta T} + \frac{\eta\rho^2}{2}$$

- For every $\mathbf{w}^\star$, if $T \geq \frac{\|\mathbf{w}^*\|^2\rho^2}{\epsilon^2}$, and $\eta = \sqrt{\frac{\|\mathbf{w}^*\|^2}{\rho^2\, T}}$, then the right-hand side is at most ϵ
- By convexity, $f(\bar{\mathbf{w}}) \leq \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{w}_t)$, hence $f(\bar{\mathbf{w}}) - f(\mathbf{w}^\star) \leq \epsilon$
- Corollary: Sub-gradient descent needs $\frac{\|\mathbf{w}^*\|^2\rho^2}{\epsilon^2}$ iterations to converge

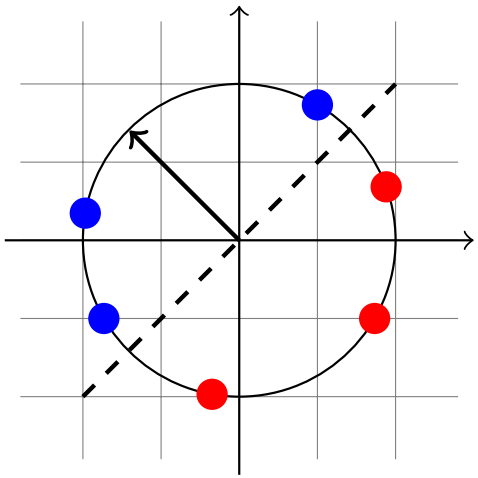
Example: Finding a Separating Hyperplane

Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ we would like to find a separating $\mathbf{w}$:

$$\forall i, \quad y_i \langle \mathbf{w}, \mathbf{x}_i \rangle > 0 \ .$$

Notation:

- Denote by $\mathbf{w}^*$ a separating hyperplane of unit norm and let $\gamma = \min_i y_i \langle \mathbf{w}^*, \mathbf{x}_i \rangle$
- W.l.o.g. assume $\|\mathbf{x}_i\| = 1$ for every i.



Separating Hyperplane using the Ellipsoid

- We can take the initial ball to be the unit ball
- The separation oracle looks for i s.t. $y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0$
- If there's no such i, we're done. Otherwise, the oracle returns $y_i \mathbf{x}_i$
- The algorithm stops after at most $2d(2d+2)\log(1/\gamma)$ iterations

Separating Hyperplane using Sub-gradient Descent

Consider the problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) \qquad \text{where} \qquad f(\mathbf{w}) \;=\; \max_{i} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$$

Separating Hyperplane using Sub-gradient Descent

Consider the problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) \qquad \text{where} \qquad f(\mathbf{w}) \;=\; \max_{i} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$$

Observe:

- f is convex

Separating Hyperplane using Sub-gradient Descent

Consider the problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) \qquad \text{where} \qquad f(\mathbf{w}) \;=\; \max_{i} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$$

Observe:

- f is convex
- A sub-gradient of f at $\mathbf{w}$ is $-y_i \mathbf{x}_i$ for some $i \in \operatorname{argmax} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$

Separating Hyperplane using Sub-gradient Descent

Consider the problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) \qquad \text{where} \qquad f(\mathbf{w}) \;=\; \max_{i} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$$

Observe:

- f is convex
- A sub-gradient of f at $\mathbf{w}$ is $-y_i \mathbf{x}_i$ for some $i \in \operatorname{argmax} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$
- f is 1-Lipschitz

Separating Hyperplane using Sub-gradient Descent

Consider the problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) \qquad \text{where} \qquad f(\mathbf{w}) \;=\; \max_{i} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$$

Observe:

- f is convex
- A sub-gradient of f at $\mathbf{w}$ is $-y_i \mathbf{x}_i$ for some $i \in \operatorname{argmax} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$
- f is 1-Lipschitz
- $f(\mathbf{w}^*) = -\gamma$

Separating Hyperplane using Sub-gradient Descent

Consider the problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) \qquad \text{where} \qquad f(\mathbf{w}) \;=\; \max_{i} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$$

Observe:

- f is convex
- A sub-gradient of f at $\mathbf{w}$ is $-y_i \mathbf{x}_i$ for some $i \in \operatorname{argmax} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$
- f is 1-Lipschitz
- $f(\mathbf{w}^*) = -\gamma$
- Therefore, after $t > \frac{1}{\gamma^2}$ iterations, we have $f(\mathbf{w}^{(t)}) < f(\mathbf{w}^*) + \gamma = 0$

Separating Hyperplane using Sub-gradient Descent

Consider the problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) \qquad \text{where} \qquad f(\mathbf{w}) \;=\; \max_{i} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$$

Observe:

- f is convex
- A sub-gradient of f at $\mathbf{w}$ is $-y_i \mathbf{x}_i$ for some $i \in \operatorname{argmax} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$
- f is 1-Lipschitz
- $f(\mathbf{w}^*) = -\gamma$
- Therefore, after $t > \frac{1}{\gamma^2}$ iterations, we have $f(\mathbf{w}^{(t)}) < f(\mathbf{w}^*) + \gamma = 0$
- So, $\mathbf{w}^{(t)}$ is a separating hyperplane

Separating Hyperplane using Sub-gradient Descent

Consider the problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) \qquad \text{where} \qquad f(\mathbf{w}) \;=\; \max_{i} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$$

Observe:

- f is convex
- A sub-gradient of f at $\mathbf{w}$ is $-y_i \mathbf{x}_i$ for some $i \in \operatorname{argmax} -y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$
- f is 1-Lipschitz
- $f(\mathbf{w}^*) = -\gamma$
- Therefore, after $t > \frac{1}{\gamma^2}$ iterations, we have $f(\mathbf{w}^{(t)}) < f(\mathbf{w}^*) + \gamma = 0$
- So, $\mathbf{w}^{(t)}$ is a separating hyperplane
- The resulting algorithm is closely related to the Batch Perceptron

The Batch Perceptron

- Initialize, $\mathbf{w}^{(1)} = \mathbf{0}$
- While exists i s.t. $y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0$ update

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

The Batch Perceptron

- Initialize, $\mathbf{w}^{(1)} = \mathbf{0}$
- While exists i s.t. $y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0$ update

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

Exercise: why did we eliminate η ?

Ellipsoid vs. Sub-gradient

For f convex and ρ-Lipschitz:

	iterations	cost of iteration
Ellipsoid	$d^2 \log\left(\frac{\rho \|\mathbf{w}^*\|}{\epsilon}\right)$	d^2+ "gradient oracle"
Sub-gradient descent	$\frac{\|\mathbf{w}^*\|^2 \rho^2}{\epsilon^2}$	$d+$"gradient oracle"

Ellipsoid vs. Sub-gradient

For f convex and ρ-Lipschitz:

	iterations	cost of iteration
Ellipsoid	$d^2 \log\left(\frac{\rho \|\mathbf{w}^*\|}{\epsilon}\right)$	$d^2 +$ "gradient oracle"
Sub-gradient descent	$\frac{\|\mathbf{w}^*\|^2 \rho^2}{\epsilon^2}$	$d +$ "gradient oracle"

For separating hyperplane:

	iterations	cost of iteration
Ellipsoid	$d^2 \log\left(\frac{1}{\gamma}\right)$	$d^2 + dm$
Sub-gradient descent	$\frac{1}{\gamma^2}$	dm

Outline

Convex Learning Problems

Definition (Convex Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called convex if the hypothesis class $\mathcal{H}$ is a convex set and for all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex function (where, for any z, $\ell(\cdot, z)$ denotes the function $f : \mathcal{H} \to \mathbb{R}$ defined by $f(\mathbf{w}) = \ell(\mathbf{w}, z)$).

Convex Learning Problems

Definition (Convex Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called convex if the hypothesis class $\mathcal{H}$ is a convex set and for all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex function (where, for any z, $\ell(\cdot, z)$ denotes the function $f : \mathcal{H} \to \mathbb{R}$ defined by $f(\mathbf{w}) = \ell(\mathbf{w}, z)$).

- The $ERM_{\mathcal{H}}$ problem w.r.t. a convex learning problem is a convex optimization problem: $\min_{\mathbf{w} \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{w}, z_i)$

Convex Learning Problems

Definition (Convex Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called convex if the hypothesis class $\mathcal{H}$ is a convex set and for all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex function (where, for any z, $\ell(\cdot, z)$ denotes the function $f : \mathcal{H} \to \mathbb{R}$ defined by $f(\mathbf{w}) = \ell(\mathbf{w}, z)$).

- The $ERM_{\mathcal{H}}$ problem w.r.t. a convex learning problem is a convex optimization problem: $\min_{\mathbf{w} \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{w}, z_i)$
- Example – least squares: $\mathcal{H} = \mathbb{R}^d$, $Z = \mathbb{R}^d \times \mathbb{R}$, $\ell(\mathbf{w}, (\mathbf{x}, y)) = (\langle \mathbf{w}, \mathbf{x} \rangle - y)^2$

Learnability of convex learning problems

- Claim: Not all convex learning problems over $\mathbb{R}^d$ are learnable
- The intuitive reason is numerical stability
- But, with two additional mild conditions, we obtain learnability
  - $\mathcal{H}$ is bounded
  - The loss function (or its gradient) is Lipschitz

Convex-Lipschitz-bounded learning problem

Definition (Convex-Lipschitz-Bounded Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called Convex-Lipschitz-Bounded, with parameters ρ, B if the following holds:

- The hypothesis class $\mathcal{H}$ is a convex set and for all $\mathbf{w} \in \mathcal{H}$ we have $\|\mathbf{w}\| \leq B$.
- For all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex and ρ-Lipschitz function.

Convex-Lipschitz-bounded learning problem

Definition (Convex-Lipschitz-Bounded Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called Convex-Lipschitz-Bounded, with parameters ρ, B if the following holds:

- The hypothesis class $\mathcal{H}$ is a convex set and for all $\mathbf{w} \in \mathcal{H}$ we have $\|\mathbf{w}\| \leq B$.
- For all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex and ρ-Lipschitz function.

Example:

- $\mathcal{H} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\| \leq B\}$
- $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq \rho\}$, $\mathcal{Y} = \mathbb{R}$,
- $\ell(\mathbf{w}, (\mathbf{x}, y)) = |\langle \mathbf{w}, \mathbf{x} \rangle - y|$

Convex-Smooth-bounded learning problem

A function f is β-smooth if it is differentiable and its gradient is β-Lipschitz.

Definition (Convex-Smooth-Bounded Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called Convex-Smooth-Bounded, with parameters β, B if the following holds:

- The hypothesis class $\mathcal{H}$ is a convex set and for all $\mathbf{w} \in \mathcal{H}$ we have $\|\mathbf{w}\| \leq B$.
- For all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex, non-negative, and β-smooth function.

Convex-Smooth-bounded learning problem

A function f is β-smooth if it is differentiable and its gradient is β-Lipschitz.

Definition (Convex-Smooth-Bounded Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called Convex-Smooth-Bounded, with parameters β, B if the following holds:

- The hypothesis class $\mathcal{H}$ is a convex set and for all $\mathbf{w} \in \mathcal{H}$ we have $\|\mathbf{w}\| \leq B$.
- For all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex, non-negative, and β-smooth function.

Example:

- $\mathcal{H} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\| \leq B\}$
- $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq \beta/2\}$, $\mathcal{Y} = \mathbb{R}$,
- $\ell(\mathbf{w}, (\mathbf{x}, y)) = (\langle \mathbf{w}, \mathbf{x} \rangle - y)^2$

Learnability

We will later show that all Convex-Lipschitz-Bounded and Convex-Smooth-Bounded learning problems are learnable, with sample complexity that depends only on ϵ, δ, B, and ρ or β.

Outline

Surrogate Loss Functions

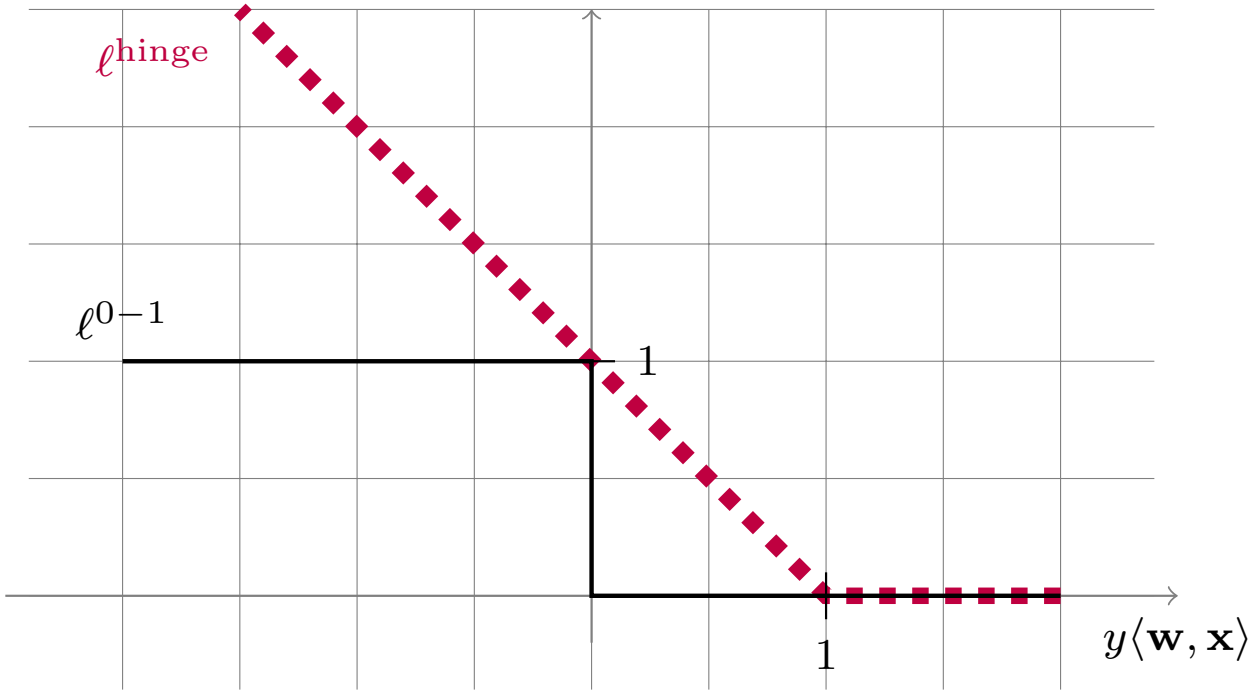
- In many natural cases, the loss function is not convex
- For example, the $0-1$ loss for halfspaces

$$\ell^{0-1}(\mathbf{w},(\mathbf{x},y)) = \mathbb{1}_{[y \neq \text{sign}(\langle \mathbf{w},\mathbf{x}\rangle)]} = \mathbb{1}_{[y\langle \mathbf{w},\mathbf{x}\rangle \leq 0]} \ .$$

- Non-convex loss function usually yields intractable learning problems
- Popular approach: circumvent hardness by upper bounding the non-convex loss function using a convex surrogate loss function

Hinge-loss

$$\ell^{\text{hinge}}(\mathbf{w}, (\mathbf{x}, y)) \stackrel{\text{def}}{=} \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\} \ .$$



Error Decomposition Revisited

- Suppose we have a learner for the hinge-loss that guarantees:

$$L_{\mathcal{D}}^{\text{hinge}}(A(S)) \;\leq\; \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) + \epsilon \;,$$

Error Decomposition Revisited

- Suppose we have a learner for the hinge-loss that guarantees:

$$L_{\mathcal{D}}^{\text{hinge}}(A(S)) \;\leq\; \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) + \epsilon \;,$$

- Using the surrogate property,

$$L_{\mathcal{D}}^{0-1}(A(S)) \;\leq\; \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) + \epsilon \;.$$

Error Decomposition Revisited

- Suppose we have a learner for the hinge-loss that guarantees:

$$L_{\mathcal{D}}^{\text{hinge}}(A(S)) \;\leq\; \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) + \epsilon \;,$$

- Using the surrogate property,

$$L_{\mathcal{D}}^{0-1}(A(S)) \;\leq\; \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) + \epsilon \;.$$

- We can further rewrite the upper bound as:

$$\begin{aligned} L_{\mathcal{D}}^{0-1}(A(S)) &\leq \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{0-1}(\mathbf{w}) + \left(\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) - \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{0-1}(\mathbf{w}) \right) + \epsilon \\ &= \epsilon_{\text{approximation}} + \epsilon_{\text{optimization}} + \epsilon_{\text{estimation}} \end{aligned}$$

Error Decomposition Revisited

- Suppose we have a learner for the hinge-loss that guarantees:

$$L_{\mathcal{D}}^{\text{hinge}}(A(S)) \;\leq\; \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) + \epsilon \;,$$

- Using the surrogate property,

$$L_{\mathcal{D}}^{0-1}(A(S)) \;\leq\; \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) + \epsilon \;.$$

- We can further rewrite the upper bound as:

$$L_{\mathcal{D}}^{0-1}(A(S)) \leq \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{0-1}(\mathbf{w}) + \left(\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) - \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{0-1}(\mathbf{w})\right) + \epsilon$$

$$= \epsilon_{\text{approximation}} + \epsilon_{\text{optimization}} + \epsilon_{\text{estimation}}$$

- The optimization error is a result of our inability to minimize the training loss with respect to the original loss.

Outline

Learning Using Stochastic Gradient Descent

- Consider a convex-Lipschitz-bounded learning problem.

Learning Using Stochastic Gradient Descent

- Consider a convex-Lipschitz-bounded learning problem.
- Recall: our goal is to (probably approximately) solve:

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

Learning Using Stochastic Gradient Descent

- Consider a convex-Lipschitz-bounded learning problem.
- Recall: our goal is to (probably approximately) solve:

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- So far, learning was based on the empirical risk, $L_S(\mathbf{w})$

Learning Using Stochastic Gradient Descent

- Consider a convex-Lipschitz-bounded learning problem.
- Recall: our goal is to (probably approximately) solve:

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- So far, learning was based on the empirical risk, $L_S(\mathbf{w})$
- We now consider directly minimizing $L_{\mathcal{D}}(\mathbf{w})$

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L_{\mathcal{D}}(\mathbf{w})$

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\nabla \ell(\mathbf{w}, z)]$

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\nabla \ell(\mathbf{w}, z)]$
- We can't calculate $\nabla L_{\mathcal{D}}(\mathbf{w})$ because we don't know $\mathcal{D}$

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\nabla \ell(\mathbf{w}, z)]$
- We can't calculate $\nabla L_{\mathcal{D}}(\mathbf{w})$ because we don't know $\mathcal{D}$
- But we can estimate it by $\nabla \ell(\mathbf{w}, z)$ for $z \sim \mathcal{D}$

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\nabla \ell(\mathbf{w}, z)]$
- We can't calculate $\nabla L_{\mathcal{D}}(\mathbf{w})$ because we don't know $\mathcal{D}$
- But we can estimate it by $\nabla \ell(\mathbf{w}, z)$ for $z \sim \mathcal{D}$
- If we take a step in the direction $\mathbf{v} = \nabla \ell(\mathbf{w}, z)$ then in expectation we're moving in the right direction

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\nabla \ell(\mathbf{w}, z)]$
- We can't calculate $\nabla L_{\mathcal{D}}(\mathbf{w})$ because we don't know $\mathcal{D}$
- But we can estimate it by $\nabla \ell(\mathbf{w}, z)$ for $z \sim \mathcal{D}$
- If we take a step in the direction $\mathbf{v} = \nabla \ell(\mathbf{w}, z)$ then in expectation we're moving in the right direction
- In other words, $\mathbf{v}$ is an unbiased estimate of the gradient

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

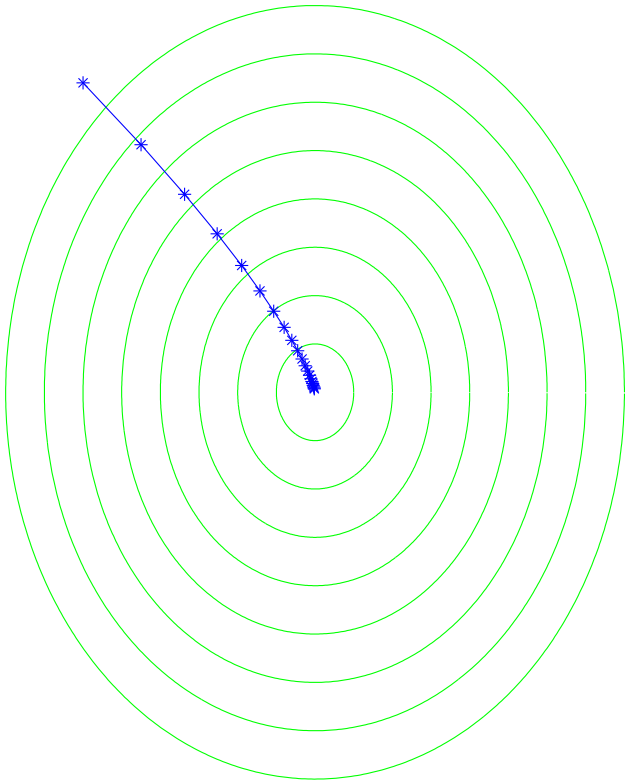
- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\nabla \ell(\mathbf{w}, z)]$
- We can't calculate $\nabla L_{\mathcal{D}}(\mathbf{w})$ because we don't know $\mathcal{D}$
- But we can estimate it by $\nabla \ell(\mathbf{w}, z)$ for $z \sim \mathcal{D}$
- If we take a step in the direction $\mathbf{v} = \nabla \ell(\mathbf{w}, z)$ then in expectation we're moving in the right direction
- In other words, $\mathbf{v}$ is an unbiased estimate of the gradient
- We'll show that this is good enough

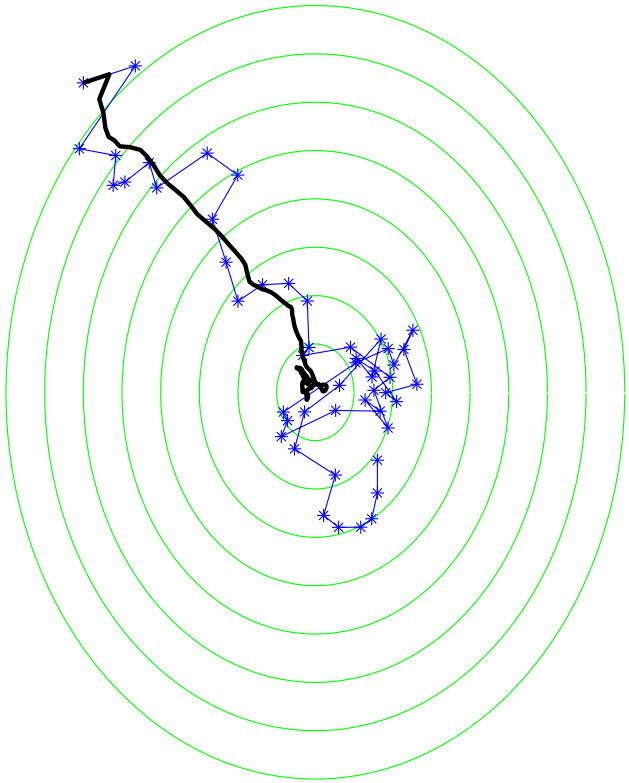
Stochastic Gradient Descent

- **initialize:** $\mathbf{w}^{(1)} = \mathbf{0}$
- **for** $t = 1, 2, \ldots, T$
  - choose $z_t \sim \mathcal{D}$
  - let $\mathbf{v}_t \in \partial \ell(\mathbf{w}^{(t)}, z_t)$ update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$
- **output** $\bar{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}^{(t)}$

Stochastic Gradient Descent

- **initialize:** $\mathbf{w}^{(1)} = \mathbf{0}$
- **for** $t = 1, 2, \ldots, T$
  - choose $z_t \sim \mathcal{D}$
  - let $\mathbf{v}_t \in \partial \ell(\mathbf{w}^{(t)}, z_t)$ update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$
- **output** $\bar{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}^{(t)}$





Analyzing SGD for convex-Lipschitz-bounded

By algebraic manipulations, for any sequence of $\mathbf{v}_1, \ldots, \mathbf{v}_T$, and any $\mathbf{w}^\star$,

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle = \frac{\|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2$$

Analyzing SGD for convex-Lipschitz-bounded

By algebraic manipulations, for any sequence of $\mathbf{v}_1, \ldots, \mathbf{v}_T$, and any $\mathbf{w}^\star$,

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle = \frac{\|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2$$

Assume that $\|\mathbf{v}_t\| \leq \rho$ for all t and that $\|\mathbf{w}^\star\| \leq B$ we obtain

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle \leq \frac{B^2}{2\eta} + \frac{\eta \, \rho^2 \, T}{2}$$

Analyzing SGD for convex-Lipschitz-bounded

By algebraic manipulations, for any sequence of $\mathbf{v}_1, \ldots, \mathbf{v}_T$, and any $\mathbf{w}^\star$,

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle = \frac{\|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2$$

Assume that $\|\mathbf{v}_t\| \leq \rho$ for all t and that $\|\mathbf{w}^\star\| \leq B$ we obtain

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle \leq \frac{B^2}{2\eta} + \frac{\eta \, \rho^2 \, T}{2}$$

In particular, for $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$ we get

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle \leq B \, \rho \, \sqrt{T} \; .$$

Analyzing SGD for convex-Lipschitz-bounded

Taking expectation of both sides w.r.t. the randomness of choosing $z_1, \ldots, z_T$ we obtain:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\left\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\right\rangle\right] \leq B\,\rho\,\sqrt{T}\,.$$

Analyzing SGD for convex-Lipschitz-bounded

Taking expectation of both sides w.r.t. the randomness of choosing $z_1, \ldots, z_T$ we obtain:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] \leq B\,\rho\,\sqrt{T}\ .$$

The law of total expectation: for every two random variables α, β, and a function g, $\mathbb{E}_\alpha[g(\alpha)] = \mathbb{E}_\beta\,\mathbb{E}_\alpha[g(\alpha)|\beta]$.

Analyzing SGD for convex-Lipschitz-bounded

Taking expectation of both sides w.r.t. the randomness of choosing $z_1, \ldots, z_T$ we obtain:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] \leq B\,\rho\,\sqrt{T}\ .$$

The law of total expectation: for every two random variables α, β, and a function g, $\mathbb{E}_\alpha[g(\alpha)] = \mathbb{E}_\beta\,\mathbb{E}_\alpha[g(\alpha)|\beta]$. Therefore

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle] = \mathop{\mathbb{E}}_{z_1,\ldots,z_{t-1}}\ \mathop{\mathbb{E}}_{z_1,\ldots,z_T}[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle \,|\, z_1, \ldots, z_{t-1}]\ .$$

Analyzing SGD for convex-Lipschitz-bounded

Taking expectation of both sides w.r.t. the randomness of choosing $z_1, \ldots, z_T$ we obtain:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] \leq B\,\rho\,\sqrt{T}\ .$$

The law of total expectation: for every two random variables α, β, and a function g, $\mathbb{E}_\alpha[g(\alpha)] = \mathbb{E}_\beta\,\mathbb{E}_\alpha[g(\alpha)|\beta]$. Therefore

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle] = \mathop{\mathbb{E}}_{z_1,\ldots,z_{t-1}}\ \mathop{\mathbb{E}}_{z_1,\ldots,z_T}[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle \,|\, z_1,\ldots,z_{t-1}]\ .$$

Once we know β the value of $\mathbf{w}^{(t)}$ is not random, hence,

$$\begin{aligned}\mathop{\mathbb{E}}_{z_1,\ldots,z_T}[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle \,|\, z_1,\ldots,z_{t-1}] &= \langle \mathbf{w}^{(t)} - \mathbf{w}^\star\ ,\ \mathop{\mathbb{E}}_{z_t}[\nabla \ell(\mathbf{w}^{(t)}, z_t)]\rangle \\ &= \langle \mathbf{w}^{(t)} - \mathbf{w}^\star\ ,\ \nabla L_{\mathcal{D}}(\mathbf{w}^{(t)})\rangle\end{aligned}$$

Analyzing SGD for convex-Lipschitz-bounded

We got:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star \;,\; \nabla L_{\mathcal{D}}(\mathbf{w}^{(t)})\rangle\right] \;\leq\; B\,\rho\,\sqrt{T}$$

Analyzing SGD for convex-Lipschitz-bounded

We got:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star \;,\; \nabla L_{\mathcal{D}}(\mathbf{w}^{(t)})\rangle\right] \;\leq\; B\,\rho\,\sqrt{T}$$

By convexity, this means

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}(L_{\mathcal{D}}(\mathbf{w}^{(t)}) - L_{\mathcal{D}}(\mathbf{w}^\star))\right] \;\leq\; B\,\rho\,\sqrt{T}$$

Analyzing SGD for convex-Lipschitz-bounded

We got:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star \;,\; \nabla L_{\mathcal{D}}(\mathbf{w}^{(t)})\rangle\right] \;\leq\; B\,\rho\,\sqrt{T}$$

By convexity, this means

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}(L_{\mathcal{D}}(\mathbf{w}^{(t)}) - L_{\mathcal{D}}(\mathbf{w}^\star))\right] \;\leq\; B\,\rho\,\sqrt{T}$$

Dividing by T and using convexity again,

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[L_{\mathcal{D}}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}^{(t)}\right)\right] \;\leq\; L_{\mathcal{D}}(\mathbf{w}^\star) + \frac{B\,\rho}{\sqrt{T}}$$

Learning convex-Lipschitz-bounded problems using SGD

Corollary

Consider a convex-Lipschitz-bounded learning problem with parameters ρ, B. *Then, for every* $\epsilon > 0$, *if we run the SGD method for minimizing* $L_{\mathcal{D}}(\mathbf{w})$ *with a number of iterations (i.e., number of examples)*

$$T \geq \frac{B^2 \rho^2}{\epsilon^2}$$

and with $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$, *then the output of SGD satisfies:*

$$\mathbb{E}\left[L_{\mathcal{D}}(\bar{\mathbf{w}})\right] \leq \min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) + \epsilon \ .$$

Summary

- Convex optimization
- Convex learning problems
- Learning using SGD