14 February 2023

CIS 5200: MACHINE LEARNING

CONVEX OPTIMIZATION

Surbhi Goel

Content here draws from material by Sham Kakade (Harvard), Elad Hazan (Princeton)

Spring 2023

OUTLINE - TODAY

- Logistics
- Kernels (quick summary)
- Convexity (recap)
- Gradient Descent
- Proof of Convergence

POWER OF KERNELS

- Show that the solution to your problem lies in the span of the training points, $w = \sum_{i=1}^{m} \alpha_i x_i$

There is a general theorem called the Representer Theorem which tells us when this is true

- Rewrite the algorithm and the predictor so that all training or test points are only accessed in inner-products ($x_i^\top x_j$) with other points

- Replace $x_i^\top x_j \rightarrow k(x_i, x_j)$ everywhere for a valid kernel k

Super Powerful!

CHALLENGE

- How do we choose a good feature map ϕ?
- Feature map is the same for all inputs!

Can learn the feature map itself → deep learning!

CONVEX OPTIMIZATION

$$\min_{w} \quad \underbrace{F(w)}_{\text{objective}}$$

$$\text{such that} \quad \underbrace{w \in \mathscr{C}}_{\text{constraint}}$$

Here $\mathscr{C} \subseteq \mathbb{R}^d$ is a *convex set* and $F : \mathbb{R}^d \to \mathbb{R}$ is a *convex function*

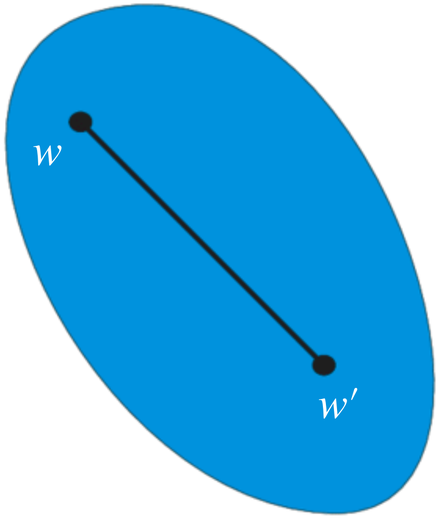
It is a powerful sub-class of optimization problems

- allows for efficient global solutions
- beautiful mathematical theory that provides guarantees
- very well researched with lots of available libraries

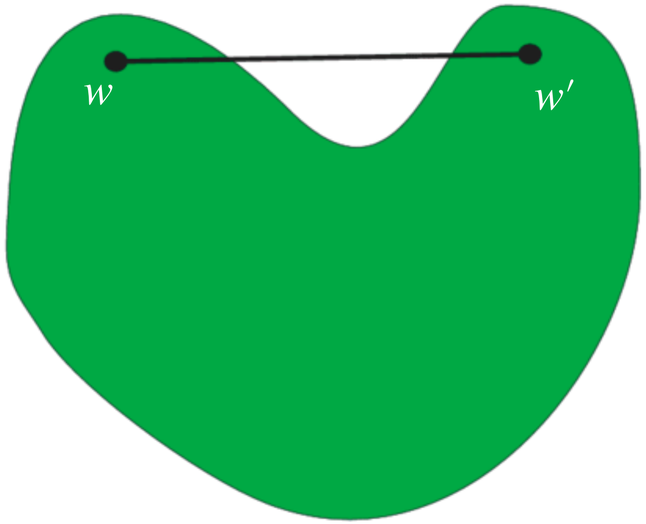
CONVEXITY - RECAP

Convex Set: For all $w, w' \in \mathscr{C}$ and $\alpha \in [0,1]$,

$$\alpha w + (1-\alpha)w' \in \mathscr{C} .$$



Convex

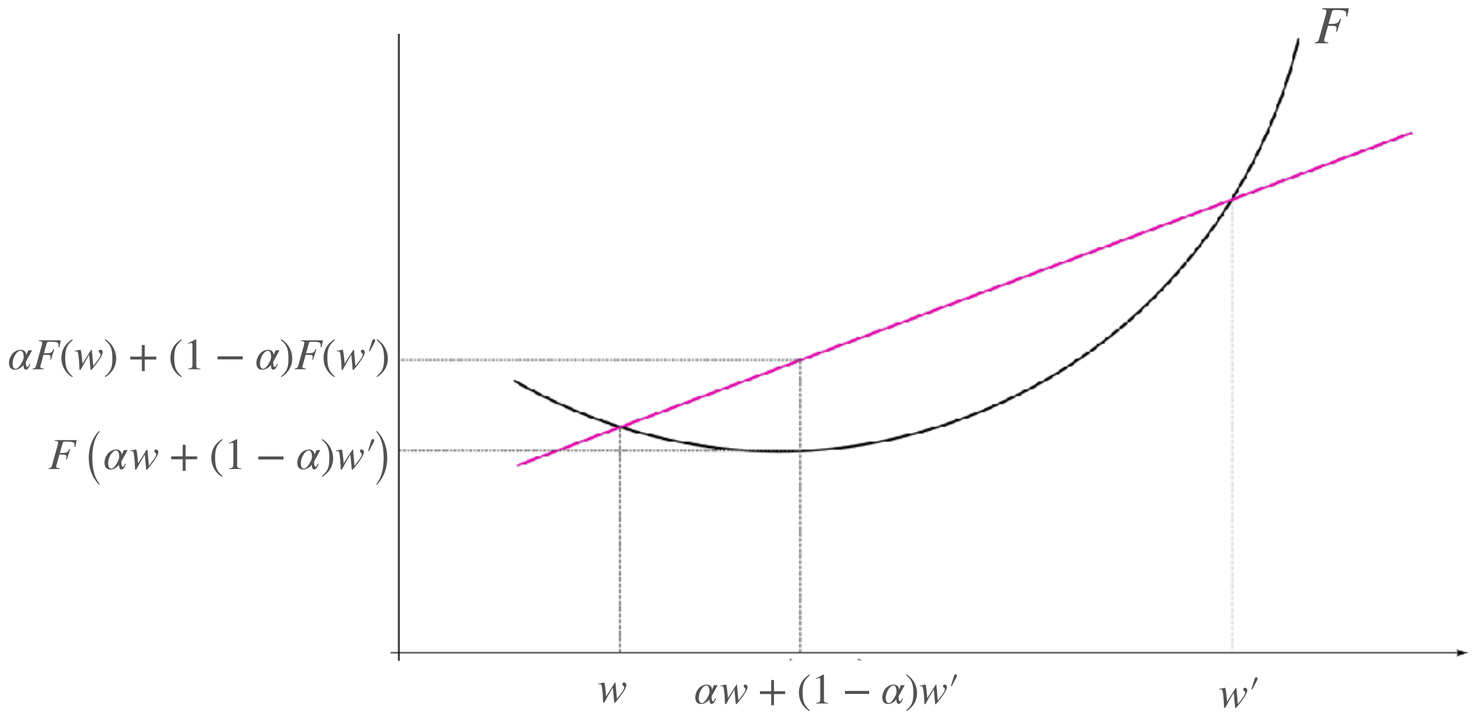


Non-convex

CONVEXITY - RECAP

Convex Function: For all $w, w' \in \mathbb{R}^d$ and $\alpha \in [0,1]$,

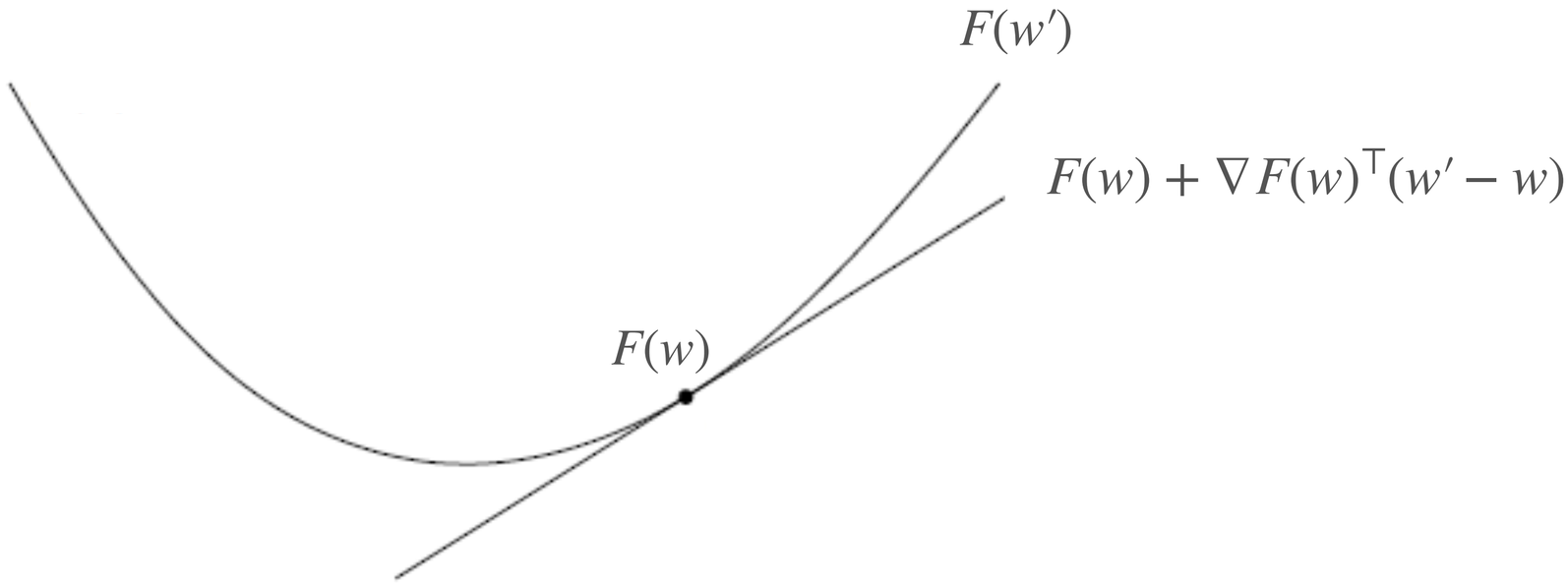
$$F\left(\alpha w + (1-\alpha)w'\right) \leq \alpha F(w) + (1-\alpha)F(w')$$



CONVEXITY - FIRST ORDER CHARACTERIZATION

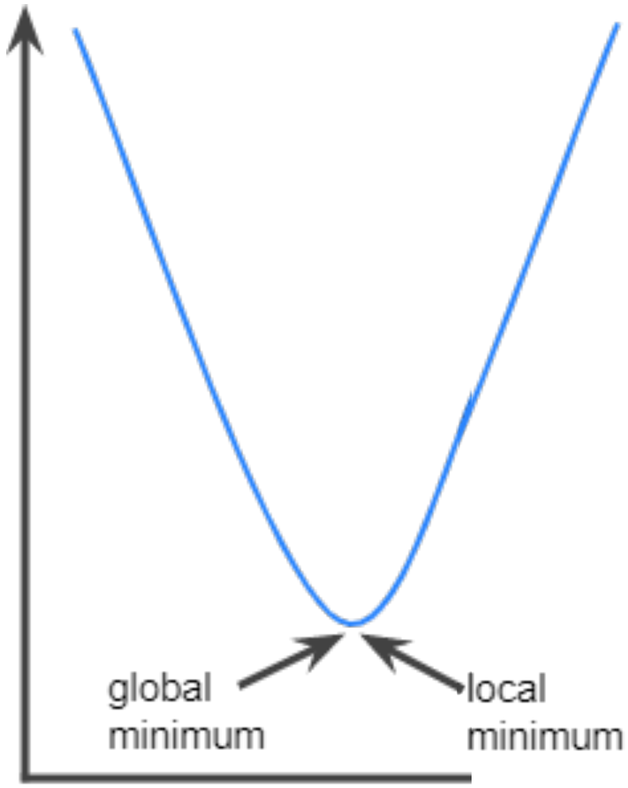
F is convex and differentiable, then for all $w, w' \in \mathbb{R}^d$

$$F(w') \geq F(w) + \nabla F(w)^\top (w' - w)$$

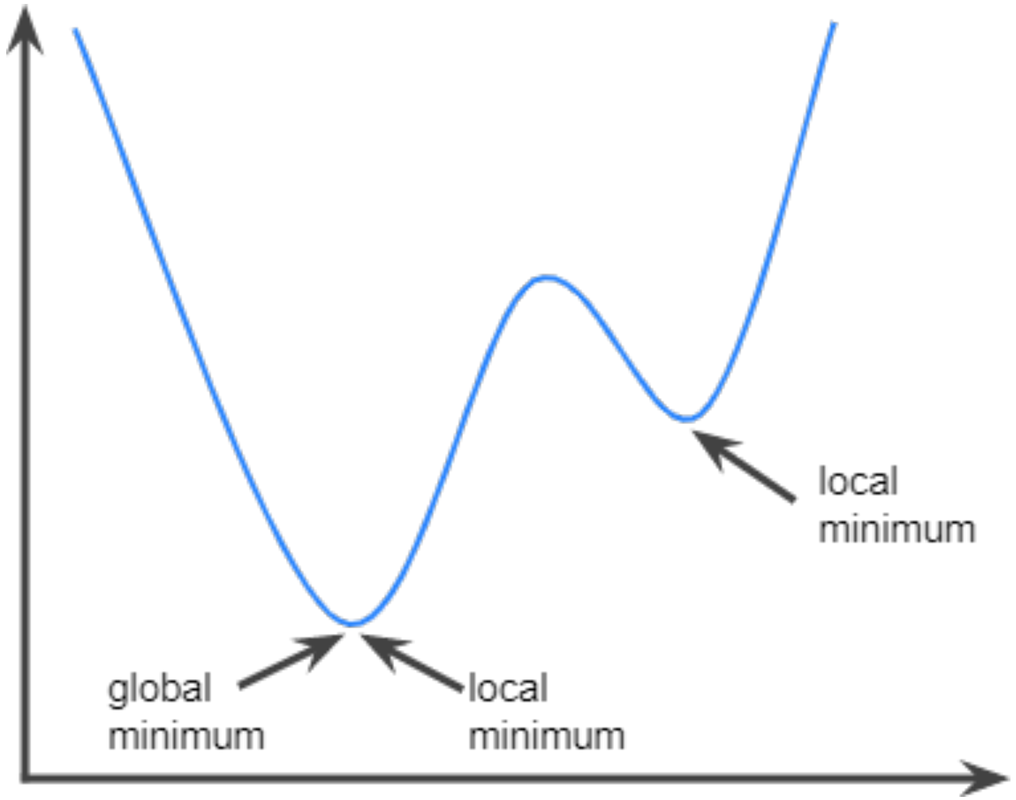


CONVEXITY - FIRST ORDER OPTIMALITY

F is convex and differentiable, any w that satisfies $\nabla F(w) = 0$ is a global minimum of F.



Convex



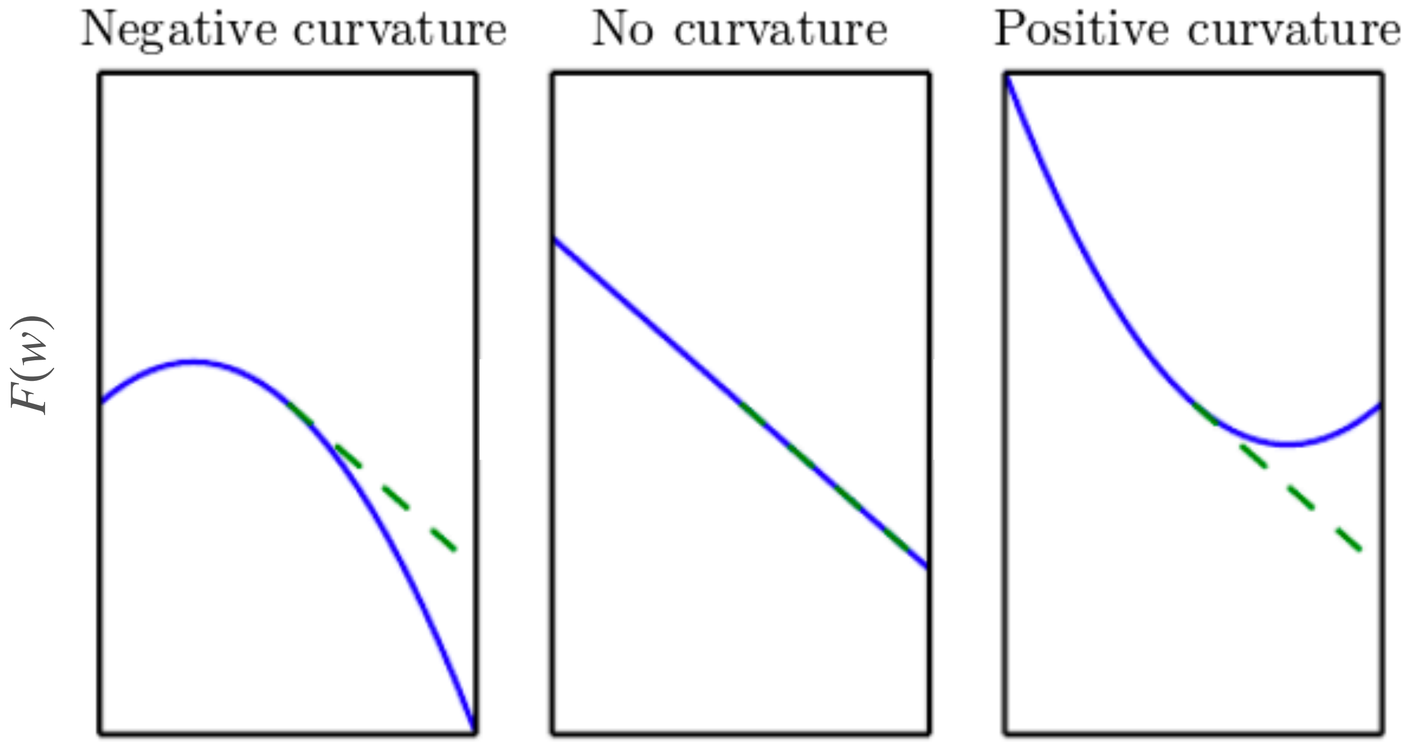
Non-convex

CONVEXITY - SECOND ORDER CHARACTERIZATION

F is convex and twice-differentiable, then for all $w \in \mathbb{R}^d$

$$\nabla^2 F(w) \succeq 0$$

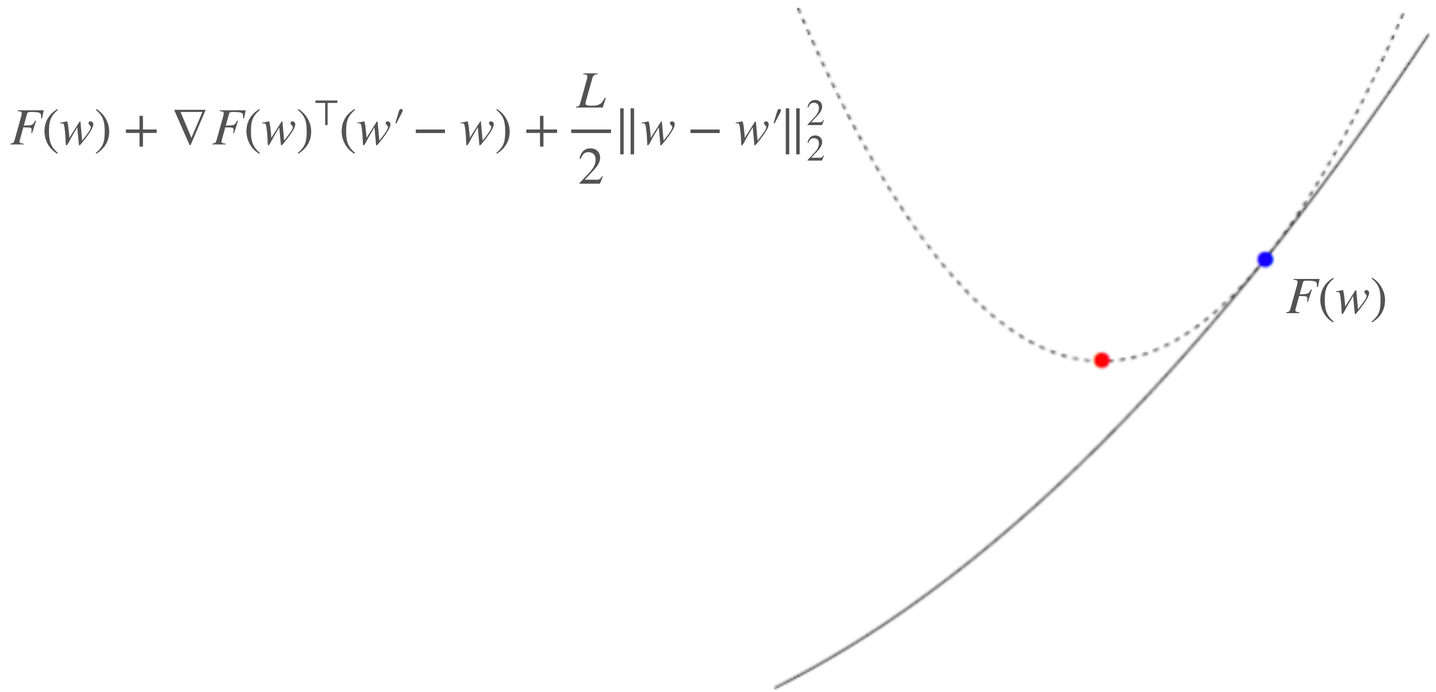
Hessian is positive semi-definite (PSD)



CONVEXITY - SMOOTHNESS

L-smooth Function: For all $w, w' \in \mathbb{R}^d$,

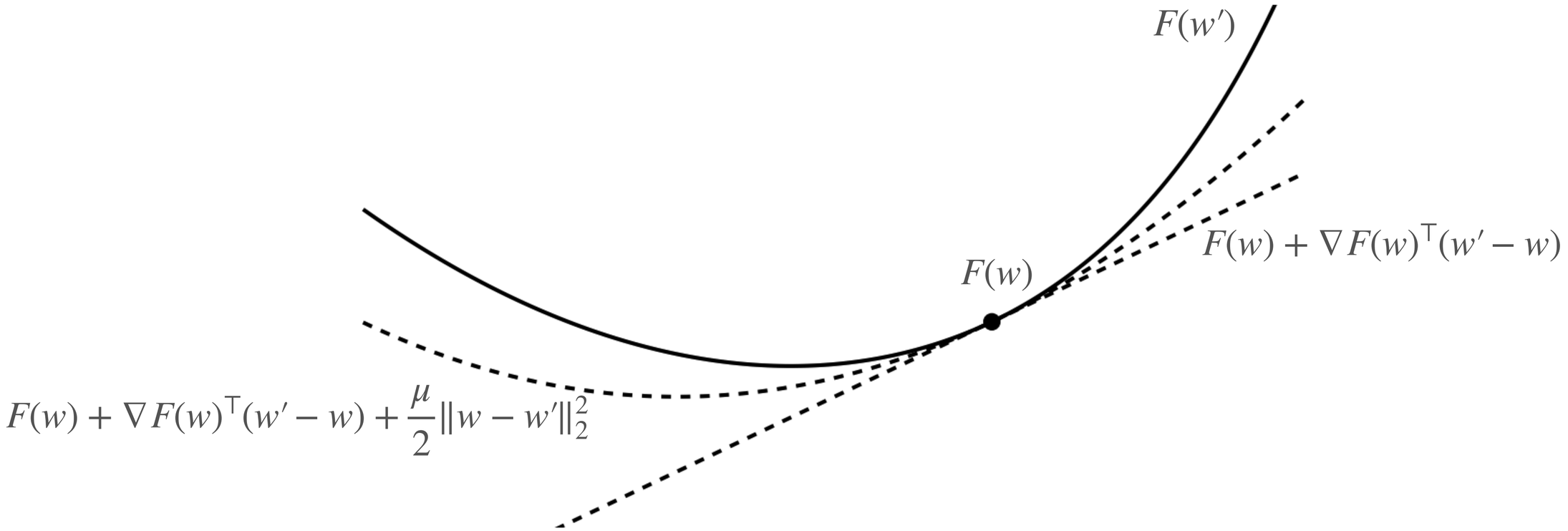
$$F(w') \leq F(w) + \nabla F(w)^\top (w' - w) + \frac{L}{2}\|w - w'\|_2^2$$



CONVEXITY - STRONG CONVEXITY

μ-strongly Convex Function: For all $w, w' \in \mathbb{R}^d$,

$$F(w') \geq F(w) + \nabla F(w)^\top (w' - w) + \frac{\mu}{2}\|w - w'\|_2^2$$



CONVEX FUNCTIONS - EXAMPLES

- ℓ_2-norm:

$$F(w) = \|x\|_2^2 = x^\top x$$

- Logistic:

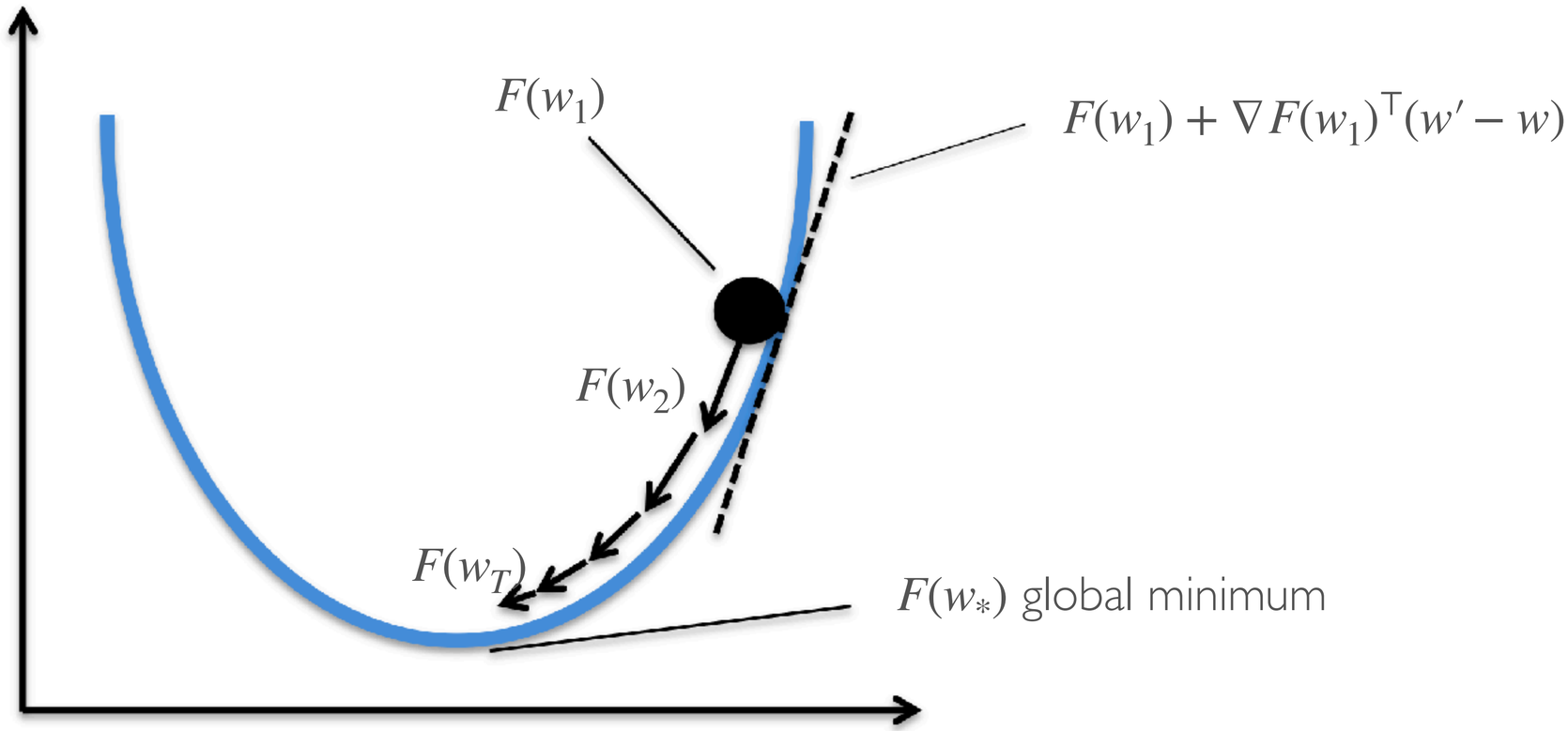
$$F(w; x, y) = \log(1 + \exp(-yw^\top x))$$

- Mean of convex functions: for convex functions $F_1, \ldots, F_m$

$$F(w) = \frac{1}{m} \sum_{i=1}^{m} F_i(w)$$

GRADIENT DESCENT - MOTIVATION

Move in the opposite direction of the gradient to decrease the function



GRADIENT DESCENT - ALGORITHM

Algorithm 1: Gradient Descent (GD)

Initialize $w_1 \in \mathbb{R}^d$

while $t = 1, 2, \ldots, T$ **do**

  Update $w_{t+1} = w_t - \eta_t \nabla F(w_t)$

end

Can stop when gradient becomes very small $\|\nabla F(w_t)\|_2 \leq \epsilon$

η_t is the learning rate or step size which governs how much to move

GRADIENT DESCENT - INTERPRETATION

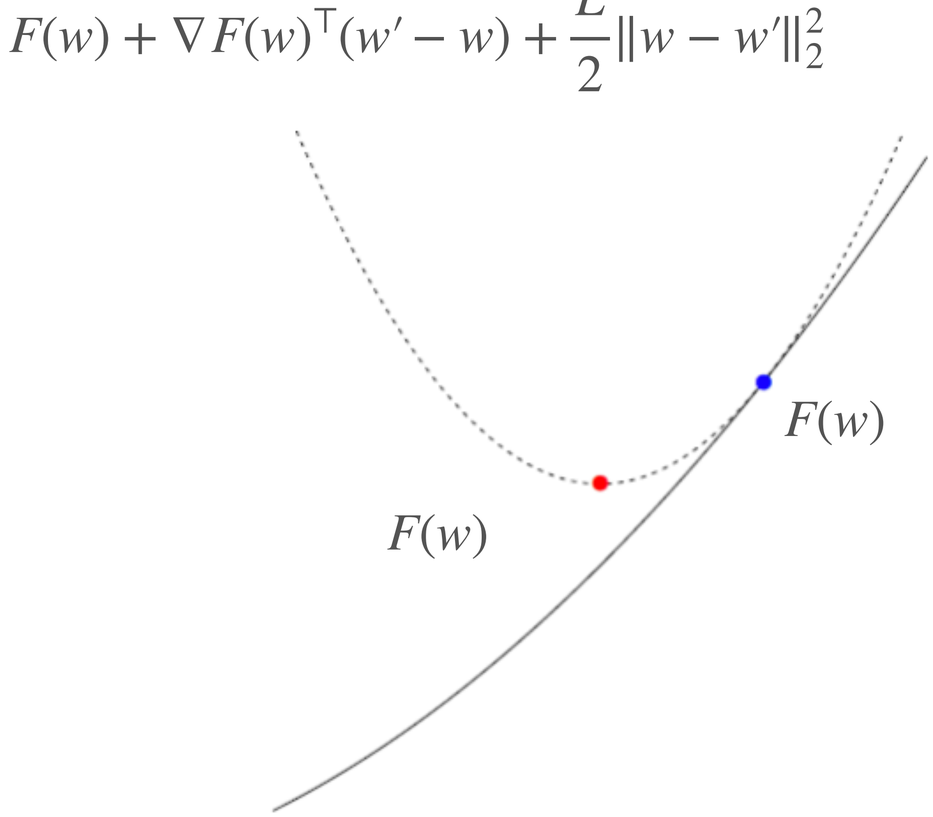
Consider L-smooth convex function F

We could locally minimize the quadratic

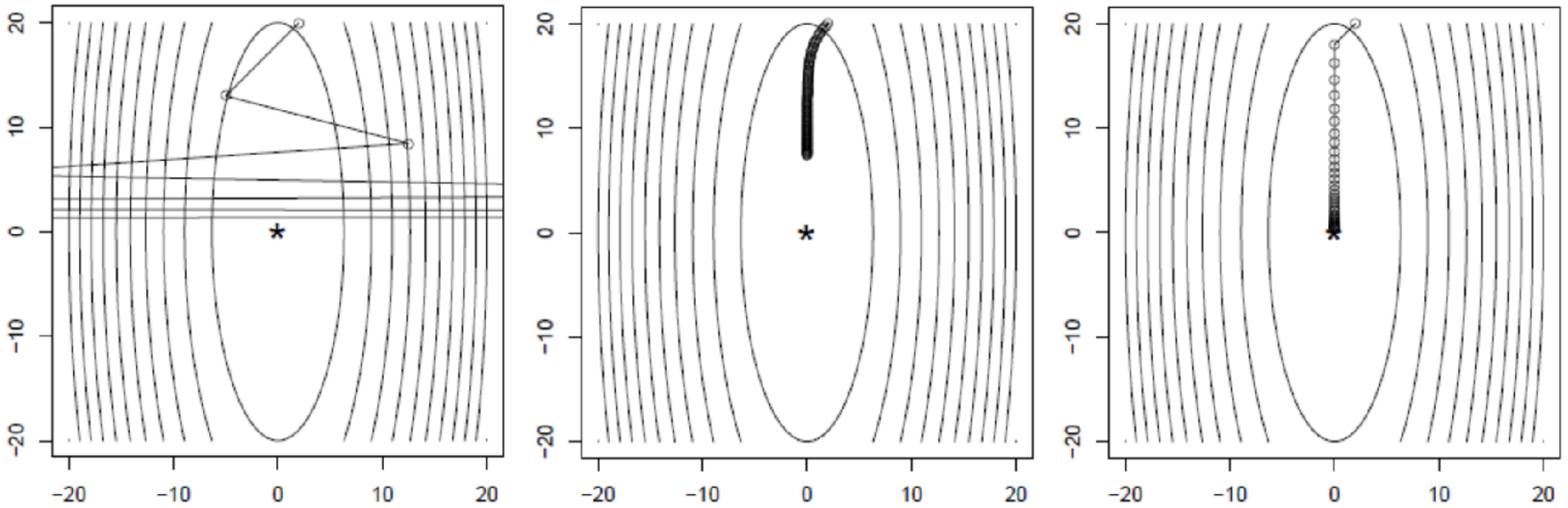
$$\min_{w'} F(w) + \nabla F(w)^\top (w' - w) + \frac{L}{2}\|w' - w\|_2^2$$

This gives $w' = w - \frac{1}{L}\nabla F(w)$

Gradient step with step size $\frac{1}{L}$



GRADIENT DESCENT - STEP SIZE



Too large

Too small

Just right

GRADIENT DESCENT - CONVERGENCE

Algorithm 1: Gradient Descent (GD)

Initialize $w_1 \in \mathbb{R}^d$

while $t = 1, 2, \ldots, T$ **do**

  Update $w_{t+1} = w_t - \eta_t \nabla F(w_t)$

end

Theorem: Suppose we run GD on L-smooth function F with fixed constant learning rate $\eta_t = 1/L$ at all time. Then at any time τ, we have

$$F(w_{\tau+1}) - F(w_*) \leq \frac{L\|w_1 - w_*\|_2^2}{2\tau}$$

w_* is the optimal

PROOF - OVERVIEW

Step 1: Upper bound the difference between function value at the next iterate and the current iterate for every time t

$$F(w_{t+1}) - F(w_t) \leq -\frac{L}{2}\|w_{t+1} - w_t\|^2$$

$F(w_t)$ is non-decreasing, we are reducing the function value

PROOF - OVERVIEW

Step 2: Upper bound the difference between function value at the next iterate and the global minimum for every time t

$$F(w_{t+1}) - F(w_*) \leq \frac{L}{2}\left(\|w_t - w_*\|_2^2 - \|w_{t+1} - w_*\|_2^2\right)$$

$\|w_t - w_*\|_2$ is non-decreasing, we are getting closer to the optimal

PROOF - OVERVIEW

Step 3: Use the previous to upper bound $F(w_{\tau+1}) - F(w_*)$, that is, the difference after τ iterates and the global minimum.

$$F(w_{\tau+1}) - F(w_*) \leq \frac{L\|w_1 - w_*\|_2^2}{2\tau}$$

PROOF - STEP 1

Step 1: Upper bound the difference between function value at the next iterate and the current iterate for every time t

$$F(w_{t+1}) - F(w_t) \leq -\frac{L}{2}\|w_{t+1} - w_t\|^2$$

$F(w_t)$ is non-decreasing, we are reducing the function value

Important properties:

$$F(w') \geq F(w) + \nabla F(w)^\top (w' - w)$$
$$F(w') \leq F(w) + \nabla F(w)^\top (w' - w) + \frac{L}{2}\|w - w'\|_2^2$$

PROOF - STEP 2

Step 2: Upper bound the difference between function value at the next iterate and the global minimum for every time t

$$F(w_{t+1}) - F(w_*) \leq \frac{L}{2}\left(\|w_t - w_*\|_2^2 - \|w_{t+1} - w_*\|_2^2\right)$$

$\|w_t - w_*\|_2$ is non-decreasing, we are getting closer to the optimal

Important properties:

$$F(w') \geq F(w) + \nabla F(w)^\top (w' - w)$$
$$F(w') \leq F(w) + \nabla F(w)^\top (w' - w) + \frac{L}{2}\|w - w'\|_2^2$$

GRADIENT DESCENT - CONVERGENCE

Algorithm 1: Gradient Descent (GD)

Initialize $w_1 \in \mathbb{R}^d$

while $t = 1, 2, \ldots, T$ **do**

  Update $w_{t+1} = w_t - \eta_t \nabla F(w_t)$

end

Theorem: Suppose we run GD on L-smooth μ-strongly convex function F with fixed constant learning rate $\eta_t = 1/L$ at all time. Then at any time τ, we have

$$\|w_{\tau+1} - w_*\|_2^2 \leq \left(1 - \frac{\mu}{L}\right)^{\tau} \|w_1 - w_*\|_2^2$$

w_* is the optimal

GRADIENT DESCENT - BENEFITS/DRAWBACKS

Algorithm 1: Gradient Descent (GD)

Initialize $w_1 \in \mathbb{R}^d$

while $t = 1, 2, \ldots, T$ **do**

  Update $w_{t+1} = w_t - \eta_t \nabla F(w_t)$

end

Easy to implement, requires only local information, very fast for strongly convex problems

Requires differentiability, many problems are not convex

GRADIENT DESCENT - RECIPE

- Write your problem as loss minimization
- As long as your loss is differentiable, run gradient descent
- Tune learning rate/step size to avoid divergence