Online Gradient Based Optimisation

Sai Ganesh Nagarajan, Assistant Professor @ IMADA

AI801 Lecture on 27.04.2026

University of
Southern Denmark

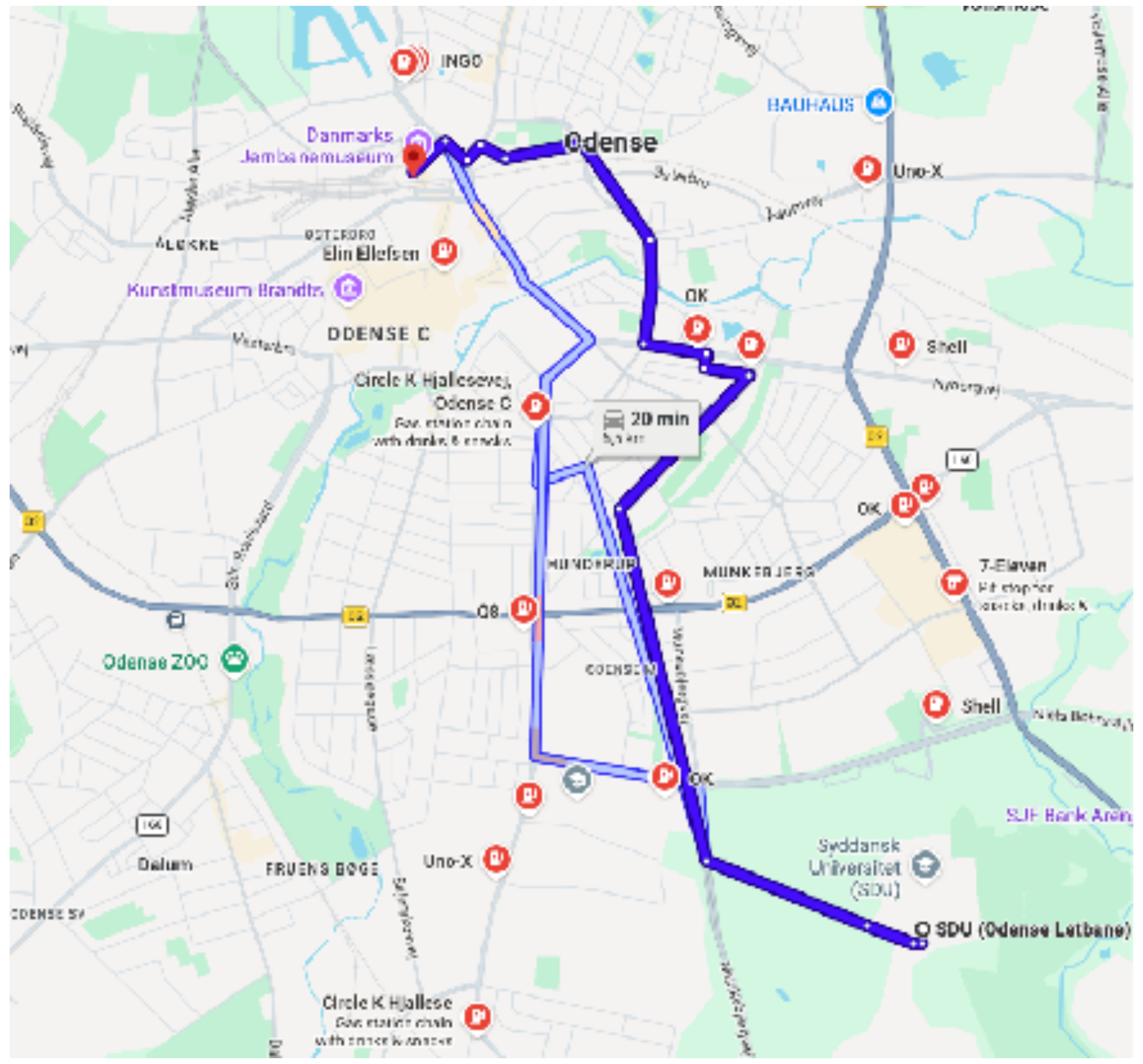
Special Topics Outline

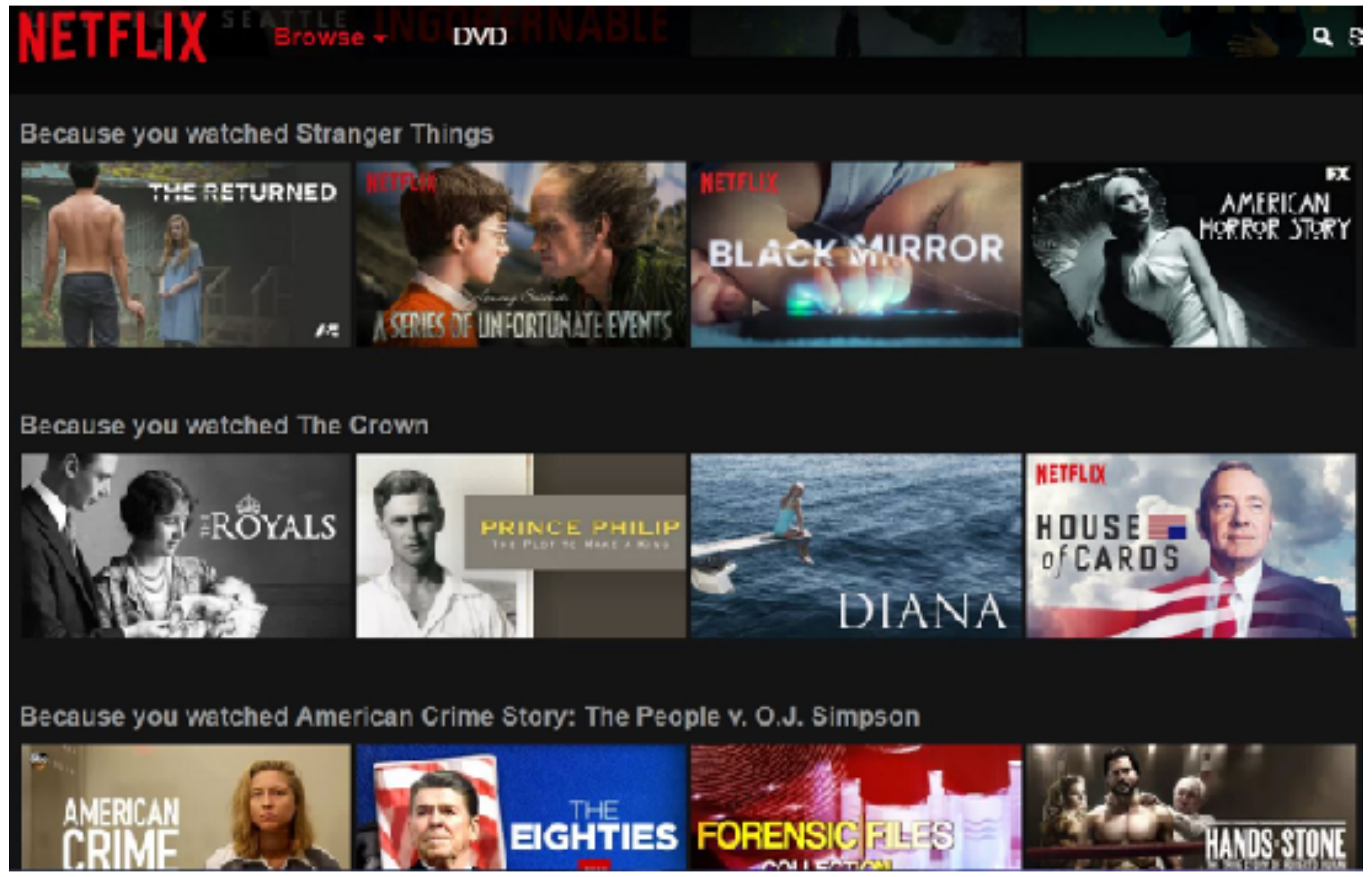
- What more can you do with Gradient Descent, especially in AI/ML?
- Three main topics:
  - Online Gradient Descent (Today)
  - Distributed Gradient Descent (May 4th)
  - Communication-Efficient Gradient Descent (May 11th)
- Cornerstone of modern machine learning (incl. deep learning, LLMs).
- Lecture slides, assignment and reading materials will be posted to the course web page (Unit 7).
- Main reference material for today is Elad Hazan's book on Online Convex Optimization.

Motivation


+123.11 %

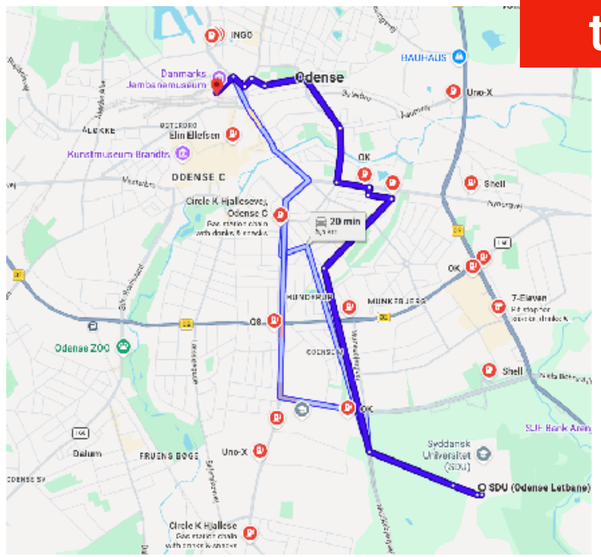

1
Google
2
3
Google

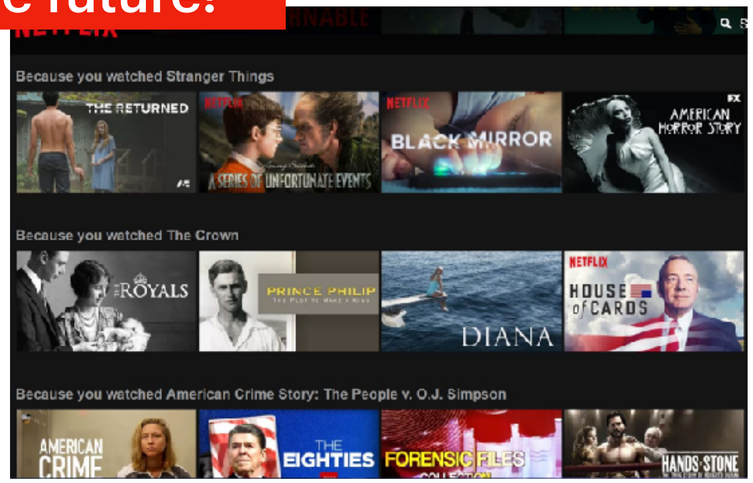

Odense
BAUHAUS
Uno-X
Shell
20 min
Odense ZOO
SDU (Odense Letbane)


NETFLIX
Browse
DVD
Because you watched Stranger Things
THE RETURNED
BLACK MIRROR
AMERICAN HORROR STORY
Because you watched The Crown
ROYALS
PRINCE PHILIP
DIANA
HOUSE of CARDS
Because you watched American Crime Story: The People v. O.J. Simpson
AMERICAN CRIME
THE EIGHTIES
FORENSIC FILES

Motivation

Take decisions now without the knowledge of the future!



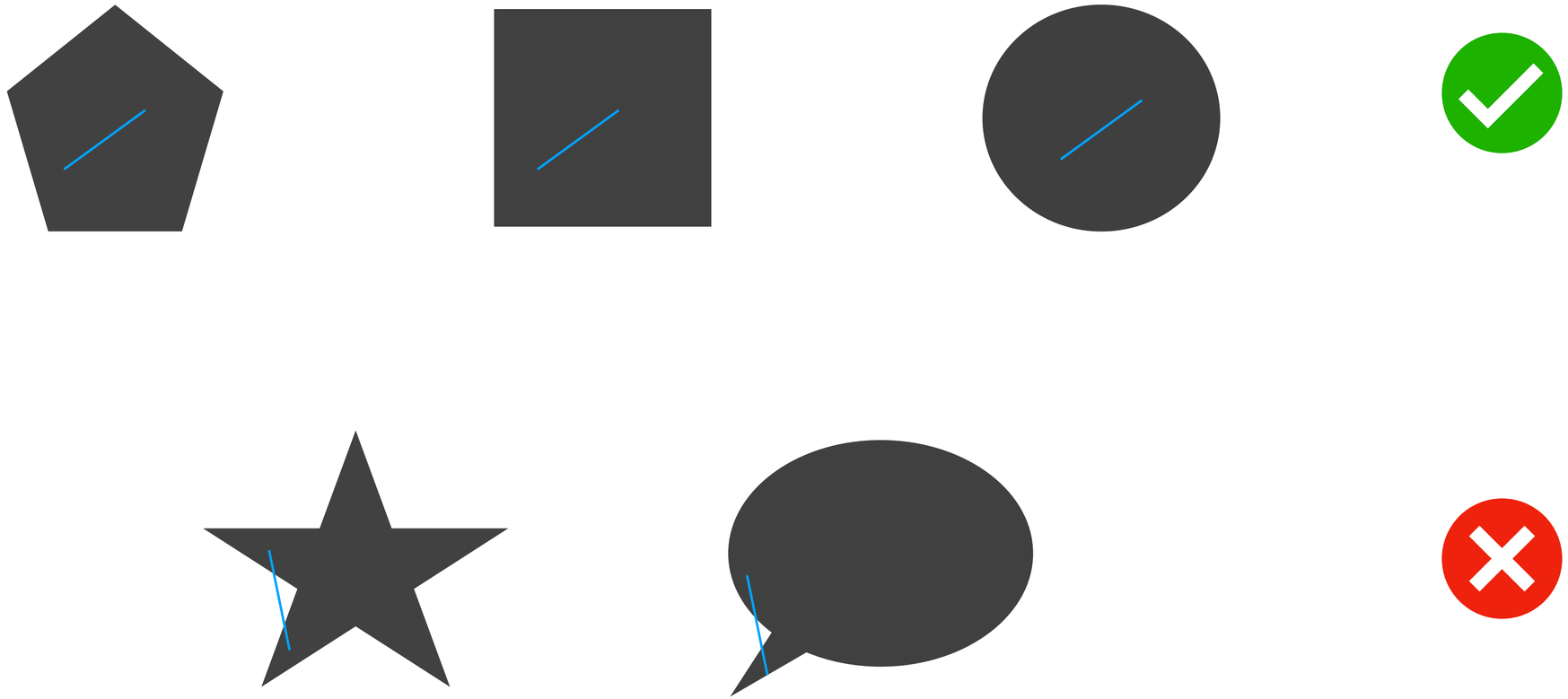


Online Optimization

- **Online** = We (player/agent) take decisions iteratively and obtain a *loss* or a *gradient* at each step, without knowing what is coming ahead!
- How do we “optimize” if we do not know *what* to optimize?
- Who decides the losses?
- Particularly, we will only be concerned with Online *convex* optimization (OCO).
- *When* can we say something meaningful about what the player can do in such a setting?

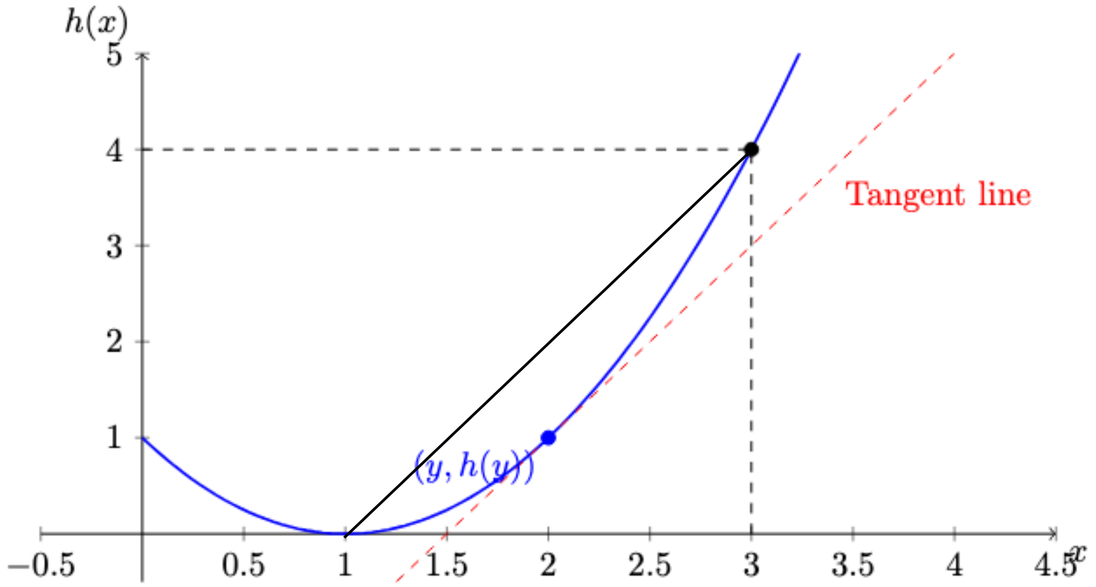
Quick Recap

Convex Sets



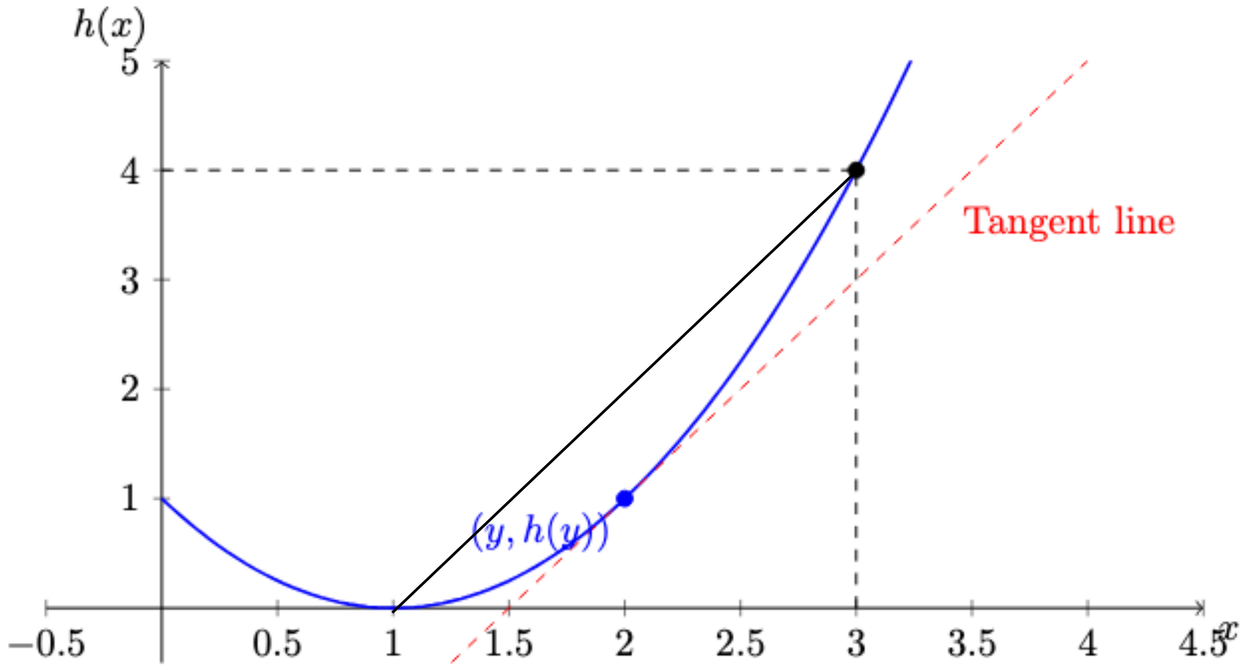
A set $\mathcal{X} \subseteq \mathbb{R}^d$ is convex if and only if for all $x, y \in \mathcal{X}$, we have: $\lambda x + (1-\lambda)y \in \mathcal{X}$, for all $\lambda \in [0,1]$

Convex Functions



A function $h : \mathcal{X} \to \mathbb{R}$ is convex, if and only if $h(\lambda x + (1-\lambda)y) \leq \lambda h(x) + (1-\lambda)h(y)$, for all $x, y \in \mathcal{X}$ and for all $\lambda \in [0,1]$

First-Order Conditions



A differentiable function $h : \mathcal{X} \to \mathbb{R}$ is convex, if and only if $h(x) \geq h(y) + \nabla h(y)^\top (x - y)$, for all $x, y \in \mathcal{X}$.

Smoothness and Strong Convexity

A differentiable function $h : \mathcal{X} \to \mathbb{R}$

is α-**strongly** convex,

if there is an $\alpha > 0$, such that:

$$h(x) \geq h(y) + \nabla h(y)^\top (x - y) + \frac{\alpha}{2}\|x - y\|_2^2,$$

for all $x, y \in \mathcal{X}$.

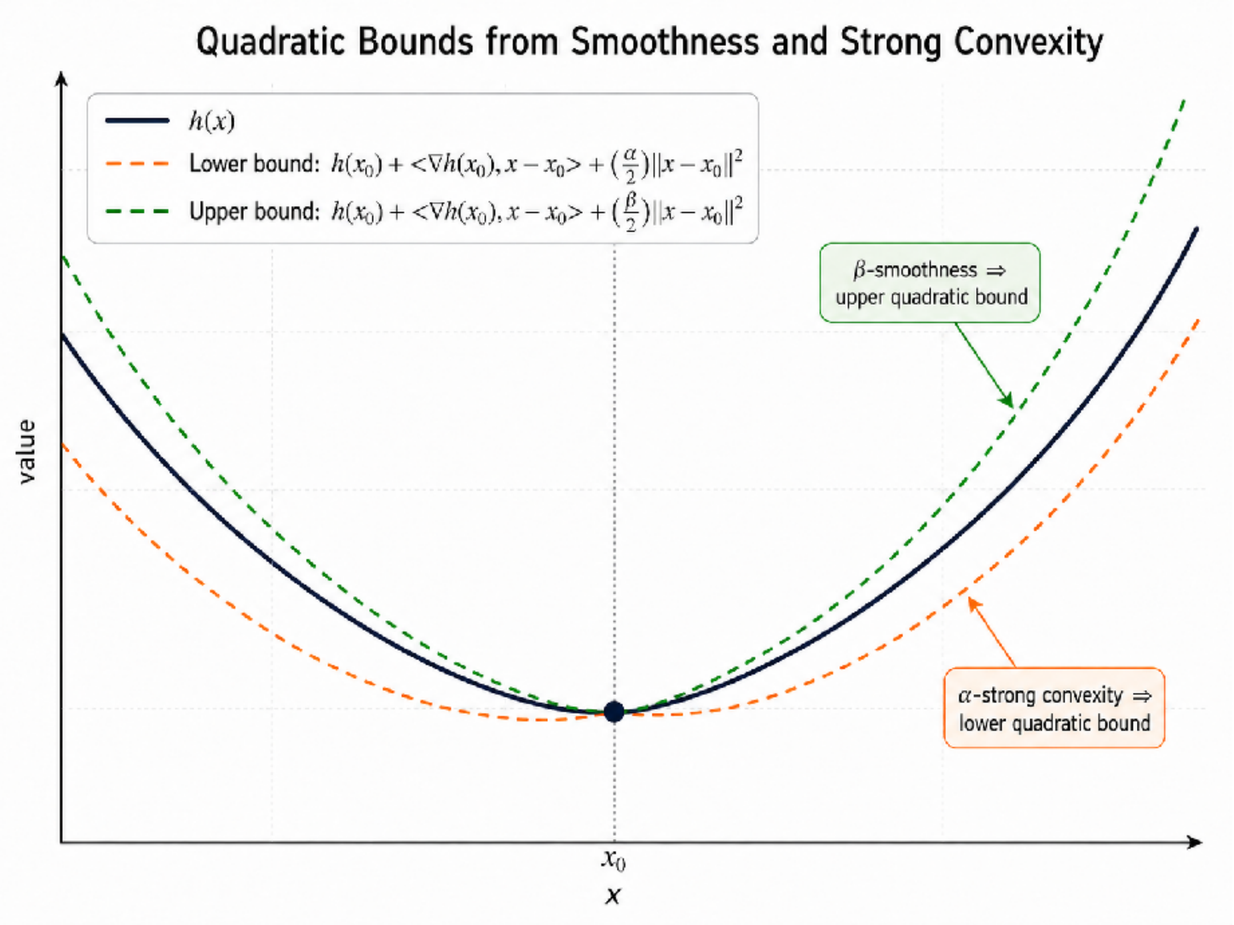
A differentiable function $h : \mathcal{X} \to \mathbb{R}$

is β-**smooth**

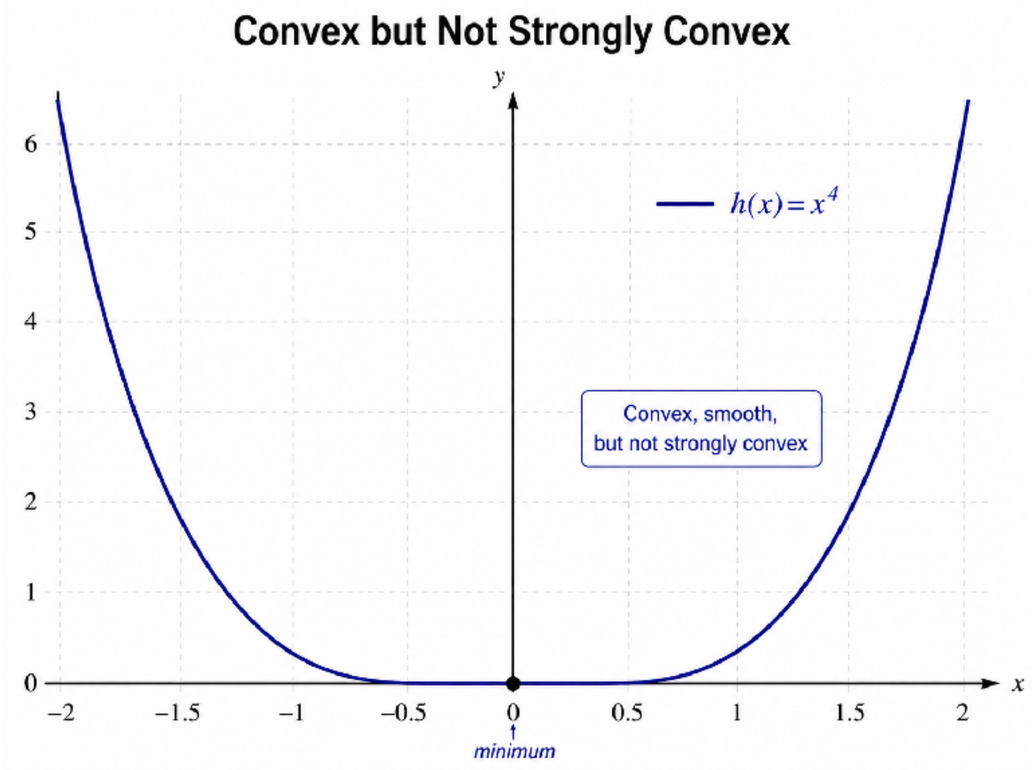
if there is an $\beta > 0$, such that:

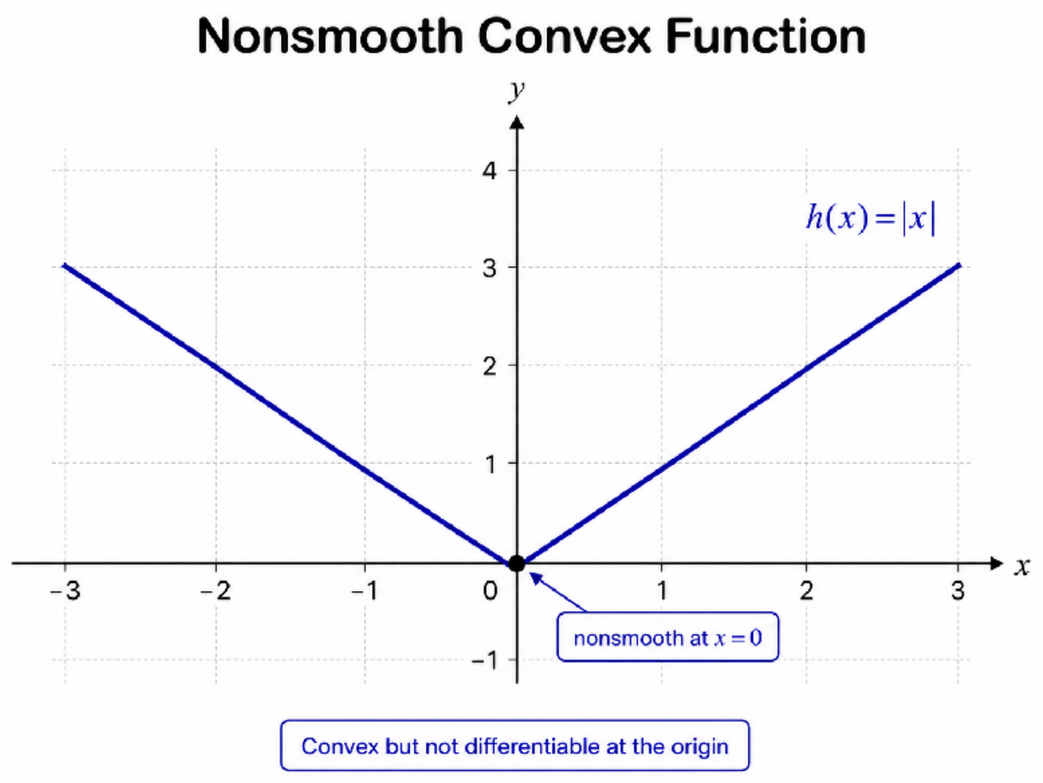
$$h(x) \leq h(y) + \nabla h(y)^\top (x - y) + \frac{\beta}{2}\|x - y\|_2^2,$$

for all $x, y \in \mathcal{X}$.

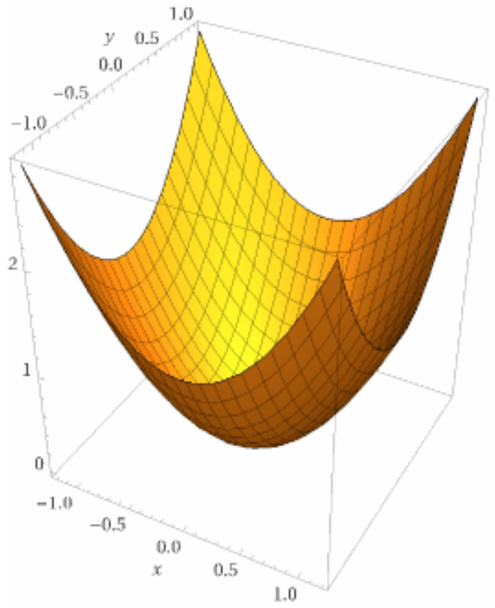


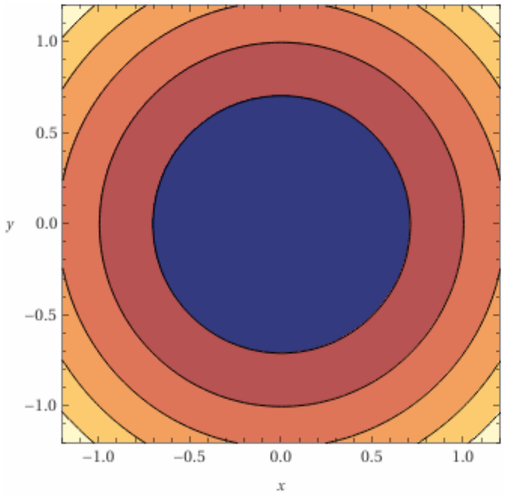
Other Convex Functions





Second-Order Conditions





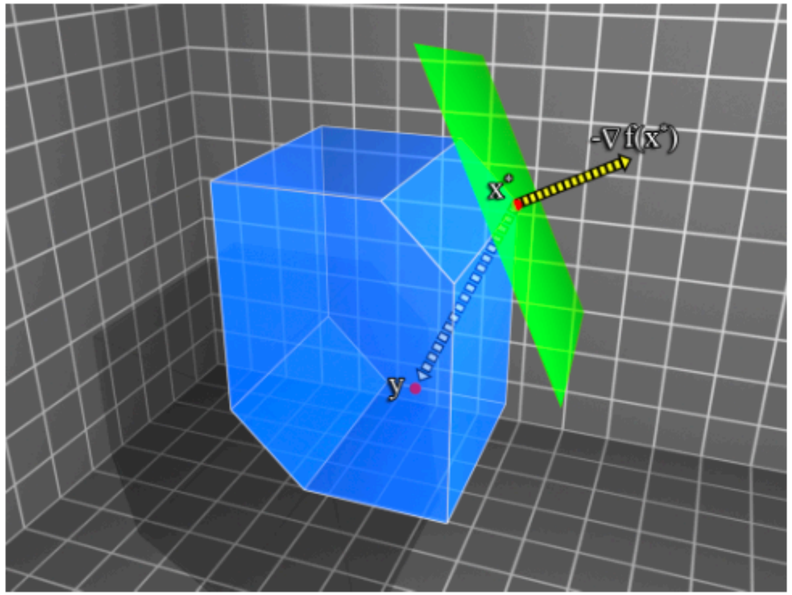
A twice differentiable function $h : \mathcal{X} \to \mathbb{R}$ is α-**strongly** convex, if there is an $\alpha > 0$, such that: $\nabla^2 h(x) \geq \alpha I$, for all $x \in \mathcal{X}$.

A twice differentiable function $h : \mathcal{X} \to \mathbb{R}$ is β-**smooth**, if there is an $\beta > 0$, such that: $\nabla^2 h(x) \leq \beta I$, for all $x \in \mathcal{X}$.

Key Facts about Optimality

- Convex functions over a compact (closed and bounded) domain always admit a global minimizer (may not be unique).
- All local minima are global minima!
- **Strongly** convex functions have a unique global minima.
- We will only consider the case when $\mathcal{X}$ is compact (e.g., polytopes, simplex, (closed) boxes etc.).

Optimality Conditions



A point $x^* \in \mathcal{X}$, is optimal for h if and only if: $\nabla h(x^*)^\top (y - x^*) \geq 0$ for all $y \in \mathcal{X}$.

Why?

Gradient Descent

Initialize with x_0

For $t = 0$ to $T - 1$ do,

$$x_{t+1} = x_t - \eta_t \nabla h(x_t)$$

Return $\bar{x}_T = \text{argmin}_{t=0,1,\ldots,T-1}\{h(x_t)\}$

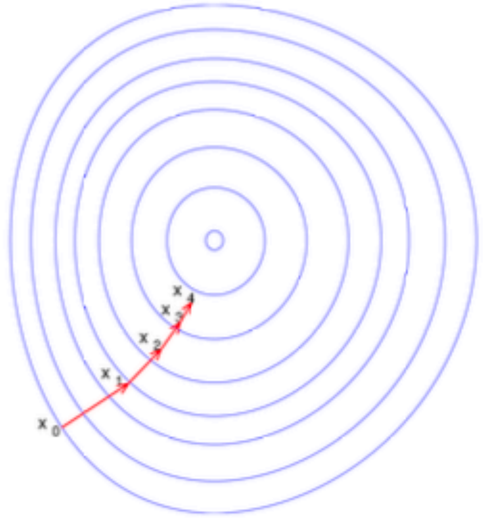


Figure 2.3: Iterates of the GD algorithm

Here we return the **best-iterate**, sometimes, we might consider,

the **last-iterate** x_T or an **average iterate** $\frac{1}{T}\sum_{t=0}^{T-1} x_t$

Convergence Rates of GD

	General	Strongly convex	Smooth	Well-conditioned
GD	$\frac{1}{\sqrt{T}}$	$\frac{1}{\alpha T}$	$\frac{\beta}{T}$	$\exp\left(-\frac{\alpha}{\beta}T\right)$

GD converges to a minimizer, i.e., $h(\bar{x}_T) - h(x^*) \leq O(r(T))$, under an appropriate choice of the sequence of step-sizes $\{\eta_t\}$.

Projected Gradient Descent

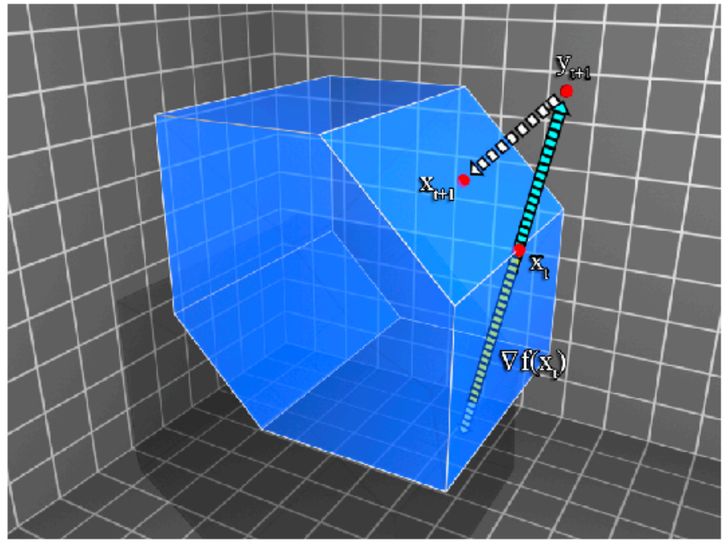
Initialize with $x_0 \in \mathcal{X}$

For $t = 0$ to $T - 1$ do,

$$y_{t+1} = x_t - \eta_t \nabla h(x_t)$$

$$x_{t+1} = \Pi_{\mathcal{X}}(y_{t+1})$$

Return $\bar{x}_T = \text{argmin}_{t=0,1,\ldots,T-1}\{h(x_t)\}$



PGD converges to a minimizer, i.e., $h(\bar{x}_T) - h(x^*) \leq O(r(T))$, with effectively the same rates (wrt dependence on T) and appropriate selection of $\{\eta_t\}$.

Online Convex Optimization

Setup

- The game proceeds in rounds $t = 0,1,\ldots,T$.
- At each iteration t, Player plays $x_t \in \mathcal{X}$ and gets $h_t(x_t)$ as feedback or loss.
- Player does not know future losses!
- We will assume that $\{h_t\}$ are a sequence of loss functions that are in some convex fixed function class (e.g., β-smooth, α-strongly convex etc.).
- But what is the right “optimizer” for the online setting?

Regret Minimization

Regret $\longrightarrow$ $$R(T) = \underbrace{\sum_{t=0}^{T-1} h_t(x_t)}_{\text{Total loss actually obtained during the T rounds.}} - \underbrace{\min_{x \in \mathcal{X}} \sum_{t=0}^{T-1} h_t(x)}_{\text{Total loss of the best fixed point (in hindsight)}}$$

Player's goal is to use $\{x_t\}_{t=0}^{T-1}$ such that: $\lim_{T \to \infty} \frac{R(T)}{T} = 0$

No-regret Algorithm: Update rules for x_t, that satisfy $R(T) = o(T)$.

Online Gradient Descent

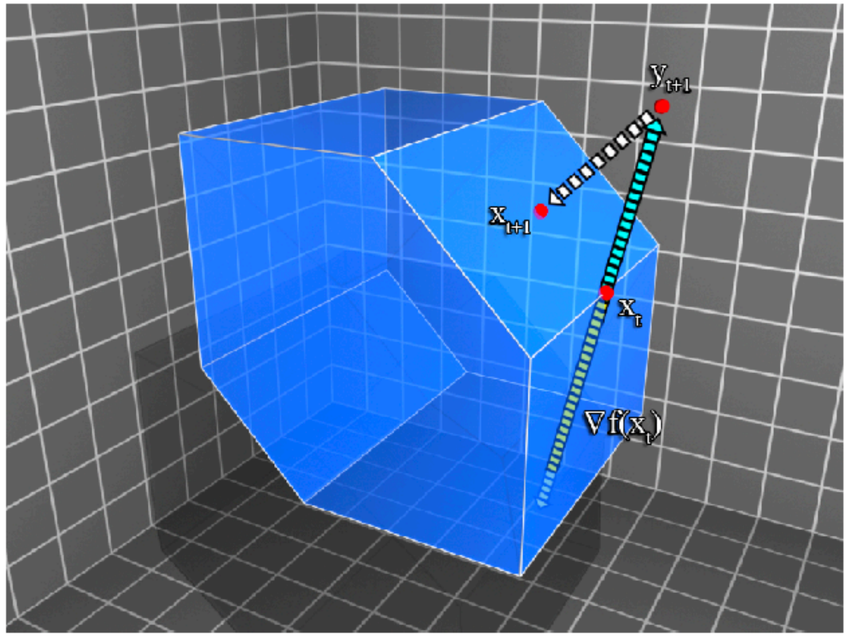
Initialize with $x_0 \in \mathcal{X}$

For $t = 0$ to $T - 1$ do,

$$y_{t+1} = x_t - \eta_t \nabla h_t(x_t)$$

$$x_{t+1} = \Pi_{\mathcal{X}}(y_{t+1})$$

Return x_T



OGD satisfies $R_T = o(T)$, with appropriate selection of $\{\eta_t\}$ and assuming uniformly bounded norm of gradients.

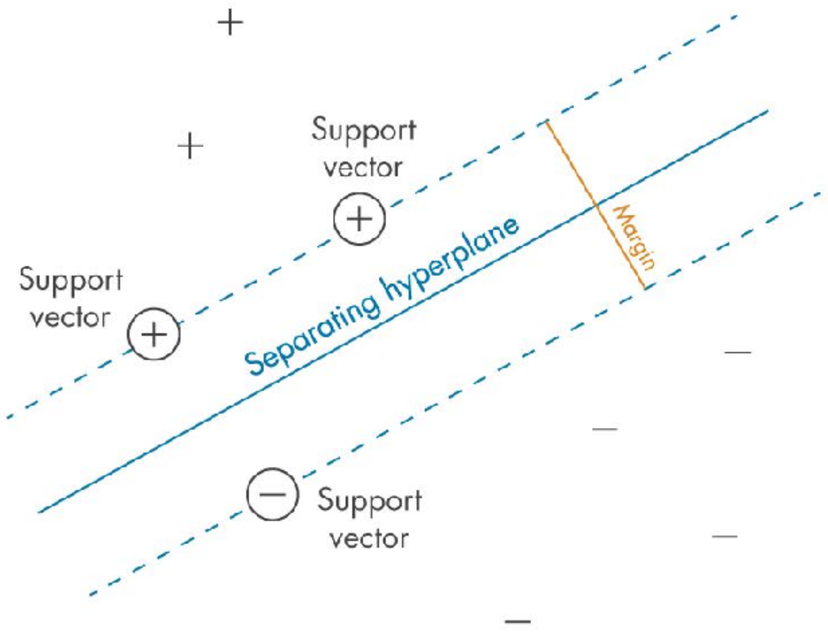
Regret of OGD

	General	Strongly convex	Smooth
Average Regret	$\frac{1}{\sqrt{T}}$	$\frac{\log T}{T}$	$\frac{1}{\sqrt{T}}$

SVM in Machine Learning

Suppose $x_t \in \mathcal{X}$ is a feature vector and $y_t \in \{-1, 1\}$ is the label such that $\{(x_t, y_t)\}$ are **i.i.d samples** from a fixed (but unknown) distribution. Let there be T data points.

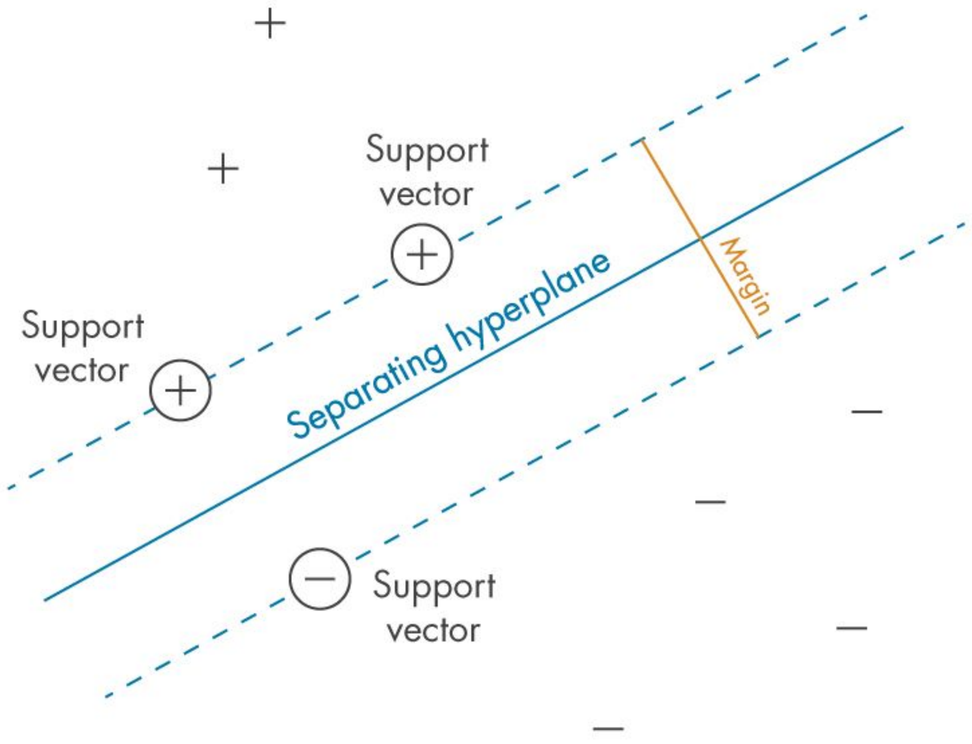
Goal: Find **a separating hyperplane** that minimizes the average hinge-loss: $h_T(w) = \frac{1}{T} \sum_{t=0}^{T-1} h(w; (x_t, y_t))$.



Support Vector Machines - Online Version

Suppose $x_t \in \mathcal{X}$ is a feature vector and $y_t \in \{-1, 1\}$ is the label that are revealed at every iteration (say by an **adversary**). Let there be T data points.

Goal: Find a sequence of hyperplanes that ensures sublinear regret against any set of T data points!



Hinge Loss

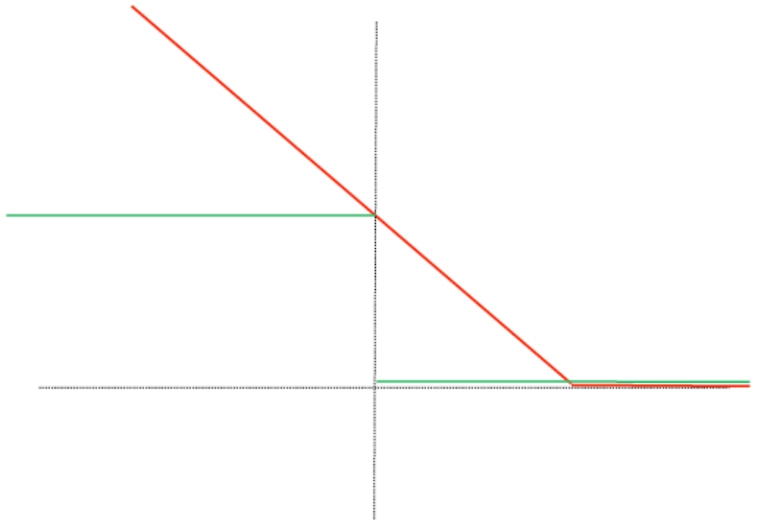


Figure 2.4: The hinge loss function versus the 0/1 loss function

SVM classifier on a data point (x, y) should satisfy, $y\big(\text{sign}(w^\top x)\big) \geq 1.$

Hinge loss: $h(w; (x, y)) = \max\{0, 1 - y(w^\top x)\}$

Mapping to Online Learning

- The game proceeds in rounds $t = 0,1,\ldots,T$.
- At each iteration t, Player plays $w_t \in \mathcal{X} := \{w : \|w\|_2 \leq B\}$ and gets $h_t(w_t) := h(w_t; (x_t, y_t))$ as feedback or loss.
- We will also assume $\|x_t\|_2 \leq G$, for all $t = 0,1,\ldots,T-1$.
- Player does not know future losses!
- The above sequence of loss functions are convex! But are they strongly convex or smooth?

Recall Online Gradient Descent

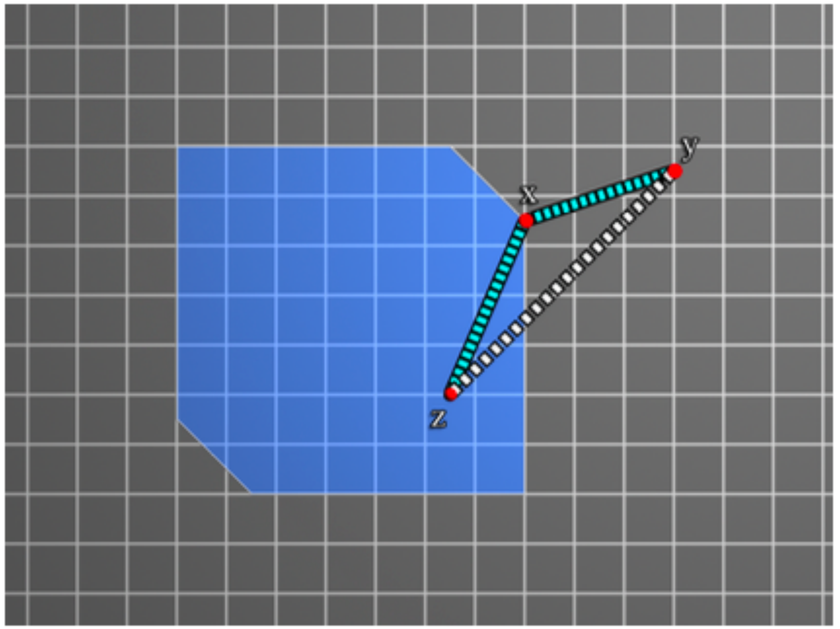
Initialize with $x_0 \in \mathcal{X}$

For $t = 0$ to $T - 1$ do,

$$y_{t+1} = x_t - \eta_t \nabla h_t(x_t)$$

$$x_{t+1} = \Pi_{\mathcal{X}}(y_{t+1})$$

Return x_T



OGD satisfies $R_T = o(T)$, with appropriate selection of $\{\eta_t\}$ and assuming uniformly bounded norm of gradients.

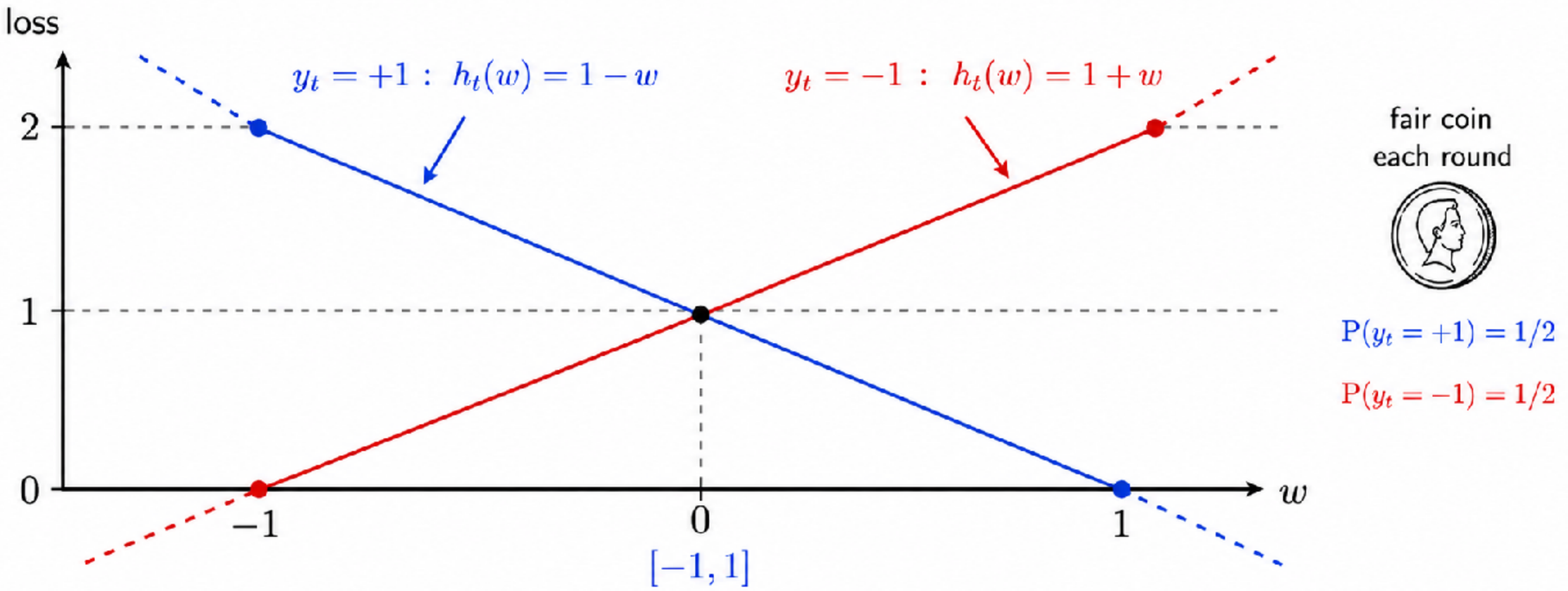
Online Gradient Descent for SVM

Under the stated assumptions, running OGD with step-size $\eta_t = \eta = B/(G\sqrt{T})$ for the online SVM problem produces a sequence of $\{w_t\}_{t=0}^{T-1}$ such that:

$$\sum_{t=0}^{T-1} h_t(w_t) - \min_{w \in \mathcal{X}} \sum_{t=0}^{T-1} h_t(w) \leq BG\sqrt{T}$$

Why?

Can we do better for SVM?

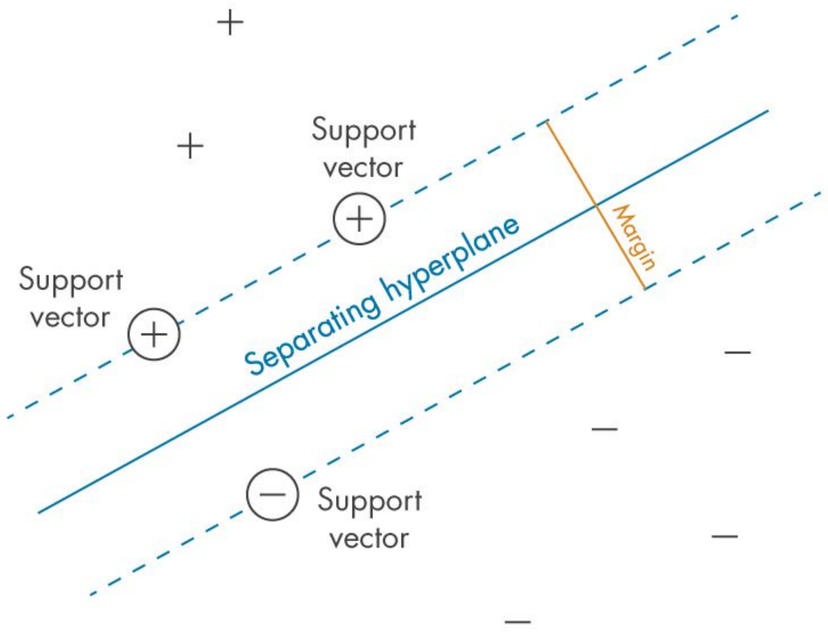


Suppose $x_t = 1$, for all t and $y_t \in \{-1,1\}$ is the label chosen by tossing a fair coin at every iteration. Let there be T data points. Then $\mathbb{E}[R(T)] = \Omega(\sqrt{T})$.

SVM in Machine Learning

Suppose $x_t \in \mathcal{X}$ is a feature vector and $y_t \in \{-1, 1\}$ is the label such that $\{(x_t, y_t)\}$ are **i.i.d samples** from a fixed (but unknown) distribution. Let there be T data points.

Goal: Find **a separating hyperplane** that minimizes the average hinge-loss: $h_T(w) = \frac{1}{T} \sum_{t=0}^{T-1} h(w; (x_t, y_t))$.



Stochastic Gradient Descent via OGD

If the (noisy) gradient at every step is **unbiased** and the **variance of the norm** of the gradient is bounded, then running OGD with these gradients and $\eta = B/(G\sqrt{T})$ leads to :

$$\mathbb{E}[h_T(\bar{w}_T)] \leq \min_{w \in \mathcal{X}} h_T(w) + 3GB/(2\sqrt{T}), \text{ where } \bar{w}_T = \sum_{t=0}^{T-1} w_t \,.$$

1. **Unbiased gradient:** $\nabla h(w) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\nabla h(w;(x,y))]$
2. **Gradient Norm Variance:** $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\|\nabla h(w;(x,y))\|_2^2] \leq G^2$

See Theorem 3.4 in Hazan's book for the proof.

Summary

- Online learning is a powerful framework for optimization against the unknown.
- We saw OGD and the guarantees for online convex optimization.
- We looked at an online SVM example and why OGD is pretty much the best you can do.
- We saw that SGD can be seen as a special case of OGD. But in the upcoming classes we will treat it a separate algorithm.
- Next class: Distributed learning