Lecture 1

- Logistics
- What is Optimization?
- What is Machine Learning (ML)?
- Examples of ML formulated as optimization problems

what is Optimization?

$\mathcal{X}$ – parameter space ; e.g. a model space for ML

$J : \mathcal{X} \to \mathbb{R}$: objective

Goal:

$$x^* = \underset{x \in \mathcal{X}}{\operatorname{argmin}} \; J(x)$$

x^* – "best" solution

Because we can equivalently find x^* as the solution to a maximization problem

$$x^* = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \; -J(x)$$

the difficulty of maximization & minimization <u>in general</u> is the same.

Challenges of optimization

- dealing with constraints
- ambiguity in matching the task w/ a specific objective function
- these problems are extremely difficult when there isn't a specific structure we can exploit
- scale of the problem

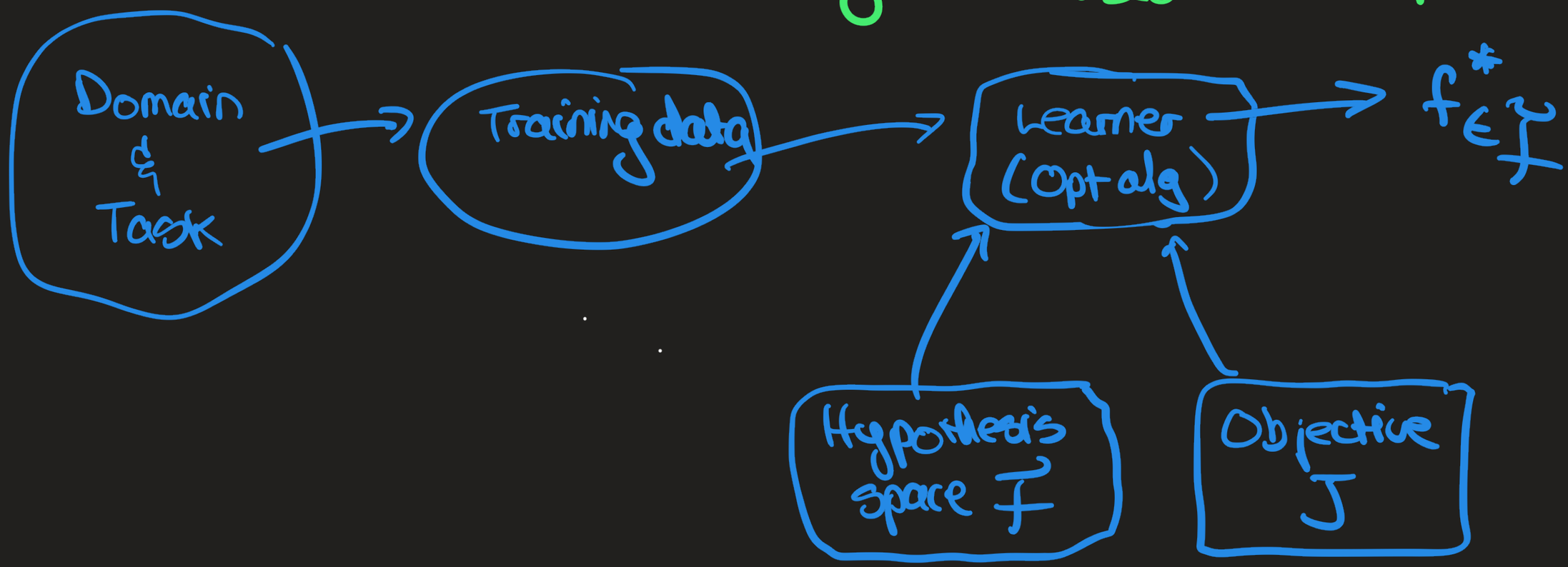
what is ML?

" finding a predictable pattern from a set of data "

predictable → accurate

pattern → $f(x) = y$

set of data → observations used to learn f



What does it mean for f to be optimal?

ℓ - loss function $\quad \ell(\cdot, \cdot)$

$f(x)$ — prediction of a candidate model f for input x

y — true label/value for that input

ℓ - quantify the mismatch b/w the prediction & the ground truth

$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \; \mathbb{E}_{\mathbb{P}} \, \ell(f(x), y)$$

called the population risk

$(x, y) \sim \mathbb{P}$ in our domain D

$$\approx \underset{f \in \mathcal{F}}{\operatorname{argmin}} \; \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

called the empirical risk

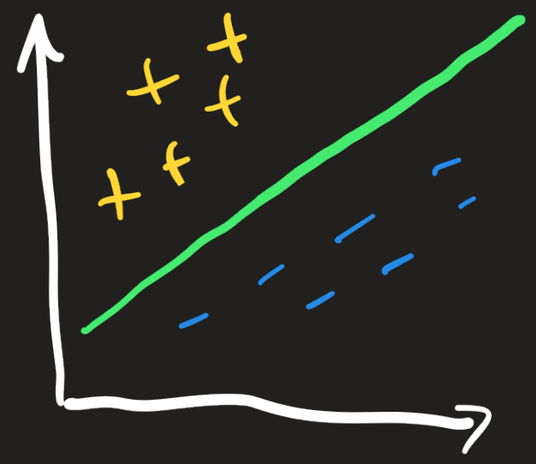
where (x_i, y_i) for $i = 1, \dots, n$ are training pairs, sampled i.i.d. from $\mathbb{P}$

High-level overview of ML

- formulate a task & pick a domain
- collect appropriate data
- choose an appropriate model class for your problem
- pick a loss function and formulate our optim prob to find f^*
- solve the optim prob

(setup for classical ML that focuses only on accuracy)

Ex Binary classification using SVMs (support vector machines)



Assumption: the two classes are linearly separable

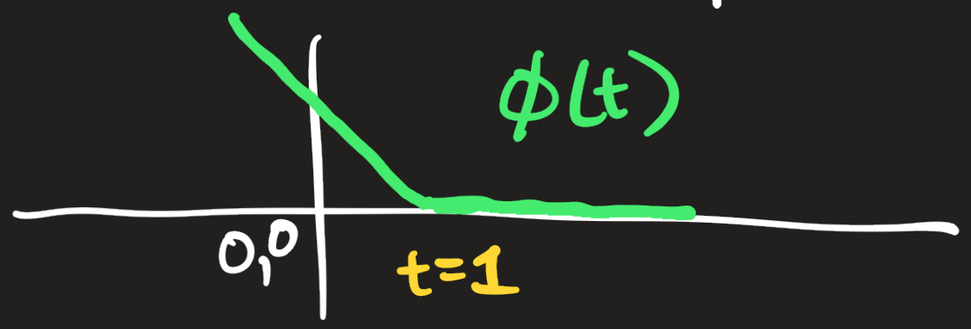
$$\mathcal{F} = \{ f : f(x) = \text{sign}(w^T x + b) \text{ for } w \in \mathbb{R}^d \text{ and } b \in \mathbb{R} \}$$

To design our loss for fitting an SVM:

- penalize our model if y_i and $w^T x_i + b$ have different signs (means $\ell(w^T x_i + b, y_i)$ should be high)
- penalize the model if $y_i (w^T x_i + b) < 1$ (we want the model to predict a large number w/ the correct sign)

- avoid trivially confident models
(do not let $\|w\|_2 \to \infty$)

Let $\phi(t) = (1-t)_+ = \max(1-t, 0)$



we call C, λ hyperparameters

Take $\ell(f(x_i), y_i) = \phi(y_i(\langle w, x_i \rangle + b))$

Two setups for the optimization problem of learning an SVM using hinge-loss:

1) $w^*, b^* = \underset{w,b \,:\, \|w\|_2 \le C}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \phi(y_i(\langle w, x_i \rangle + b))$

2) $w^*, b^* = \underset{w,b}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \phi(y_i(\langle w, x_i \rangle + b)) + \lambda \|w\|_2^2$

Ex Ordinary Least Squares Regression

Given a continuous target $y \in \mathbb{R}$, the task is to predict y as a linear combination of the features $x \in \mathbb{R}^d$

e.g. y = life expectancy

x = vector of demographic descriptors

Assume that there are true coefficients $\beta \in \mathbb{R}^d$ so that

$$y = \langle x, \beta \rangle + b = \langle [x, 1], [\beta\ b] \rangle$$

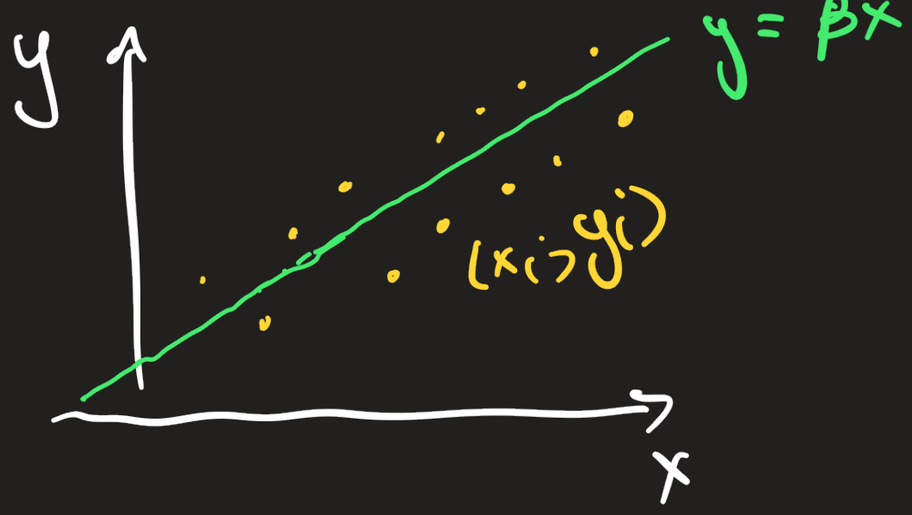
$$= \langle x_{aug}, \beta_{aug} \rangle$$

(WLOG assume we have augmented linear models w/ a bias term)

By augmenting we simplify to

$$y = \langle x, \beta \rangle$$

But our training data is corrupted by noise



$$y_i = \langle \beta^*, x_i \rangle + \varepsilon_i$$

for $i = 1, \dots, n$ where

$$\varepsilon \sim \mathcal{N}(0, \frac{1}{n})$$

Q: how can we recover β?

Alg for computing $\beta_{ols} \approx \beta^*$

$$\beta_{ols} = \underset{\beta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (\langle x_i, \beta \rangle - y_i)^2$$

Alg for computing β_{OLS} (Ordinary Least Squares)

Notice

$$\frac{1}{n}\sum_{i=1}^{n}(\langle x_i, \beta\rangle - y_i)^2 = \frac{1}{n}\|y - X\beta\|_2^2$$

where

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \text{and} \quad X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$$

and note β minimizes the objective $J(\beta) = \frac{1}{n}\|y - X\beta\|_2^2$ iff

$$\nabla_\beta J(\beta) = 0 = \frac{1}{n} X^T(X\beta - y) = 0$$

so we have

$$X^T(X\beta_{OLS} - y) = 0 \quad \text{characterizes } \beta_{OLS}$$

$$\Rightarrow \beta_{OLS} = (X^TX)^{-1}X^Ty$$