Important Announcements and Notes (Sep 26)

- $\mathcal{D} = \left((\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n) \right) \in (\mathcal{X} \times \mathcal{Y})^n$

  if $\mathcal{X} = \mathbb{R}^d$ then $\vec{x}_1 \in \mathcal{X}$

  $\vec{x}_1 = (x_{11}, x_{12}, \dots, x_{1d})^T$, $\vec{x}_n = (x_{n1}, x_{n2}, \dots x_{nd})^T$

- $f(\vec{x})$ where $\vec{x} \in \mathcal{X}$

  $\vec{x} = (x_1, x_2, \dots, x_d)^T$

  $x_1 \neq \vec{x}_1$, $x_2 \neq \vec{x}_2$

Important Announcements and Notes (Sep 24)

- The predictor output by the Learner will be $\hat{f}$ from now on. $\mathcal{A}(D) = \hat{f}$
- The dataset $D = ((\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n))$ is used by the Learner to output a predictor $\hat{f}$
- The predictor $\hat{f}(\vec{X})$ takes as input any $\vec{X} \in \mathcal{X}$

$\vec{X}$ and $\vec{X}_i$ are different r.v.

- if $X \in \mathcal{X}$ is a r.v. then it has a distribution $\mathbb{P}$
- otherwise $x \in \mathcal{X}$ is an: outcome, or
  instance of X, or
  a fixed value of X

$X \in \mathcal{X}$ and $x \in \mathcal{X}$ seem the same because we are being imprecise and X is actually a special function

$\mathbb{P}(X \in E)$ is valid $\mathbb{P}(x \in E)$ is not valid

$\mathbb{E}[X]$ is valid $\mathbb{E}[x]$ is not valid

$f(X) = X^2$ is a r.v. $f(x) = x^2$ is not a r.v.

- $D \in (\mathcal{X} \times \mathcal{Y})^n$ is a r.v. with distribution $\mathbb{P}_D$ representing the dataset
- $\mathcal{D} \in (\mathcal{X} \times \mathcal{Y})^n$ is a fixed dataset (an instance of D)

$\mathcal{A}(D)$ is a r.v. $\mathcal{A}(\mathcal{D})$ is not a r.v.

Supervised Learning: Learning from a randomly sampled batch of labeled data

What does Learning well mean?
i.e. What is the objective of Learning?

Dataset (tuple of tuples) $\xrightarrow{\text{Learner (function)}}$ Predictor = Model = Hypothesis (function)

$D = ((\vec{X}_1, Y_1), (\vec{X}_2, Y_2), \ldots, (\vec{X}_n, Y_n))$, $\quad \vec{X}_1 = (X_{1,1}, \ldots, X_{1,d})^T$

$D \in (\mathcal{X} \times \mathcal{Y})^n$

$(\vec{X}_i, Y_i) \sim \mathbb{P}_{\vec{X},Y}$

independent for all $i \in \{1, \ldots, n\}$

D n feature-label pairs

$\mathcal{X}$ set of features

$\mathcal{Y}$ set of labels/targets

Ex: $\mathcal{X} = \mathbb{R}^2$, $\mathcal{Y} = \mathbb{R}$

$\boxed{\hat{f} : \mathcal{X} \to \mathcal{Y}}$

f a function from features to labels

Ex $\hat{f}(x) = 2x + 1$, $\mathcal{X} = \mathbb{R}$

$\boxed{\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to \{f \mid f : \mathcal{X} \to \mathcal{Y}\}}$

$\mathcal{A}$ a function from datasets to predictors

Ex: $\mathcal{A}(D) = \hat{f}$ where $\hat{f} : \mathcal{X} \to \mathcal{Y}$

$$\hat{f}(x) = \begin{cases} y_i & \text{if } x = x_i \text{ for some } i \in \{1, \ldots, n\} \\ 0 & \text{otherwise} \end{cases}$$

Setting:

We are given a random dataset of size n

$$D = ((\vec{X}_1, Y_1), \dots, (\vec{X}_n, Y_n)) \in (\mathcal{X} \times \mathcal{Y})^n$$

where $(\vec{X}_i, Y_i) \sim \mathbb{P}_{\vec{X},Y}$ are independent for all $i \in \{1, \dots, n\}$

$\vec{X}_i$ feature vector

Y_i label or target

$\boxed{\text{We will always assume the features are vectors.}}$

Ex (of features and labels/targets):

$\vec{X}_i \in \mathbb{R}^3$ # of rooms, # of floors, age of a house

$Y_i \in \mathbb{R}$ price

$\vec{X}_i \in \mathbb{R}^2$ amount of chemical 1, amount of chemical 2 in a wine

$Y_i \in \{0,1\}$ type of wine

$(1, \dots, 70)^T$

$\vec{X}_i \in \mathbb{R}^{400}$ pixel value of a $20 \times 20 = 400$ pixel image

$Y_i \in \{\text{cat}, \text{dog}, \text{bird}\}$ type of animal

What is a feature and what is a label is a design choice. Usually a feature is info that is easy to gather. And the label is hard, which is why you want to predict it

Objective (informal):

Define a learner $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to \{f \mid f : \mathcal{X} \to \mathcal{Y}\}$ such that the predictor $\hat{f}$ is good.

Where $\mathcal{A}(D) = \hat{f}$

What is a good predictor $f : \mathcal{X} \to \mathcal{Y}$?

Forget about the dataset D for now. We just want to study a predictor f

Ex: $f(\vec{x})$ is a predictor that takes as input the # of room $\vec{x}$ and outputs a price

Suppose we are given the # of rooms $\vec{x} = 2$ of a house (might not be in D)

We are not given the price $Y = 300$

What would a good predictor $f(\vec{x})$ do?

Ans: $f(2) = 300$

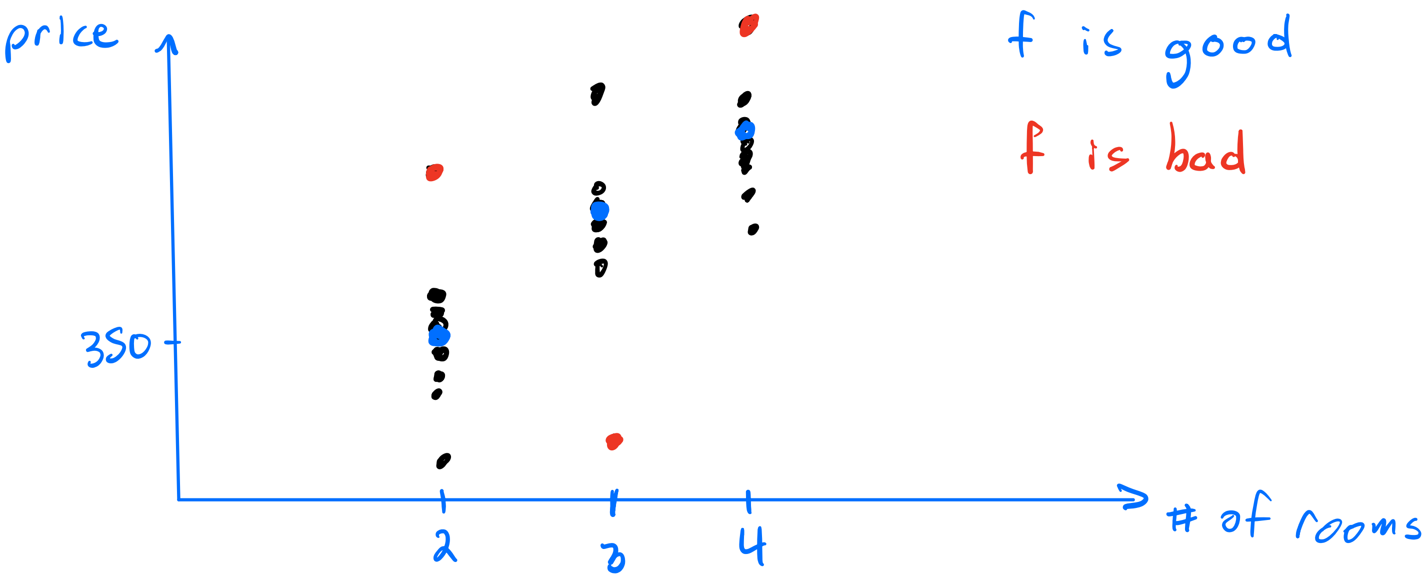
What if you were given another house?

# of rooms = 2 price = 400

what would a good f be?

Ans: maybe $f(2) = \frac{400+300}{2} = 350$

What if we got even more houses?



$\vec{X}, Y$ are random which means they can potentially be any feature-label pair

Ans: We will care about $f(\vec{X})$ being good on average

How do we measure how close $f(\vec{X})$ is to Y?

Ans: We use a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$

The choice of ℓ depends on your problem

Regression: $Y \in \mathcal{Y}$ represent something with a notion of order

(Usually $\mathcal{Y}$ is $\mathbb{R}$ or some interval)

Ex: house prices, stock prices, energy consumption, weather prediction

We use:

$$\ell(f(\vec{X}), Y) = |f(\vec{X}) - Y| \quad \text{absolute loss}$$

or

$$\ell(f(\vec{X}), Y) = (f(\vec{X}) - Y)^2 \quad \text{squared loss}$$

A good predictor f should have a small loss ℓ on average (expectation)

$$L(f) = \mathbb{E}\left[\ell(f(\vec{X}), Y)\right]$$

$$(\vec{X}, Y) \sim P_{\vec{X}, Y}$$

Ex: (squared loss)

$$\mathbb{E}\left[\ell(f(\vec{X}), Y)\right] = \int_{\mathcal{X}} \int_{\mathcal{Y}} (f(\vec{x}) - y)^2 \overbrace{p(\vec{x}, y)\, dy\, d\vec{x}}^{p(y|x)\, p(x)\, dy\, dx}$$

$$= \int_{\mathcal{X}} \left(\int_{y} (f(\vec{x}) - y)^2 p(y|\vec{x}) dy \right) p(\vec{x}) dx$$

Objective (almost formal):

Define a learner $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to \{f \mid f : \mathcal{X} \to \mathcal{Y}\}$

such that $L(\hat{f})$ is small

where $\mathcal{A}(D) = \hat{f}$

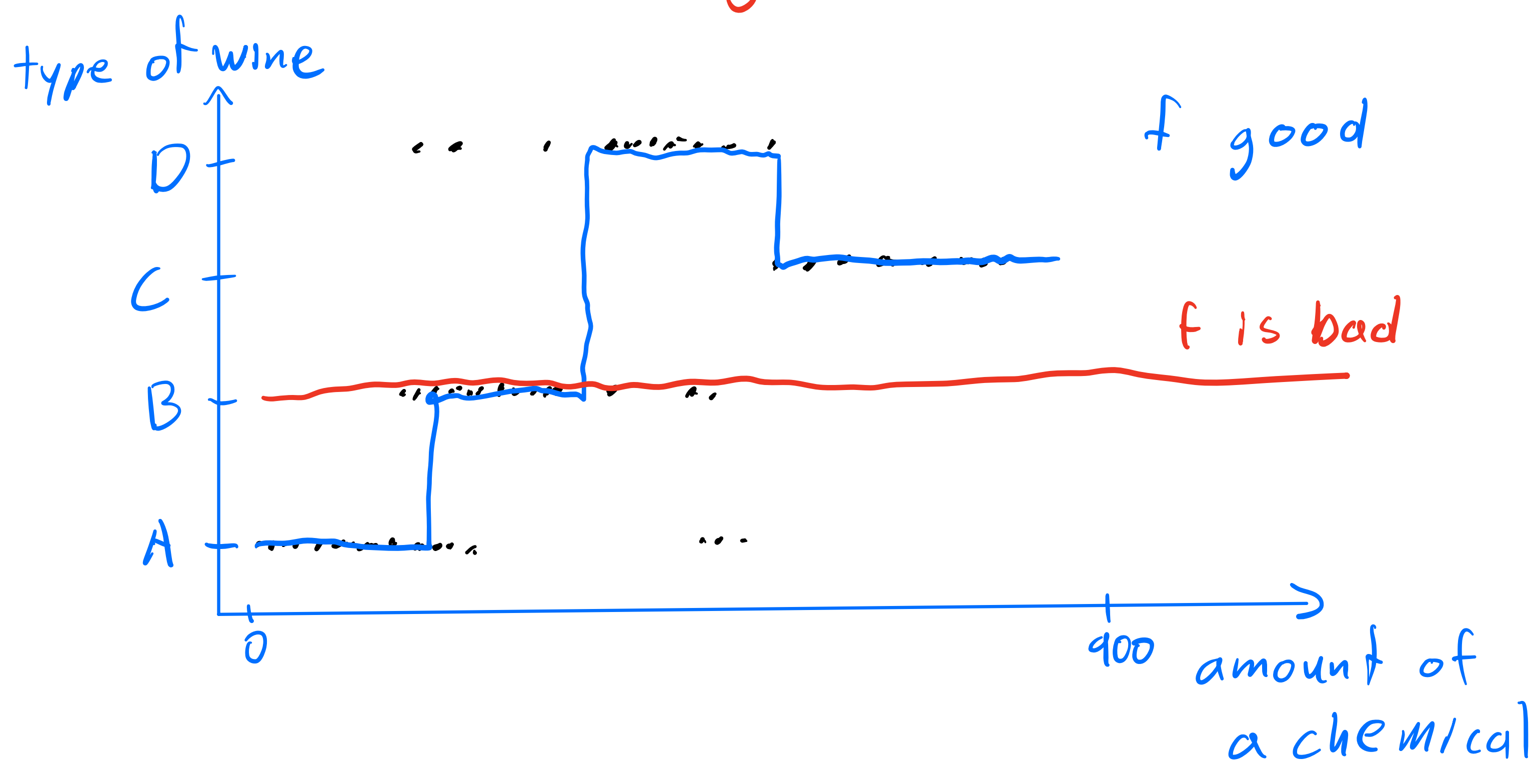
Classification: if $Y \in \mathcal{Y}$ represents something without order

(Usually $\mathcal{Y}$ is finite)

Ex: type wines, type of image, type of email, type of disease

Ex: $f(\vec{x})$ is a predictor that takes as input the amount of a chemical in a wine and outputs the type of wine

Suppose you got multiple wines, what would a good f be



for ℓ we use:

$$\ell(f(\vec{X}),Y) = \begin{cases} 0 & \text{if } f(\vec{X}) = Y \\ 1 & \text{otherwise} \end{cases}$$

0-1 loss

Ex: $L(f)$ if we use 0-1 loss $\quad \mathcal{Y} = \{A, B, C, D\}$

$$L(f) = \mathbb{E}\left[\ell(f(\vec{X}),Y)\right] = \int_{\mathcal{X}} \sum_{y \in \mathcal{Y}} \ell(f(\vec{x}),y)\, p(x,y)\, dx$$

$$= \int_{\mathcal{X}} \left(\sum_{y \in \mathcal{Y}} \ell(f(\vec{x}),y)\, p(y|x) \right) p(x)\, dx$$

Objective (almost formal):

Define a learner $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \Rightarrow \{f \mid f : \mathcal{X} \Rightarrow \mathcal{Y}\}$

such that $L(\hat{f})$ is small

where $\mathcal{A}(D) = \hat{f}$

D is random! It can potentially be any dataset

If D changes then $\hat{f}$ also changes.

Can $\mathcal{A}(D) = \hat{f}$ be good for all values of D

Ans: No you can't. There is a tradeoff.

Instead we will care about $\mathcal{A}(D) = \hat{f}$ being good on average (expectation) over datasets

$$\mathbb{E}\left[L(\mathcal{A}(D))\right]$$

Objective (formal):

Define a learner $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to \{f \mid f : \mathcal{X} \to \mathcal{Y}\}$

such that $\mathbb{E}[L(\mathcal{A}(D))]$ is small

First we will assume we have a fixed Dataset $D = \mathcal{D}$ (not random) and see how to define $\mathcal{A}(\mathcal{D}) = \hat{f}$ such that $L(\hat{f})$ is small

Our Approach:

Let f^* be the f that minimizes $L(f)$

So $\mathcal{A}(\mathcal{D}) = \hat{f} = f^*$

We don't know what $L(f)$ is for any f

Since we don't know $\mathbb{P}_{X,Y}$

risk $\quad L(f) = \mathbb{E}\left[\, \ell(f(\vec{X}), Y) \right]$

Defining $\mathcal{A}(D)$: Empirical Risk Minimization (ERM)

Estimation:

Use D to estimate $L(f)$ for all $f \in \mathcal{F} \subset \{f \mid f: \mathcal{X} \to \mathcal{Y}\}$

call the estimate $\hat{L}(f)$

Optimization:

pick $\hat{f}$ to be the $f \in \underset{\text{Function class}}{\mathcal{F}}$ that minimizes $\hat{L}(f)$

Ex: Let $\mathcal{F}$ be all linear functions

ERM picks the line that best fits the data