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

What does learning well mean?

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

$\vec{x}_1 = (x_{1,1}, \dots, x_{1,d})^T$

$$D = \left((\vec{x}_1, Y_1), (\vec{x}_2, Y_2), \dots, (\vec{x}_n, Y_n)\right)$$

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

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

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

D n feature-label pairs

$\mathcal{X}$ set of features

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

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

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

f a function from features to labels

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

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

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

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

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

Setting

We are given a random dataset

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

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

$\vec{X}_i \in \mathbb{R}^d$ feature vectors

Y_i labels, targets

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

$\vec{X}_i \in \mathbb{R}^{400}$ 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 $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \Rightarrow \{f \mid f : \mathcal{X} \Rightarrow \mathcal{Y}\}$

such that the predictor $\hat{f}$ is good

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

Ex: $f(\vec{x})$ suppose $\vec{x} = 2$ # of rooms

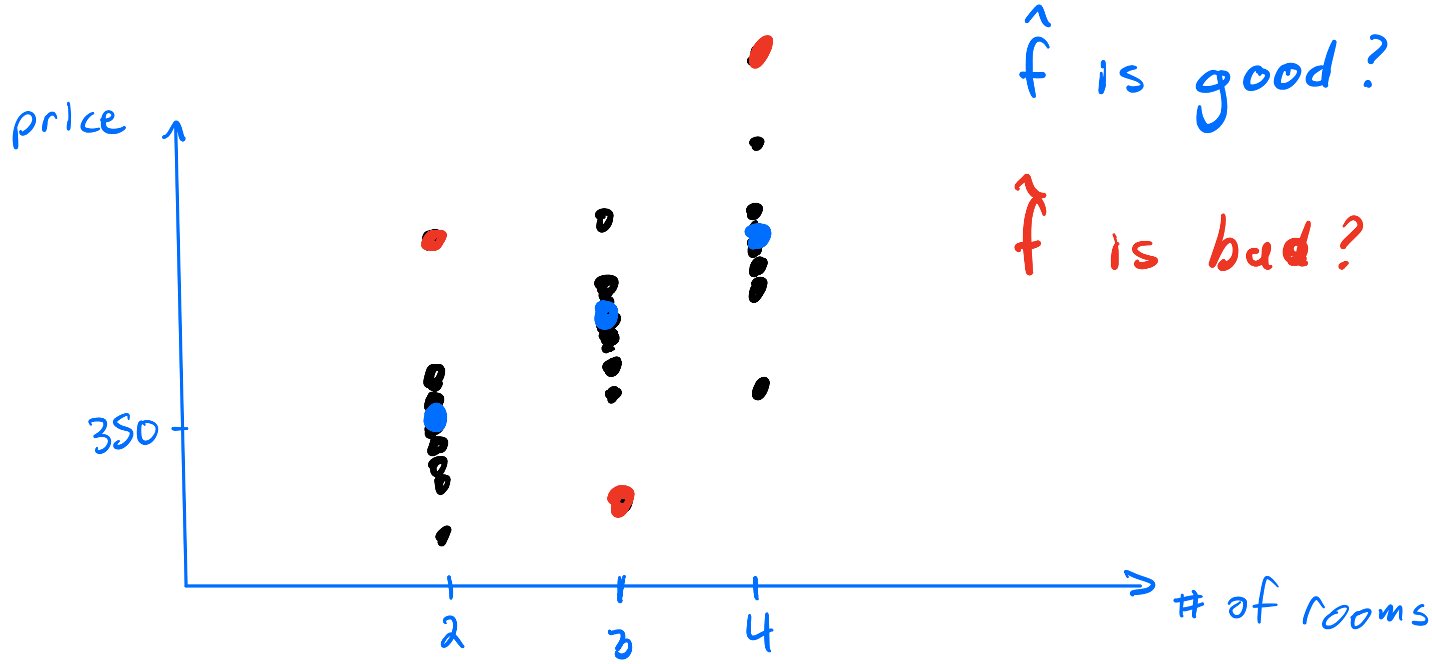
the label is \$300K (predictor doesn't know this)

$f(2) = \$300k$

you get another house with $\vec{x} = 2$

but the label is \$400k

$f(2) = \frac{400 + 300}{2}$?



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

A good predictor $\hat{f}$ should have a small loss in expectation

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

number

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

"Regression"

$$\overset{L(f)}{\overset{\parallel}{\mathbb{E}\left[\ell(\hat{f}(\vec{x}), y)\right]}} = \int_x \int_y \ell(\hat{f}(\vec{x}), y)\, \overbrace{p(\vec{x}, y)}^{p(y|\vec{x})\,p(\vec{x})}\, dy\, d\vec{x}$$

$$= \int_x \left(\int_y \ell(\hat{f}(\vec{x}), y)\, p(y|\vec{x})\, dy \right) p(\vec{x})\, d\vec{x}$$

Regression: $\mathcal{Y}$ 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

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

Objective (Still Informal)

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

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

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

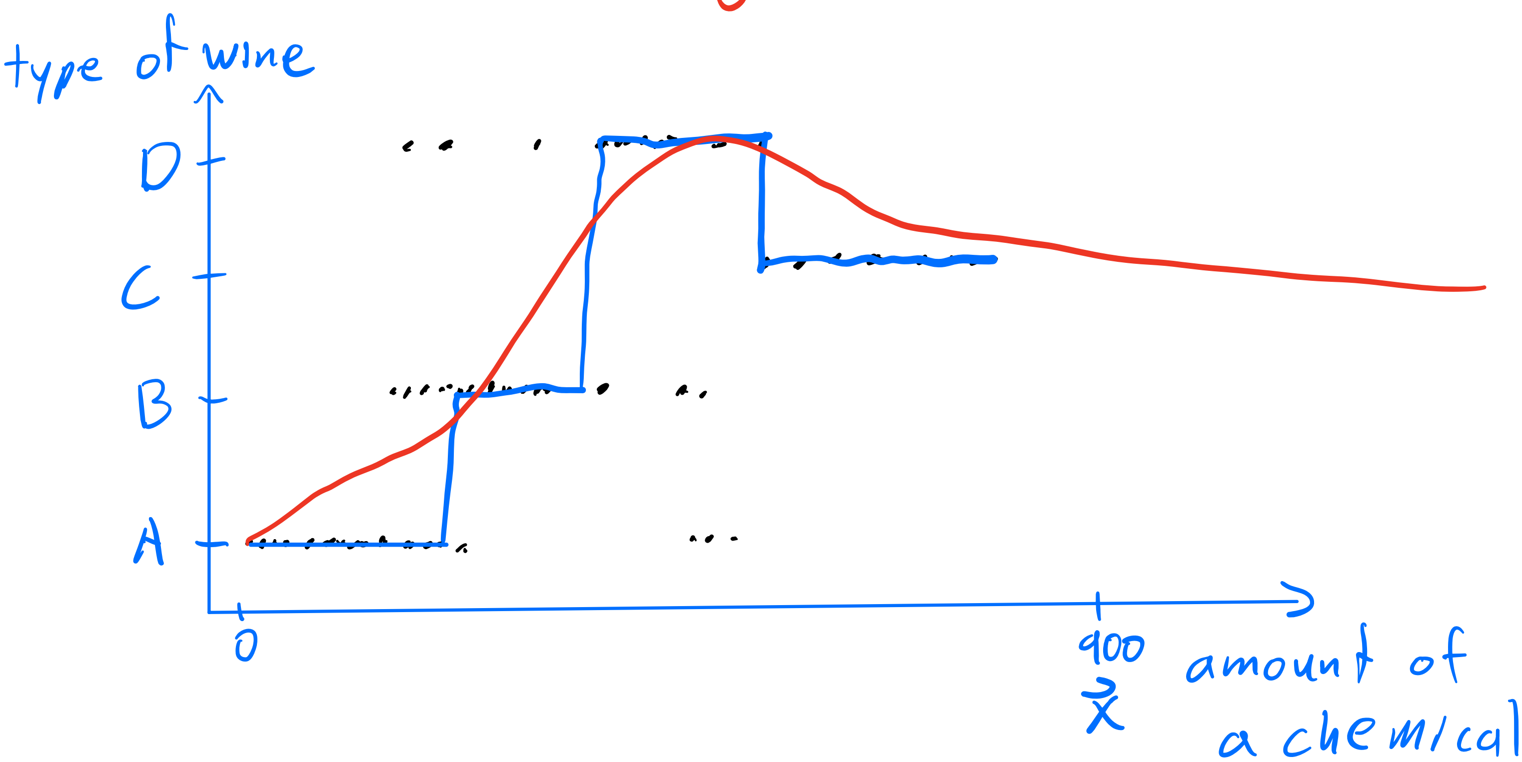
Classification: if $Y \in \mathcal{Y}$ represents something without order

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

Ex (of $\mathcal{Y}$): 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 $\mathcal{Y} = \{A, B, C, D\}$

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

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

Objective (formal)

Define a learner $\mathcal{A}: (\mathcal{X} \times \mathcal{Y})^n \Rightarrow \{f \mid f: \overset{\text{ex: } \mathbb{R}^d}{\mathcal{X}} \Rightarrow \mathcal{Y}\}$

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

For now we will mostly fix a dataset D

D "random dataset"

What is the Learner?

we can't explicitly calculate $L(f)$ for any $f \in \{f \mid f: \mathcal{X} \Rightarrow \mathcal{Y}\}$

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

$L(f)$ "risk of f"

Defining $\mathcal{A}(D)$

Empirical Risk Minimization (ERM)

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

function class

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

Optimization: pick $\hat{f}$ to be the $f \in \mathcal{F}$ that minimizes $\hat{L}(f)$

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

ERM picks the line that best fits the data