1/26/15

SUPERVISED LEARNING

(1) Basic Setup + Notation

→ Input / Feature / Feature-vector / Attributes

$\vec{x} \in \mathbb{R}^d$ (continuous features)
$\in \{0,1\}^d$ (binary features)
} sometimes both

→ Output / Target / Labels

y or $\hat{y} \in \mathbb{R}$ in case of Regression
$\in \{0,1\}$ in case of binary classification
(or maybe $\{-1,+1\}$)

$\in \{1,2,\ldots,k\}$ in case of multiclass Classification

→ Unknown Target Function

$f: X \rightarrow Y$

X: input space
Y: output space

→ Given: Dataset $D = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_N, y_N)\}$
or "Training Data"

$(\vec{x}_1, y_1)$: sample a data-point

→ Goal: Given dataset D of samples $(x_i, f(x_i))$ from an unknown target f, predict $f(x)$ on a NEW point x.

→ [Note: we will use d: # features, n (or N): # samples / data-points]

→ Approach:

① Pick a model / Hypothesis Space

$$H = \{ g : X \to Y \}$$

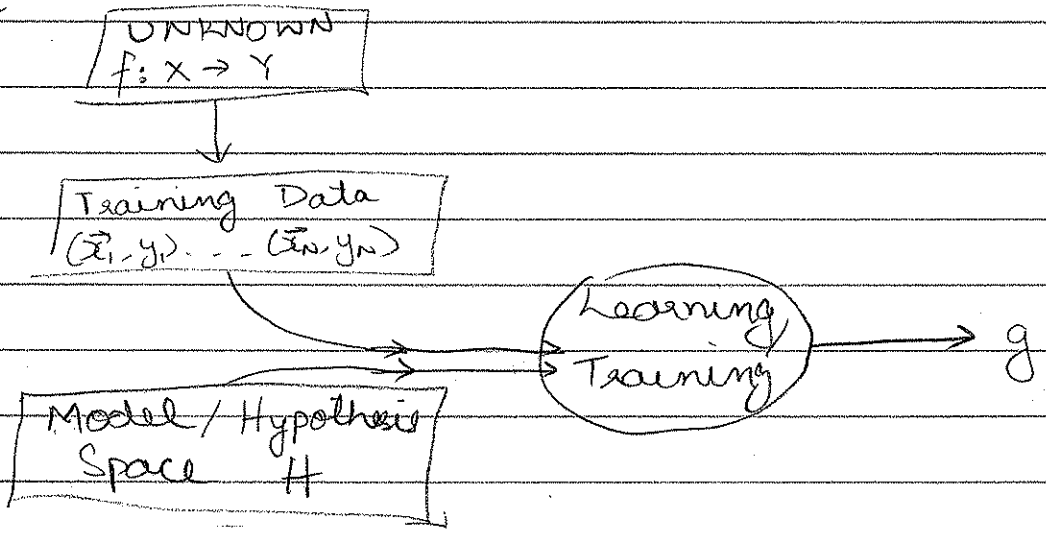
a set of mappings from X to Y

[Usually we will NOT allow all possible mappings. You'll see.]

② Search / Optimize in H to find "best" $g \in H$ that approximates f.

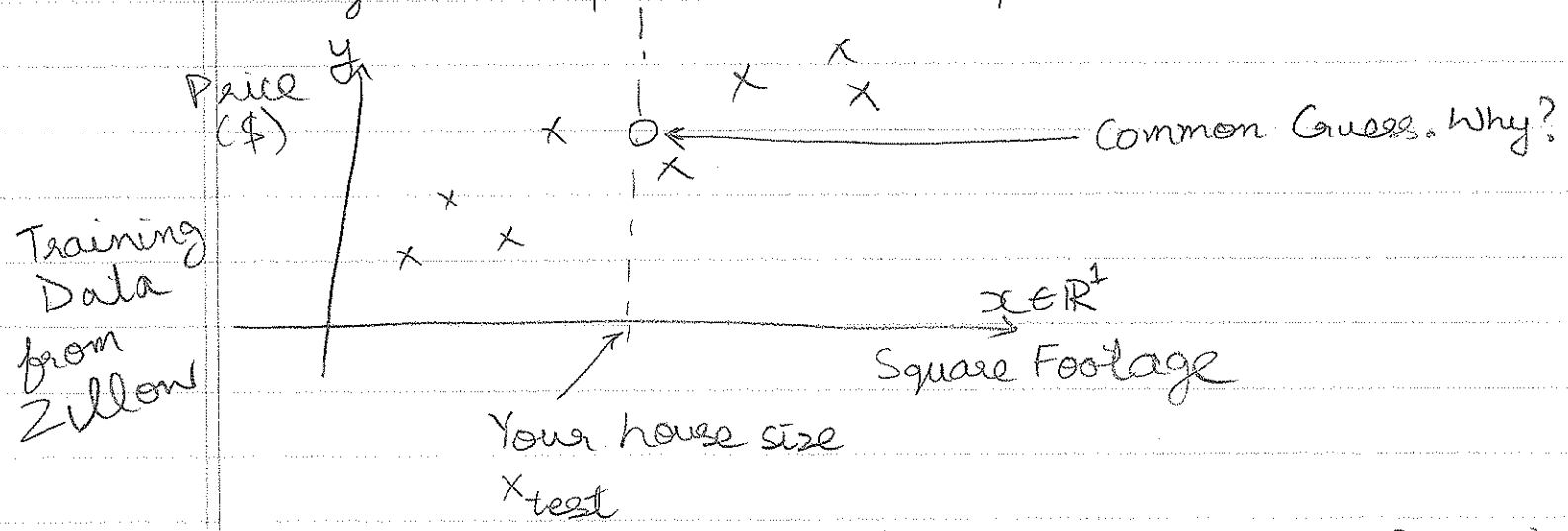
⇒ Learning = Search in H

Picture



(2) Examples

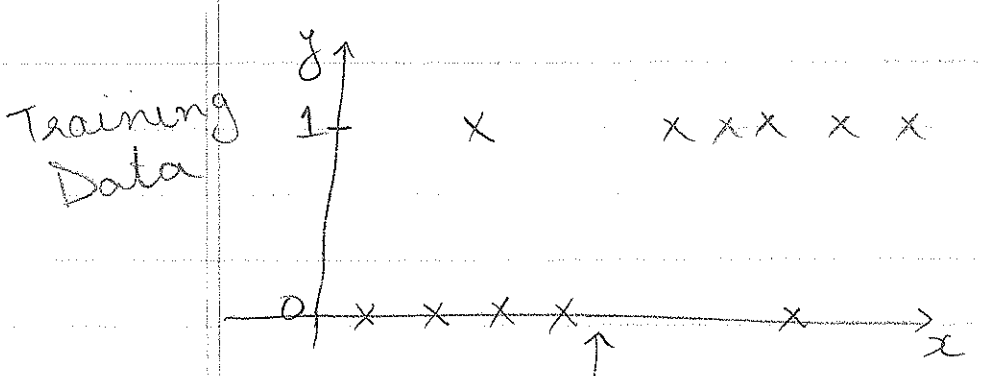
→ Regression: predict house price



Goal: Predict selling price $y_{test} = f(x_{test})$

→ Classification:

$x \in \mathbb{R}^1$: size of tumor
$y \in \{0, 1\}$: Benign or Malignant



This seems to be the transition point. Larger tumor than this = trouble

③ Learning = Search in H

Seems straightforward. Why is this still a research problem?

Q: How many functions possible?
i.e. how large is the largest H?

Let's start with the "easy" countable case

How many Boolean fns on d-variables?

x_1	x_2	. . . . .	x_d	$y = f(x_1, \ldots, x_d)$
0	0	- - - -	0	
⋮				
1	1	- - - -	1	

2^d choices each 0 or 1

$$\#\text{fns} = |H| = 2^{2^d}$$ Ouch!

Can BIG DATA help? [Not without more assumptions]

n data-points = n rows in the table filled (assuming unique data)

"compatible" functions that agree with the data

$$= 2^{[2^d - n]}$$

Two problems: (much much smaller)

① typically $n \ll 2^d$

$d \sim$ hundreds

$n \sim$ millions / billions (?) or 2^{10-20}

In some domains (e.g biology)

$n < d$

n: subjects in a study; d: measurements on each subject

② $2^{[2^d - n]}$

difference sitting in exponent.
Even if off by 10 samples, that's a lot!

Take-away: No Assumptions = No Learning

④ How do we measure performance?

Loss / Error Function

$L(y^{gt}, \hat{y})$ = penalty / cost for predicting $\hat{y}$ when the "true" or correct one is y^{gt}

Usually, Loss = 1 - Accuracy.

→ Regression: Typical Loss

$$L(y, \hat{y}) = (y - \hat{y})^2 \quad L_2 \text{ or Squared Loss}$$

$$|y - \hat{y}| \quad \text{Absolute Loss}$$

→ Classification:

Typically count # mis-classifications

$$L(y, \hat{y}) = \begin{cases} 1 & \text{if } y \neq \hat{y} \\ 0 & y = \hat{y} \end{cases} \quad 0/1 \text{ - Loss}$$

→ Typically, Loss "decomposes" over the dataset

$$\text{Loss}(D) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

Training Label — Prediction

Sometimes losses are "corpus" losses, ie. they are not "additive over each instance.

Example: Next time

Area - Under - ROC curve