Machine Learning
CS60050

Computational Learning Theory
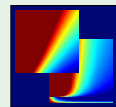### (Error and Noise)

Learning From Data

Yaser S. Abu-Mostafa
California Institute of Technology
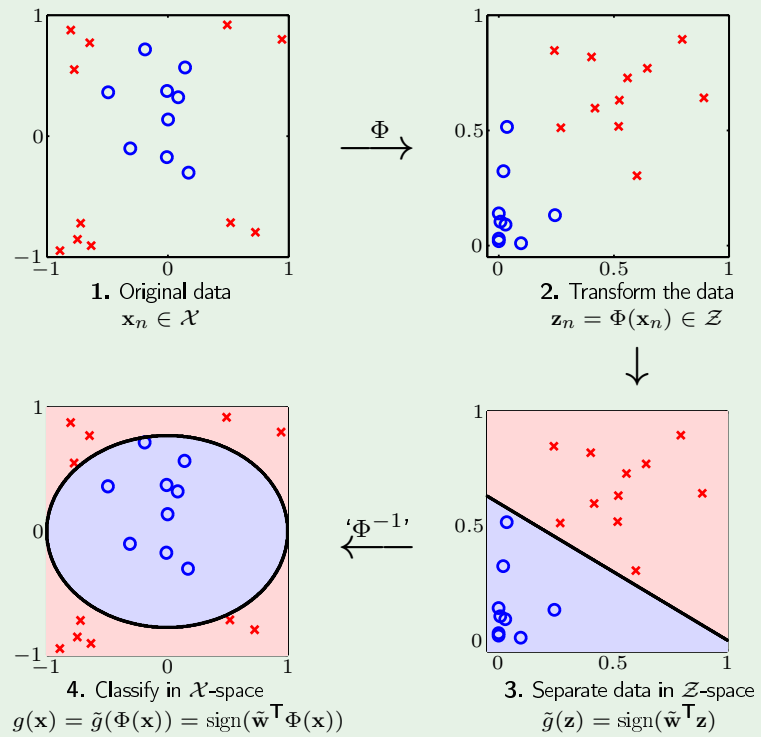
Lecture 4: **Error and Noise**

Outline

- Nonlinear transformation (continued)

- Error measures

- Noisy targets

- Preamble to the theory

1. Original data
$\mathbf{x}_n \in \mathcal{X}$

2. Transform the data
$\mathbf{z}_n = \Phi(\mathbf{x}_n) \in \mathcal{Z}$

Φ

4. Classify in $\mathcal{X}$-space
$g(\mathbf{x}) = \tilde{g}(\Phi(\mathbf{x})) = \mathrm{sign}(\tilde{\mathbf{w}}^\mathsf{T}\Phi(\mathbf{x}))$

'Φ^{-1}'

3. Separate data in $\mathcal{Z}$-space
$\tilde{g}(\mathbf{z}) = \mathrm{sign}(\tilde{\mathbf{w}}^\mathsf{T}\mathbf{z})$

What transforms to what

$$\mathbf{x} = (x_0, x_1, \cdots, x_d) \quad \xrightarrow{\ \Phi\ } \quad \mathbf{z} = (z_0, z_1, \cdots\cdots\cdots, z_{\tilde{d}})$$

$$\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N \quad \xrightarrow{\ \Phi\ } \quad \mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N$$

$$y_1, y_2, \cdots, y_N \quad \xrightarrow{\ \Phi\ } \quad y_1, y_2, \cdots, y_N$$
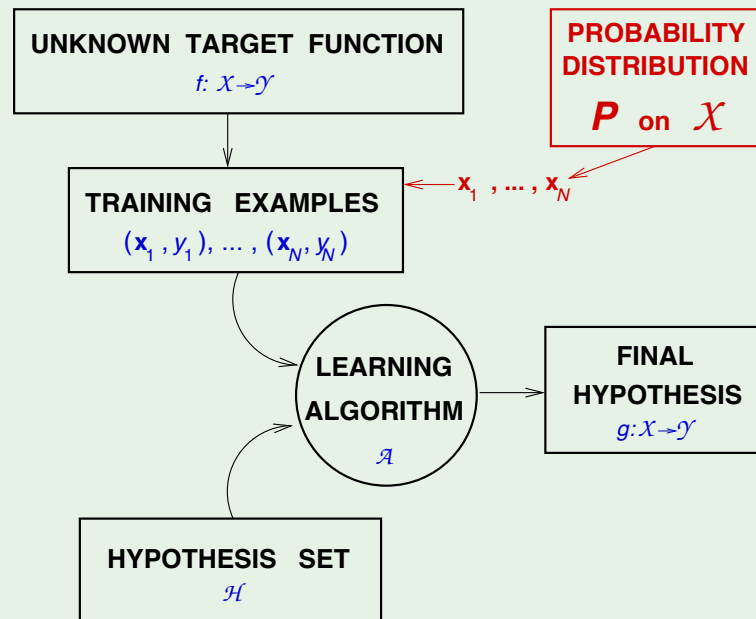
No weights in $\mathcal{X}$ $\qquad\qquad \tilde{\mathbf{w}} = (w_0, w_1, \cdots\cdots\cdots, w_{\tilde{d}})$

$$g(\mathbf{x}) \quad = \quad \mathrm{sign}\big(\tilde{\mathbf{w}}^{\mathsf{T}} \Phi(\mathbf{x})\big)$$

Outline

- Nonlinear transformation (continued)

- Error measures

- Noisy targets

- Preamble to the theory

The learning diagram - where we left it

Error measures

What does "$h \approx f$" mean?

Error measure: $E(h, f)$

Almost always *pointwise definition*: $\text{e}\left(h(\mathbf{x}), f(\mathbf{x})\right)$

Examples:

$\qquad$ Squared error: $\qquad \text{e}\left(h(\mathbf{x}), f(\mathbf{x})\right) = \left(h(\mathbf{x}) - f(\mathbf{x})\right)^2$

$\qquad$ Binary error: $\qquad \text{e}\left(h(\mathbf{x}), f(\mathbf{x})\right) = [\![h(\mathbf{x}) \neq f(\mathbf{x})]\!]$

From pointwise to overall

Overall error $E(h, f)$ = average of pointwise errors $e\big(h(\mathbf{x}), f(\mathbf{x})\big)$.
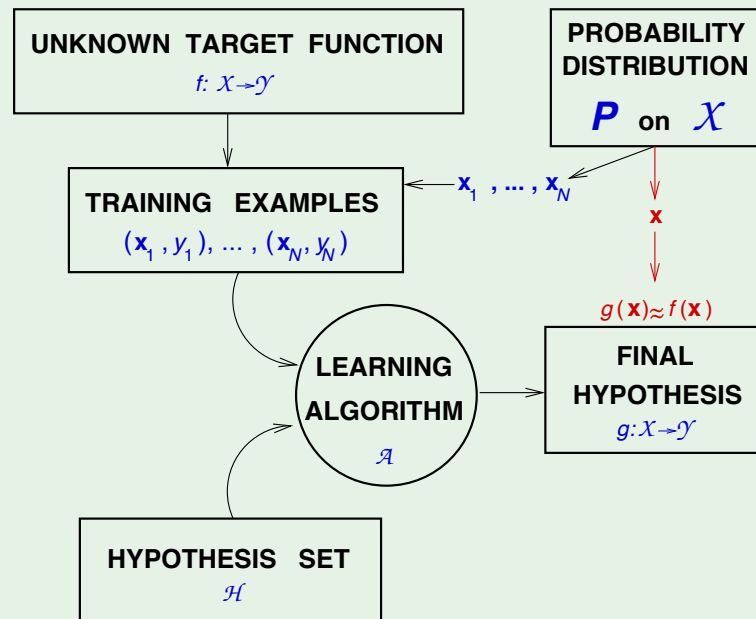
In-sample error:

$$E_{\text{in}}(h) = \frac{1}{N} \sum_{n=1}^{N} e\big(h(\mathbf{x}_n), f(\mathbf{x}_n)\big)$$

Out-of-sample error:

$$E_{\text{out}}(h) = \mathbb{E}_{\mathbf{x}}\Big[e\big(h(\mathbf{x}), f(\mathbf{x})\big)\Big]$$

The learning diagram - with pointwise error

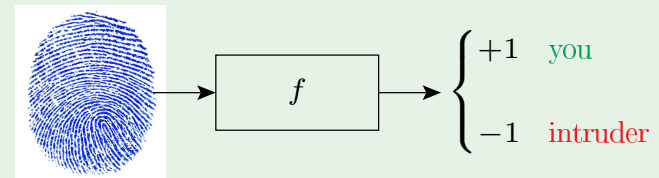# How to choose the error measure

Fingerprint verification:

Two types of error:

    false accept and *false reject*
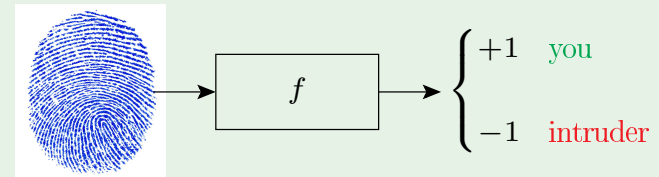
How do we penalize each type?

		f	
		$+1$	-1
h	$+1$	no error	*false accept*
	-1	*false reject*	no error

The error measure - for supermarkets

Supermarket verifies fingerprint for discounts

False reject is costly; customer gets annoyed!

False accept is minor; gave away a discount
and intruder left their fingerprint ☺

$$
\begin{array}{cc|cc}
 & & \multicolumn{2}{c}{f} \\
 & & +1 & -1 \\
\hline
h & +1 & 0 & 1 \\
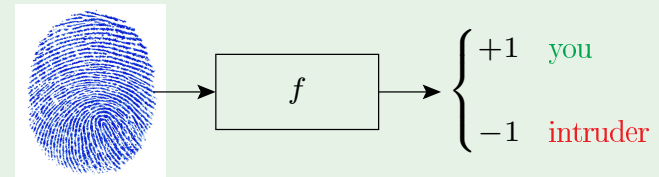 & -1 & 10 & 0 \\
\end{array}
$$

The error measure –   for the CIA

CIA verifies fingerprint for security

False accept is a disaster!

False reject can be tolerated
Try again; you are an employee ☺

$$
\begin{array}{c|cc}
 & \multicolumn{2}{c}{f} \\
 & +1 & -1 \\
\hline
\;+1 & 0 & 1000 \\
h\;-1 & 1 & 0
\end{array}
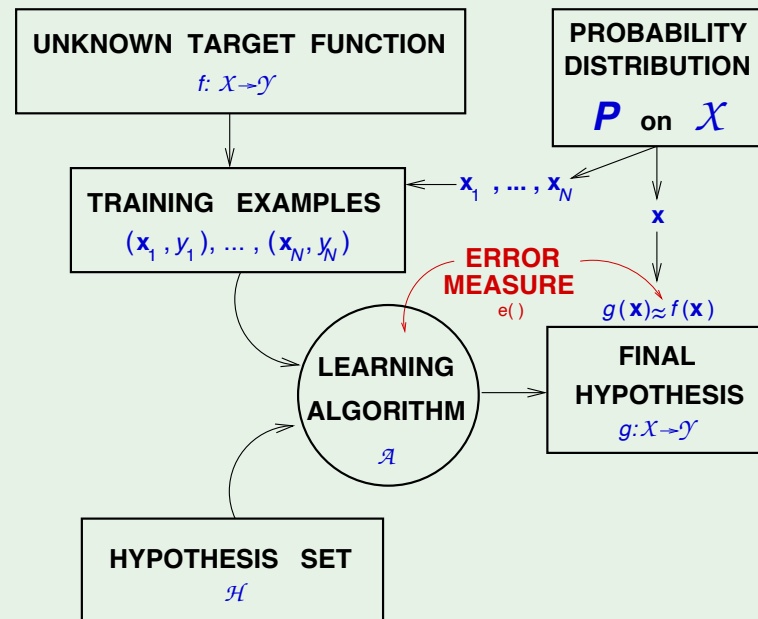$$

Take-home lesson

The error measure should be <u>specified by the user</u>.

Not always possible. Alternatives:

    Plausible measures: squared error $\equiv$ Gaussian noise

    Friendly measures: closed-form solution, convex optimization

The learning diagram - with error measure

Noisy targets

The 'target function' is not always a *function*

Consider the credit-card approval:

age	23 years
annual salary	$30,000
years in residence	1 year
years in job	1 year
current debt	$15,000
· · ·	· · ·

two 'identical' customers $\longrightarrow$ two different behaviors

Target 'distribution'

Instead of $y = f(\mathbf{x})$, we use target *distribution*:
$$P(y \mid \mathbf{x})$$

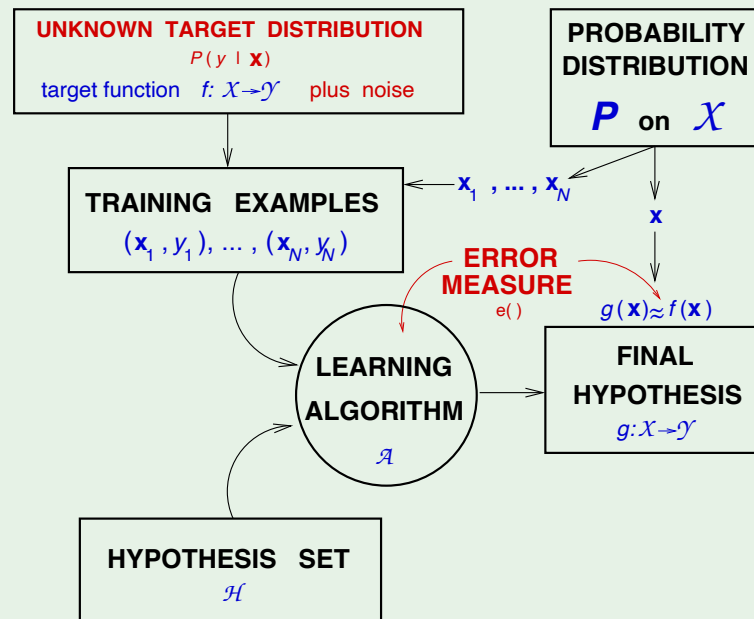$(\mathbf{x}, y)$ is now generated by the joint distribution:
$$P(\mathbf{x})P(y \mid \mathbf{x})$$

Noisy target = deterministic target $f(\mathbf{x}) = \mathbb{E}(y|\mathbf{x})$ plus noise $y - f(\mathbf{x})$

Deterministic target is a special case of noisy target:
$$P(y \mid \mathbf{x}) \text{ is zero except for } y = f(\mathbf{x})$$

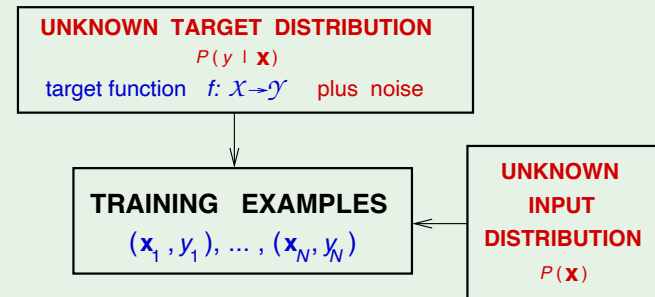# The learning diagram - including noisy target

Distinction between $P(y|\mathbf{x})$ and $P(\mathbf{x})$

Both convey probabilistic aspects of $\mathbf{x}$ and y

The target distribution $P(y \mid \mathbf{x})$
is what we are trying to learn

The input distribution $P(\mathbf{x})$
quantifies relative importance of $\mathbf{x}$

Merging $P(\mathbf{x})P(y|\mathbf{x})$ as $P(\mathbf{x}, y)$
mixes the two concepts

UNKNOWN TARGET DISTRIBUTION
$P(y \mid \mathbf{x})$
target function $f: \mathcal{X} \to \mathcal{Y}$ plus noise

TRAINING EXAMPLES
$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$

UNKNOWN INPUT DISTRIBUTION
$P(\mathbf{x})$

Outline

- Nonlinear transformation (continued)

- Error measures

- Noisy targets

- Preamble to the theory

What we know so far

Learning is feasible. It is likely that

$$E_{\text{out}}(g) \approx E_{\text{in}}(g)$$

Is this learning?

We need $g \approx f$, which means

$$E_{\text{out}}(g) \approx 0$$

The 2 questions of learning

$E_{\text{out}}(g) \approx 0$ is achieved through:

$$\underbrace{E_{\text{out}}(g) \approx E_{\text{in}}(g)}_{\text{Lecture 2}} \qquad \text{and} \qquad \underbrace{E_{\text{in}}(g) \approx 0}_{\text{Lecture 3}}$$

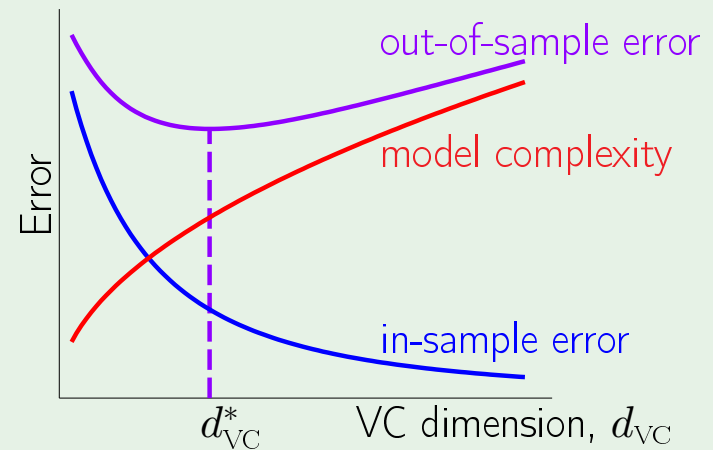Learning is thus split into 2 questions:

1. Can we make sure that $E_{\text{out}}(g)$ is close enough to $E_{\text{in}}(g)$?

2. Can we make $E_{\text{in}}(g)$ small enough?

What the theory will achieve

Characterizing the feasibility of learning for
infinite M

Characterizing the tradeoff:

Model complexity $\uparrow$	E_{in}	$\downarrow$
Model complexity $\uparrow$	$E_{\text{out}} - E_{\text{in}}$	$\uparrow$

Thank You!