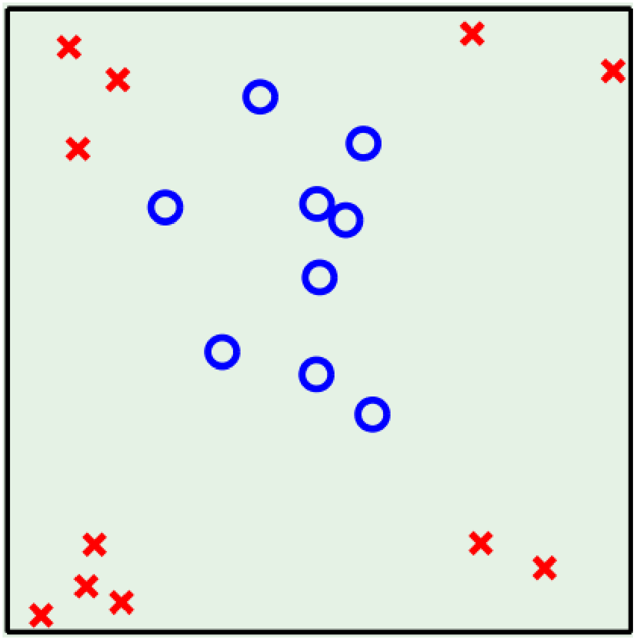
Error and Noise

outline

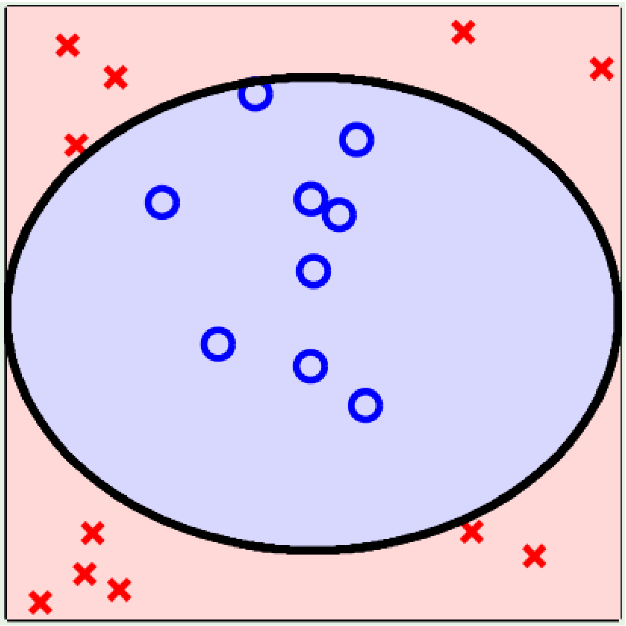
- Nonlinear transformation
- Error measures
- Noisy targets
- Preambles to the theory

Linear is limited

Data



Hypothesis



Linear in what?

- Linear regression implements $\sum_{j=0}^{d} w_j x_j$
- Linear classification implements $sign\left(\sum_{j=0}^{d} w_j x_j\right)$

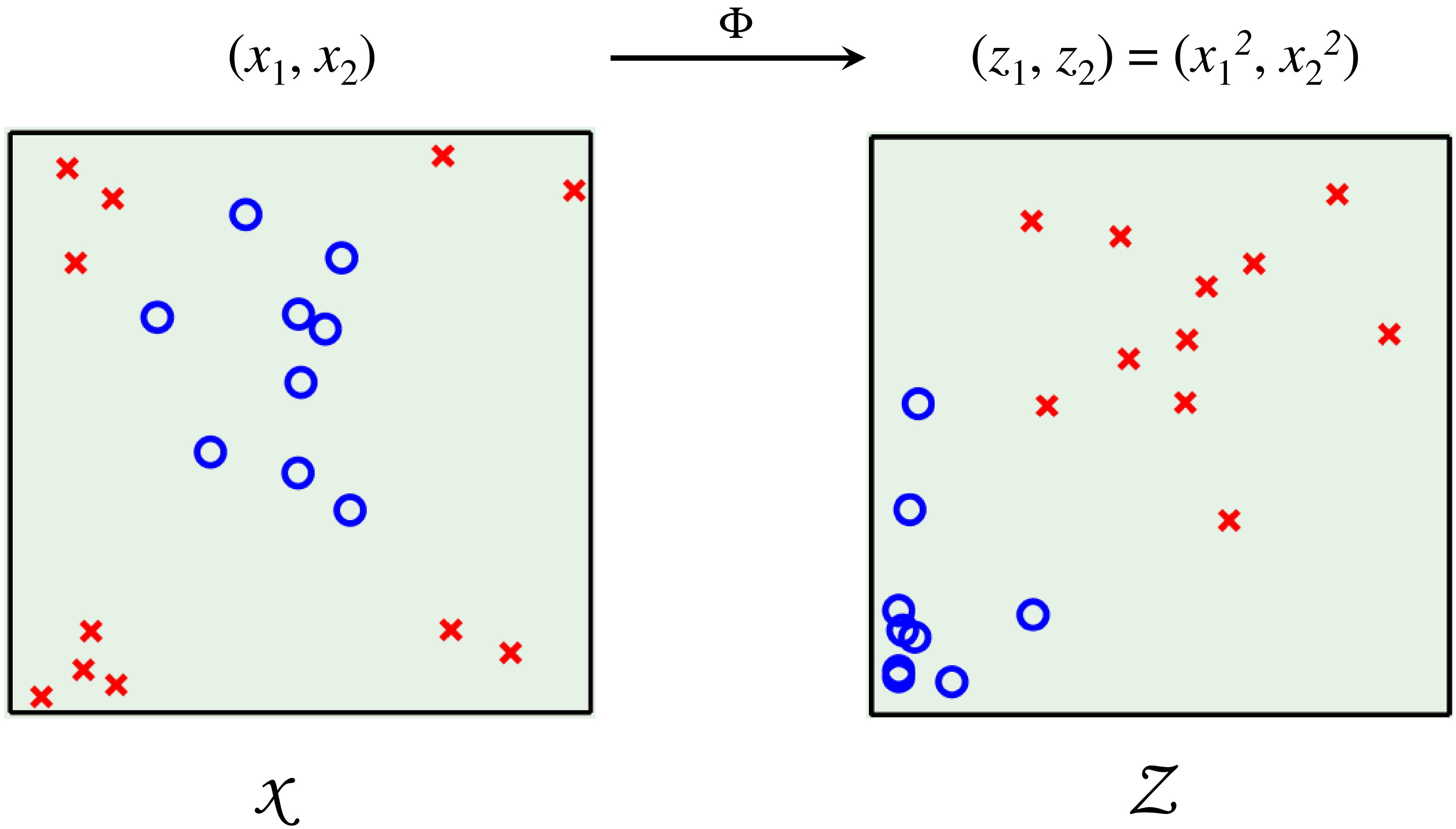
  Both are functions linear in x's.

  More importantly from the learning point of view,

  they are linear in w's (parameters in learning).

- Both algorithms, PLA and Regression, work because of linearity in the weights. So, we can apply any nonlinear transformation to data (**x**'s are constant) and still remain in the realm of linear models.

Transform data nonlinearly



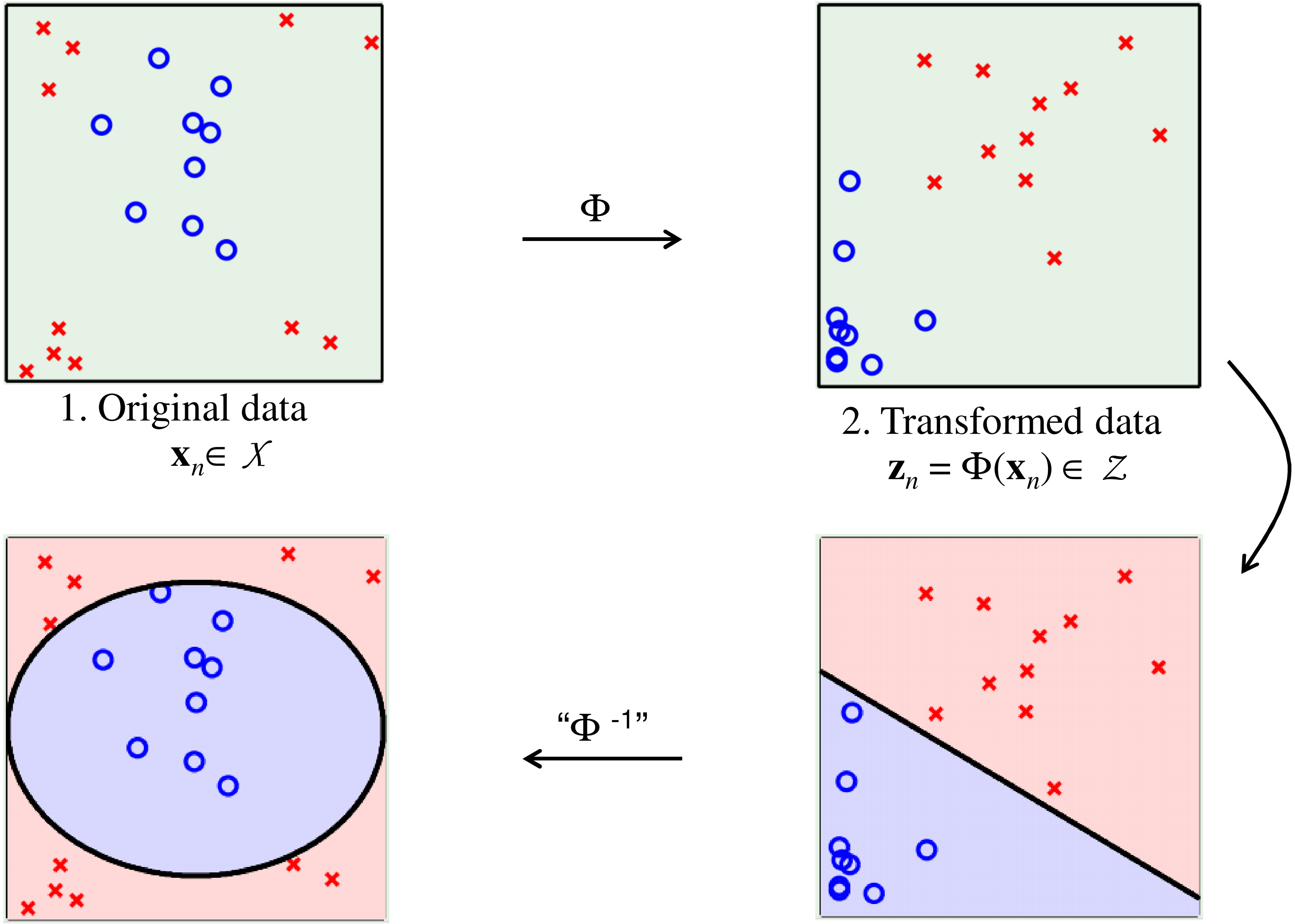
None Linear transformation

- None-linear transforms give the ability to linearly separate any data points by moving to a new space with sufficiently larger dimensions.

- A tool to get more sophisticated surfaces in the initial space while we are still able to use simple linear techniques.

- Any transformation $\Phi: \mathcal{X} \rightarrow \mathcal{Z}$ on data preserves the linearity of the model, $\mathbf{w}^T\mathbf{x}$.

The cycle of nonlinear transformation



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

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

3. Separate data in $\mathcal{Z}$-space
$g'(\mathbf{z}) = \text{sign}\,(\mathbf{w}'^{\mathrm{T}}\mathbf{z})$

4. Classify in $\mathcal{X}$-space
$g(\mathbf{x}) = g'(\Phi(\mathbf{x})) = \text{sign}\,(\mathbf{w}'^{\mathrm{T}}\Phi(\mathbf{x}))$

What transforms to what?

$$\mathbf{x} = (x_0, x_1, \ldots, x_\mathrm{d}) \xrightarrow{\Phi} \mathbf{z} = (z_0, z_1, \ldots, z_{\mathrm{d}'})$$

$$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \xrightarrow{\Phi} \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$$

$$y_1, y_2, \ldots, y_n \xrightarrow{\Phi} y_1, y_2, \ldots, y_n$$

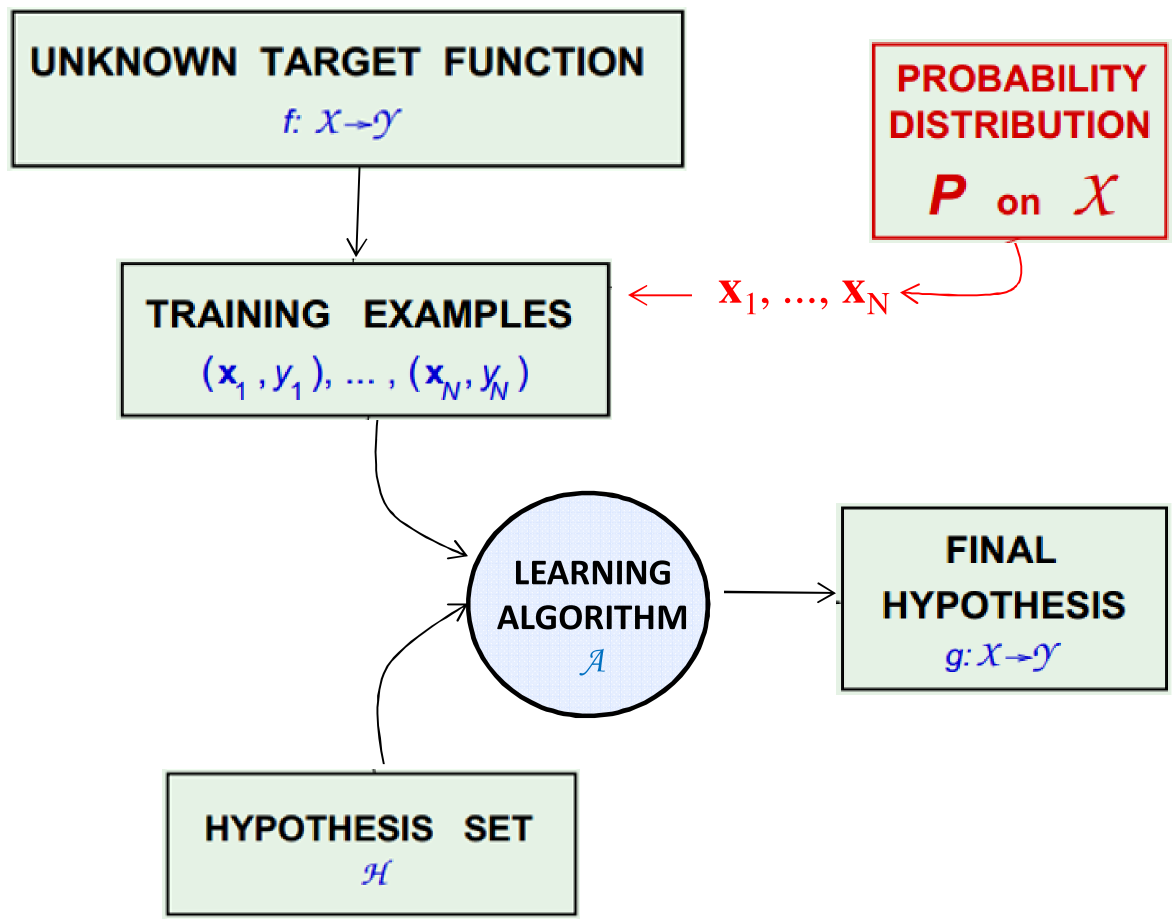
No weights in $\mathcal{X}$ $\qquad$ $\mathbf{w}' = (w_0, w_1, \ldots, w_{\mathrm{d}'})$

Hypothesis:

$$g(\mathbf{z}) = \mathrm{sign}\,(\mathbf{w}'^{\mathrm{T}}\mathbf{z})$$

$$g(\mathbf{x}) = \mathrm{sign}\,(\mathbf{w}'^{\mathrm{T}}\mathbf{\Phi}(\mathbf{x}))$$

The learning diagram so far



Error measures

- Error measures try to answer the following question:
  - What does " $h \approx f$ " mean?

- Error measure: $E\,(h, f)$

- Pointwise definition $e\,(h(\mathbf{x}), f(\mathbf{x}))$

- Examples
  - Squared error $e\,(h(\mathbf{x}), f(\mathbf{x})) = (\ h(\mathbf{x}) - f(\mathbf{x})\ )^2$
  - Binary error $e\,(h(\mathbf{x}), f(\mathbf{x})) = [\ h(\mathbf{x}) \neq f(\mathbf{x})\ ]$

Overall error

- $E\,(h, f)$ = average of pointwise errors $e(h(\mathbf{x}), f(\mathbf{x}))$

- In-sample error:

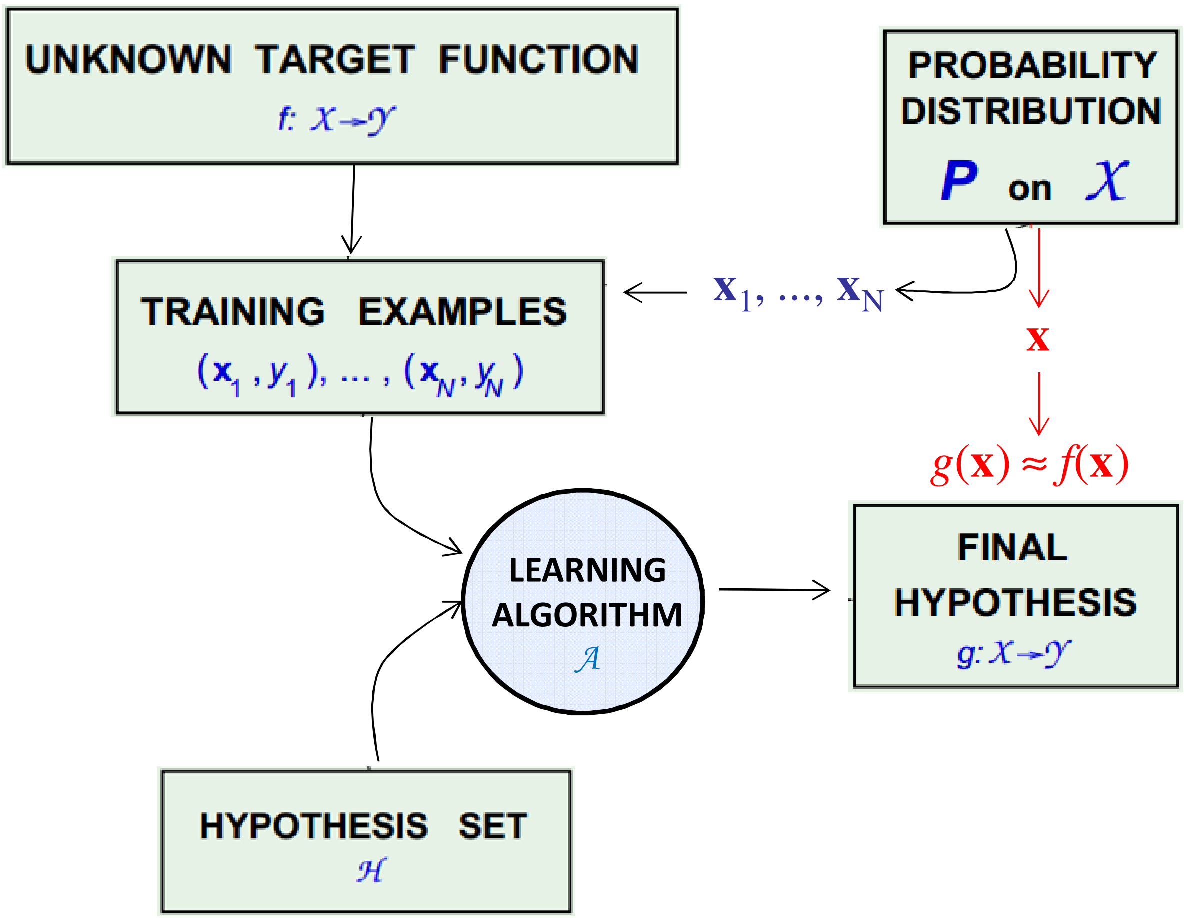
$$E_{in}(h) = \frac{1}{n}\sum_{i=1}^{n} e(h(\mathbf{x}_i), f(\mathbf{x}_i))$$

- Out-of- sample error:

$$E_{out}(h) = E_{\mathbf{x}}\left[e(h(\mathbf{x}), f(\mathbf{x}))\right]$$

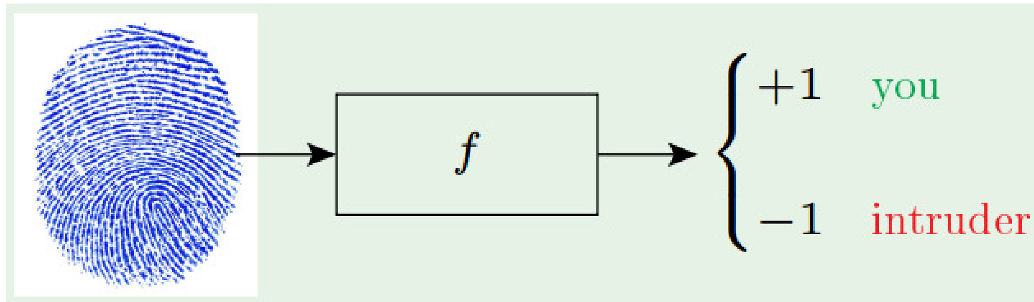
Expected value w.r.t. $\mathbf{x}$.
($\mathbf{x}$ is a general point in X space)

The learning diagram – with pointwise error



The error cost

- Fingerprint verification:



- Two types of error:
  - *false accept* (false positive - FP) and
  - *false reject* (false negative - FN)

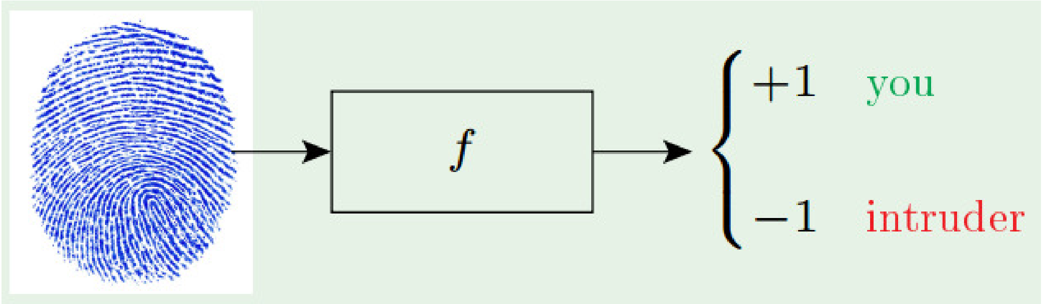
		f: $+1$	f: -1
h	$+1$	no error	*false accept*
	-1	*false reject*	no error

Confusion Matrix

- How to penalize each type?

The error cost – for supermarkets

- Supermarket verifies fingerprints for discounts

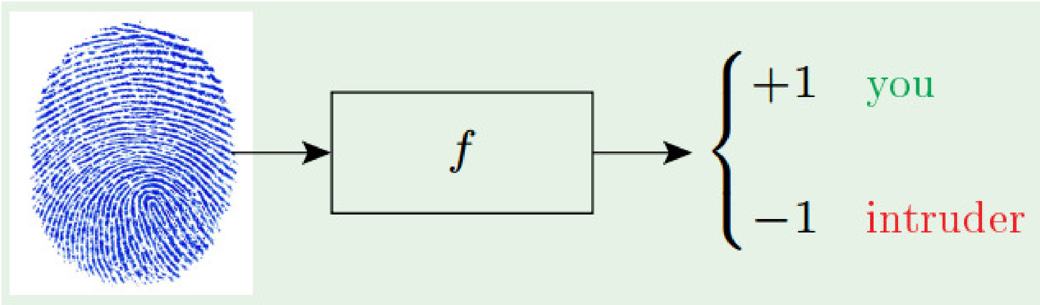


- False reject is costly; risk of loosing a customer!
- False accept is minor; gave away a discount, and intruder left their fingerprint!

		f	
		+1	−1
h	+1	0	1
	−1	10	0

The error cost – for CIA

- CIA verifies fingerprints for security

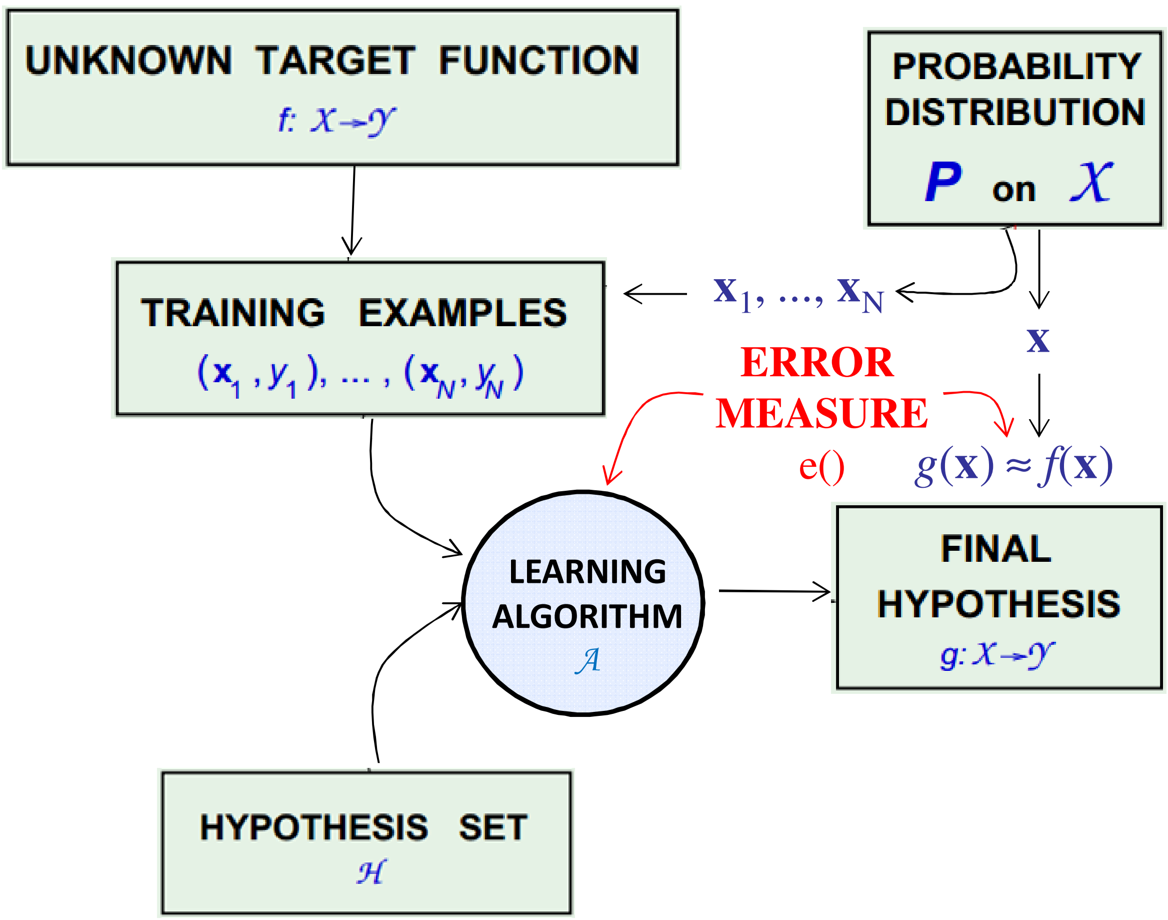


- False accept is a disaster!
- False reject can be tolerated. Try again!

		f: +1	f: −1
h	+1	0	1000
	−1	1	0

In practical problems: *the error measure and the cost should be specified* *<u>by the user</u>*.

The learning diagram – with error measure



Noisy targets

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

- Consider the credit-card approval

age	gender	Annual salary	Years in residence	Years in job	Current debt	Class
23	male	$30,000	1	3	$15,000	+1
23	male	$30,000	1	3	$15,000	-1
…	…	…	…	…	…	

- Two identical customers $\rightarrow$ two different behaviors

Target ‘distribution’

- Instead of *target function* $y = f(\mathbf{x})$
  we use *target distribution*

  $P(y \mid \mathbf{x})$

- $(\mathbf{x}, y)$ is now generated by the joint distribution (assuming independence):

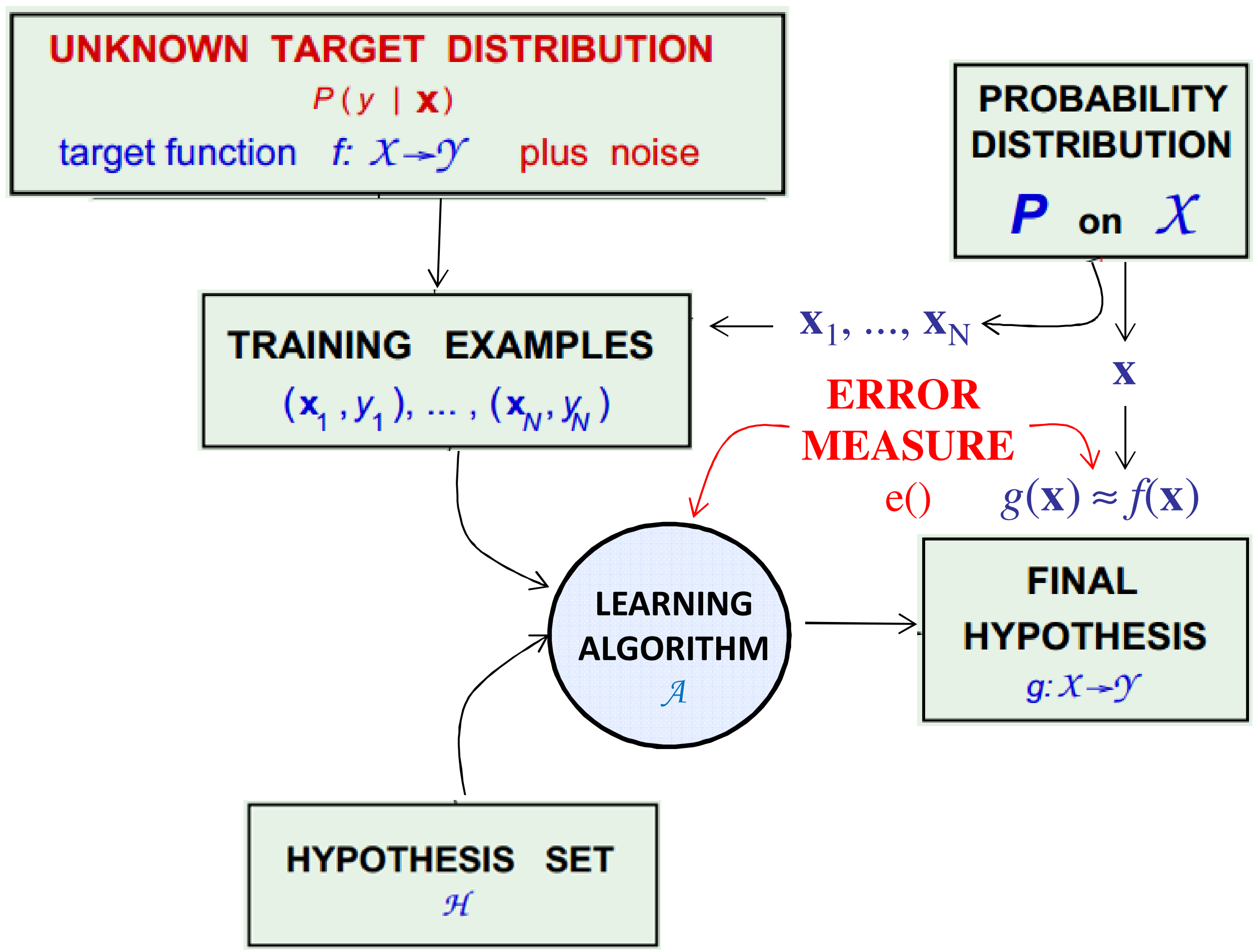
  $P(\mathbf{x})\, P(y \mid \mathbf{x})$

- Noisy target = deterministic target + noise

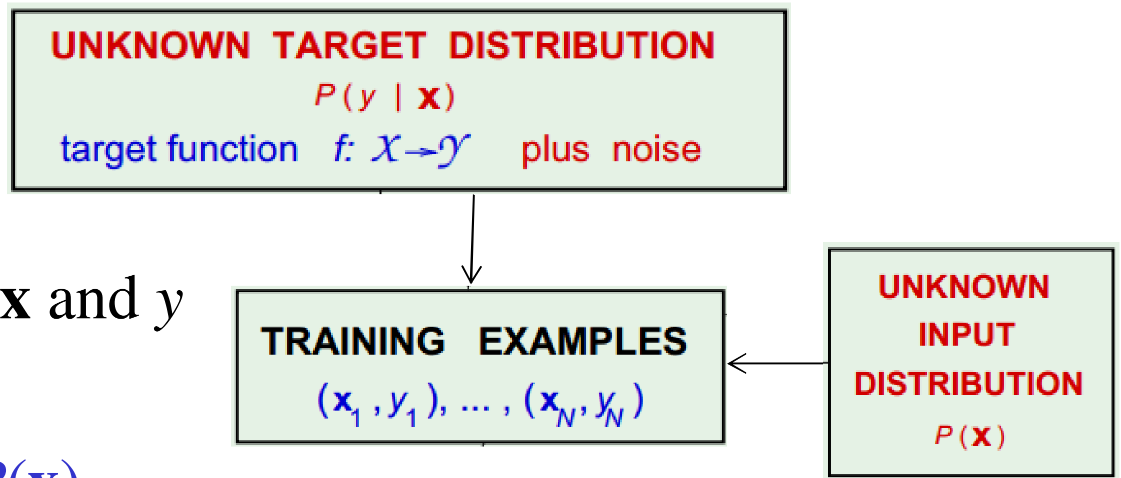
  $$\underbrace{\text{deterministic target}}_{f(\mathbf{x}) = E(y \mid \mathbf{x})} \qquad \underbrace{\text{noise}}_{y - f(\mathbf{x})}$$

- Deterministic target is a special case of noisy target,
  where $P(y \mid \mathbf{x})$ 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 input distribution $P(\mathbf{x})$ quantifies relative importance of $\mathbf{x}$
- The target distribution $P(y \mid \mathbf{x})$ is what we are trying to learn
- Merging $P(\mathbf{x})\, P(y \mid \mathbf{x})$ as $P(\mathbf{x}, y)$ mixes two concepts inherently different

Preamble to the theory

- What we know so far:
  - Learning is feasible, in a probabilistic sense.

    It is likely that

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

(good)

Generalization

  - Is this learning?

    Indeed, we need to learn a g such that

$$g \approx f$$

which means

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

(good)

Learning

The 2 questions of learning

- $E_{\text{out}}(g) \approx 0$ (*the learning*)
  is achieved through two conditions
  1. $E_{\text{out}}(g) \approx E_{\text{in}}(g)$
     (developed using Hoeffding)
     and
  2. $E_{\text{in}}(g) \approx 0$

Theoretical result
(Lecture 2)

Practical result
(Lecture 3)

- Learning reduces to 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

Two important things the theory does:

1. Characterize the feasibility of learning for infinite M

   (M is the number of hypothesis)

2. Characterizing the tradeoff:

   Model complexity $\uparrow$ $\quad E_{\text{in}} \downarrow$

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