Big Data Analytics

Learning theory

Souhaib Ben Taieb

March 19, 2021

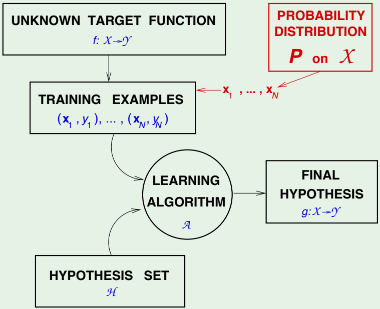
University of Mons

Table of contents

Table of contents

Learning diagram

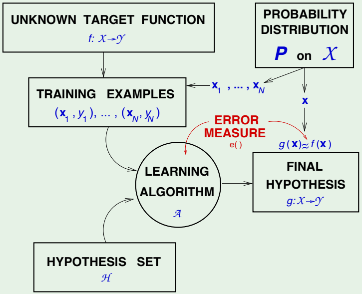
The learning diagram - where we left it



 6/22

Learning diagram (with error measure)

The learning diagram - with error measure

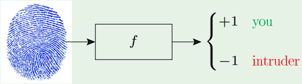


© AML Creator: Yaser Abu-Mostafa - LFD Lecture 4 14/22

How to choose the error measure?

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

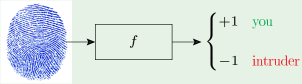
The supermarket example

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 ☺



		f: $+1$	f: -1
h	$+1$	0	1
	-1	10	0

The CIA example

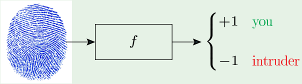
The error measure - for the CIA

CIA verifies fingerprint for security

False accept is a disaster!

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

		f	
		$+1$	-1
h	$+1$	0	1000
	-1	1	0



Noisy targets

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

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:

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

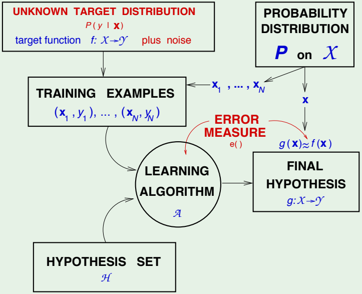
Noisy target = deterministic target $f(\mathbf{x}) = \mathbb{E}(y|\mathbf{x})$ plus noise $y - f(\mathbf{x})$

Deterministic target is a special case of noisy target:

$$P(y \mid \mathbf{x}) \text{ is zero except for } y = f(\mathbf{x})$$

Final learning diagram

The learning diagram - including noisy target



© Creator: Yaser Abu-Mostafa - LFD Lecture 4

17/22

Table of contents

In-sample and out-of-sample errors

Consider

$$f = \operatorname*{argmin}_{h:\mathcal{X}\to\mathcal{Y}} E_{\text{out}}(h),$$

$$g^* = \operatorname*{argmin}_{h\in\mathcal{H}} E_{\text{out}}(h),$$

and

$$g = \operatorname*{argmin}_{h\in\mathcal{H}} E_{\text{in}}(h).$$

Approximation-generalization tradeoff

The difference between the out-of-sample error of g and f can be decomposed as follows

$$E_{\text{out}}(g) - E_{\text{out}}(f) = \underbrace{[E_{\text{out}}(g^*) - E_{\text{out}}(f)]}_{\text{Approximation error}} + \underbrace{[E_{\text{out}}(g) - E_{\text{out}}(g^*)]}_{\text{Estimation error}}$$

- **Approximation error** is how far the entire hypothesis set is from f. Larger hypothesis sets have lower approximation error.
- **Estimation error** is how good g is with respect to the best in the hypothesis set. Larger hypothesis sets have higher estimation error because it is harder to find a good prediction function based on limited data.

This is called the **approximation-generalization** tradeoff.

Quantifying the approximation-generalization tradeoff

The VC analysis is one approach to quantify the tradeoff:

- $d_{\mathrm{VC}} \uparrow \implies$ better chance of **approximating** f ($E_{\mathrm{in}} \approx 0$)
- $d_{\mathrm{VC}} \downarrow \implies$ better chance of **generalizing** to out-of-sample ($E_{\mathrm{in}} \approx E_{\mathrm{out}}$)

The VC analysis uses binary errors (classification).

The VC analysis only depends on $\mathcal{H}$ (through d_{VC}):

$$E_{\mathrm{out}} \leq E_{\mathrm{in}} + \Omega(d_{\mathrm{VC}})$$

$\implies$ Independent of the target function f, the input distribution $p(x)$ and the learning algorithm $\mathcal{A}$.

Quantifying the approximation-generalization tradeoff

The **bias-variance** analysis approach is another way to quantify the tradeoff:

- How well **can** the learning approximate f
  ... as opposed to how well **did** the learning approximate f in-sample (E_{in})
- How close **can** you get to that approximation with a finite data set
  ... as opposed to how close **is** E_{in} to E_{out}

The bias-variance analysis applies to **squared errors** (classification and regression).

The bias-variance analysis can take into account the **learning algorithm** $\mathcal{A}$.

- Different learning algorithms can have different E_{out} when applied to the same $\mathcal{H}$!

Bias and variance decomposition

Start with E_{out}

$$E_{\text{out}}(g^{(\mathcal{D})}) = \mathbb{E}_{\mathbf{x}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$$

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[E_{\text{out}}(g^{(\mathcal{D})})\right] &= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\mathbf{x}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right] \\ &= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right]\end{aligned}$$

Now, let us focus on:

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$$

The average hypothesis

The average hypothesis

To evaluate $\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$

we define the 'average' hypothesis $\bar{g}(\mathbf{x})$:

$$\bar{g}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}\left[g^{(\mathcal{D})}(\mathbf{x})\right]$$

Imagine **many** data sets $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_K$

$$\bar{g}(\mathbf{x}) \approx \frac{1}{K}\sum_{k=1}^{K} g^{(\mathcal{D}_k)}(\mathbf{x})$$

Using the average hypothesis

Using $\bar{g}(\mathbf{x})$

$$
\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] &= \mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}) + \bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] \\
&= \mathbb{E}_{\mathcal{D}}\Big[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2 + \left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2 \\
&\qquad + \ 2\ \left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)\ \left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)\Big] \\
&= \mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right] + \left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2
\end{aligned}
$$

Bias and variance

Bias and variance

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]}_{\text{var}(\mathbf{x})} + \underbrace{\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2}_{\text{bias}(\mathbf{x})}$$

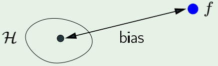
$$\text{Therefore,} \quad \mathbb{E}_{\mathcal{D}}\left[E_{\text{out}}(g^{(\mathcal{D})})\right] = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right]$$

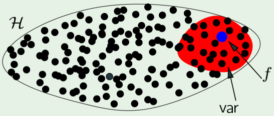
$$= \mathbb{E}_{\mathbf{x}}[\text{bias}(\mathbf{x}) + \text{var}(\mathbf{x})]$$

$$= \quad \text{bias} \quad + \quad \text{var}$$

The tradeoff

$$\text{bias} = \mathbb{E}_{\mathbf{x}}\left[\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] \qquad \text{var} = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]\right]$$





$\downarrow \qquad \mathcal{H} \uparrow \qquad \uparrow$

Example: sine target

Example: sine target

$f : [-1, 1] \rightarrow \mathbb{R} \qquad f(x) = \sin(\pi x)$

Only two training examples! $\quad N = 2$

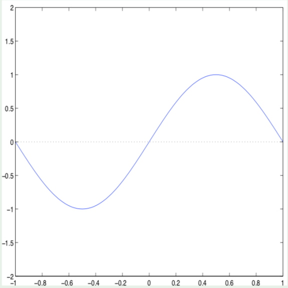
Two models used for learning:

$\mathcal{H}_0: \quad h(x) = b$

$\mathcal{H}_1: \quad h(x) = ax + b$

Which is better, $\mathcal{H}_0$ or $\mathcal{H}_1$?

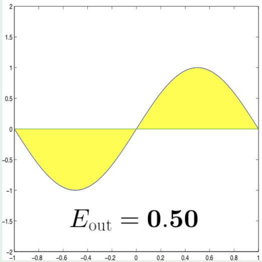
f



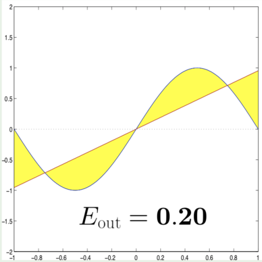
Example: Approximation

Approximation - $\mathcal{H}_0$ versus $\mathcal{H}_1$

$\mathcal{H}_0$

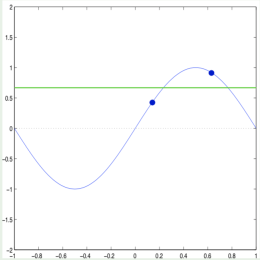
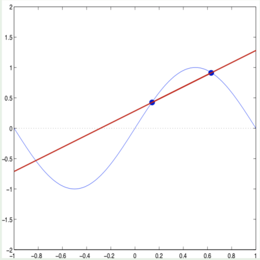


$\mathcal{H}_1$



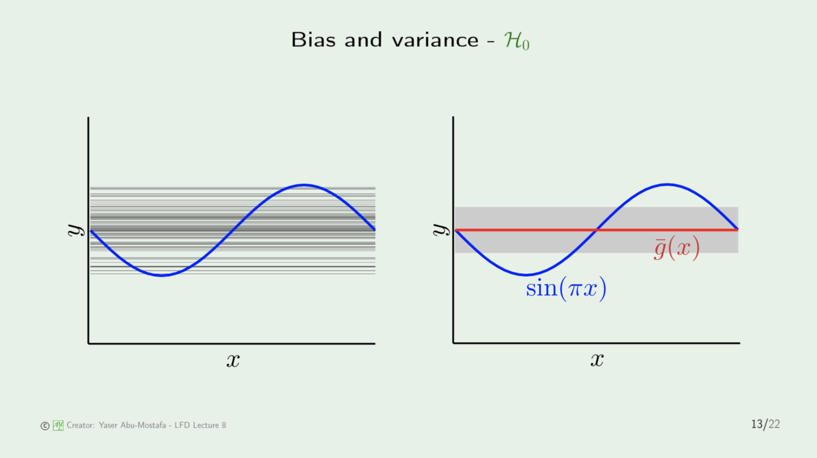
Example: Learning

Learning - $\mathcal{H}_0$ versus $\mathcal{H}_1$

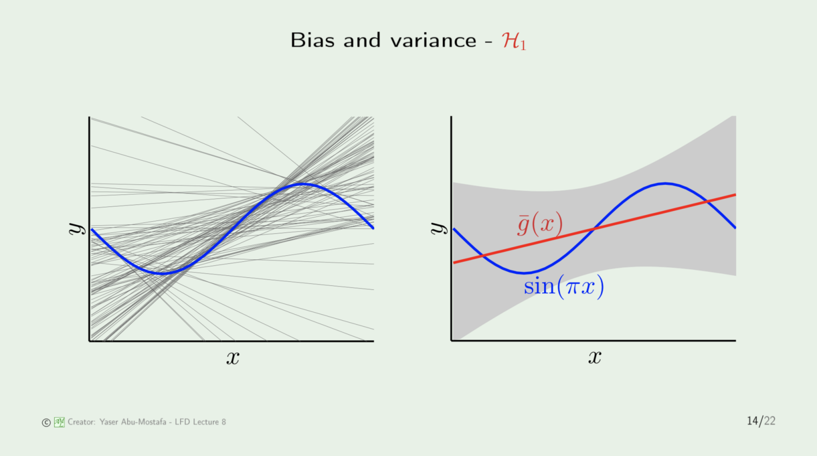
$\mathcal{H}_0$	$\mathcal{H}_1$
	

12/22

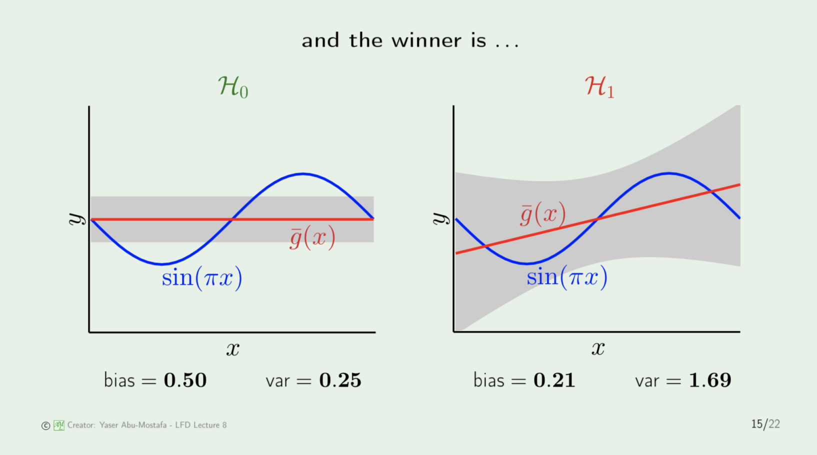
Example: Bias and variance



Example: Bias and variance



Example: Bias and variance tradeoff



Lesson learned

Lesson learned

Match the **'model complexity'**

to the **data resources**, not to the **target complexity**

Table of contents

Learning curves

Expected E_{out} and E_{in}

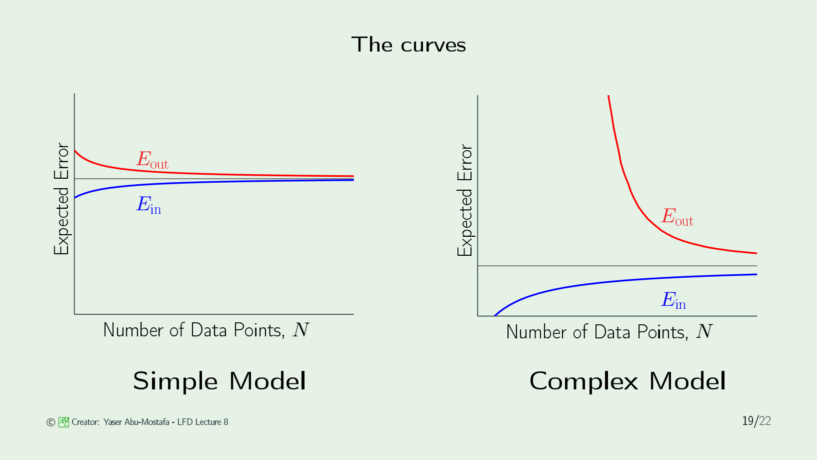
Data set $\mathcal{D}$ of size N

Expected out-of-sample error $\mathbb{E}_{\mathcal{D}}[E_{\text{out}}(g^{(\mathcal{D})})]$

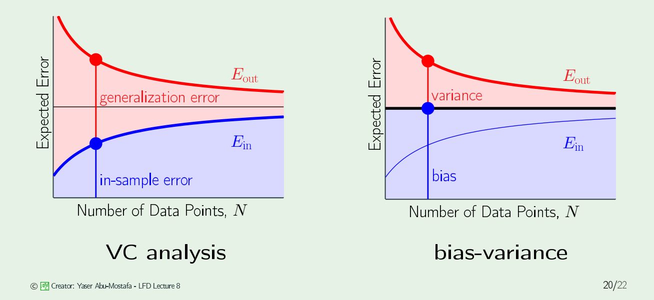
Expected in-sample error $\mathbb{E}_{\mathcal{D}}[E_{\text{in}}(g^{(\mathcal{D})})]$

How do they vary with N?

Learning curves



VC versus bias-variance analysis



- **VC**[1]: Pick $\mathcal{H}$ that can generalize and has a good chance to fit the data.
- **Bias-variance**[2]: Pick $(\mathcal{H}, \mathcal{A})$ to approximate f and not behave wildly.

[1]We take the expected values of all quantities with respect to $\mathcal{D}$ of size N.

[2]we assume, for every N, the average learned hypothesis $\bar{g}$ has the same performance as the best approximation to f in the learning model.

Linear regression case

Linear regression case

Noisy target $y = \mathbf{w}^{*\intercal}\mathbf{x}$ + noise

Data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$

Linear regression solution: $\mathbf{w} = (\mathrm{X}^\intercal\mathrm{X})^{-1}\mathrm{X}^\intercal\mathbf{y}$

In-sample error vector $= \mathrm{X}\mathbf{w} - \mathbf{y}$

'Out-of-sample' error vector $= \mathrm{X}\mathbf{w} - \mathbf{y}'$

Learning curves for linear regression

Learning curves for linear regression

Best approximation error $= \sigma^2$

Expected in-sample error $= \sigma^2 \left(1 - \frac{d+1}{N}\right)$

Expected out-of-sample error $= \sigma^2 \left(1 + \frac{d+1}{N}\right)$

Expected generalization error $= 2\sigma^2 \left(\frac{d+1}{N}\right)$