Lecture #2 | Part 3

Bias-Variance Trade-off

Reading: - Murphy 4.7.6.
- MIT Notes 2.8

Code Demo: bias-variance_demo.ipynb

- Let's start w/ recalling that for: **Optimal Prediction**, the goal is:
predict $Y \in \mathbb{R}$ given $X \in \mathbb{R}^d$ if $(X, Y) \sim P_{XY}$ ← true distribution out of which dataset $\mathcal{D}$ is sampled from. We do not have access to it (do not know it)

- We want to find function η that minimizes:

$$E_{XY}\left[(Y-\eta(X))^2\right] = E_X\left[E_{Y|X}\left[(Y-\eta(x))^2 \mid X=x\right]\right]$$

- $E_{XY}\left[(Y-\eta(X))^2\right]$: also known/called the risk $R(\eta)$
- $E_X\left[E_{Y|X}\left[(Y-\eta(x))^2 \mid X=x\right]\right]$: we condition on X trick (See Mostafa, pp. 61)

Outer expectation is just an average over distribution X ⇒ to minimize $R(\eta)$, it suffices to minimize the inner conditional expectation for each fixed value $X=x$.

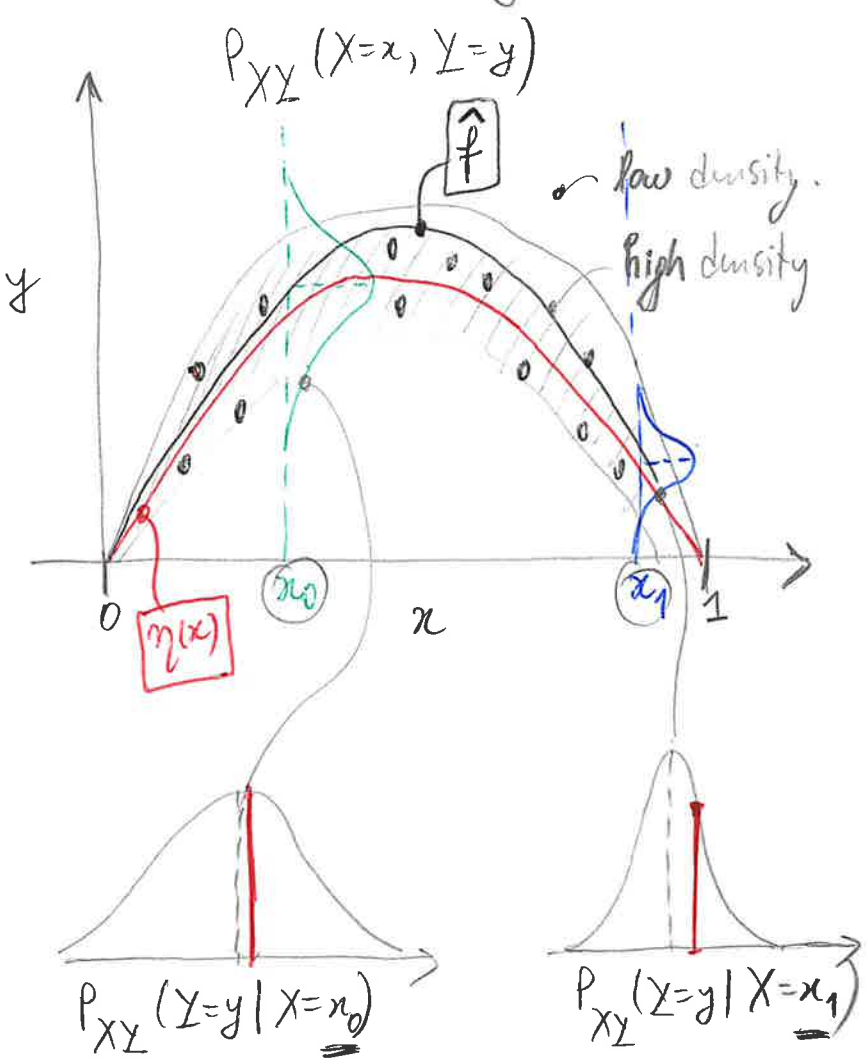
For any fixed value $X=x$, $\eta(x) = c_x$: a simple number because once x is fixed dependence on η is only through the scalar value $\eta(x)$.

c_x minimizes $E_{Y|X}\left[(Y-c_x)^2 \mid X=x\right]$, so, to get it, take derivative and set to 0:

$$\begin{aligned} 0 &= \frac{d}{dc_x} E_{Y|X}\left[(Y-c_x)^2 \mid X=x\right] \\ &= E_{Y|X}\left[\frac{d}{dc_x}(Y-c_x)^2 \mid X=x\right] \\ &= E_{Y|X}\left[-2(Y-c_x) \mid X=x\right] \\ &= -2\,E_{Y|X}\left[Y \mid X=x\right] + 2c_x \end{aligned}$$

⇒ Squared Error Optimal Predictor: $\boxed{c_x = \eta(x) = E_{Y|X}\left[Y \mid X=x\right]}$

Statistical Learning



- Find η that minimizes:

$$E_{XY}\left[(Y-\eta(X))^2\right]$$

- Ideally, we want to find:

$$\eta(x) = E_{Y|X}[Y \mid X=x]$$

- But, we only have samples:

$$(x_i, y_i) \overset{iid}{\sim} P_{XY}, \quad i=1,2,3,\dots,n$$

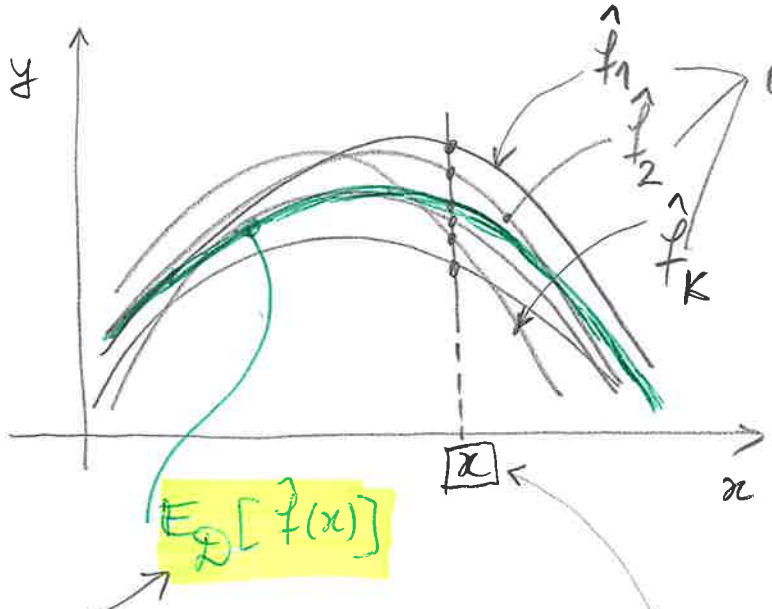
- And are restricted to a function class (e.g. linear)
- So, we compute:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

$\hat{f} \neq \eta(x)$, but this approach is the best we can do.

We care about future predictions: $E_{XY}\left[(Y-\hat{f}(X))^2\right]$

Note: Each sample/draw $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ results in different $\hat{f}$



obtained by applying the learning algorithm on each $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K$

(See Mostefa pp 63)
(See Bishop pp. 149)

Gives an "average function", which can be interpreted in the following operational way:

- Generate many datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K$ and apply learning algo to produce $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_K$
- Estimate "average function" for any x by: $\hat{f}(x) \cong \frac{1}{K} \sum_{j=1}^{K} \hat{f}_j(x)$
- We are viewing $\hat{f}(x)$ as a random variable; with the randomness coming from the randomness in the dataset.
- $E_{\mathcal{D}}[\hat{f}(x)]$ is the expected value of this random variable (for a particular x)

Note: The "average function" $\hat{f}(x)$ need not be in the model's hypothesis set, even though it is the average of functions that are.

Empirical Reconstruction Error (ERE)

- Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, the ERE is the finite-sample average:

$$\text{ERE}(f, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$: minimized in practice

→ This is what we actually compute from data.
→ In Statistical Learning Theory, it is called the "empirical risk"

Expected (population) Reconstruction Error

- if instead we take the expectation under the true data-generating P_{XY}, we get the population risk:

$$R(f) = \mathbb{E}_{XY}\left[(Y - f(X))^2\right]$$: analyzed in theory

→ This is what we would minimize if we had access to the true distribution!!
→ Its minimizer is the Bayes-optimal predictor $\eta(x) = \mathbb{E}[Y|X=x]$

Relationship between the two:

- $\text{ERE}(f, \mathcal{D})$ is a Monte Carlo approximation of $R(f)$
- By the Law of Large Numbers, for fixed f

$$\text{ERE}(f, \mathcal{D}) \xrightarrow[n\to\infty]{} R(f)$$

- So, minimizing ERE is the empirical counterpart of minimizing the true squared error loss/risk!!

Proof Sketch: • Start w/ $\text{ERE}(f; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$

• Take expectation wrt data-generating distribution:

$$\mathbb{E}_{XY}\left[\text{ERE}(f; \mathcal{D})\right] = \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2\right] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}\left[(y_i - f(x_i))^2\right] =$$

↑ by linearity of expectation

$$= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}\left[(y - f(x))^2\right] = \mathbb{E}\left[(y - f(x))^2\right]$$

↑ by iid: $\mathbb{E}(x_i), \forall i \equiv \mathbb{E}(x), \forall x$

Bias-Variance Trade off Decomposition

- We wish to break down the squared error of model $\hat{f}(\cdot)$ that makes a prediction of our target y given input data point $\boldsymbol{x}$.

- True predictor $E[Y|X]$ not available; unknown to us:

$$\eta(x) = \mathbb{E}_{Y|X}[Y|X=x]$$

- But, for given dataset $\mathcal{D}$, we can apply learning algo to produce

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

And the idea is to use this as a substitute/surrogate for the ground true predictor.

- The generalization error we want to minimize is the usual expected squared error:

$$\mathbb{E}_{XY}[(Y-\eta(X))^2] \overset{\text{based on what we discussed on paper 2,3,3}}{\cong} \mathbb{E}_{XY}[(Y-\hat{f}(X))^2] \overset{\text{use tower trick}}{=}$$

(see also MIT Notes pp.23)

$$= \mathbb{E}_{Y|X}\left[\mathbb{E}_{\mathcal{D}}\left[(Y-\hat{f}_{\mathcal{D}}(x))^2\right] \middle| X=x\right]$$

$\hat{f}_{\mathcal{D}}$: to acknowledge dependence of our fitted model $\hat{f}$ on the dataset $\mathcal{D}$.

$$= \mathbb{E}_{Y|X}\left[\mathbb{E}_{\mathcal{D}}\left[\underbrace{(Y-\eta(x))}_{a} + \underbrace{(\eta(x)-\hat{f}_{\mathcal{D}}(x))}_{b}\right)^2\right] \middle| X=x\right]$$

$$= \mathbb{E}_{Y|X}\left[\mathbb{E}_{\mathcal{D}}\left[\underbrace{(Y-\eta(x))^2}_{a^2} + \underbrace{2(Y-\eta(x))(\eta(x)-\hat{f}_{\mathcal{D}}(x))}_{2ab \to 0 \text{ (see next page)}} + \underbrace{(\eta(x)-\hat{f}_{\mathcal{D}}(x))^2}_{b^2}\right] \middle| X=x\right]$$

$$= \underbrace{\mathbb{E}_{Y|X}[(Y-\eta(x))^2|X=x]}_{\text{irreducible error}} + \underbrace{\mathbb{E}_{\mathcal{D}}[(\eta(x)-\hat{f}_{\mathcal{D}}(x))^2]}_{\text{learning error}}$$

← we want to minimize, bcoz we can control!!!

Caused by (stochastic) label noise

Caused by:
- either using too 'simple' model
- or not enough data to learn model accurately

Show: $2ab \to 0$

$$\mathbb{E}_{Y|X}\left[\mathbb{E}_{\mathcal{D}}\left[2\underbrace{(Y-\eta(x))}_{\text{no } \mathcal{D}}\underbrace{(\eta(x)-\hat{f}_{\mathcal{D}}(x))}_{\text{no } Y}\right] \mid X=x\right]$$

$$\mathbb{E}_{Y|X}\left[\mathbb{E}_{\mathcal{D}}\left[2\underbrace{(Y-\eta(x))}\right] \mid X=x\right]$$

$$\mathbb{E}_{Y|X}\left[Y - \mathbb{E}[Y|X=x] \mid X=x\right] = \quad \eta(x)=\mathbb{E}_{Y|X}[Y|X=x]$$

$$= \mathbb{E}_{Y|X}[Y|X=x] - \mathbb{E}_{Y|X}[Y|X=x]$$

$$= 0$$

Y & $\mathcal{D}$ are independent of each other w.r.t. the sources of randomness we are considering.

Now, looking at the **Learning error** seperately:

$$E_{\mathcal{D}}\left[(\eta(x)-\hat{f}_{\mathcal{D}}(x))^2\right] = E_{\mathcal{D}}\left[(\eta(x)-E_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] + E_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]-\hat{f}_{\mathcal{D}}(x))^2\right] =$$

a b — added and substracted trick

$$= E_{\mathcal{D}}\left[(\eta(x)-E_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2 + 2(\eta(x)-E_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])\cdot(E_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]-\hat{f}_{\mathcal{D}}(x)) + (E_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]-\hat{f}_{\mathcal{D}}(x))^2\right]$$

a^2 — $ab \to$ because $b \to 0$ — b^2

$$= \underbrace{(\eta(x)-E_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2}_{\text{bias squared}} + \underbrace{E_{\mathcal{D}}\left[(E_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]-\hat{f}_{\mathcal{D}}(x))^2\right]}_{\text{Variance}}$$

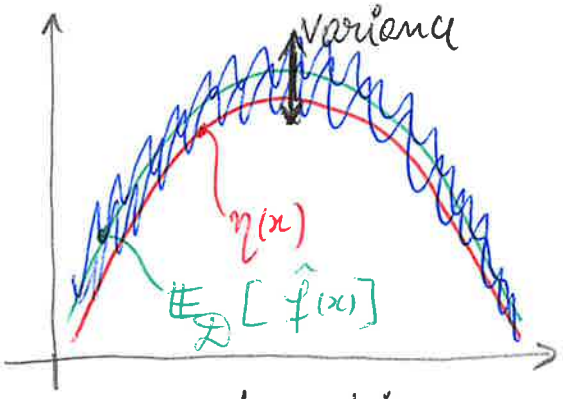
(See MIT Notes pp. 23)
(See Bishop, pp. 149)

- we arrived at important result:

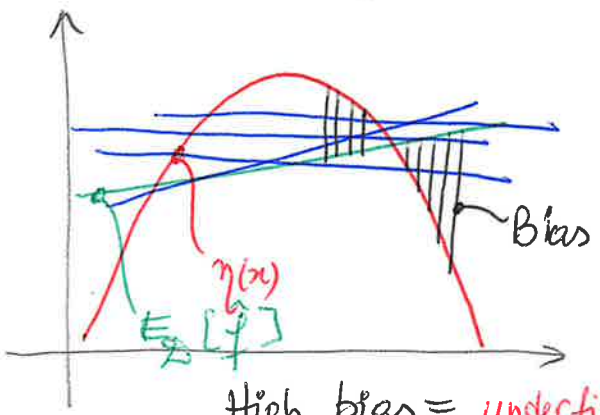
Learning error = bias^2 + Variance

Bias – Represent the extent to which the average prediction over **all** datasets differs from the *ideal* regression function/predictor $\eta(x)$
– if it is large $\Rightarrow$ function class $\hat{f}$ cannot describe data well.

Variance – for a particular sample dataset $\mathcal{D} \sim P_{XY}$, indicates how different is $\hat{f}_{\mathcal{D}}$ from average $\hat{f}$; if varying wildly $\Rightarrow$ error!



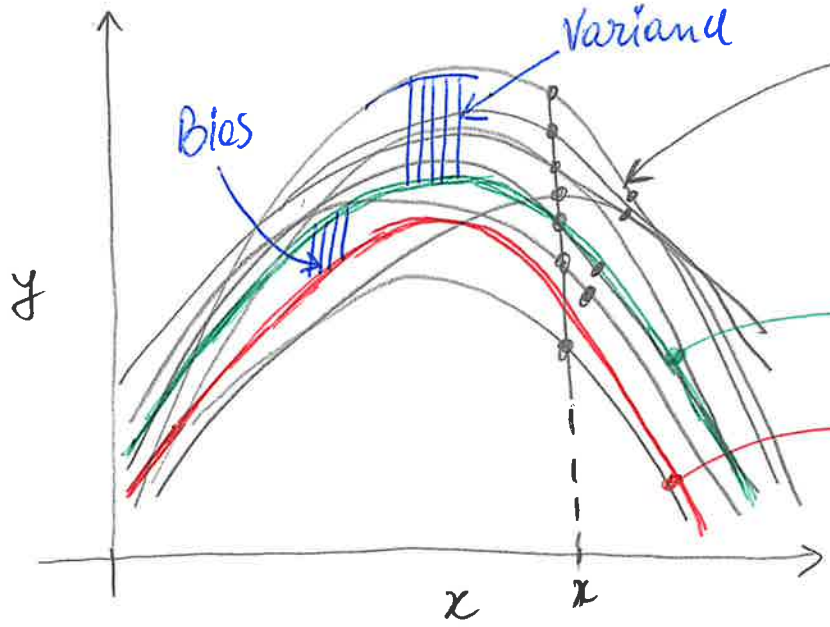
Low bias



High bias ≡ underfitting
High Variance ≡ overfitting.

$$\mathbb{E}_{\mathcal{D}}\left[\eta(x) - \hat{f}_{\mathcal{D}}(x)\right)^2] = \left(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]\right)^2 + \mathbb{E}_{\mathcal{D}}\left[\left(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)\right)^2\right]$$

Learning error = Biased squared + Variance.



$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

$$\mathbb{E}_{\mathcal{D}}[\hat{f}(x)]$$

$$\eta(x) = \mathbb{E}_{Y|X}[Y \mid X = x]$$

So, **True Generalization Error:**

(notation in Bishop, pp. 149)

$$\mathbb{E}_{Y|X}\left[\mathbb{E}_{\mathcal{D}}[(Y - \hat{f}_{\mathcal{D}}(x))^2] \mid X = x\right] = \mathbb{E}_{Y|X}\left[(Y - \eta(x))^2 \mid X = x\right] +$$

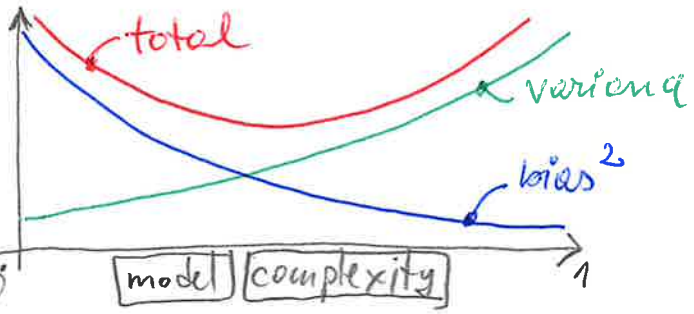
irreducible error

Learning error

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

Biased squared | Variance

- Main takeaway:
- Model w/ optimal **error** predictive ability is the one that leads to best balance between bias & variance:



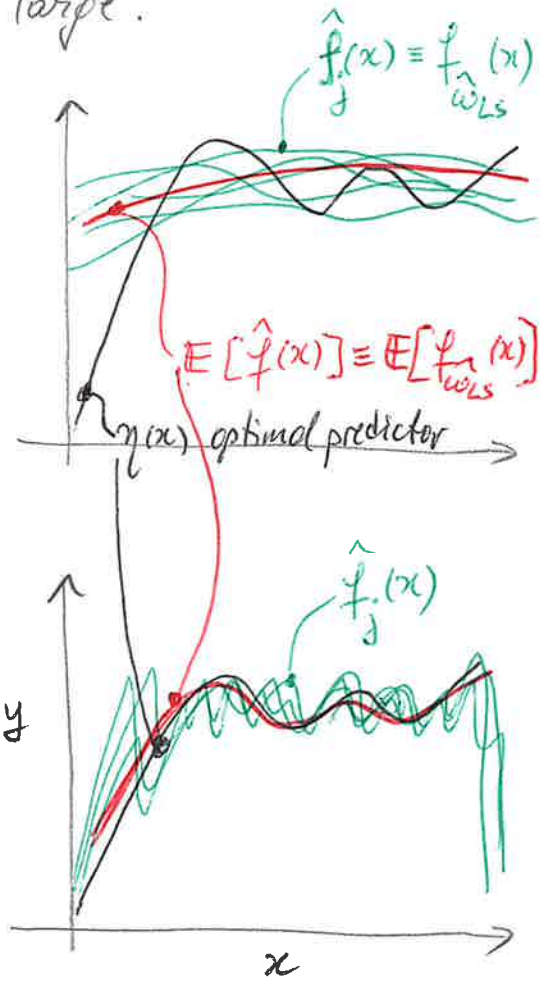
(*)

"Bias" queries wrong in Training

"Variance" queries wrong in Testing.

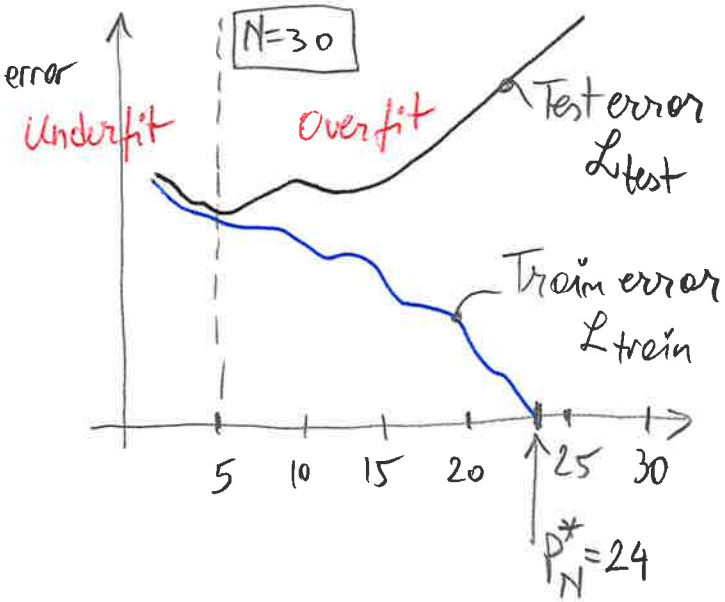
Summary

→ With simple model (model complexity is low)
- Bias^2 of model of our predictor is large.
- variance of predictor is small.
- if more samples (larger n):
  - Bias remains the same
  - Variance goes down.
  - overall test-error the same.



→ With complex model (higher than that of optimal predictor $\eta(x)$)
- Bias^2 of our predictor is small
- Variance is large
- if more samples (larger n):
  - Bias remains the same
  - Variance goes down
  - Overall test error goes down.

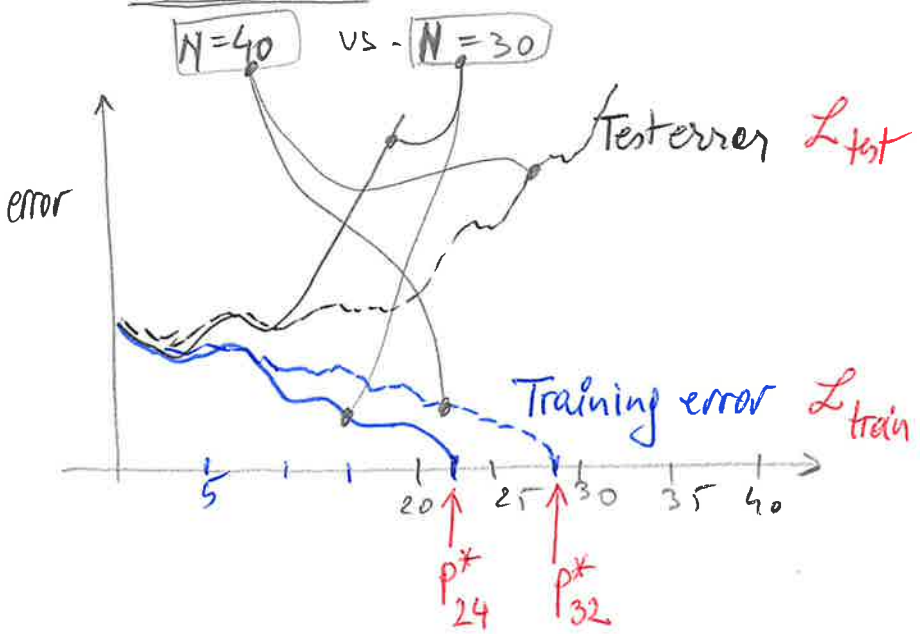
→ Optimal model complexity depends on data size (i.e., degree of polynomial in our code demo)



- Given sample size N, there is a threshold p_N^* where train error = 0
- Training error is always monotonically non-increasing
- Test error goes down, then up and fluctuates.

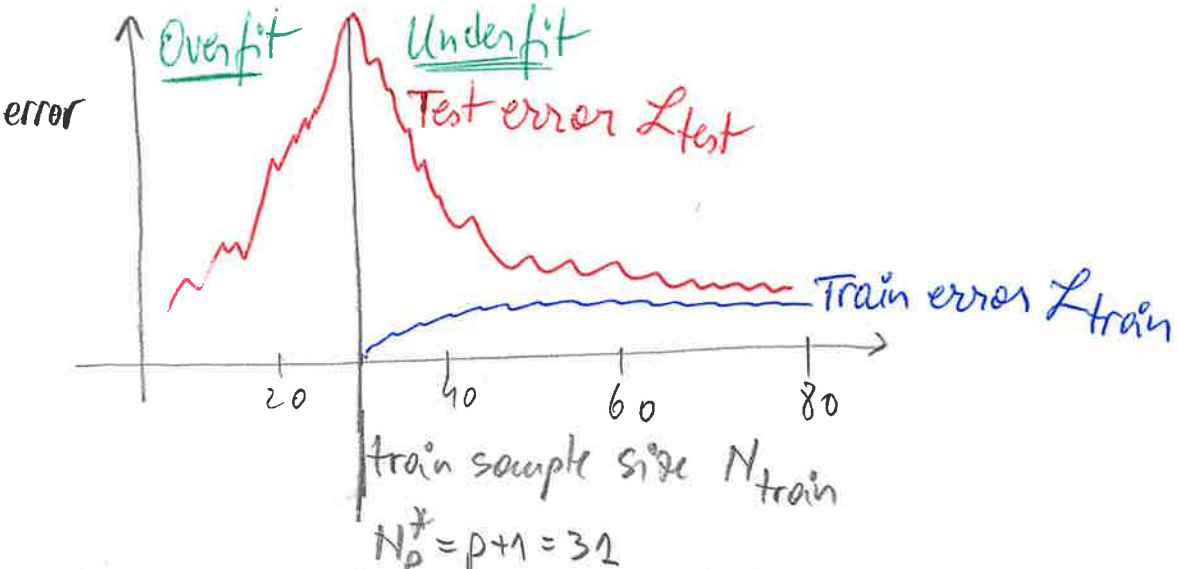
Variance decreases with more data

→ lets us fit more complex models



- Threshold p^*_N moves to the right, as dataset size increases.
- Training error tends to increase, because more points need to fit.
- Test error tends to decrease, because variance decreases.

Choose model complexity $p=30$, vary data size n



- Threshold N^*_p below which training error is zero (extreme overfit)
- Above threshold, test error tends to decrease.
- Training error tends to increase (harder to fit so much data w/ simple model!)