Evaluating Predictors / Models

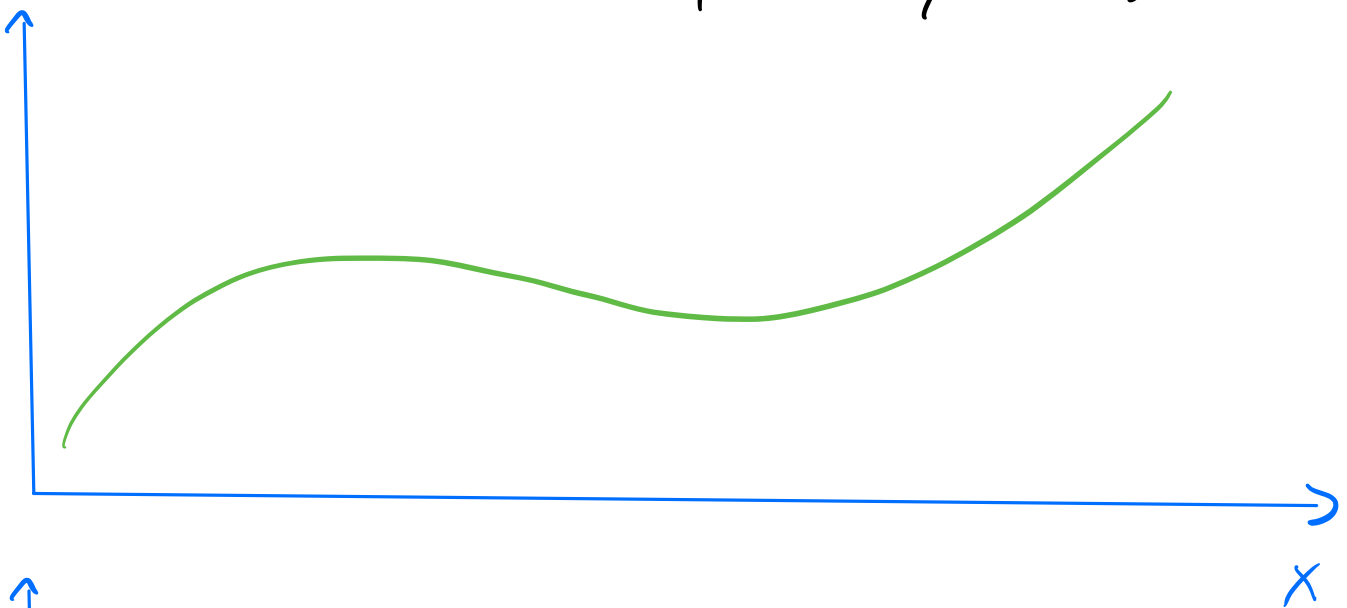
$$\mathcal{A}(D) = \hat{f}_D = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \hat{L}(f) \qquad D \text{ is a r.v.}$$

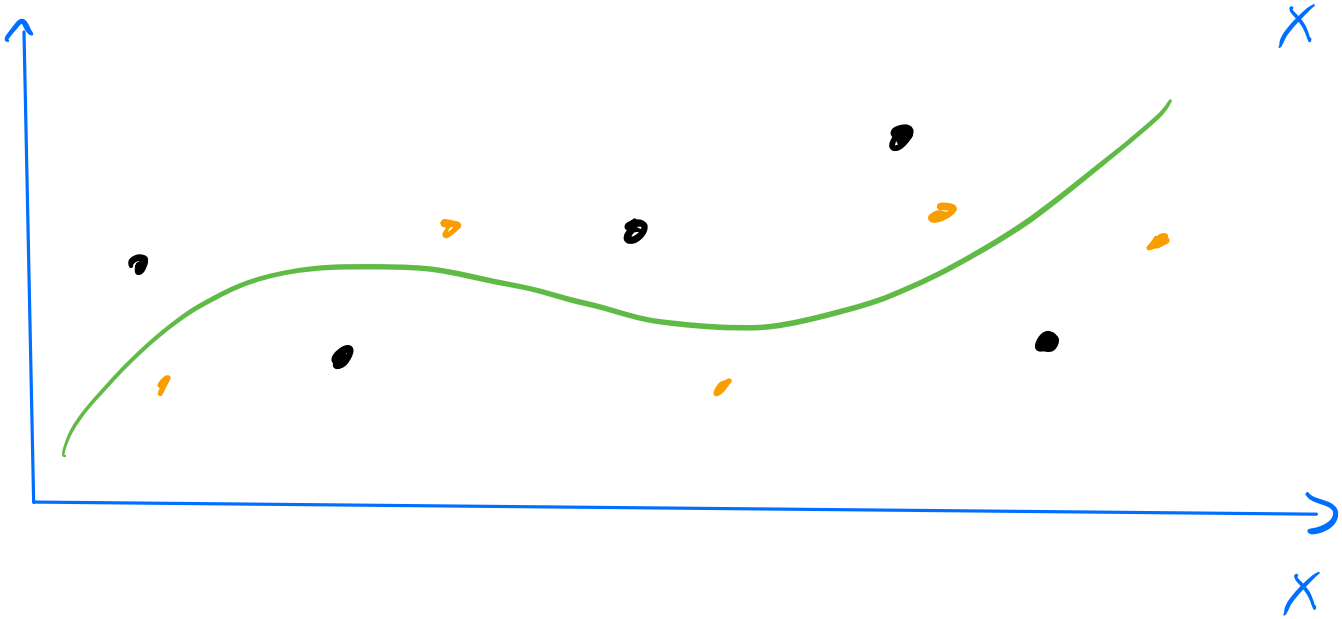
$$\hat{L}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i), \quad L(f) = \mathbb{E}[\ell(f(X), Y)]$$

$$\hat{L}(\hat{f}_D) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_D(X_i), Y_i)$$

Pick $f \in \mathcal{F}$ then gather D

(i.e. f is chosen independently of D)

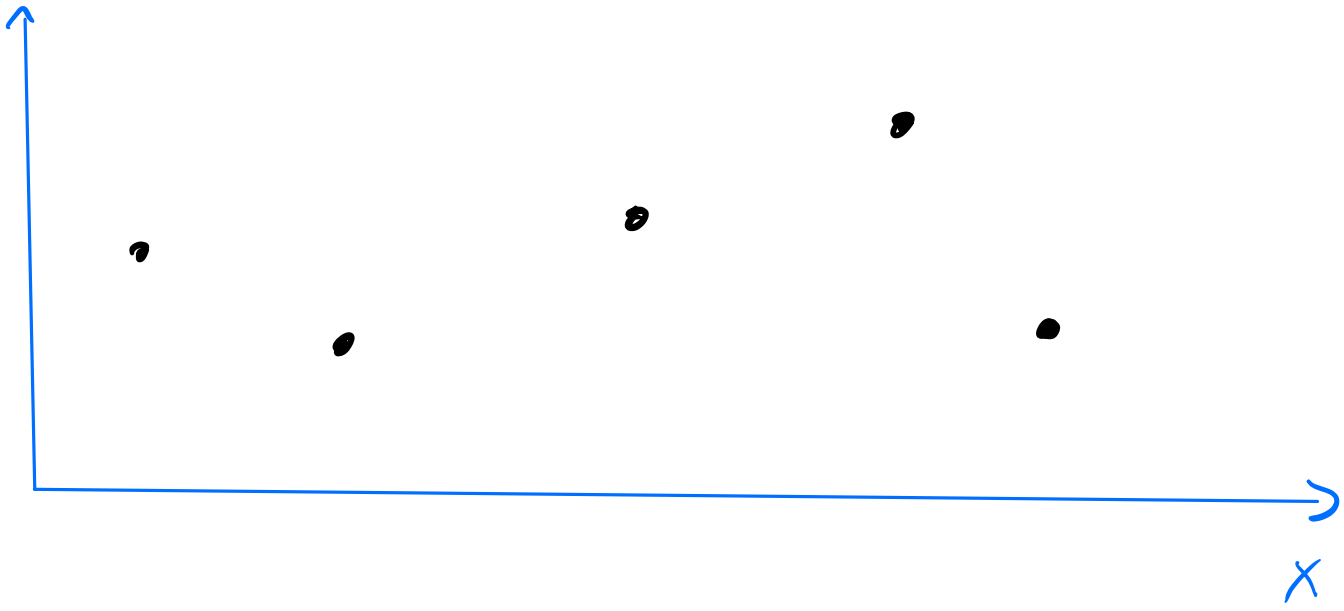


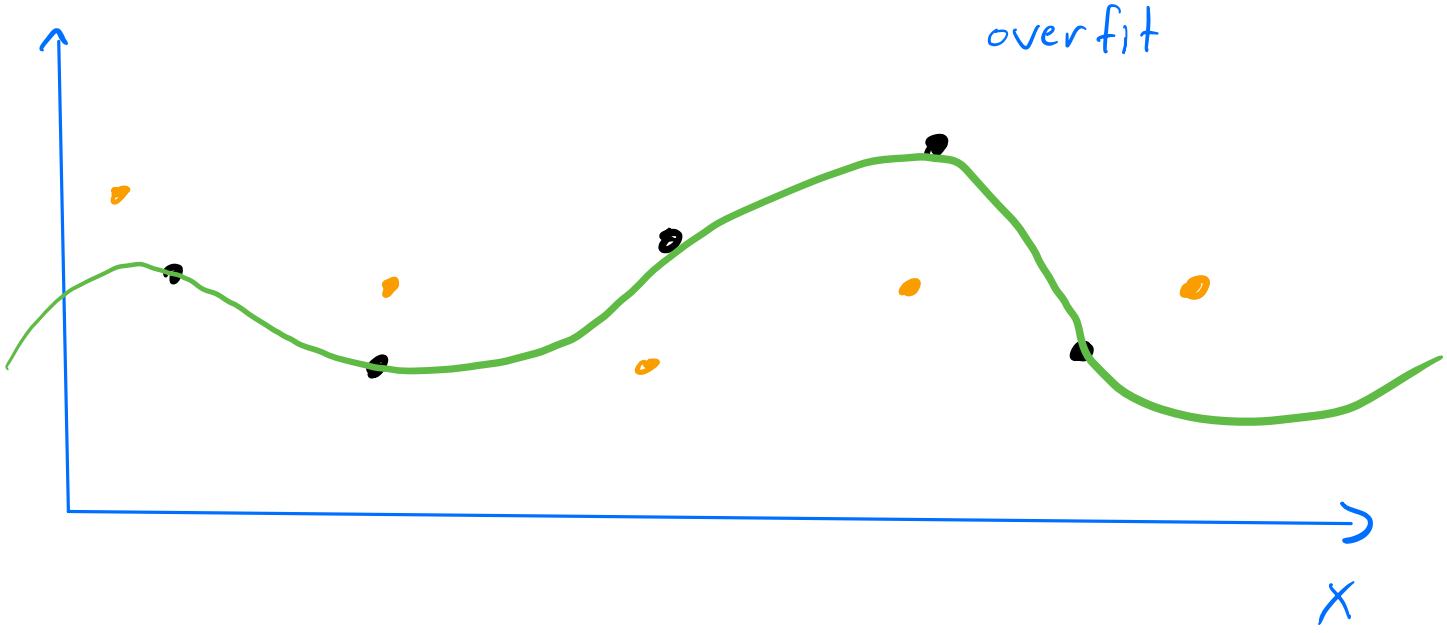


Then $E[\hat{L}(f)] = L(f)$

$$\text{Var}[\hat{L}(f)] = \frac{1}{n}\text{Var}[\ell(f(\vec{X}), Y)]$$

We are gathering D first and then picking $\hat{f}_D \in \mathcal{F}$! (i.e. $\hat{f}_D$ depends on D)





In general

$$\mathbb{E}[\hat{L}(\hat{f}_0)] \neq \mathbb{E}[L(\hat{f}_0)]$$

$\ell(\hat{f}_0(\vec{X}_i), Y_i)$ are not i.i.d.

$\hat{f}_0$ depends $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$

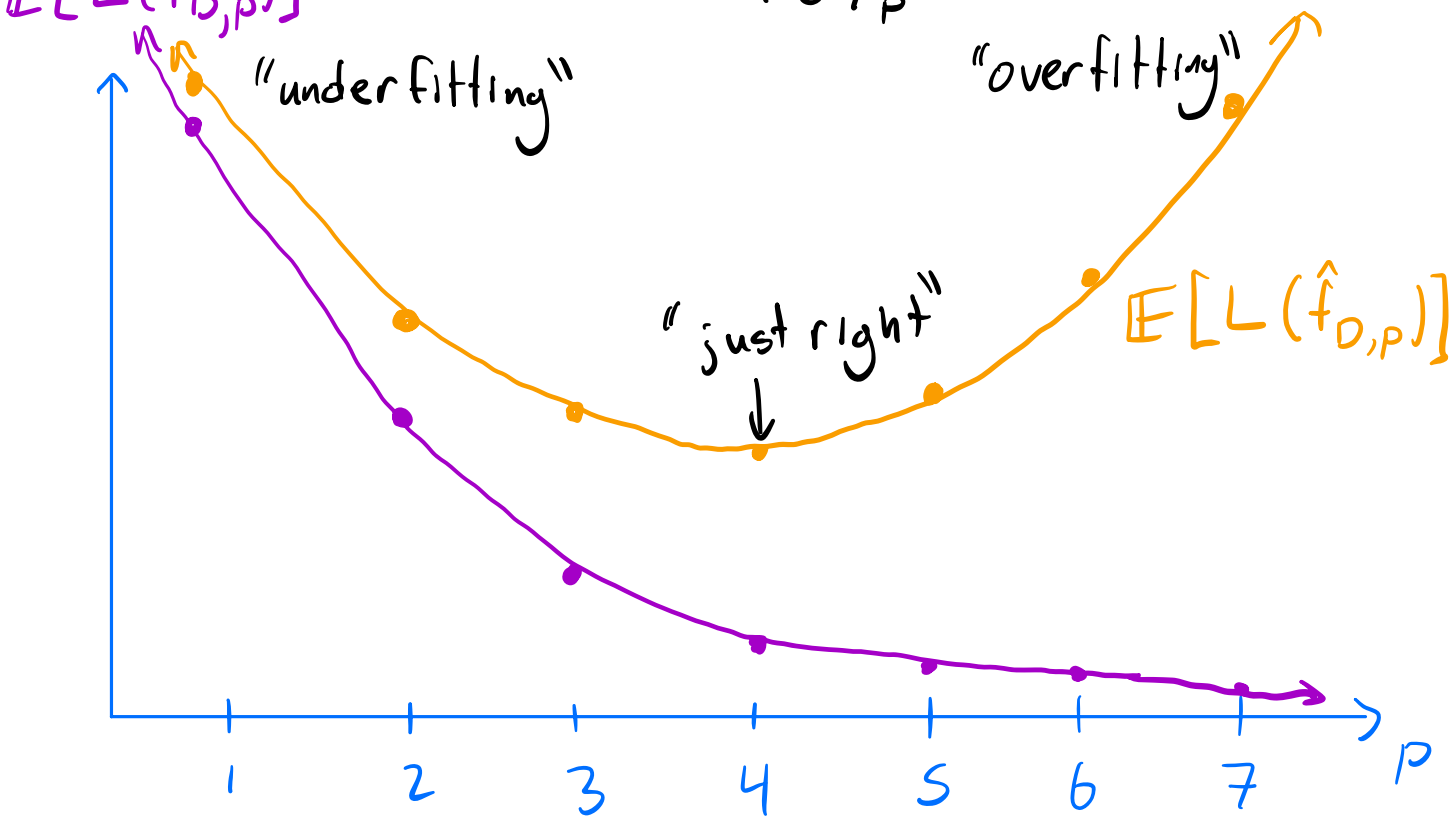
It can be shown that

$\mathbb{E}[L(\hat{f}_0)] - \mathbb{E}[\hat{L}(\hat{f}_0)]$ is:

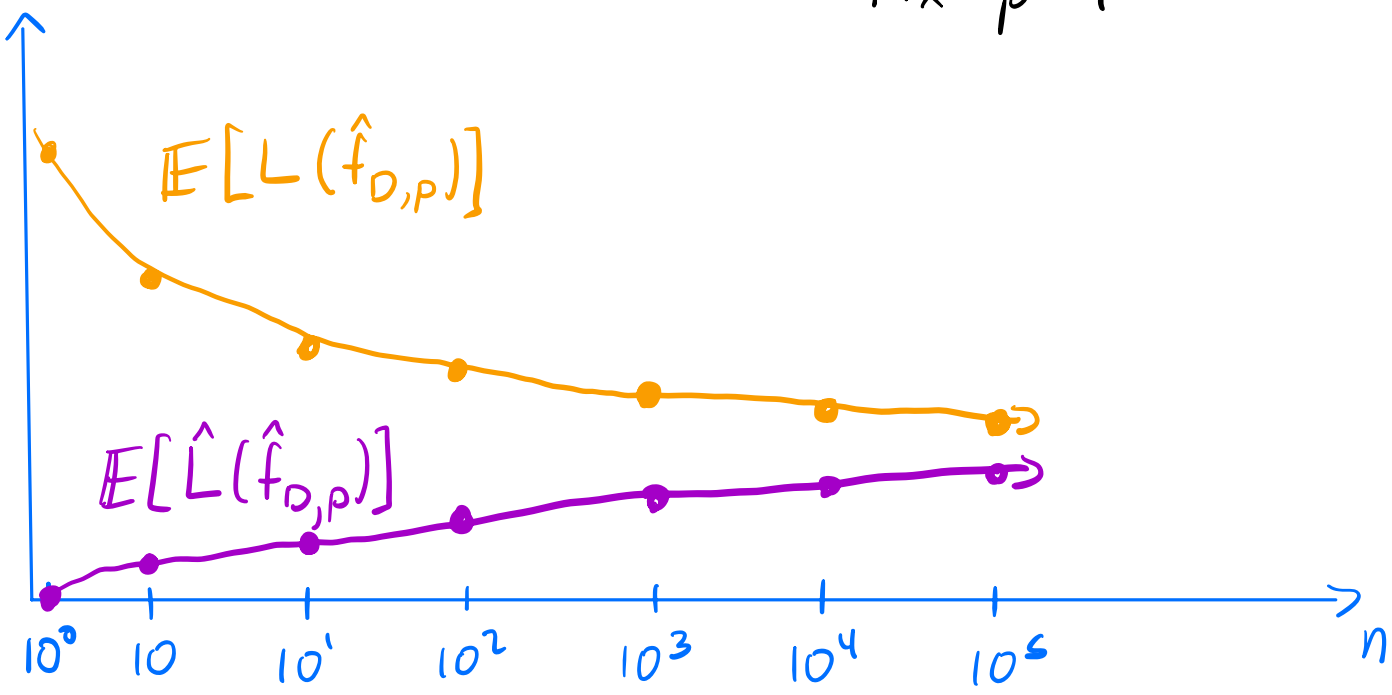
- non-negative (i.e. $\mathbb{E}[\hat{L}(\hat{f}_0)] \leq \mathbb{E}[L(\hat{f}_0)]$)
- increases as $\mathcal{F}$ gets more complex
- decreases as n (# of data points) increases

$$\hat{f}_{D,p} = \underset{f \in \mathcal{F}_p}{\operatorname{argmin}} \hat{L}(f)$$

Fix $n = 100$



Fix $p = 4$



In practice we only have a fixed dataset D

Train on a D_{train} and evaluate on D_{test}:

$$D_{train} = ((\vec{X}_1, Y_1), \ldots, (\vec{X}_{n-m}, Y_{n-m}))$$

$$D_{test} = ((\vec{X}_{n-m+1}, Y_{n-m+1}), \ldots, (\vec{X}_n, Y_n))$$

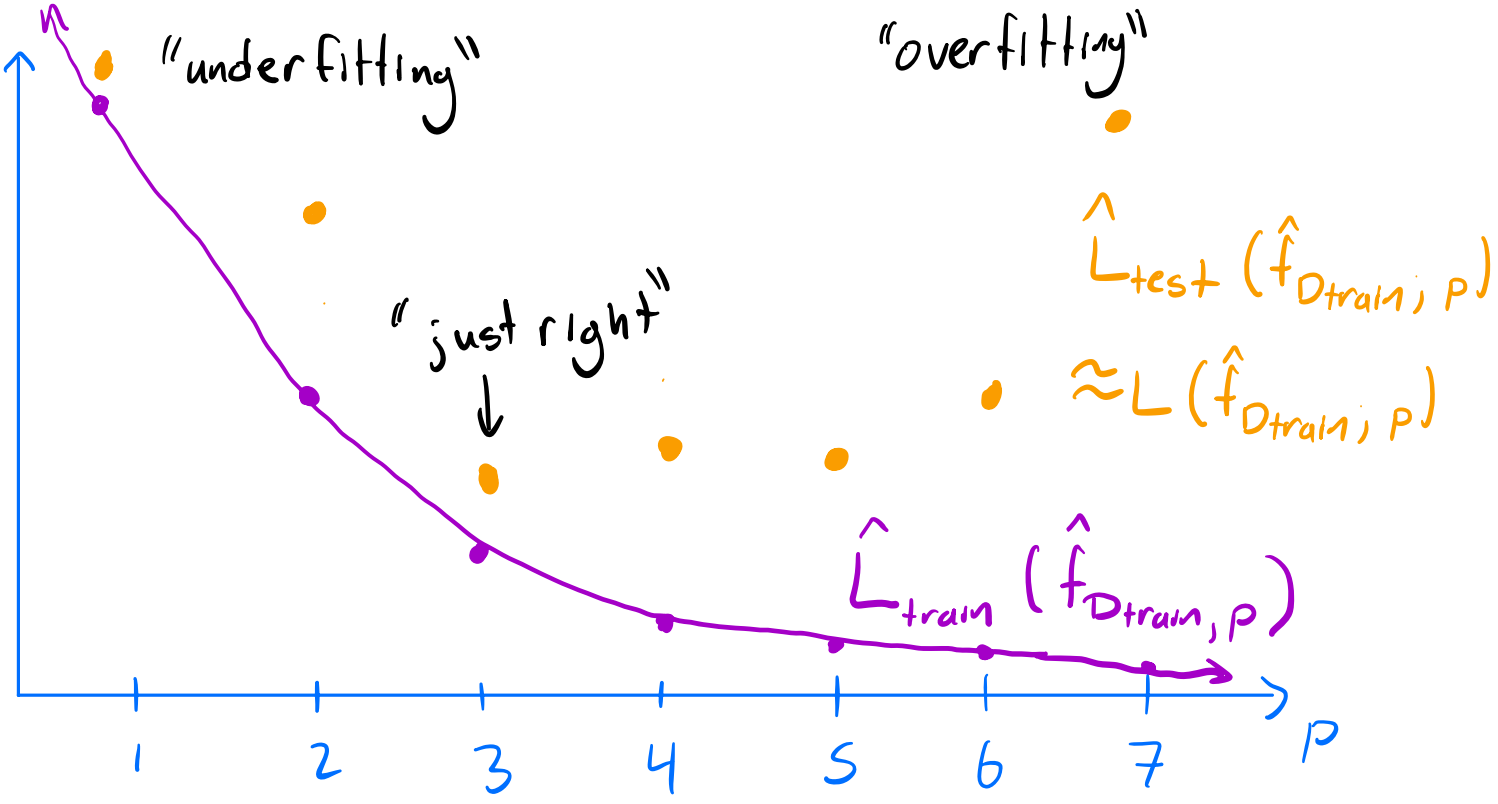
$|D_{train}| = n-m$, $|D_{test}| = m$

Then $\mathcal{A}(D_{train}) = \hat{f}_{D_{train}} = \underset{f \in \mathcal{F}}{\arg\min}\ \hat{L}_{train}(f)$

$$\hat{L}_{train}(\hat{f}_{D_{train}}) = \frac{1}{n-m} \sum_{i=1}^{n-m} \ell(\hat{f}_{D_{train}}(X_i), Y_i)$$

$$\hat{L}_{test}(\hat{f}_{D_{train}}) = \frac{1}{m} \sum_{i=n-m+1}^{n} \ell(\hat{f}_{D_{train}}(X_i), Y_i)$$

$$\mathcal{A}_p(D_{train}) = \hat{f}_{D_{train},p} = \underset{f \in \mathcal{F}_p}{\operatorname{argmin}} \hat{L}_{train}(f)$$

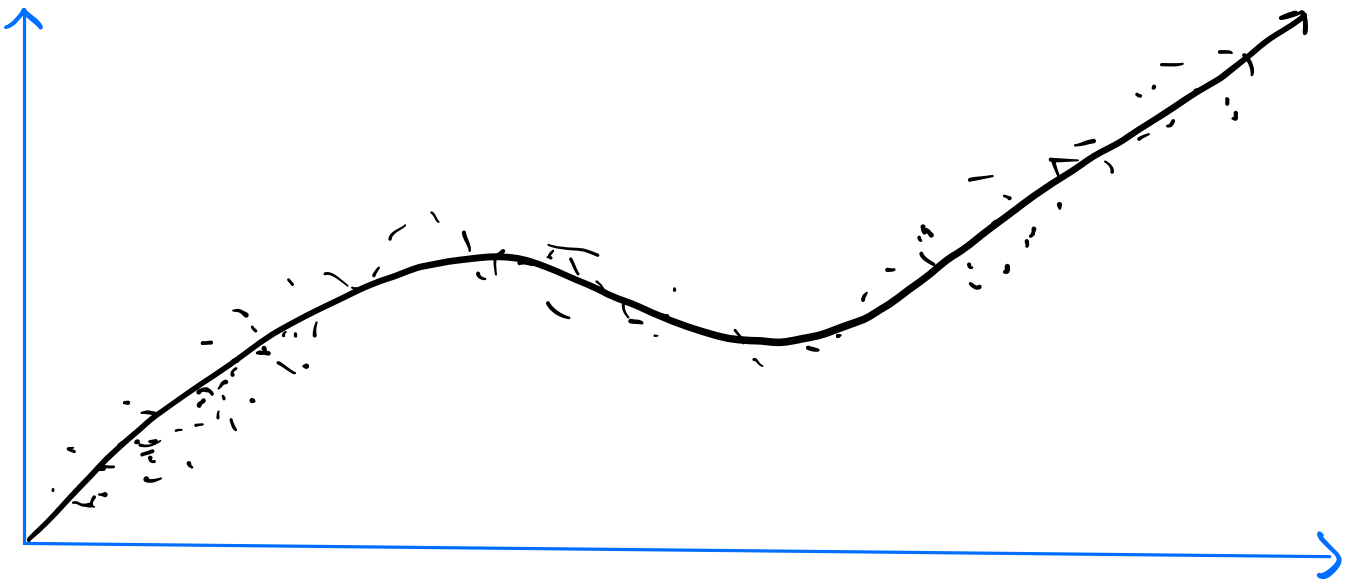


Decomposing $\mathbb{E}[L(A(D))]$ into Error Terms

- You know $\mathbb{P}_{X,Y}$

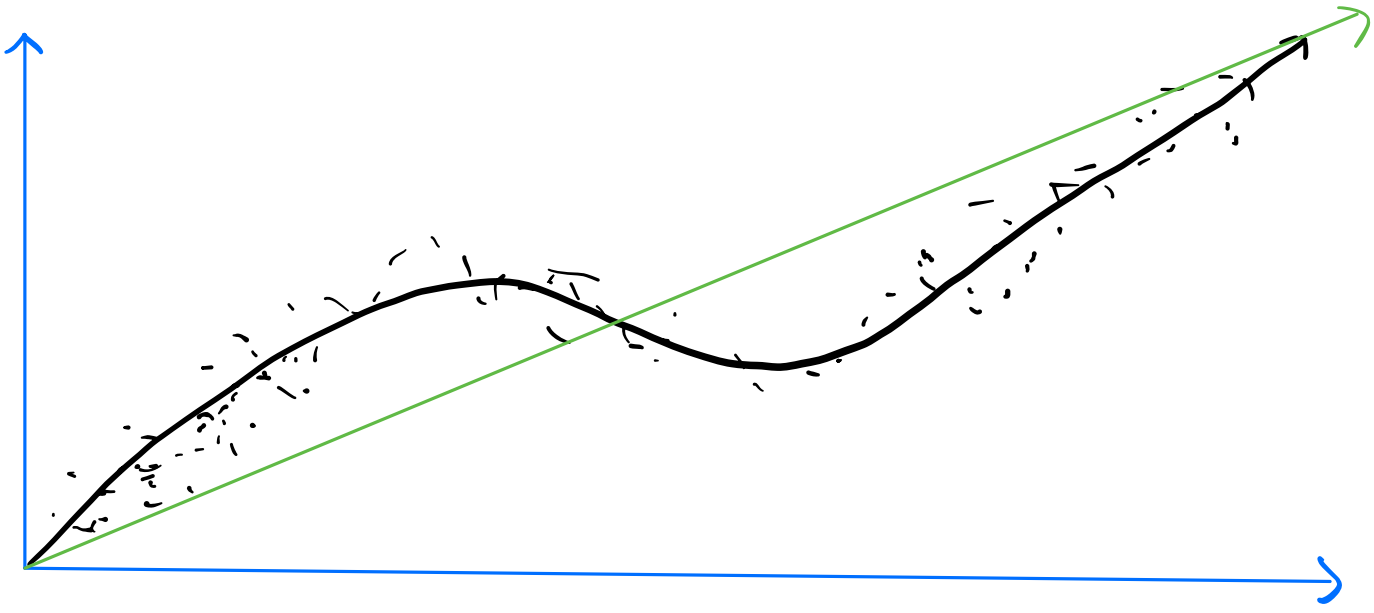
$$f_{Bayes} = \underset{f \in \{f \mid f: X \to Y\}}{\operatorname{argmin}} L(f)$$

Bayes optimal predictor



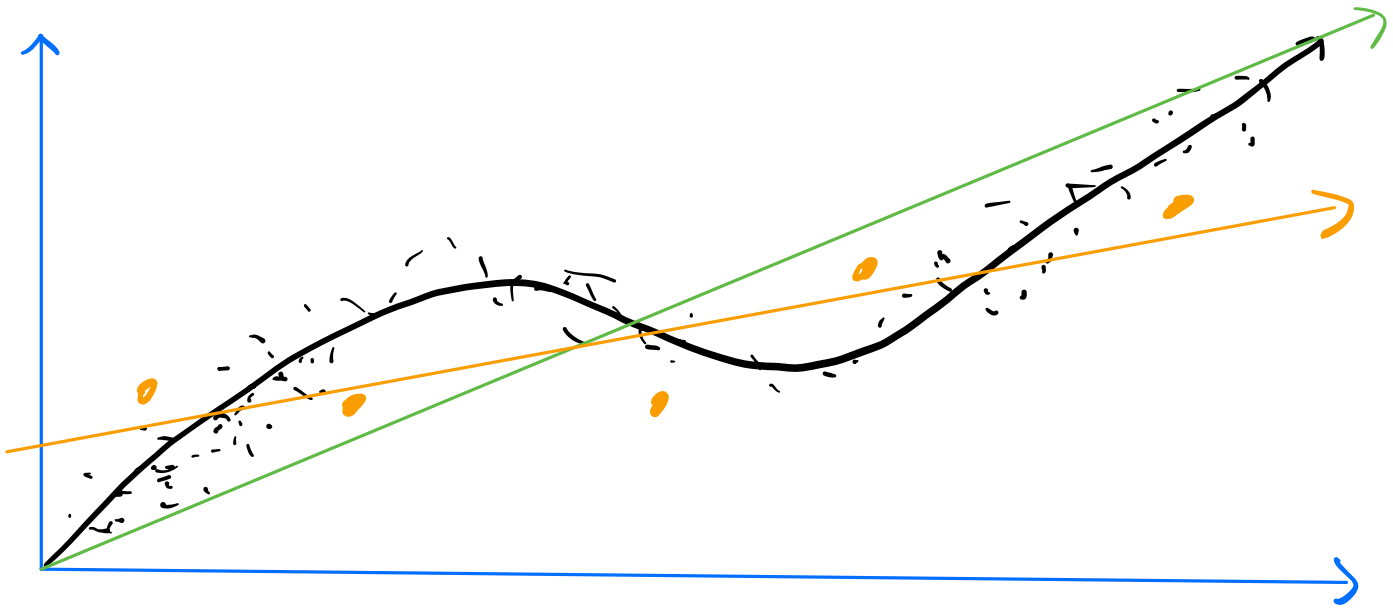
- You know $\mathbb{P}_{\vec{X},Y}$ but we have to output from $\tilde{F}_1$

$$f^* = \underset{f \in \tilde{F}_1}{\operatorname{argmin}} L(f)$$

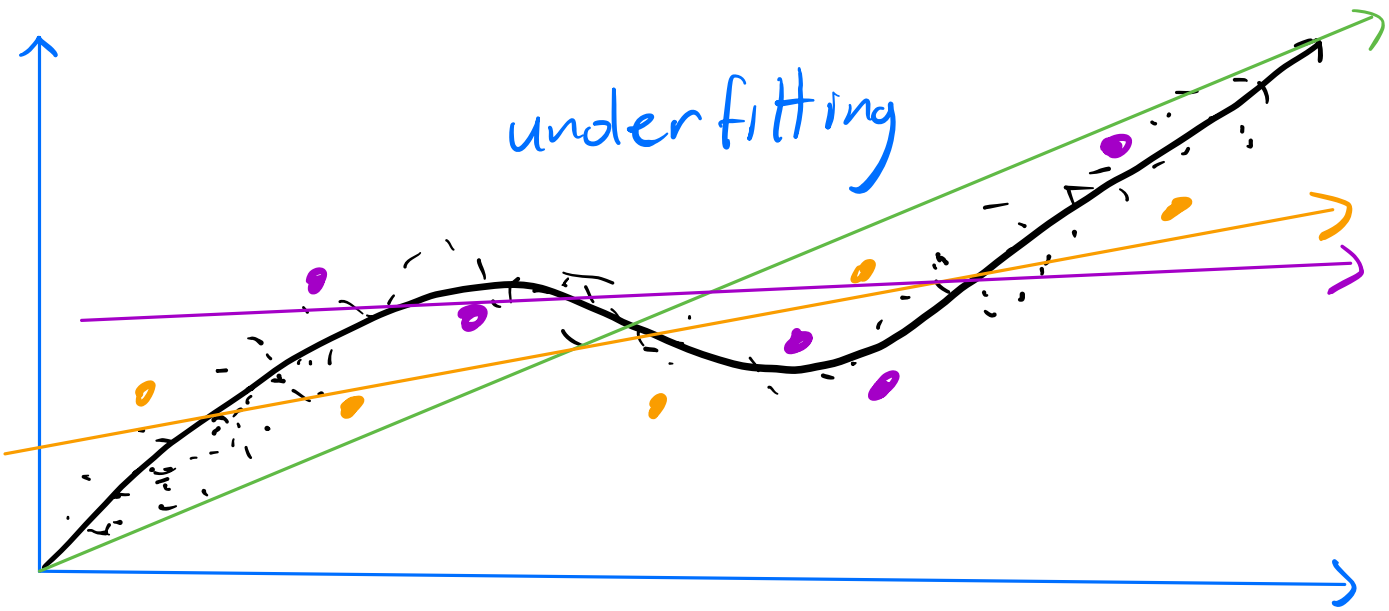


- You don't know $P_{X,Y}$ but we have to output from $\tilde{F}_1$ and we have a dataset D_1

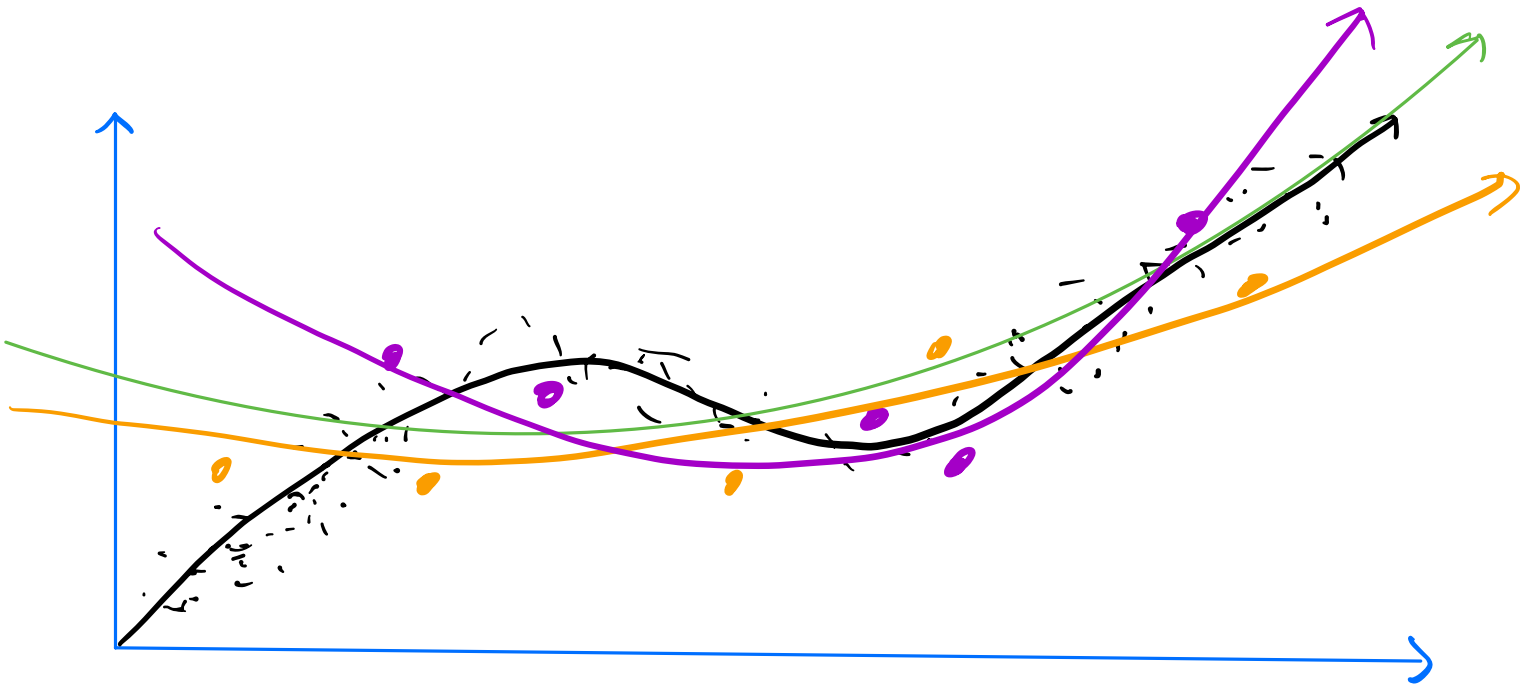
$$\hat{f} = \underset{f \in \tilde{F}_1}{\operatorname{argmin}} \hat{L}(f)$$



D_2

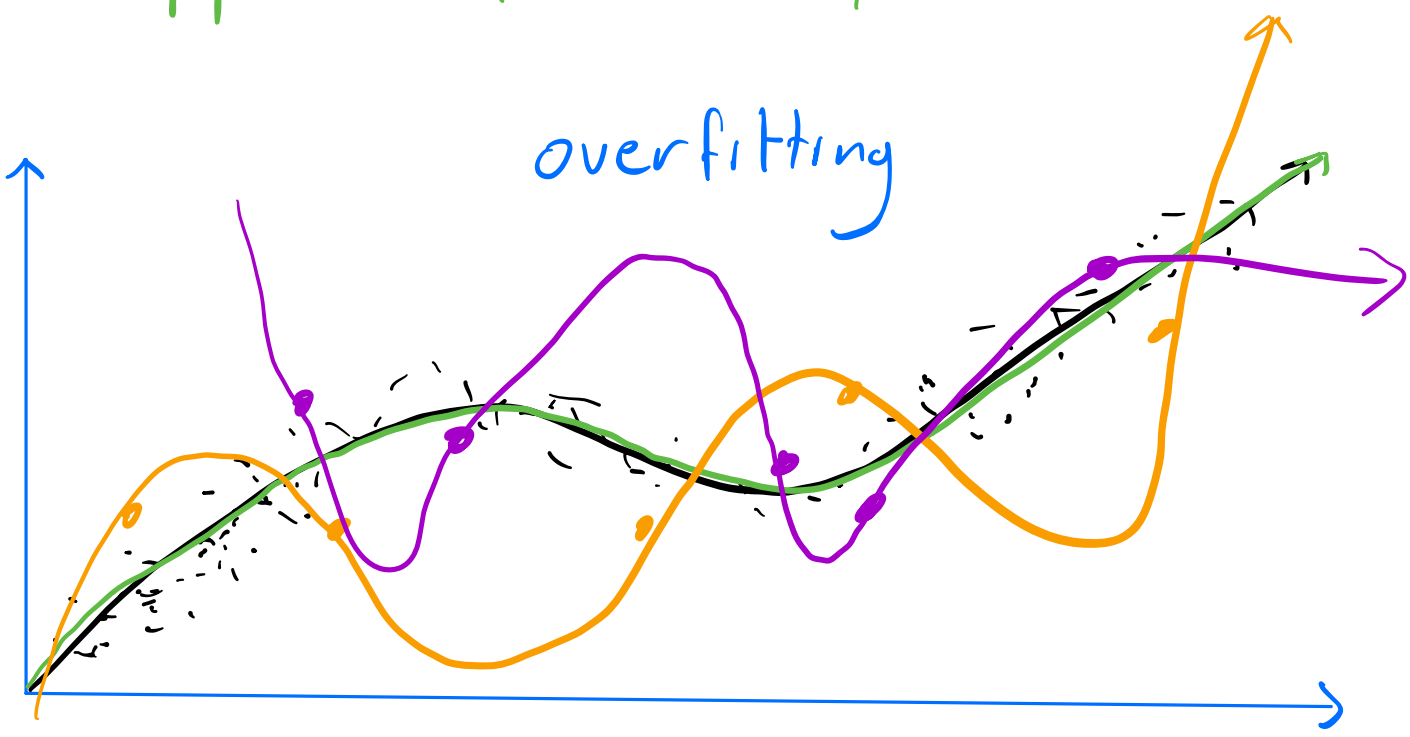


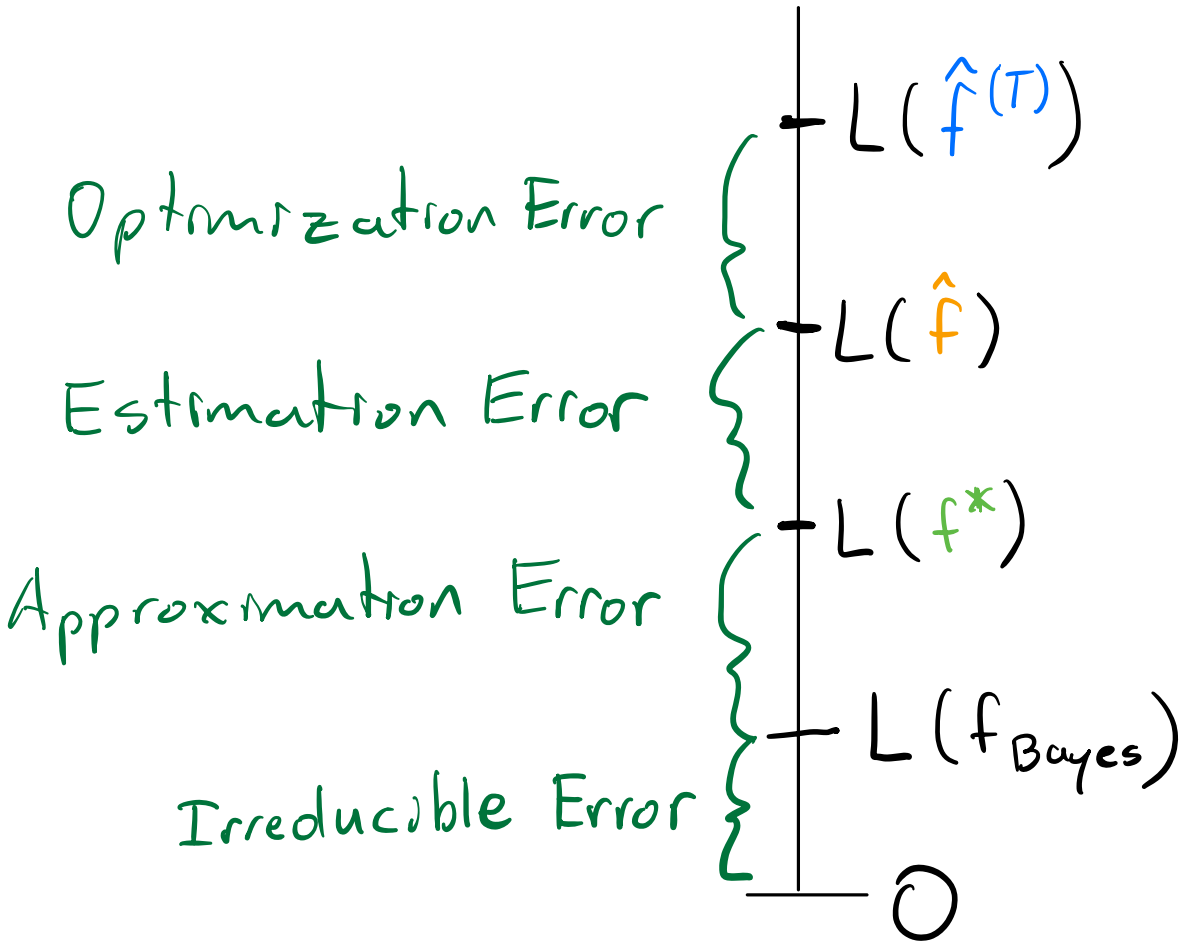
Suppose $\mathcal{F}_2$



Suppose $\mathcal{F}_{10}$ and $f_{Bayes} \in \mathcal{F}_{10}$

overfitting




$L(\hat{f}^{(T)})$
Optimization Error
$L(\hat{f})$
Estimation Error
$L(f^*)$
Approximation Error
$L(f_{Bayes})$
Irreducible Error
0

$f_{Bayes} = \underset{f \in \{f \mid f: X \to Y\}}{\operatorname{argmin}} L(f)$ $\quad f^* = \underset{f \in \mathcal{F}_1}{\operatorname{argmin}} L(f)$ $\quad \hat{f}_D = \underset{f \in \mathcal{F}_1}{\operatorname{argmin}} \hat{L}(f)$

$$\mathbb{E}[L(\hat{f}_D)] = \underbrace{\mathbb{E}[L(\hat{f}_D)] - L(f^*)}_{\text{Estimation error (EE)}} + \underbrace{L(f^*) - L(f_{Bayes})}_{\text{Approximation Error (AE)}} + \underbrace{L(f_{Bayes})}_{\text{Irreducible Error}}$$

What affects the errors:

Irreducible Error: Due to inherent noise in the labels

- Can decrease if you gather more informative features

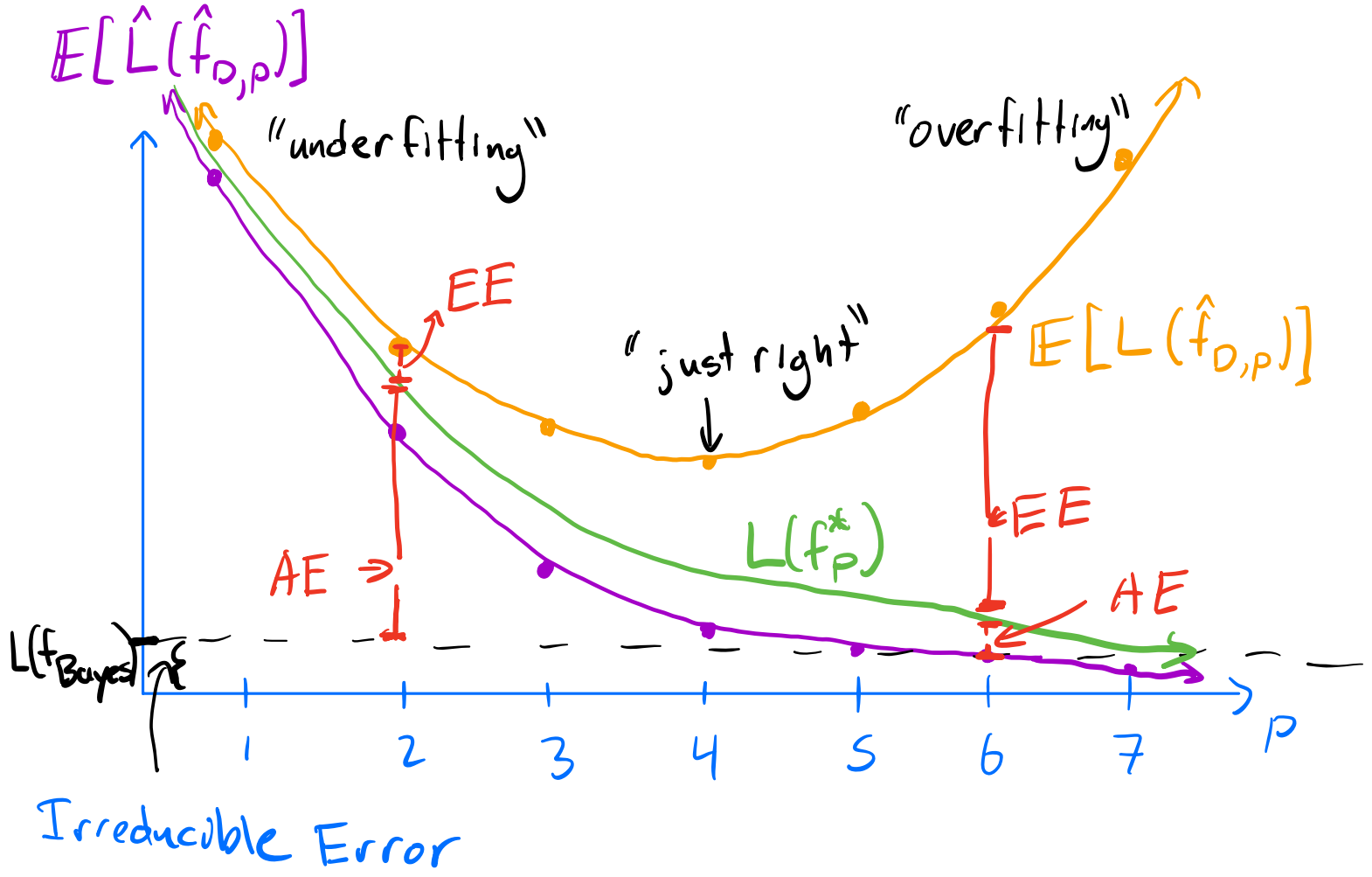
Approximation Error: Due to function class $\mathcal{F}$

- Decrease or stays the same if you make $\mathcal{F}$ larger

Estimation Error: Due to using a dataset D

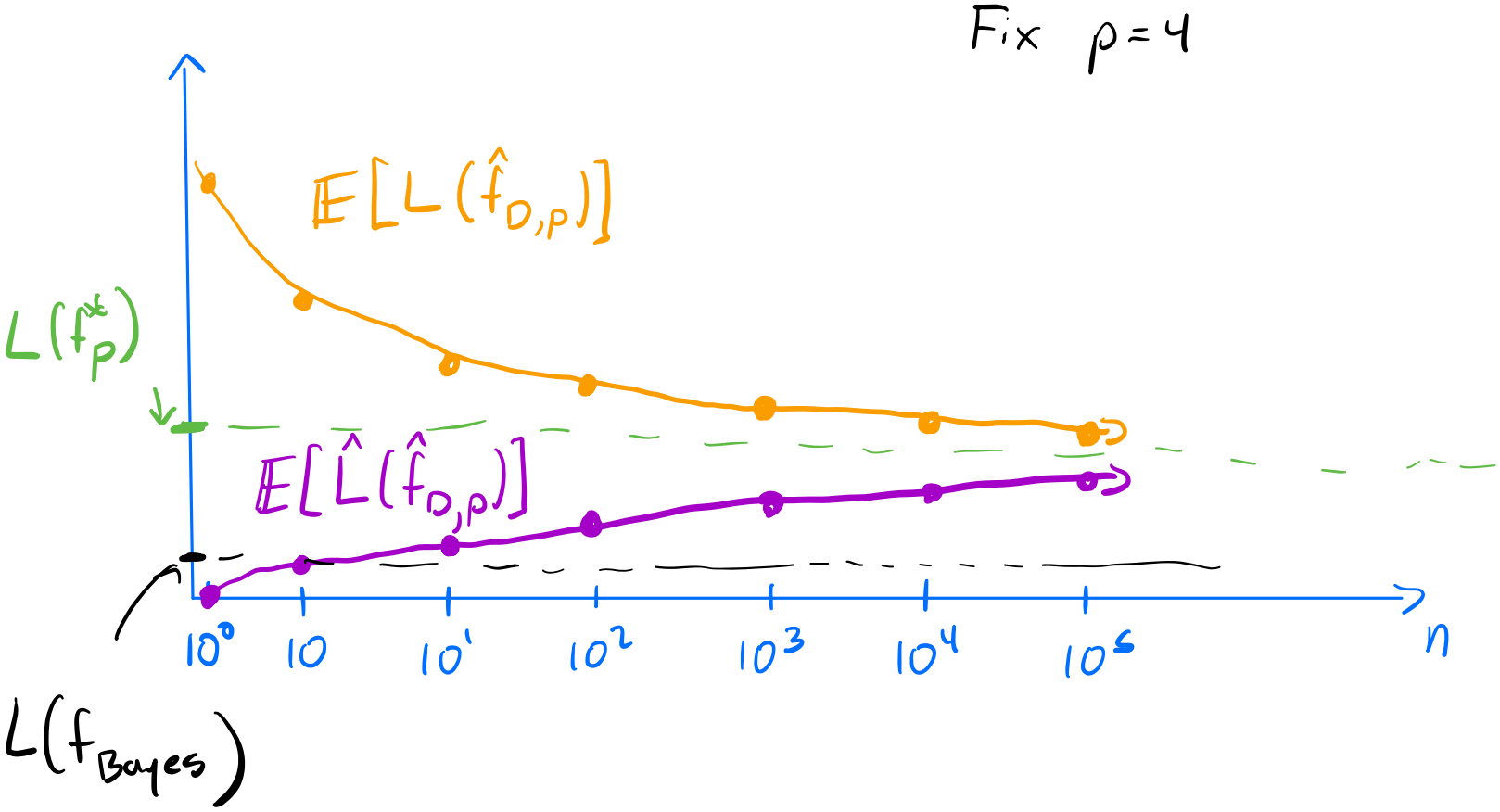
- Decrease or stays the same if n increases
- Increase or stay the same if $\mathcal{F}$ gets larger

$$\hat{f}_{D,p} = \underset{f \in \mathcal{F}_p}{\operatorname{argmin}} \ \hat{L}(f) \qquad \text{Fix } n=100$$



Underfitting: High AE, Low EE

Overfitting: Low AE, High EE



Bias-Variance Decomposition

$$\mathbb{E}[L(\hat{f}_D)] = \mathbb{E}[L(\hat{f}_D)] - L(f_{Bayes}) + L(f_{Bayes})$$

regression and squared loss

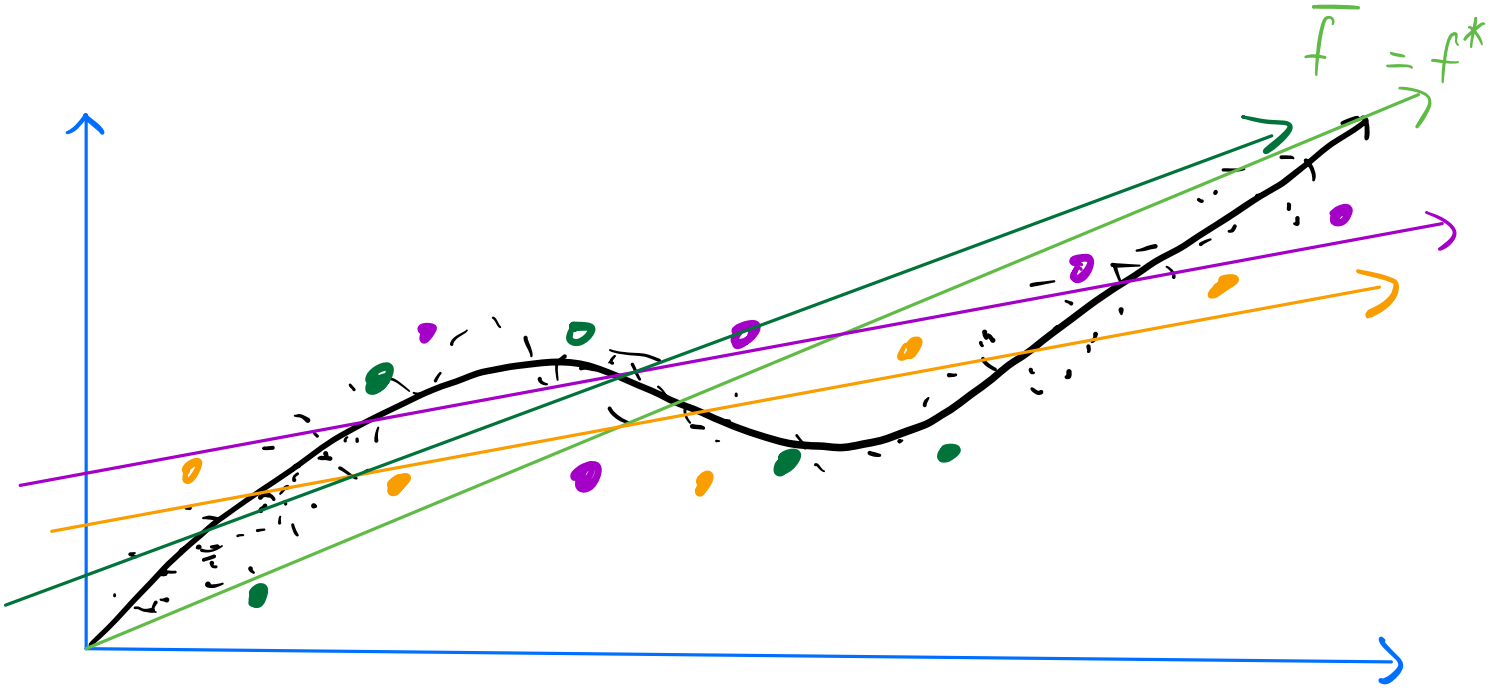
$$= \underbrace{\mathbb{E}\left[\mathbb{E}\left[(\hat{f}_D(x) - \bar{f}(x))^2 \mid X\right]\right]}_{\text{Variance} = EE} + \underbrace{\mathbb{E}\left[(\bar{f}(X) - f_{Bayes}(X))^2\right]}_{\text{Bias} = AE}$$

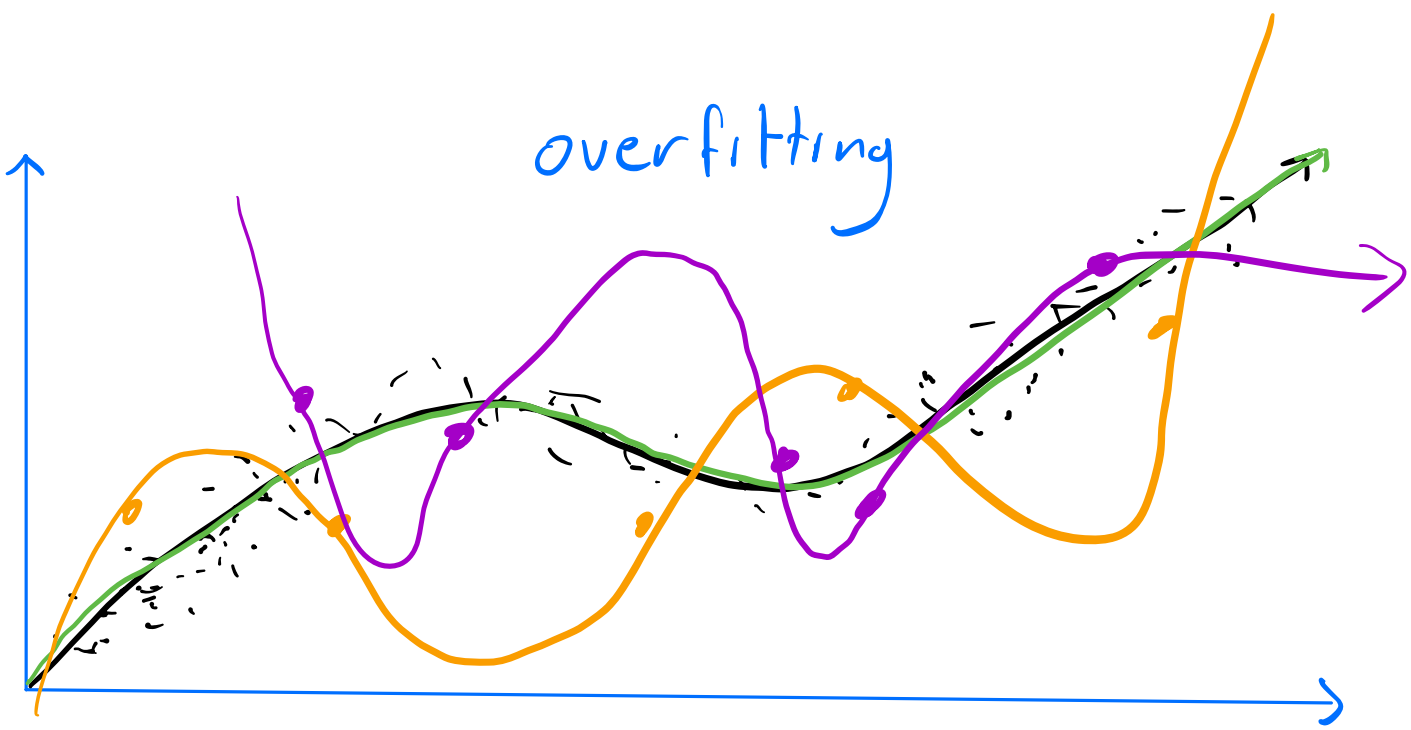
$$+ L(f_{Bayes})$$

$$\bar{f}(x) = \mathbb{E}[\hat{f}_D(x) \mid X = x]$$ expected predictor

$$= f^*(x)$$

$$\hat{f} = \underset{f \in \mathcal{F}_1}{\operatorname{argmin}} \hat{L}(f)$$

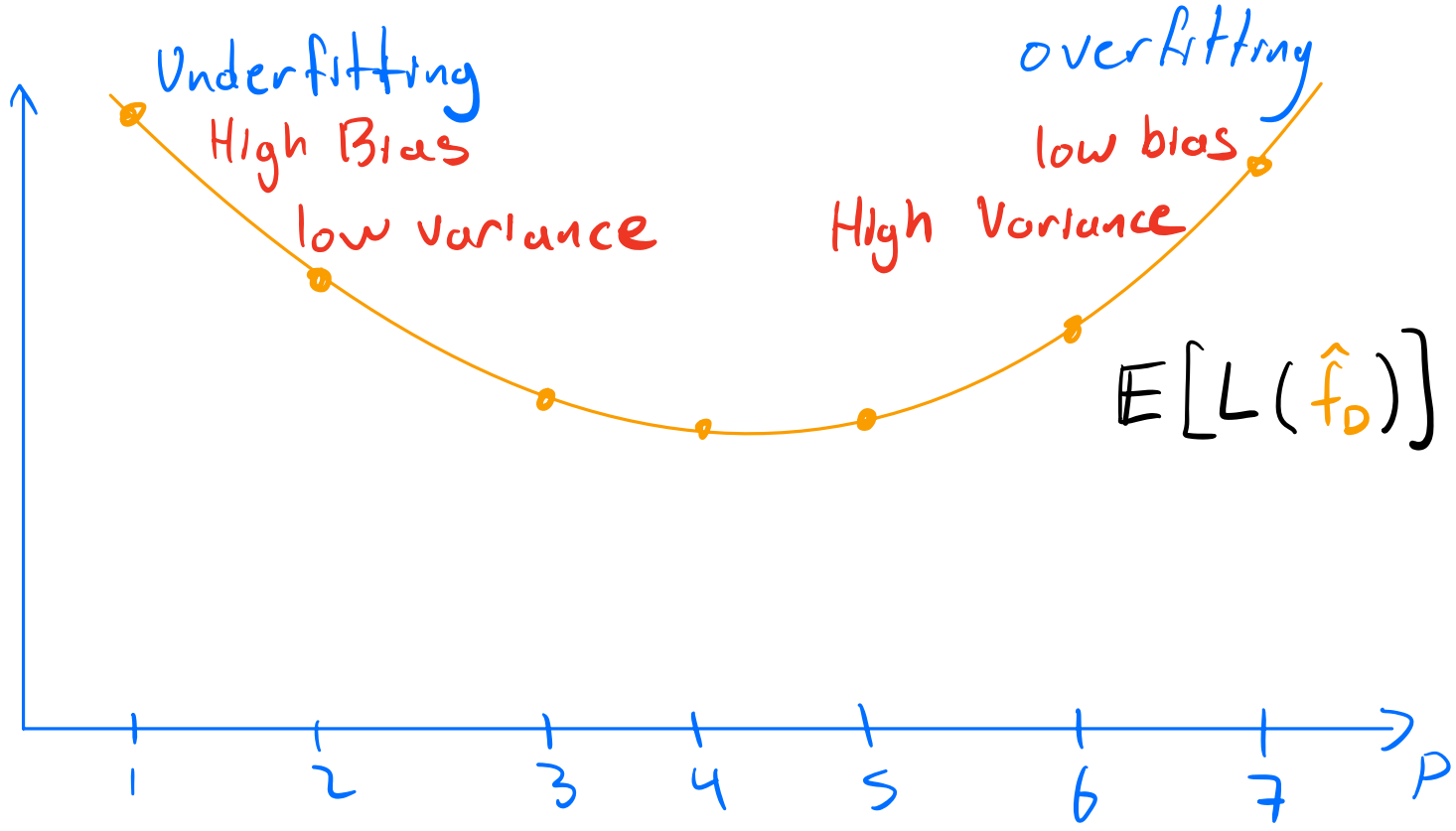




Bias ↓ if $\mathcal{F}$ ↑

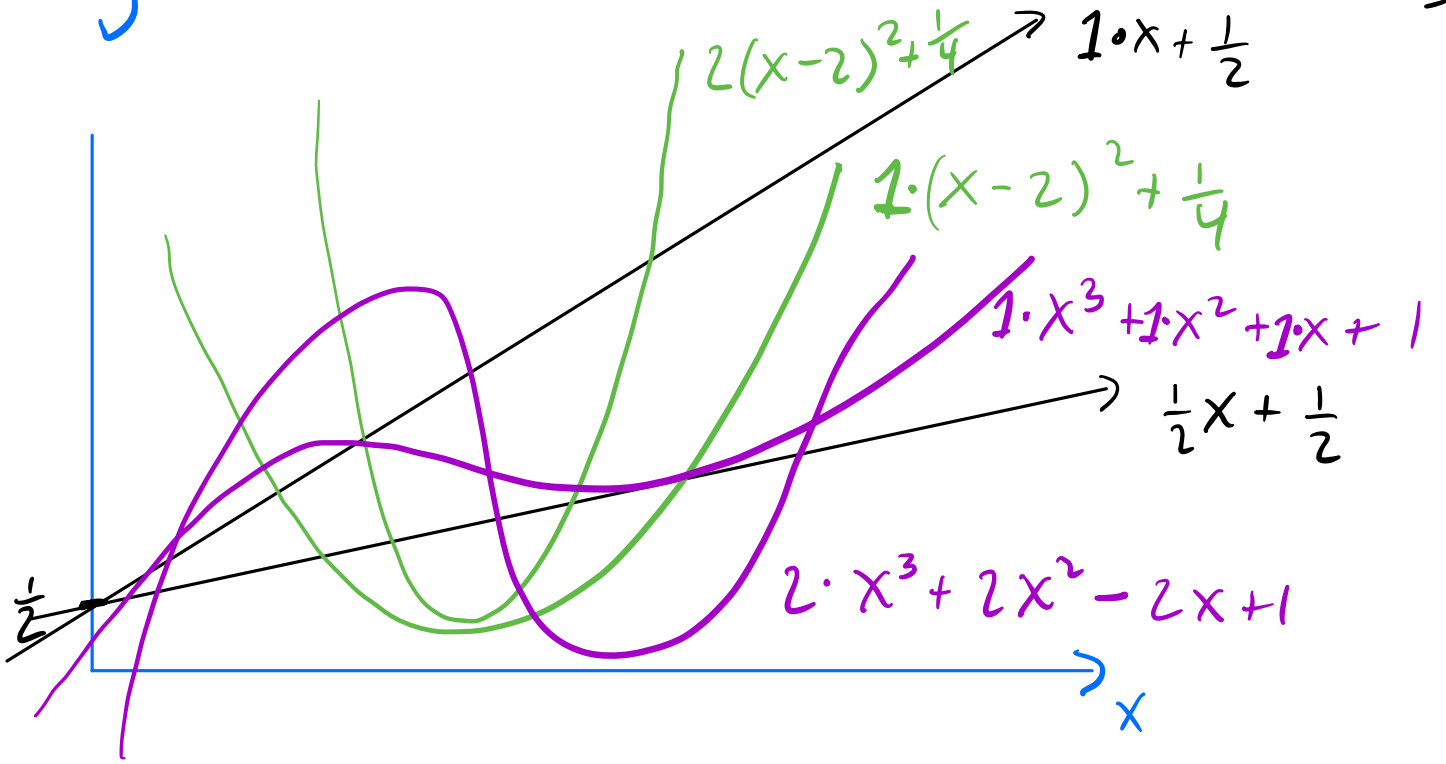
Variance ↓ if n ↑

↑ if $\mathcal{F}$ ↑



Regularization

$\hat{f}(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$



let $\vec{w} = (w_0, \ldots, w_{\bar{p}-1})^T \in \mathbb{R}^{\bar{p}}$

Observation: large values of $w_0, \ldots, w_{\bar{p}-1}$ leads to a more complex $f_p(\vec{x}) = \phi_p(\vec{x})^T \vec{w}$

$$\tilde{F}_{p,C} = \left\{ f \mid f : \mathbb{R}^{d+1} \to \mathbb{R}, \text{ s.t. } f(\vec{x}) = \phi_p(\vec{x})^T \vec{w}, \right.$$
$$\left. \vec{w} \in \mathbb{R}^{\bar{p}} \quad \text{and} \quad \sum_{j=1}^{\bar{p}-1} w_j^2 \leq C, \; C > 0 \right\}$$

$$\mathcal{A}_{p,\lambda}(D) = \hat{f}_{p,\lambda} \text{ where } \hat{f}_{p,\lambda}(\vec{x}) = \phi_p(\vec{x})^T \hat{\vec{w}}_\lambda$$

regularizer

$$\hat{\vec{w}}_\lambda = \underset{\vec{w} \in \mathbb{R}^{\bar{p}}}{\operatorname{argmin}} \left(\frac{1}{n} \sum_{i=1}^{n} \ell(\phi_p(\vec{x}_i)^T \vec{w}, y_i) + \frac{\lambda}{n} \sum_{j=1}^{\bar{p}-1} w_j^2 \right)$$

"Ridge Regression" $g(\vec{w})$ convex!

doesn't use w_0

$\lambda \in (0, \infty)$ regularization parameter

$$\hat{L}_\lambda(\vec{w}) = \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell(\phi_p(\vec{x}_i)^T \vec{w}, y_i)}_{\hat{L}(\vec{w})} + \frac{\lambda}{n} \sum_{j=1}^{\bar{p}-1} w_j^2$$

set $p=1 \Rightarrow \bar{p} = d+1$

$$\hat{\vec{w}}_\lambda = \underset{\vec{w} \in \mathbb{R}}{\operatorname{argmin}} \hat{L}_\lambda(\vec{w})$$

$$\vec{w}^{(t+1)} = \vec{w}^{(t)} - \eta^{(t)} \nabla \hat{L}_\lambda(\vec{w}^{(t)})$$

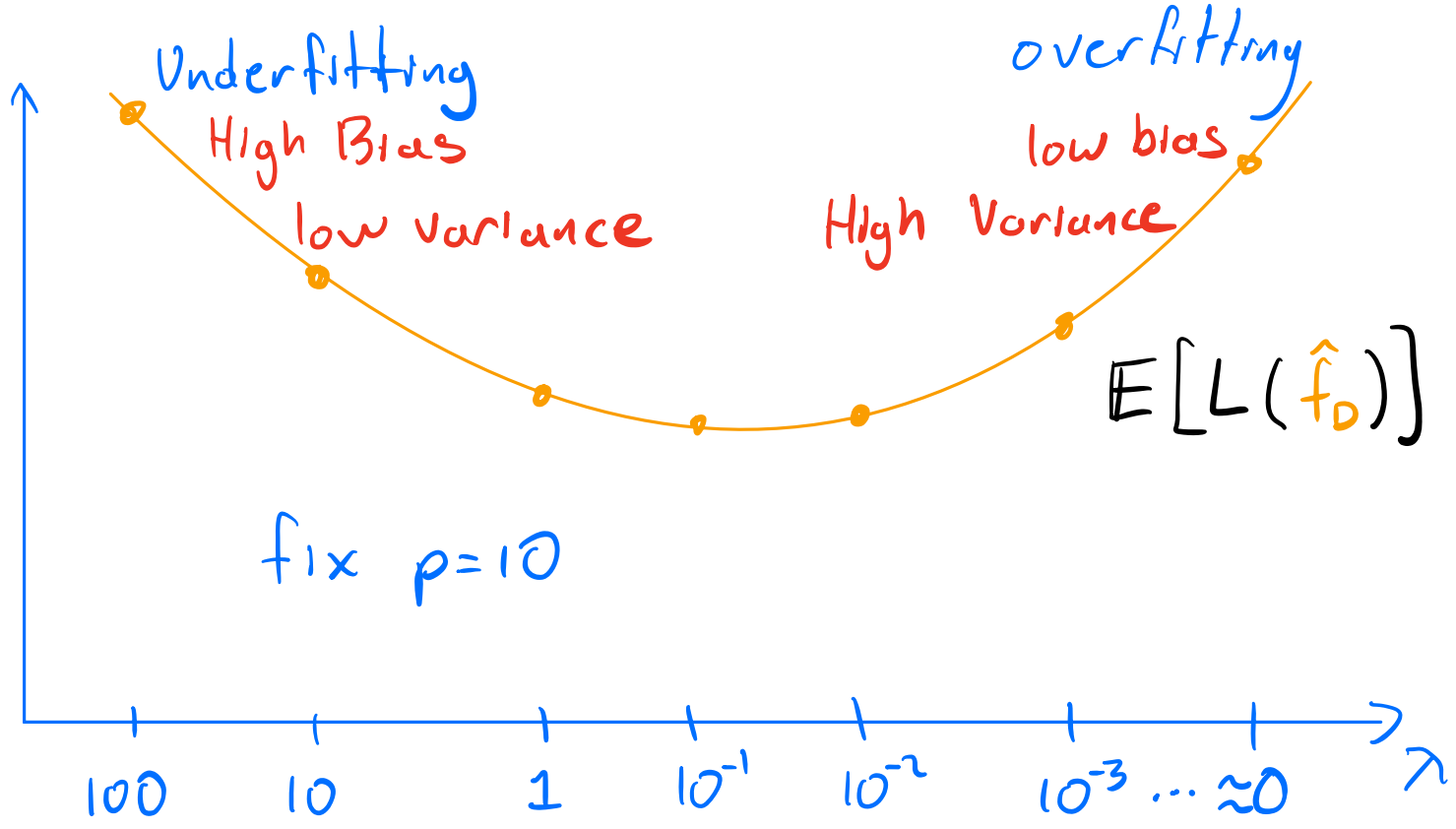
$$\nabla \hat{L}_\lambda = \left(\frac{\partial \hat{L}_\lambda}{\partial w_0}, \frac{\partial \hat{L}_\lambda}{\partial w_1}, \ldots, \frac{\partial \hat{L}_\lambda}{\partial w_d} \right)^T$$

$$\frac{\partial \hat{L}_\lambda}{\partial w_0} = \frac{\partial \hat{L}}{\partial w_0} = \frac{2}{n}\sum_{i=1}^{n}(\vec{x}_i^{\top}\vec{w} - y_i)x_{i0}$$

for $k \in \{1, \ldots, d\}$

$$\frac{\partial \hat{L}_\lambda}{\partial w_k} = \frac{\partial \hat{L}}{\partial w_k} + \frac{\partial\left(\frac{\lambda}{n}\sum_{j=1}^{d} w_j^2\right)}{\partial w_k}$$

$$= \frac{2}{n}\sum_{i=1}^{n}(\vec{x}_i^{\top}\vec{w} - y_i)x_{ij} + \frac{2\lambda}{n}w_k$$



Algorithm: BGD Linear Regression Learner
(with a constant step size, and regularization)

input: $D = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$, η, T, λ

$\vec{w}$ = random vector in $\mathbb{R}^{d+1}$

for $t = 1, \ldots, T$

$$\nabla \hat{L}(\vec{w}) = \frac{2}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i) \vec{x}_i$$

$$\nabla \hat{L}(\vec{w})[1:d] \mathrel{+}= \frac{2\lambda}{n} \vec{w}[1:d]$$

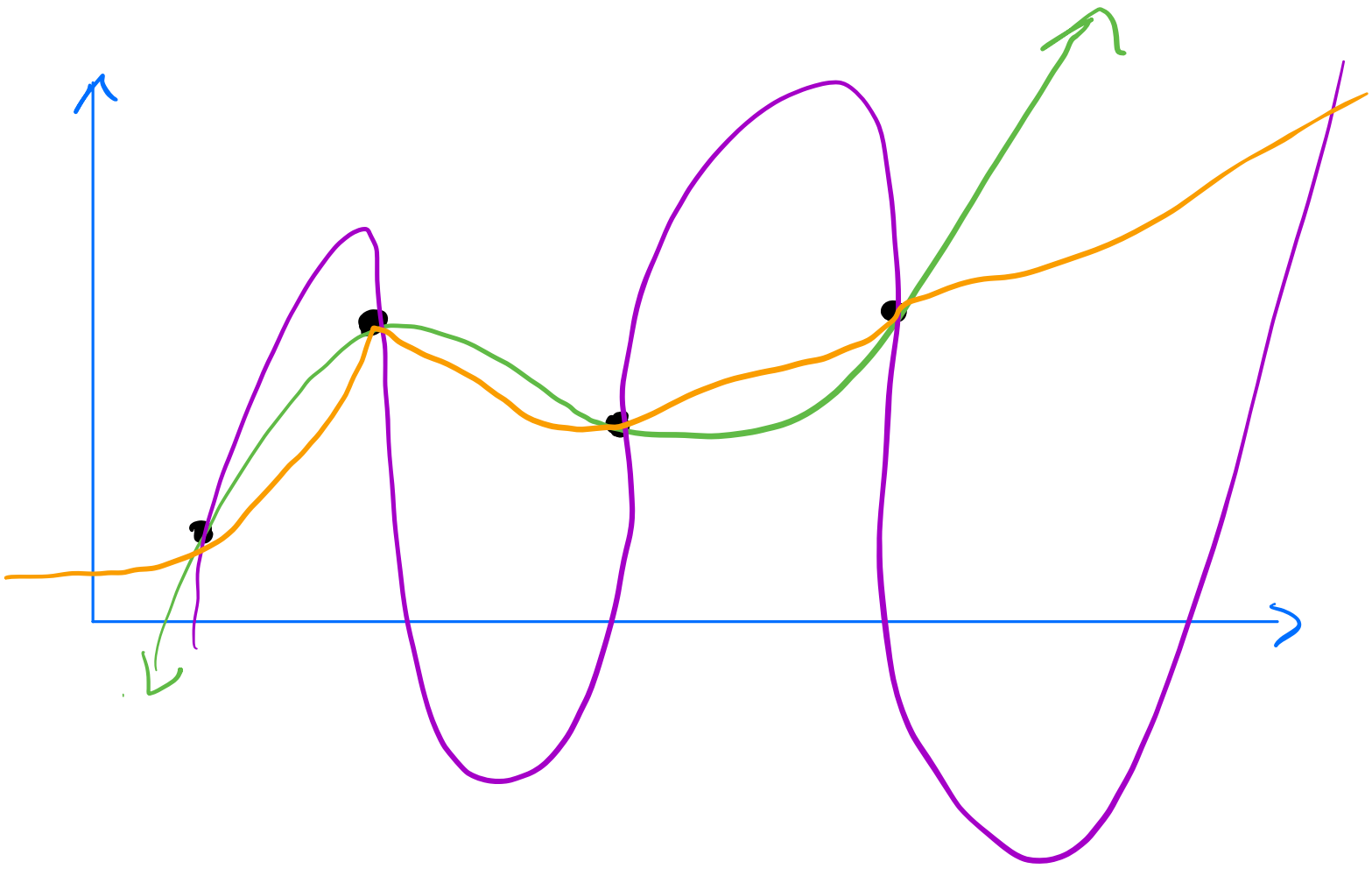
$$\vec{w} = \vec{w} - \eta \nabla \hat{L}(\vec{w})$$

return $\hat{f} : \mathbb{R}^{d+1} \to \mathbb{R}$ where $\hat{f}(\vec{x}) = \vec{x}^T \vec{w}$

if $\lambda \uparrow$ Bias $\uparrow$
Var $\downarrow$
AE is same
EE is complicated

if $\lambda \neq 0$ then $\bar{f} \neq f^*$

Double Descent (Not graded)



$$\hat{f} \in \underset{f \in \mathcal{F}}{\operatorname{argmin}} \hat{L}(f)$$