Evaluating Predictors/Models

Objective (formal):

Define a learner $\mathcal{A}:(\mathcal{X}\times\mathcal{Y})^n \to \{f \mid f:\mathcal{X}\to\mathcal{Y}\}$

such that $\mathbb{E}[L(\mathcal{A}(D))]$ is small

Defining $\mathcal{A}(D)$: Empirical Risk Minimization (ERM)

Estimation:

Use D to estimate $L(f)$ for all $f \in \mathcal{F} \subset \{f \mid f:\mathcal{X}\to\mathcal{Y}\}$

call the estimate $\hat{L}(f)$

Optimization:

pick $\hat{f}$ to be the $f \in \mathcal{F}$ that minimizes $\hat{L}(f)$

Function class

When should we expect ERM to work well?

- When $\mathcal{F}$ contains an f that can make $L(f)$ small
- When $\hat{L}(\hat{f})$ is a good estimate of $L(\hat{f})$

Is $\hat{L}(\hat{f}_D)$ really a good estimate of $L(\hat{f}_D)$?

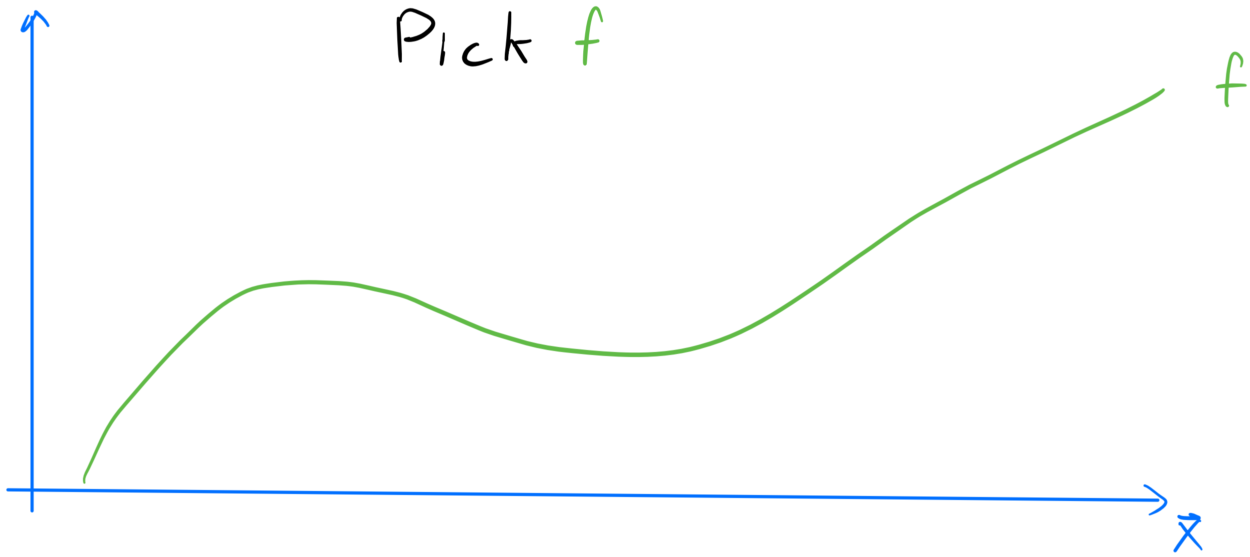
where $\mathcal{A}(D) = \hat{f}_D \in \mathcal{F}$ D is a r.v. $(\vec{X}, Y) \sim \mathbb{P}_{\vec{X},Y}$

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

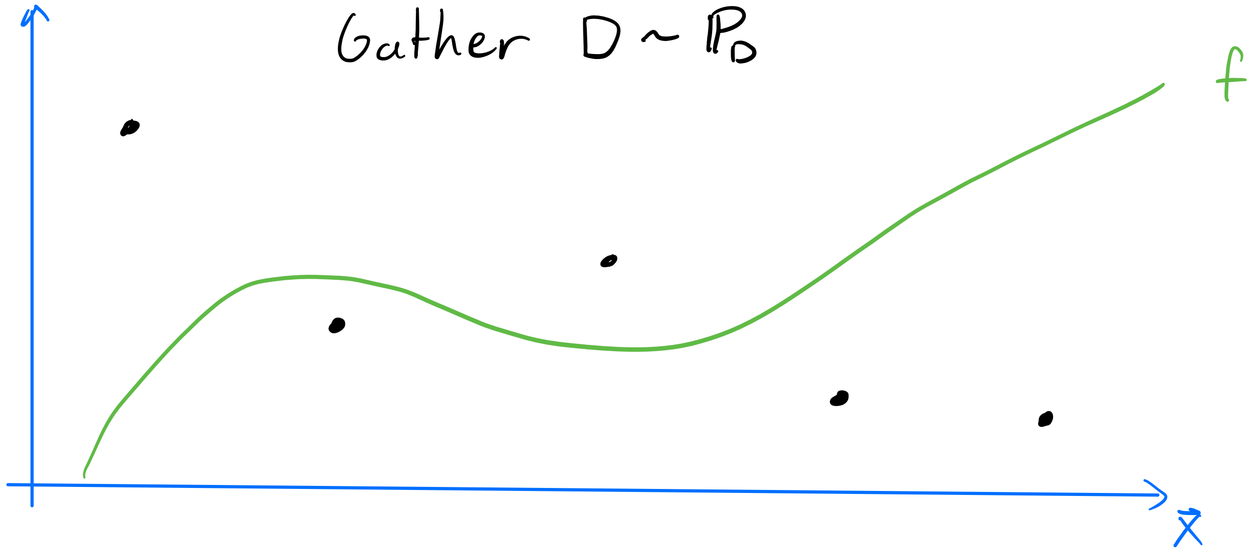
If we pick $f \in \mathcal{F}$ and then gather D

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

Pick f



Gather $D \sim \mathbb{P}_D$

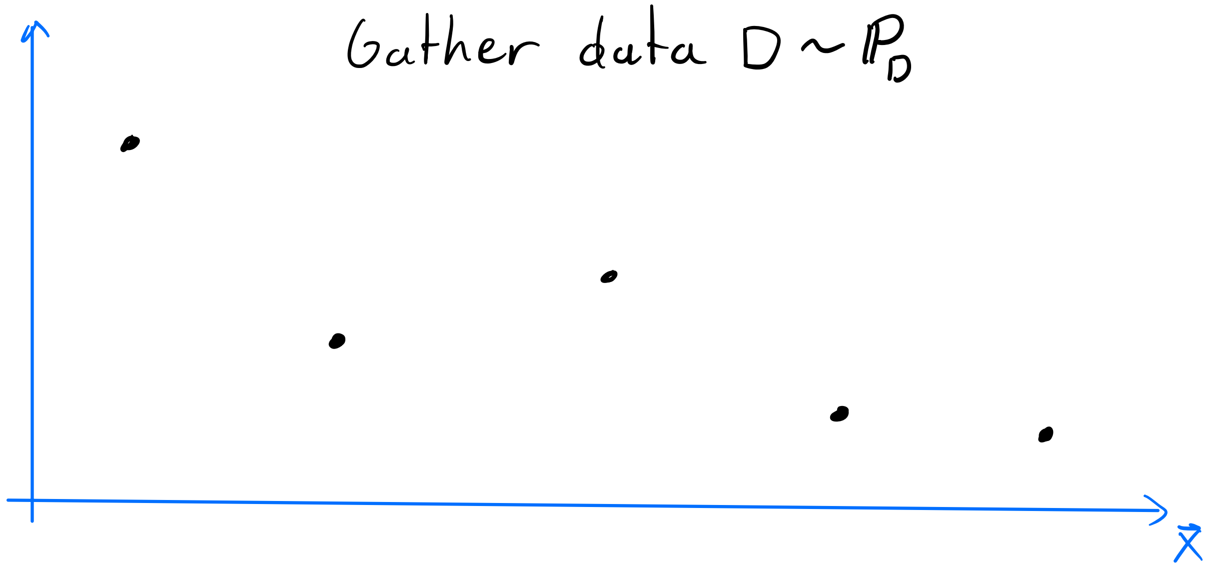


Then: $\mathbb{E}[\hat{L}(f)] = L(f)$

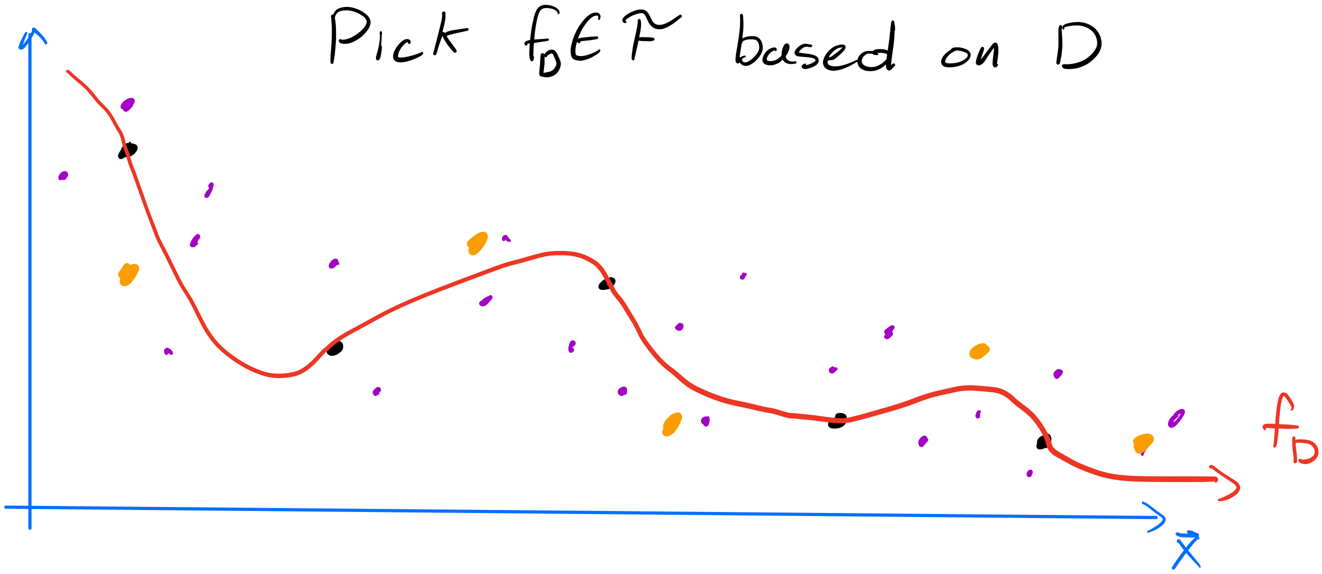
$$\mathrm{Var}[\hat{L}(f)] = \frac{1}{n}\mathrm{Var}[\ell(f(\vec{X}_1), Y_1)]$$

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

Gather data $D \sim \mathbb{P}_D$



Pick $\hat{f}_D \in \mathcal{F}$ based on D



Then, in general:

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

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

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

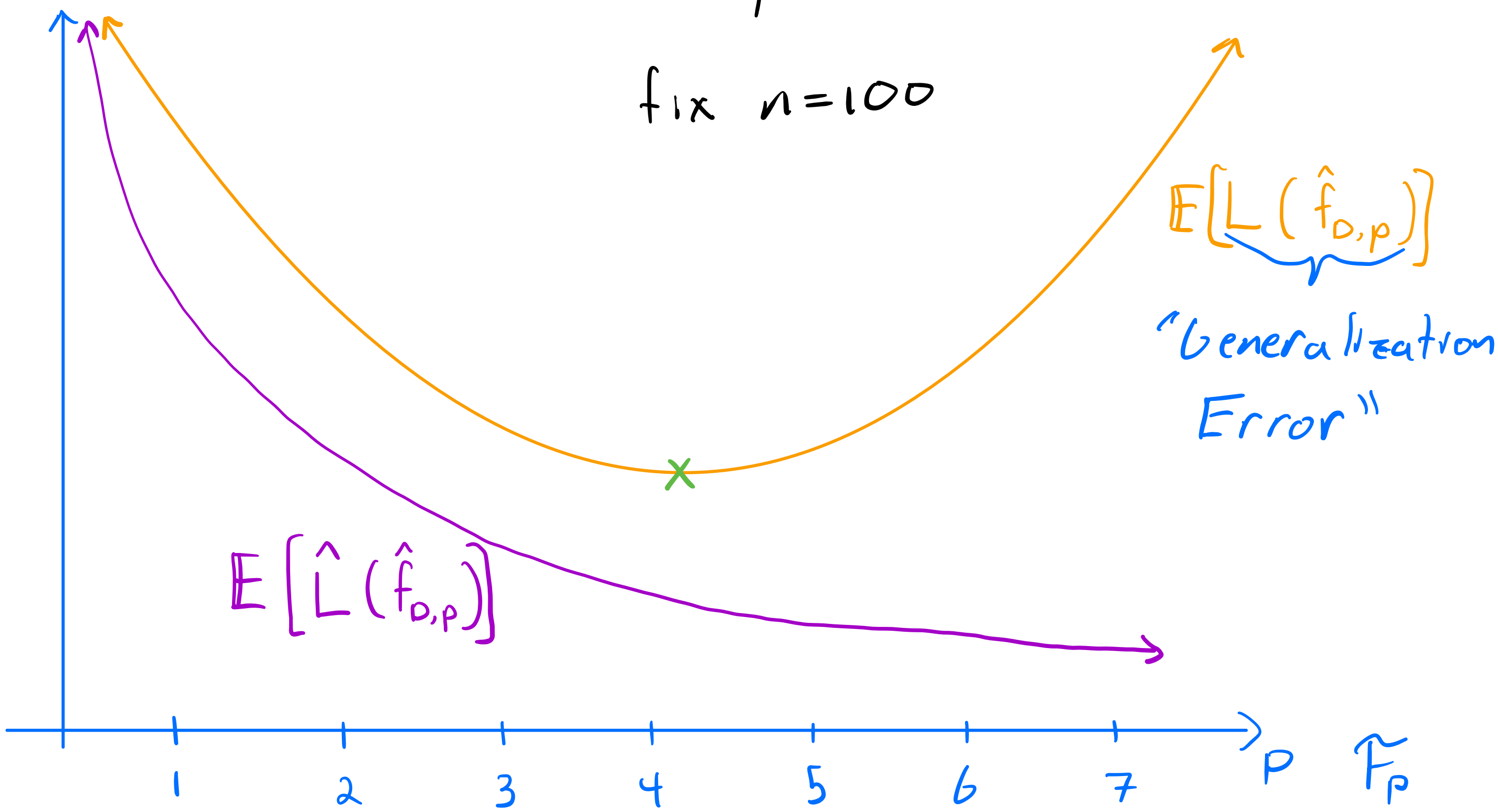
It can be shown that:

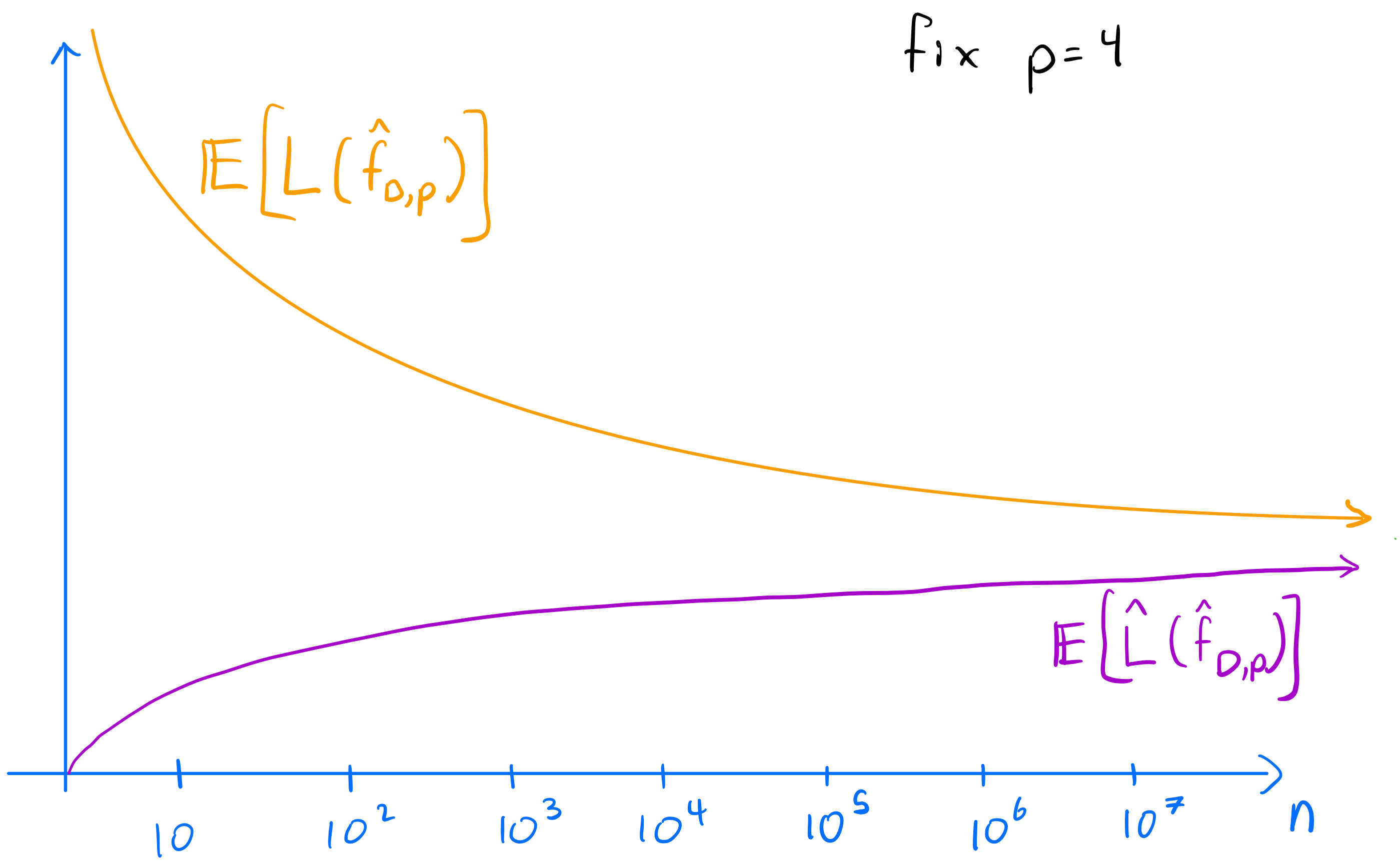
$$\mathbb{E}[L(\hat{f}_D)] - \mathbb{E}[\hat{L}(\hat{f}_D)]$$

- increases as $\mathcal{F}$ gets more complex
- decreases as n increases

Ex:

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

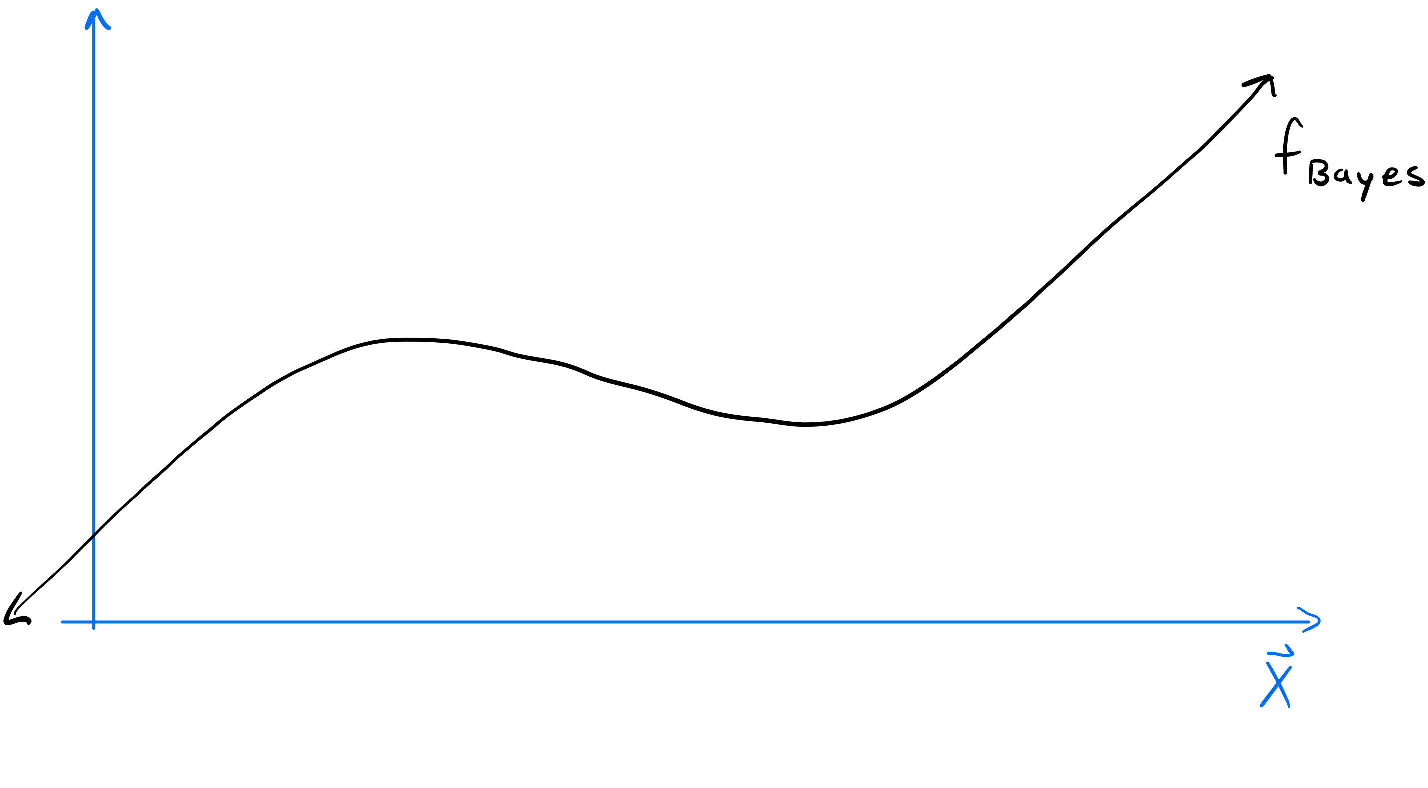




Decomposing $\mathbb{E}[L(\mathcal{A}(D))]$ into Error Terms

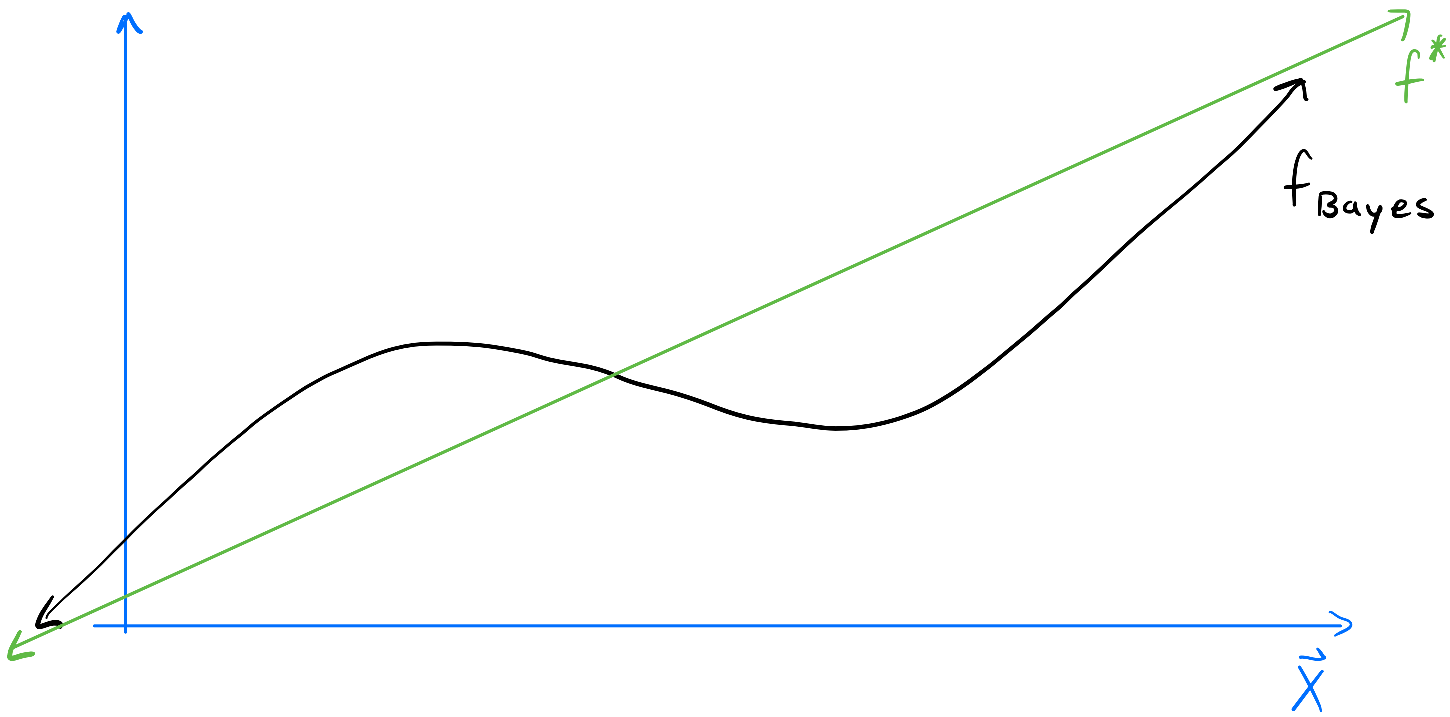
Suppose we knew $\mathbb{P}_{\vec{X},Y}$ what would we choose for $\mathcal{A}$?

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



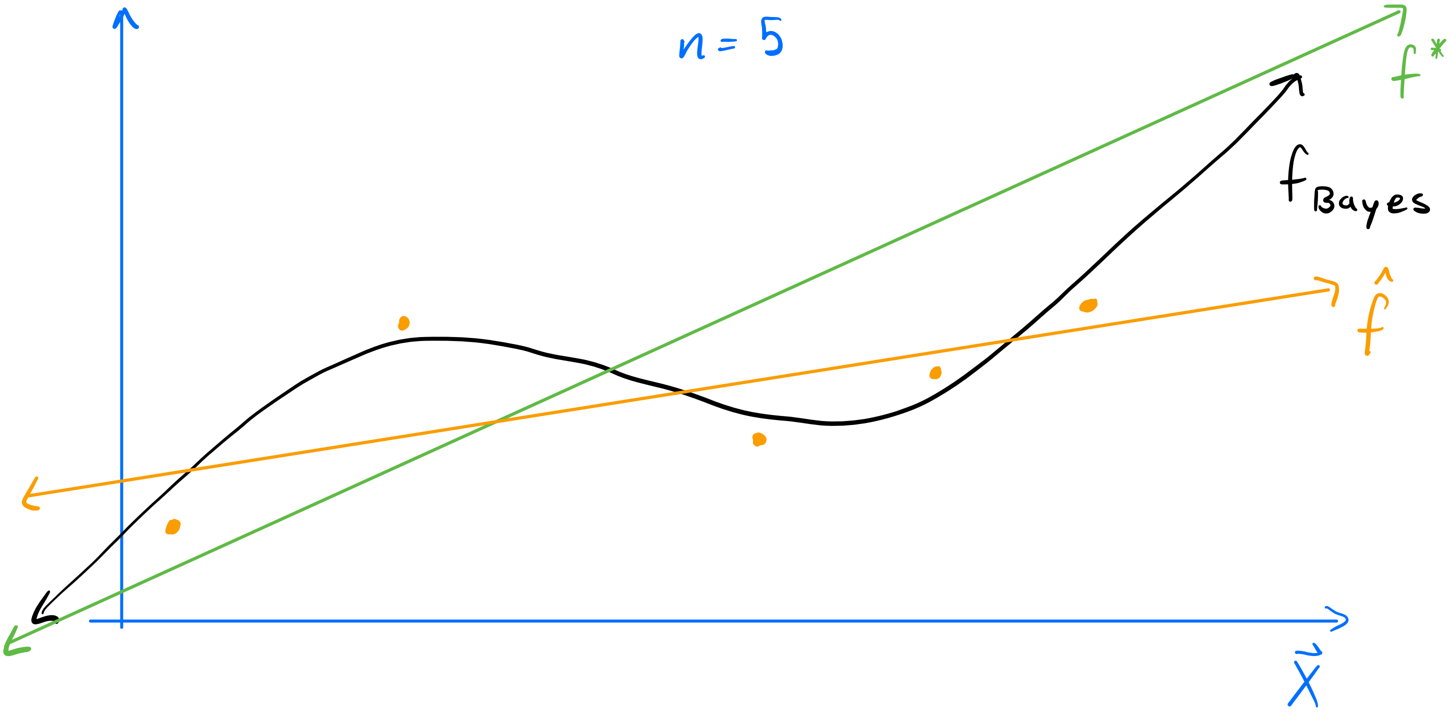
Suppose we knew $\mathbb{P}_{\vec{X},Y}$ but $\mathcal{A}:(\mathcal{X}\times\mathcal{Y})^n \to \underbrace{\mathcal{F}_1}_{\text{linear functions}}$

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



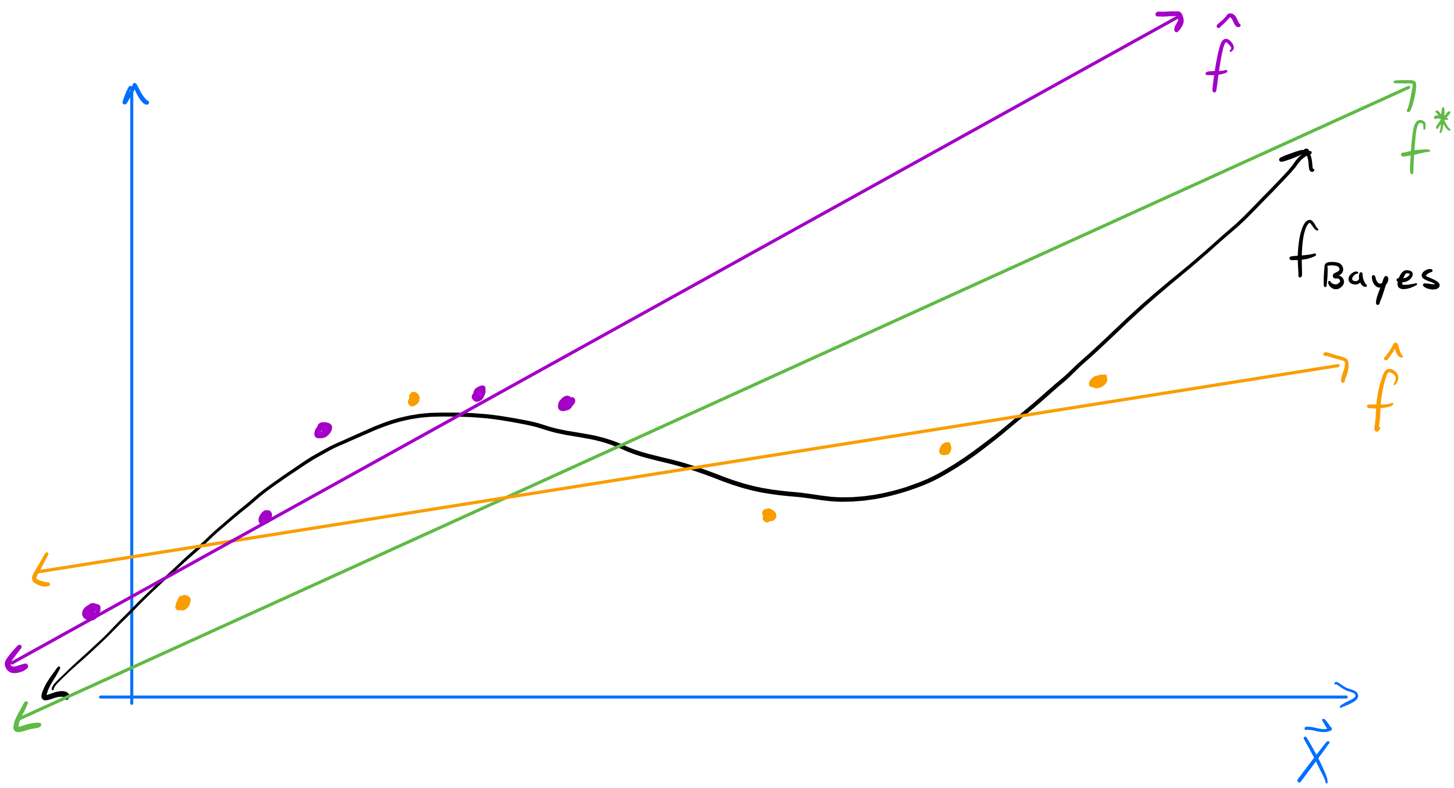
Suppose we didn't know $\mathbb{P}_{\vec{X},Y}$ but we had a dataset $D_1 = ((\vec{X}_1, y_1), \ldots, (\vec{X}_n, y_n))$ and $\mathcal{A}:(\mathcal{X}\times\mathcal{Y})^n \to \mathcal{F}_1$

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

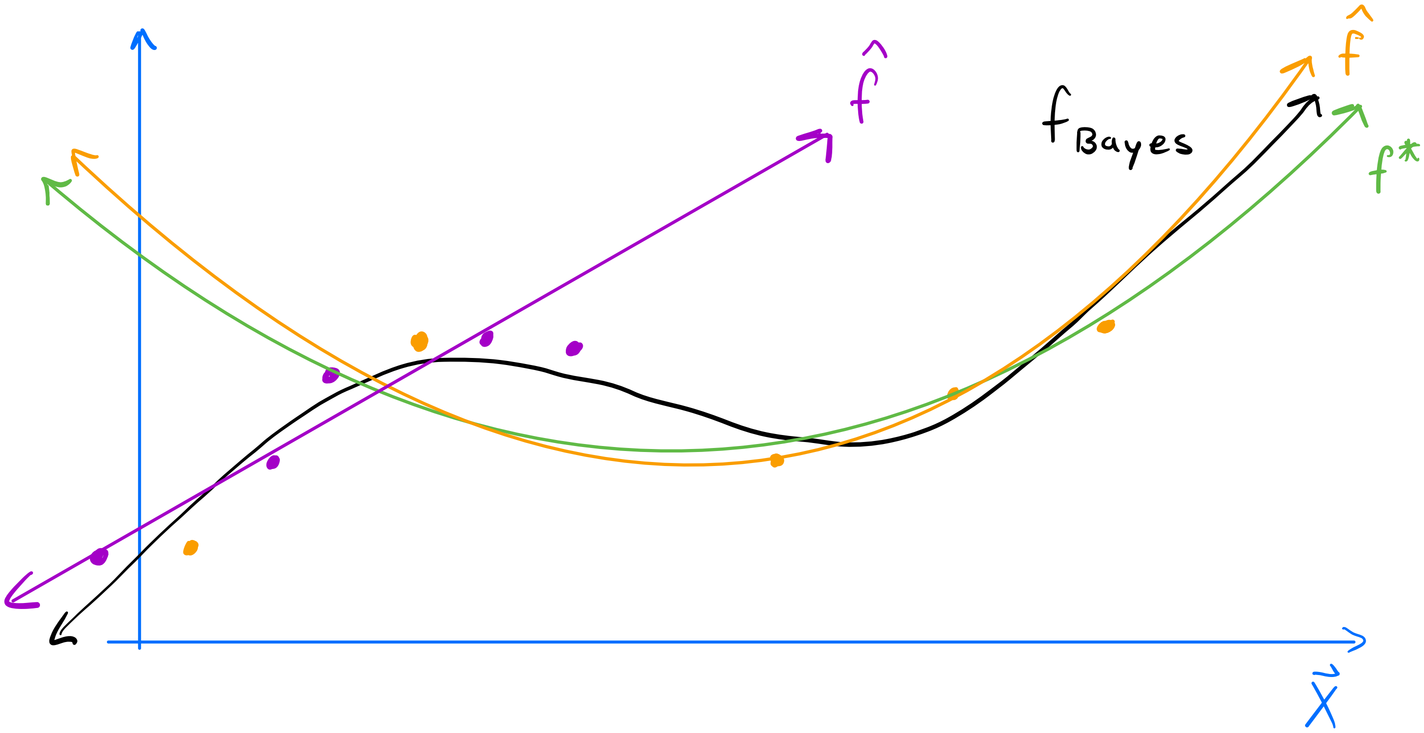


How about for a different dataset

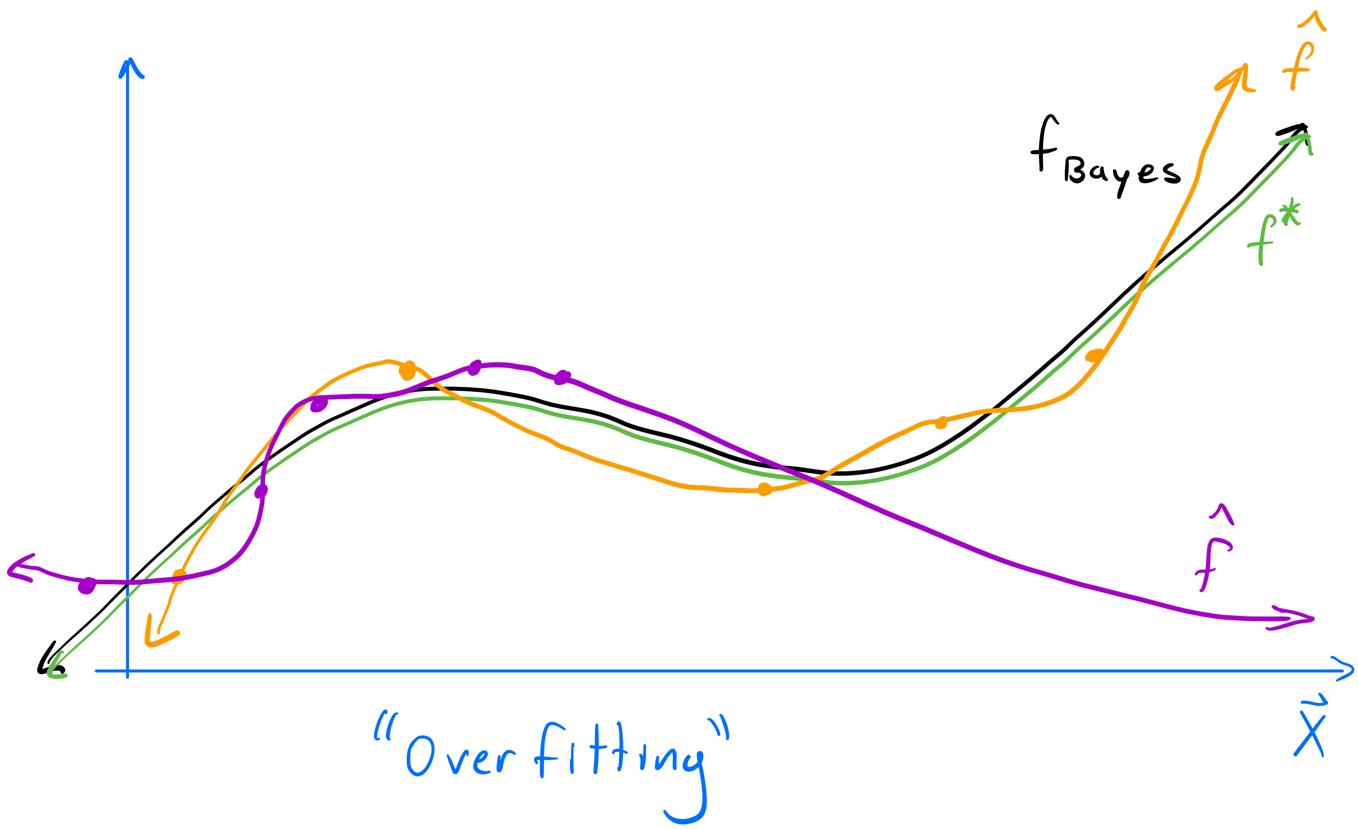
$D_2 = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$



How about for F_2



How about if F_{10}



Decomposing $\mathbb{E}[L(\mathcal{A}(D))]$

Let $\mathcal{A}(D) = \hat{f}_D$

$\mathcal{F}$ any function class

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

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

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

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

What affects the different types of errors?

Irreducible Error: Due to inherent noise in labels

- Decreases if you gather more/better feature info
- Usually not possible to do "irreducible"

Approximation Error: Due to a small $\mathcal{F}$

- Decreases if you make $\mathcal{F}$ larger

Estimation Error: Due to random dataset D

- Decreases if you increase n
- Increases if you increase $\mathcal{F}$

High EE: small n, large $\mathcal{F}$

High AE: f_{Bayes} complex, $\mathcal{F}$ simple

Understanding EE:

$\hat{L}(\hat{f}_D) \le \hat{L}(f)$ for all $f \in \mathcal{F}$

EE: $\mathbb{E}[L(\hat{f}_D)] - L(f^*)$

$$= \mathbb{E}[L(\hat{f}_D)] - \mathbb{E}[\hat{L}(\hat{f}_D)] + \overbrace{\mathbb{E}[\hat{L}(\hat{f}_D)] - \mathbb{E}[\hat{L}(f^*)]}^{\le 0}$$

$$+ \mathbb{E}[\hat{L}(f^*)] - L(f^*)$$

$$\le \underbrace{\mathbb{E}[L(\hat{f}_D)] - \mathbb{E}[\hat{L}(\hat{f}_D)]} + \mathbb{E}[\hat{L}(f^*)] - L(f^*)$$

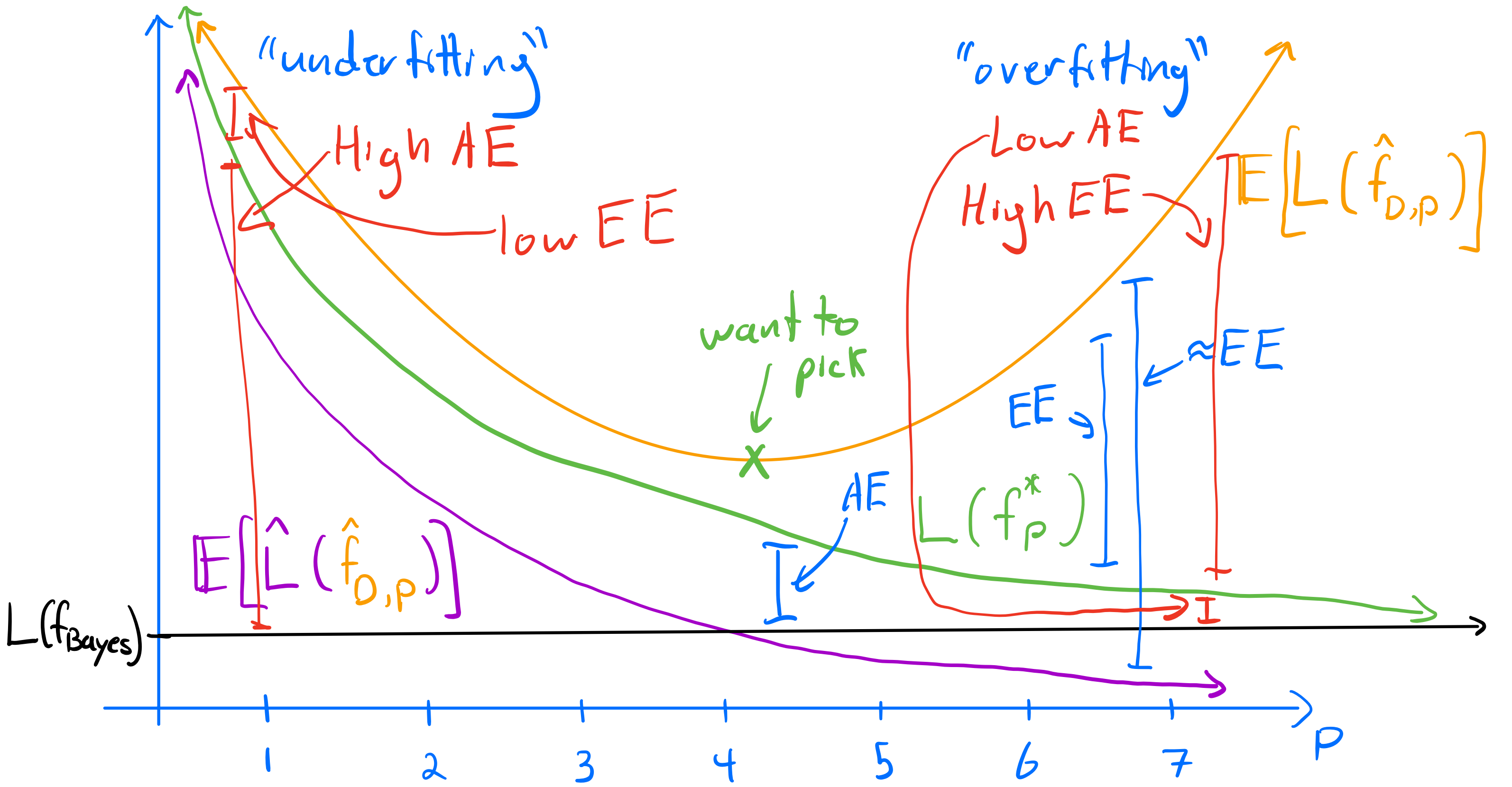
decreases as n increases

increases as $\mathcal{F}$ gets larger

$$\mathcal{A}(D) = \hat{f}_{D,p} = \underset{f \in \mathcal{F}_p}{\operatorname{argmin}} \hat{L}(f) \qquad \mathcal{F}_1 \subset \cdots \subset \mathcal{F}_p$$

$$\mathbb{E}[L(\hat{f}_D)] - \mathbb{E}[\hat{L}(\hat{f}_D)]$$

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

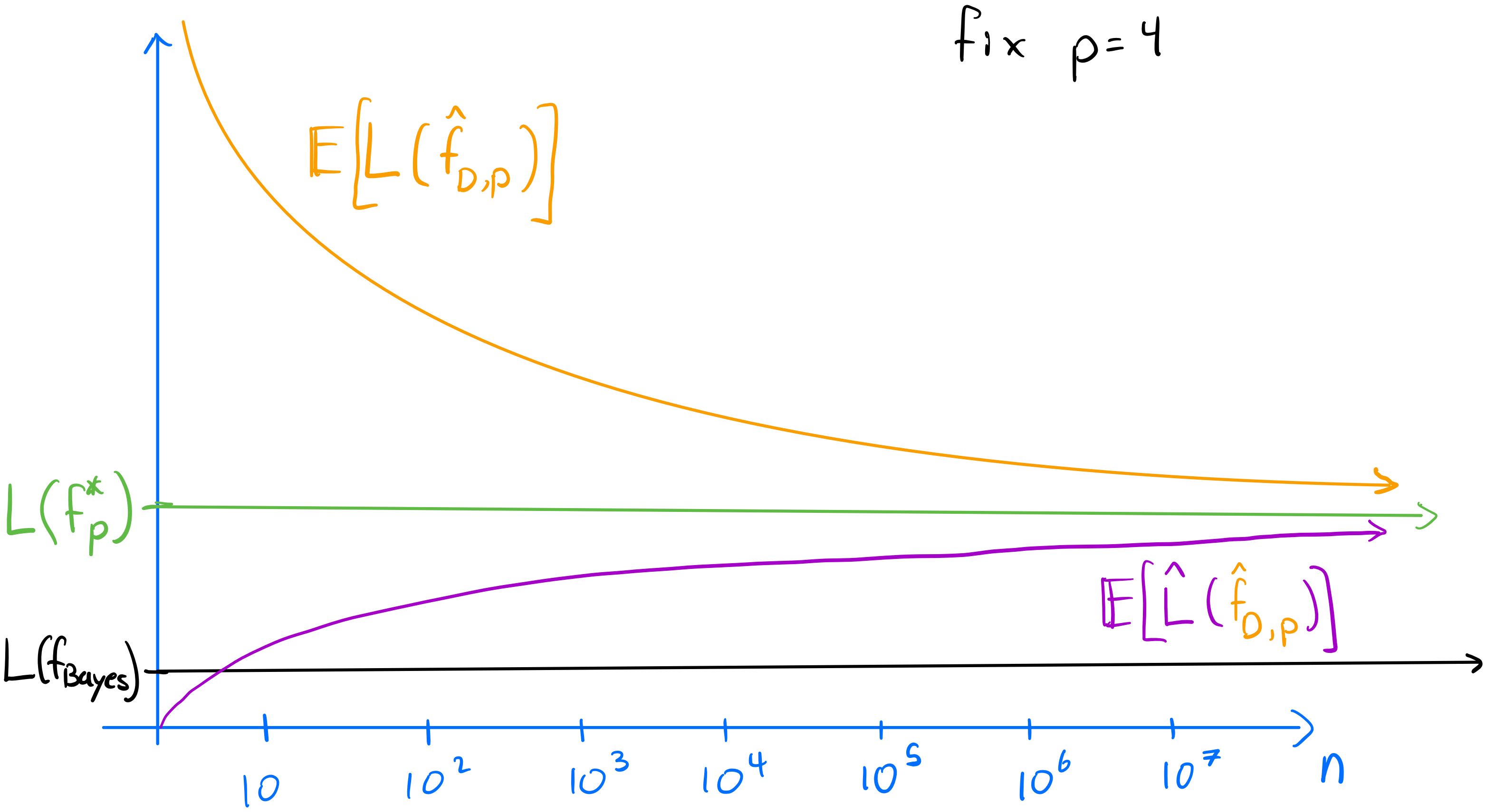


Underfitting: $\mathcal{F}$ is too simple (small) compared to n

- High AE, Low EE

Overfitting: $\mathcal{F}$ is too complex (large) compared to n

- Low AE, High EE


fix $p=4$
$\mathbb{E}[L(\hat{f}_{D,p})]$
$L(f_p^*)$
$\mathbb{E}[\hat{L}(\hat{f}_{D,p})]$
$L(f_{Bayes})$
10
10^2
10^3
10^4
10^5
10^6
10^7
n

In practice we only have a fixed dataset D

How can we tell if we are overfitting or underfitting if we can't calculate $L(\hat{f}_D)$?

Estimate $L(\hat{f}_D)$ with a different dataset D_{test}

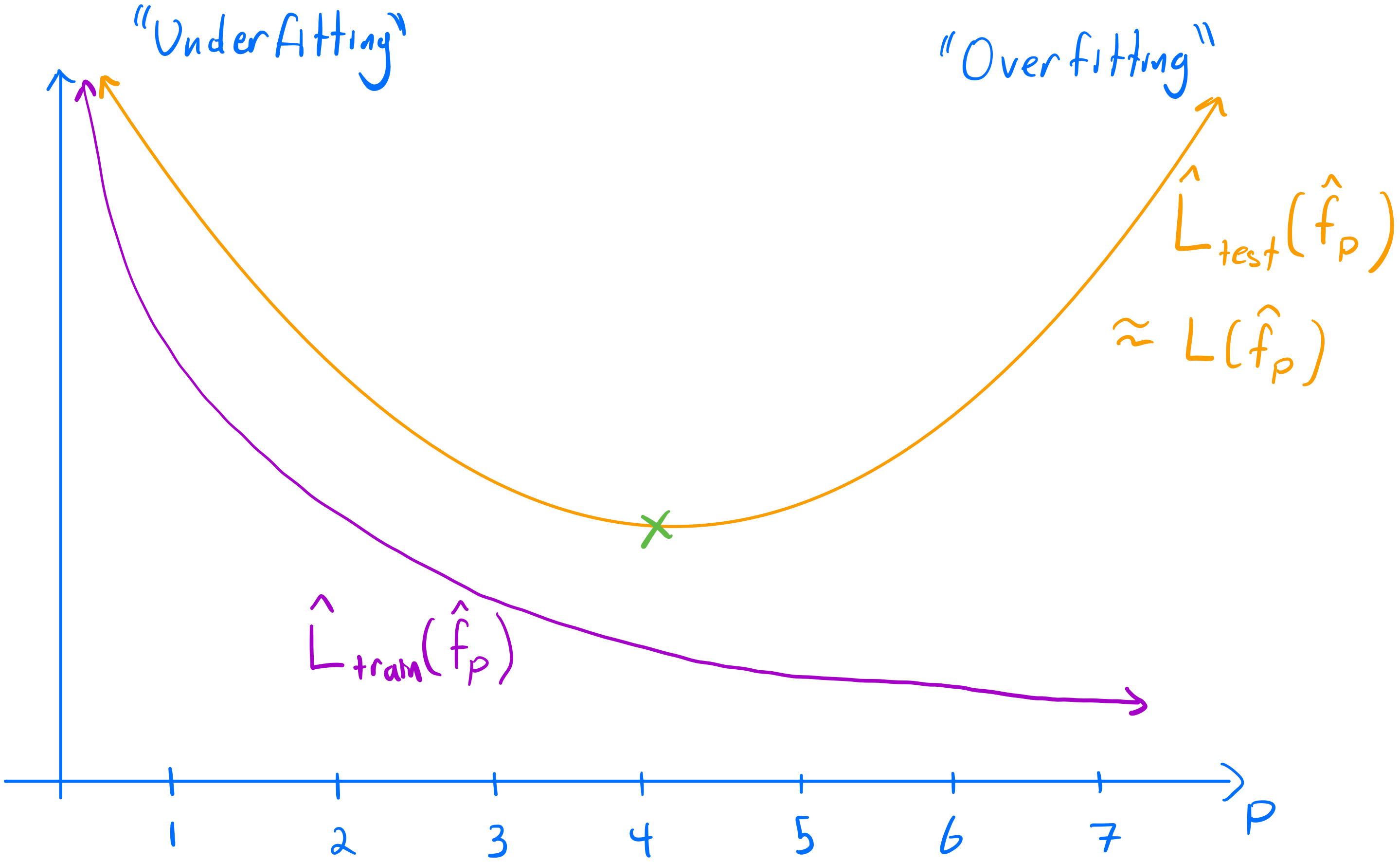
Since we can't gather new data we split D into D_{train}, D_{test}

$$D_{train} = ((\vec{x}_1, y_1), \ldots, (\vec{x}_{n-m}, y_{n-m}))$$

$$D_{test} = ((\vec{x}_{n-m+1}, y_{n-m+1}), \ldots, (\vec{x}_n, y_n))$$

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

$$\mathcal{A}(D_{train}) = \hat{f}_p = \underset{f \in \mathcal{F}_p}{\operatorname{argmin}} \hat{L}_{train}(f) \qquad \mathcal{F}_1 \subset \cdots \subset \mathcal{F}_p$$



Bias-Variance Tradeoff

$\mathbb{E}[L(\hat{f}_D)]$ Let ℓ be squared loss

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

if $\bar{f} = f^*$

$$= \overbrace{\mathbb{E}\Big[\underbrace{\mathbb{E}\big[(\hat{f}_D(\vec{x}) - \bar{f}(\vec{x}))^2 \mid \vec{x}\big]}_{\text{Variance} = \mathrm{Var}[\hat{f}_D(\vec{x}) \mid \vec{x}]}\Big]}^{=EE} + \overbrace{\mathbb{E}\Big[\big(\underbrace{\bar{f}(\vec{x}) - f_{Bayes}(\vec{x})}_{\text{Bias}}\big)^2\Big]}^{=AE} + \underbrace{L(f_{Bayes})}_{IE}$$

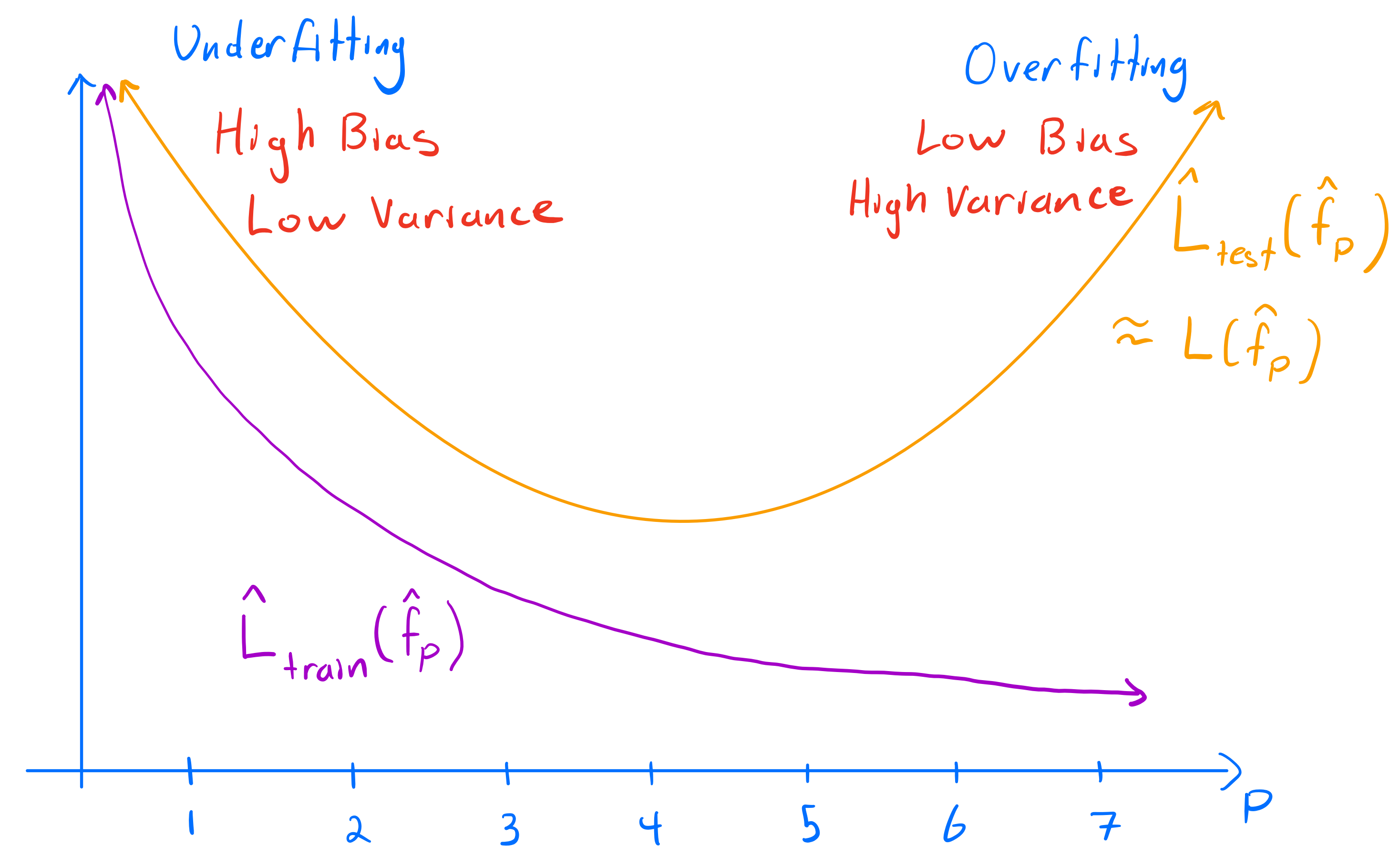
where $\bar{f}(\vec{x}) = \mathbb{E}[\hat{f}_D(\vec{x}) \mid \vec{x}]$ "expected predictor"

Effects of changing $\mathcal{F}$, n on Bias, Variance follow the same trend as for AE, EE:

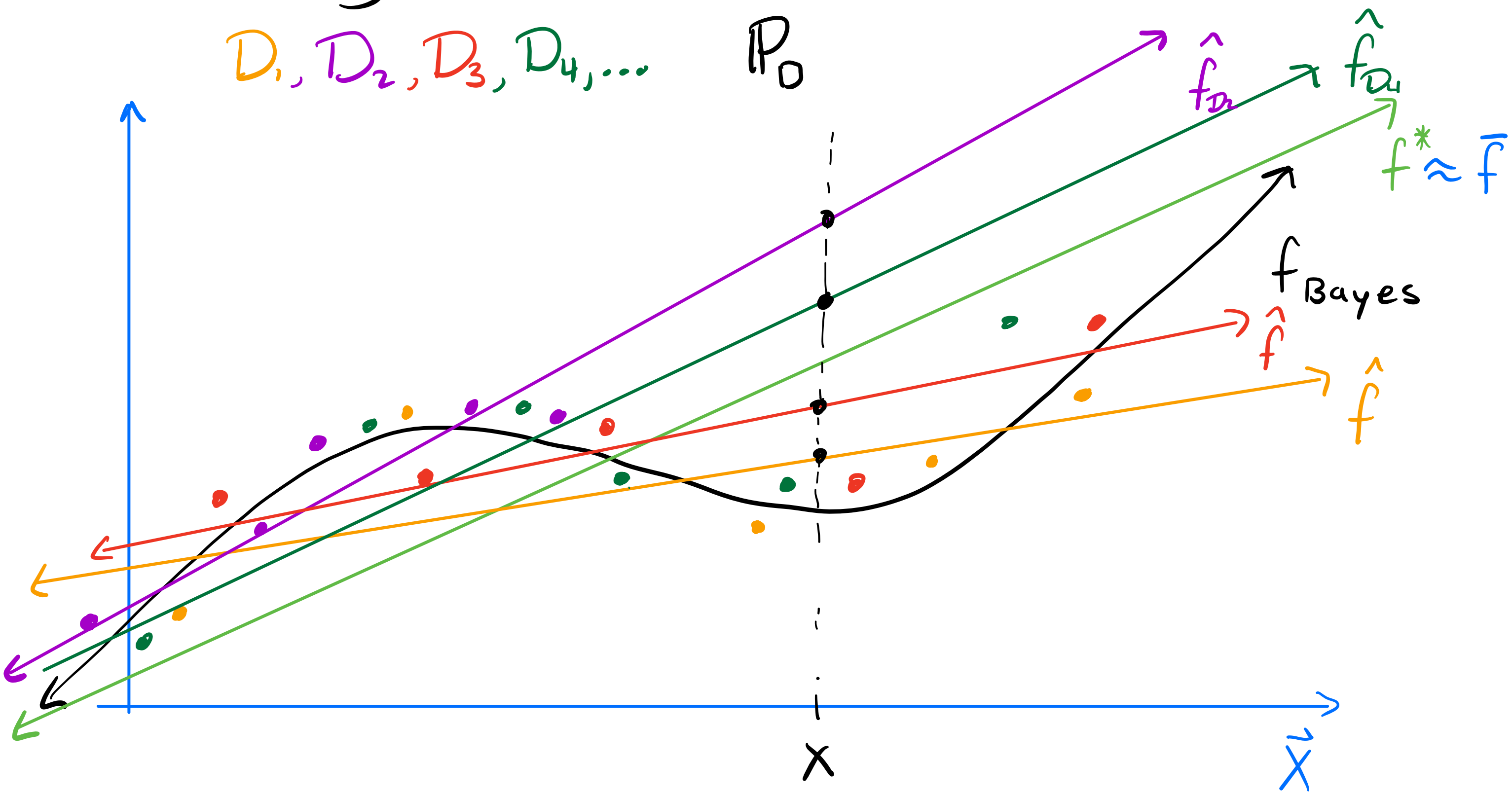
Bias $\downarrow$ if $\mathcal{F} \uparrow$

Variance $\downarrow$ if $n \uparrow$

$\uparrow$ if $\mathcal{F} \uparrow$



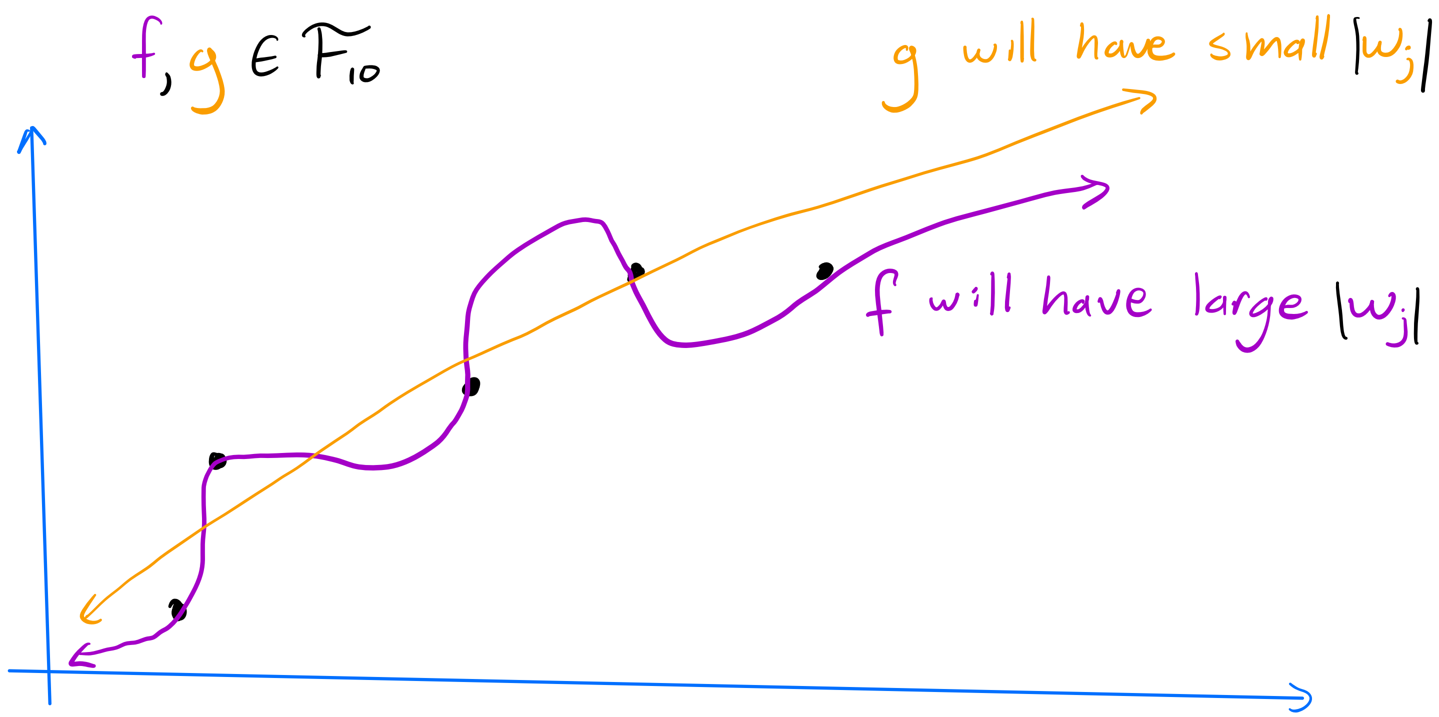
Visualizing $\bar{f}(x) = \mathbb{E}[\hat{f}_D(X)|X]$



Regularization

let $\vec{w} = (w_0, w_1, \ldots, w_{\bar{p}-1})^T \in \mathbb{R}^{\bar{p}}$

Observation: large values of $|w_0|, |w_1|, \ldots, |w_{\bar{p}-1}|$ leads to more complex $f_p(\vec{x}) = \phi_p(\vec{x})^T \vec{w}$



Regularization: penalize large weights

If $f_p \in \mathcal{F}_p$:

$$\underbrace{\hat{L}_\lambda(f_p)}_{\text{Regularized estimated loss}} = \overbrace{\frac{1}{n}\sum_{i=1}^{n} \ell(f_p(\vec{x}_i), y_i)}^{=\hat{L}(f_p)} + \underbrace{\frac{\lambda}{n}\sum_{j=1}^{\bar{p}-1} w_j^2}_{\text{"Regularizer"}}$$

Regularization parameter $\lambda \in [0, \infty)$

$j=0$ not included

Minimizing $\hat{L}_\lambda(f)$ instead of $\hat{L}(f)$ is called
"Ridge Regression"

Let $\hat{f}_\lambda = \arg\min_{f \in \mathcal{F}} \hat{L}_\lambda(f)$, $f^* = \arg\min_{f \in \mathcal{F}} L(f)$

fix $\mathcal{F}$

If λ increases, then $\hat{f}_\lambda$ gets simpler

, $\bar{f}_\lambda$ gets simpler

, but f^* does not change

$\bar{f}_\lambda \neq f^*$ unless $\lambda = 0$

Bias vs. Variance

Bias: $(\bar{f}_\lambda(\vec{x}) - f_{Bayes}(\vec{x}))^2$

- Decreases if λ decreases

Variance: $E[(\hat{f}_{D,\lambda}(\vec{x}) - \bar{f}_\lambda(\vec{x}))^2 | \vec{x}]$

- Increases if λ decreases
- Decreases if n increases

Minimizing $\hat{L}_\lambda(f)$

$$\hat{\vec{w}}_\lambda = \arg\min_{\vec{w} \in \mathbb{R}^{d+1}} \hat{L}_\lambda(\vec{w}) \quad \text{using squared loss, } F_1$$

$$\text{where} \quad \hat{L}_\lambda(\vec{w}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}(\vec{x}_i^T\vec{w} - y_i)^2}_{\hat{L}(\vec{w})} + \underbrace{\frac{\lambda}{n}\sum_{j=1}^{d} w_j^2}_{g(\vec{w})}$$

There is a closed form solution but it is more complicated so we use gradient descent to find the minimum instead

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

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

$$\frac{\partial \hat{L}_\lambda}{\partial w_k}(\vec{w}) = \frac{\partial \hat{L}}{\partial w_k}(\vec{w}) + \frac{\partial g}{\partial w_k}(\vec{w}) \qquad k \in \{0, \ldots, d\}$$

$$\frac{\partial g}{\partial w_0}(\vec{w}) = 0$$

$$\frac{\partial g}{\partial w_k}(\vec{w}) = \frac{\lambda}{n} \cdot 2 w_k \qquad K = \{1, \ldots, d\}$$

$$\frac{\partial L_\lambda}{\partial w_0}(\vec{w}) = \frac{2}{n}\sum_{i=1}^{n}(\vec{x}_i^T\vec{w} - y_i)\,x_{i0}$$

$$\frac{\partial L_\lambda}{\partial w_k}(\vec{w}) = \frac{2}{n}\sum_{i=1}^{n}(\vec{x}_i^T\vec{w} - y_i)\,x_{ik} + \frac{2\lambda}{n} w_k \qquad k \in \{1,\dots,d\}$$

$$\hat{f}_\lambda = \underset{f \in F_{10}}{\operatorname{argmin}}\ \hat{L}_\lambda(f)$$