Important Announcements

March 4

- BGD: $\nabla \hat{L}(\vec{w}) = \left(\frac{\partial}{\partial w_0} \hat{L}(\vec{w}), \ldots \right.$
- $O(n^3)$ for inverse calculation
- MBGD $\nabla \hat{L}_m(\vec{w})$

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 $f \in \underbrace{\mathcal{F}}_{\text{Function class}}$ that minimizes $\hat{L}(f)$

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 $L(\hat{f})$

Evaluating Predictors/Models

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

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

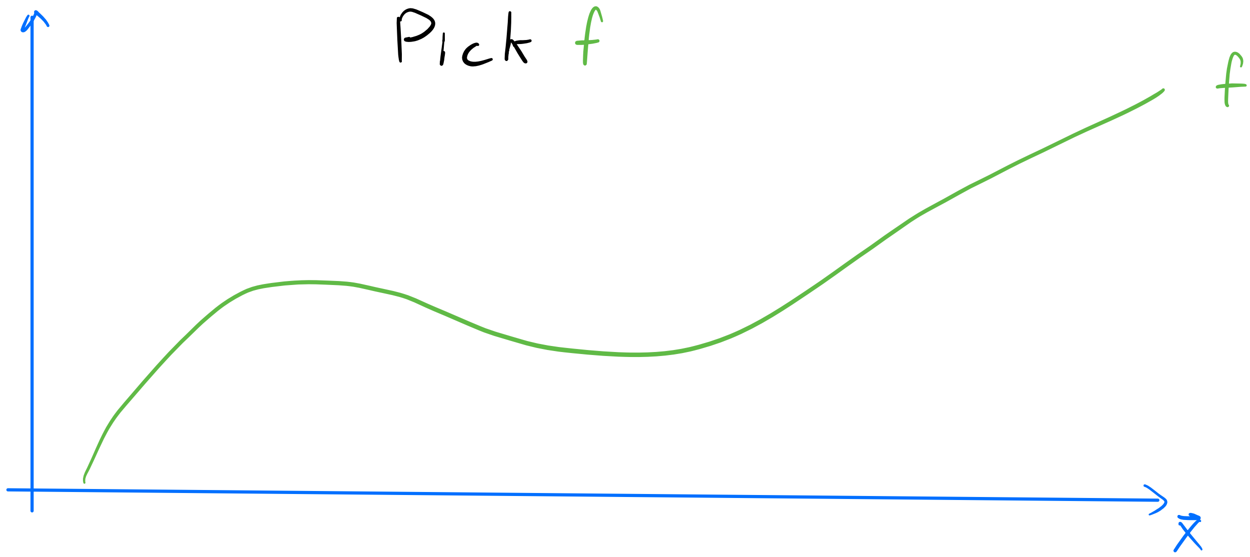
$$\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(\vec{X}_i), Y_i) \qquad L(f) = \mathbb{E}[\ell(f(\vec{X}), Y)]$$

(estimate of $L(f)$) (expected loss)

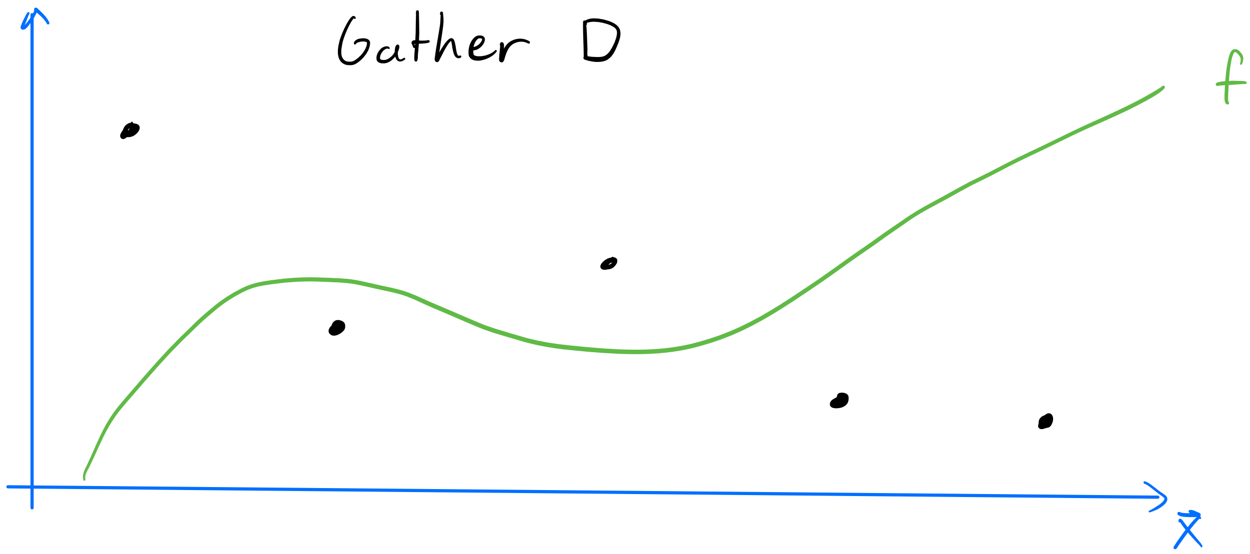
If we pick $f \in \mathcal{F}$ and then gather D

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

Pick f



Gather D



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

$$\mathrm{Var}[\hat{L}(f)] = \mathrm{Var}\left[\frac{1}{n}\sum_{i=1}^{n} \ell(f(\vec{X}_i), Y_i)\right]$$

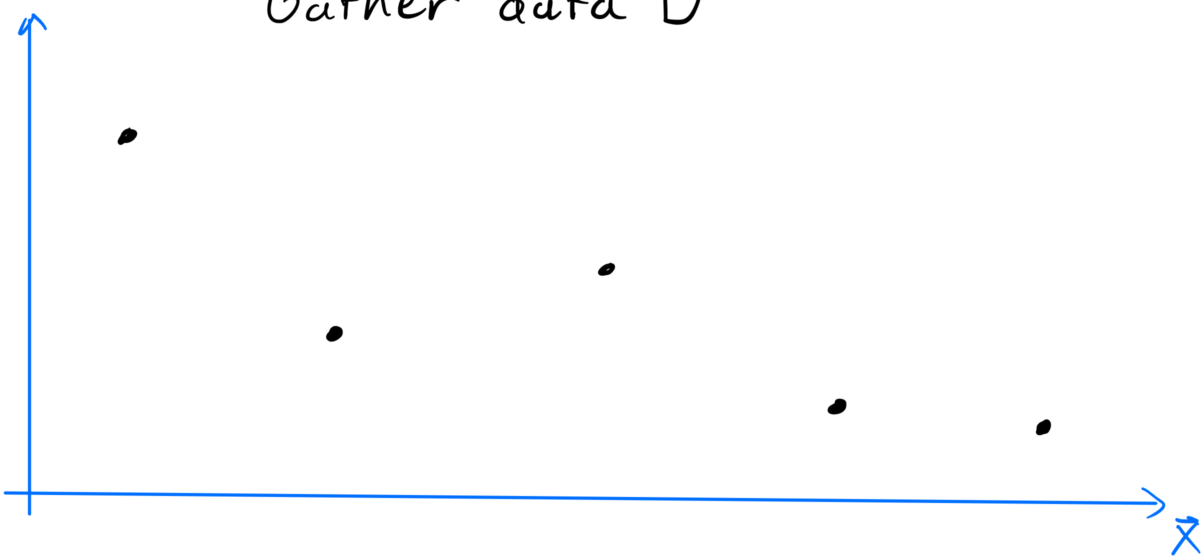
$$= \frac{1}{n^2}\sum_{i=1}^{n} \mathrm{Var}[\ell(f(\vec{X}_i), Y_i)]$$

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

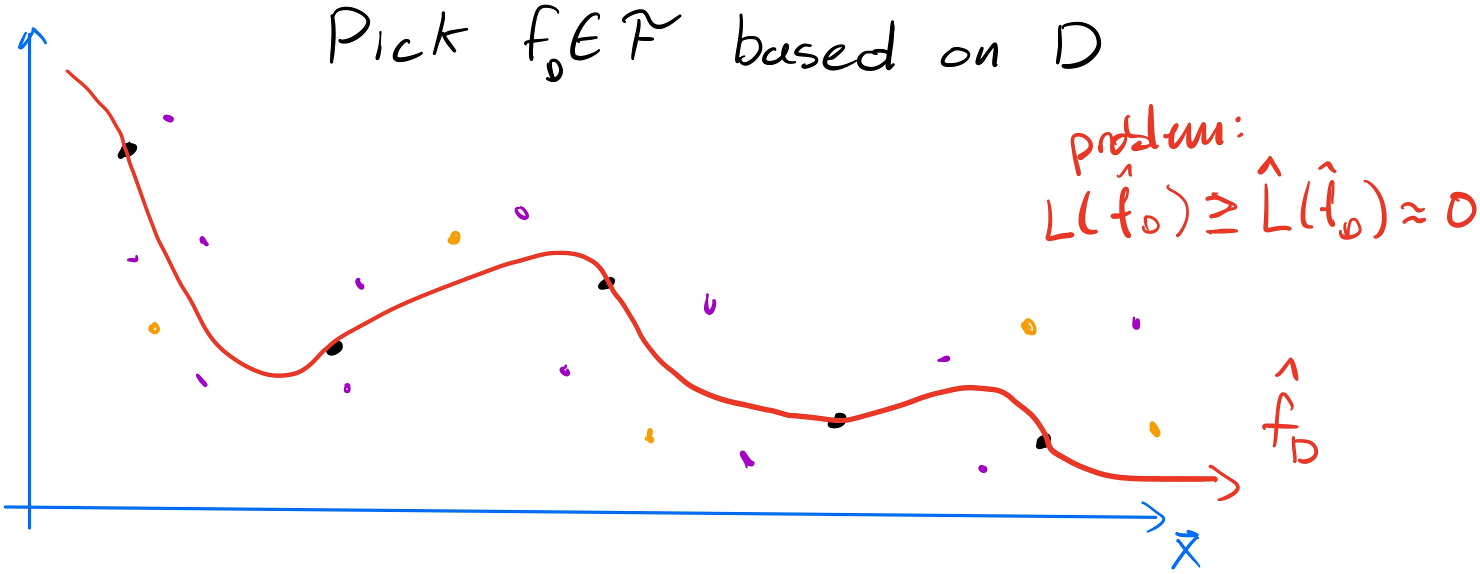
Since $\ell(f(\vec{X}_i), Y_i)$ are independent for all $i \in \{1, \dots, n\}$

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

Gather data D



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



Then:

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

$$\neq \mathbb{E}[\ell(f(\vec{X}_1), Y_1)]$$

$$\mathrm{Var}[\hat{L}(\hat{f}_D)] = \mathrm{Var}\left[\frac{1}{n}\sum_{i=1}^{n} \ell(\hat{f}_D(\vec{X}_i, Y_i))\right]$$

$$\neq \frac{1}{n^2}\sum_{i=1}^{n} \mathrm{Var}[\ell(\hat{f}_D(\vec{X}_i), Y_i)]$$

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

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

Instead:

This difference increases as $\mathcal{F}$ gets more complex

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

$\rightarrow$ decreases as n increases

$$\mathrm{Var}[\hat{L}(\hat{f}_D)] \leq \frac{1}{n} \max_{f \in \mathcal{F}} \mathrm{Var}[\ell(f(\vec{X}_1), Y_1)] + \mathrm{Var}[L(\hat{f}_D)]$$

As $\mathcal{F}$ gets larger, $\mathrm{Var}[\hat{L}(\hat{f}_D)]$ increases

$\Rightarrow$ i.e. $\hat{L}(\hat{f}_D)$ becomes a worse estimate of $L(\hat{f}_D)$

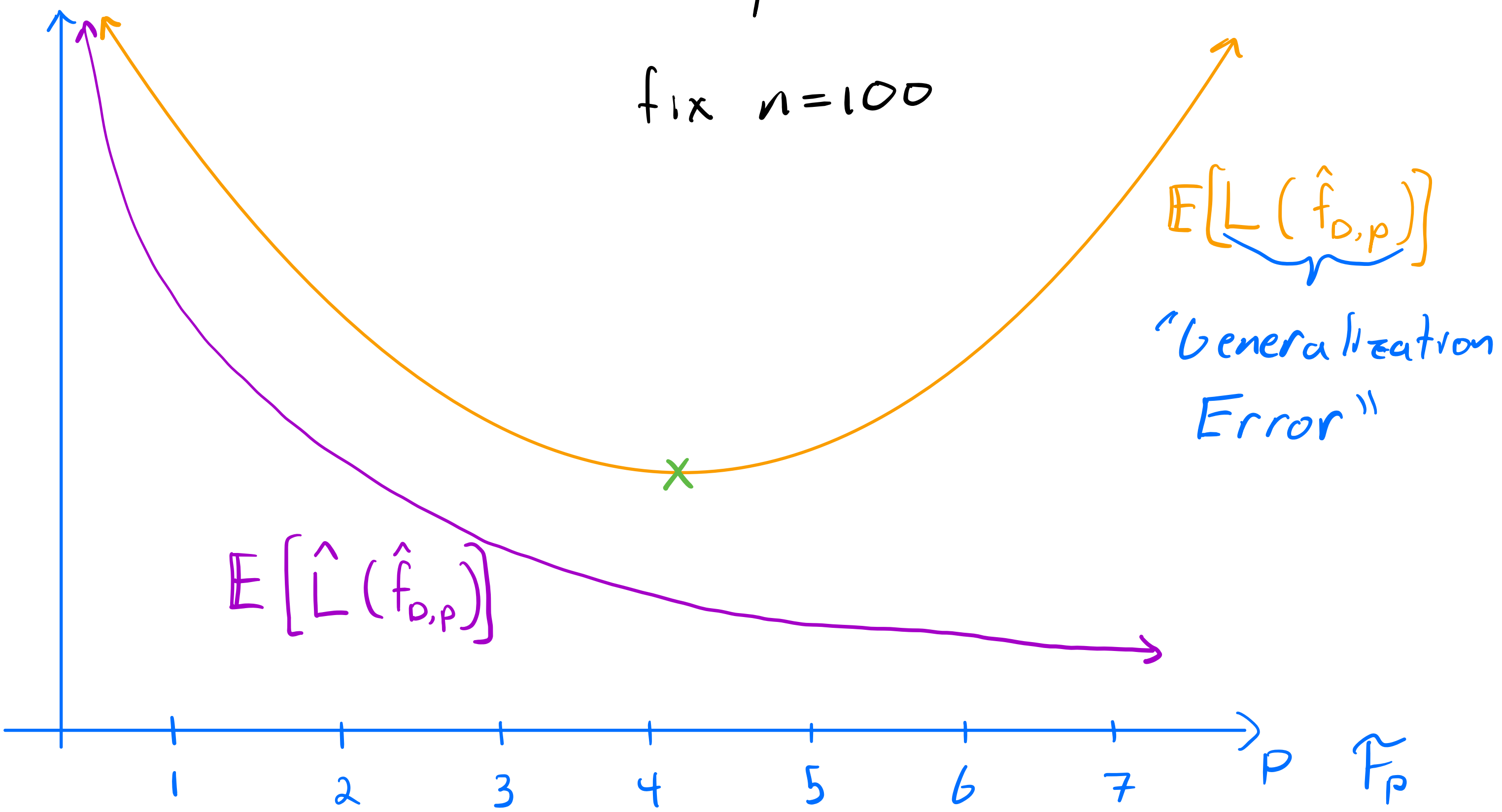
As n gets larger, $\mathrm{Var}[\hat{L}(\hat{f}_D)]$ decreases

$\Rightarrow$ $\hat{L}(\hat{f}_D)$ becomes a better estimate of $L(\hat{f}_D)$

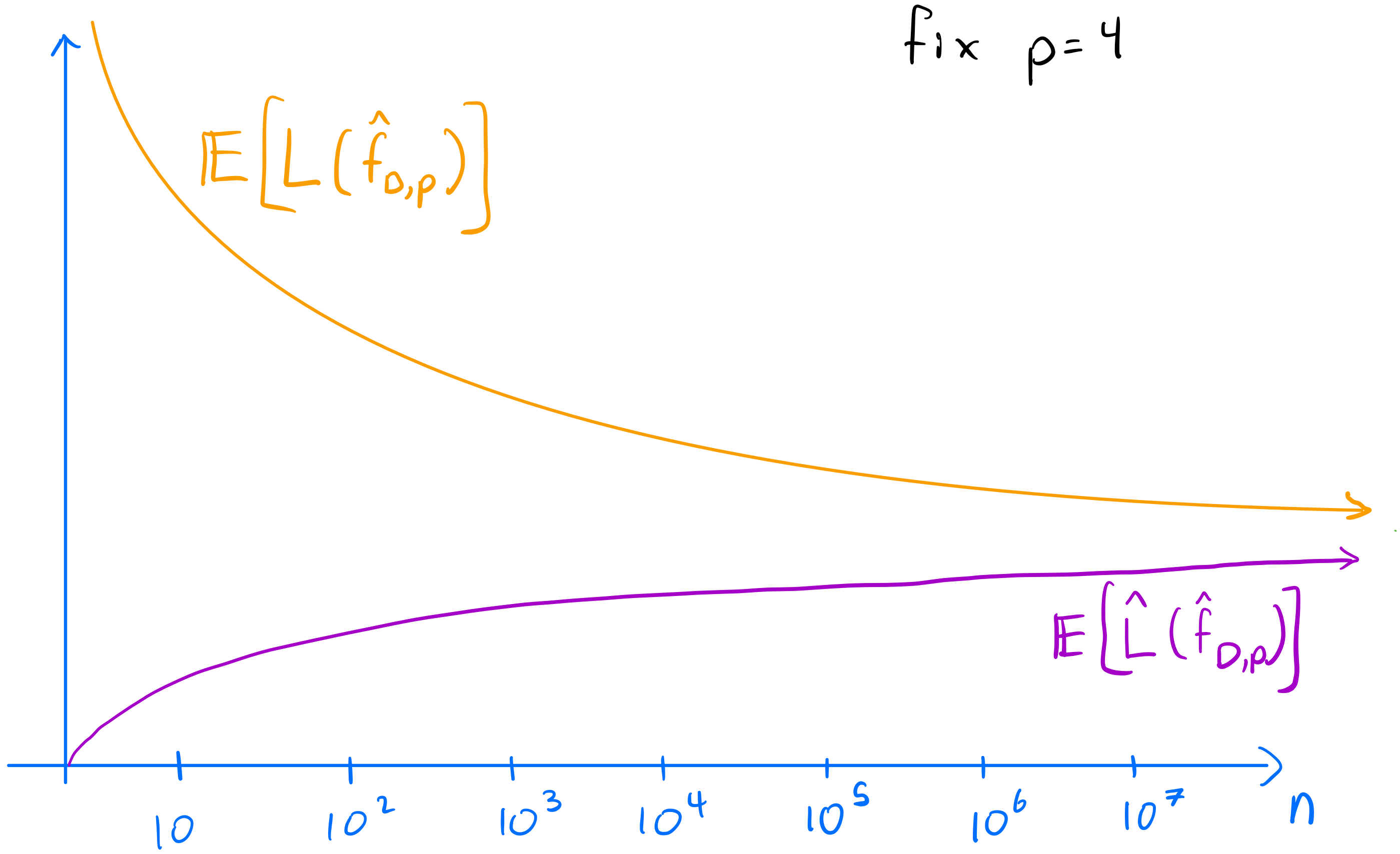
Ex:

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

fix $n=100$



fix $p=4$



Objective (formal):

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

such that $\underbrace{\mathbb{E}[L(\mathcal{A}(D))]}_{\text{(expected loss)}}$ is small over the datasets

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

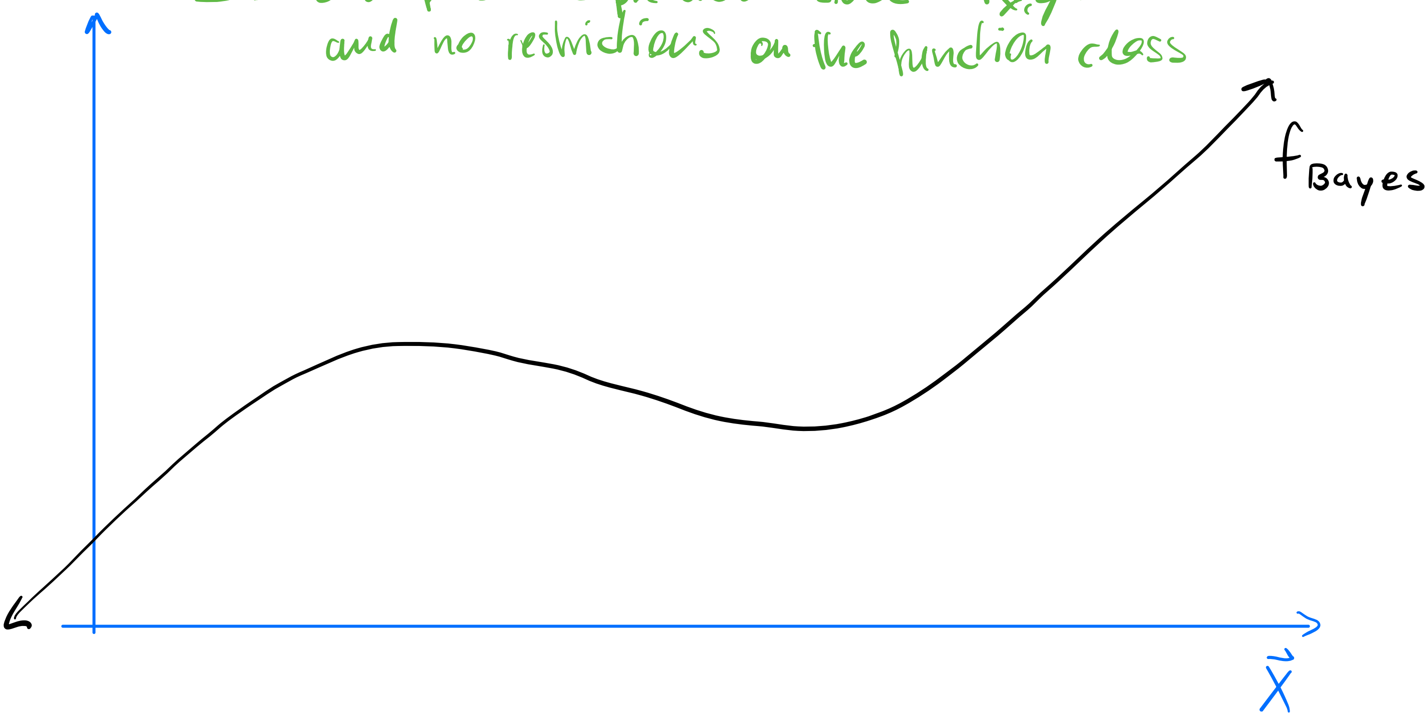
$$f_{Bayes} = \underset{f \in \{f \mid f : \mathcal{X} \to \mathcal{Y}\}}{\text{argmin}} \; L(f) \quad \text{(true expected loss)}$$

"Bayes optimal predictor"

no restriction on the function class

where $L(f) = \mathbb{E}[\ell(f(\vec{X}), Y)]$ and $(\vec{X}, Y) \sim \mathbb{P}_{\vec{X},Y}$

=> best possible predictor since $P_{\vec{X},Y}$ known and no restrictions on the function class

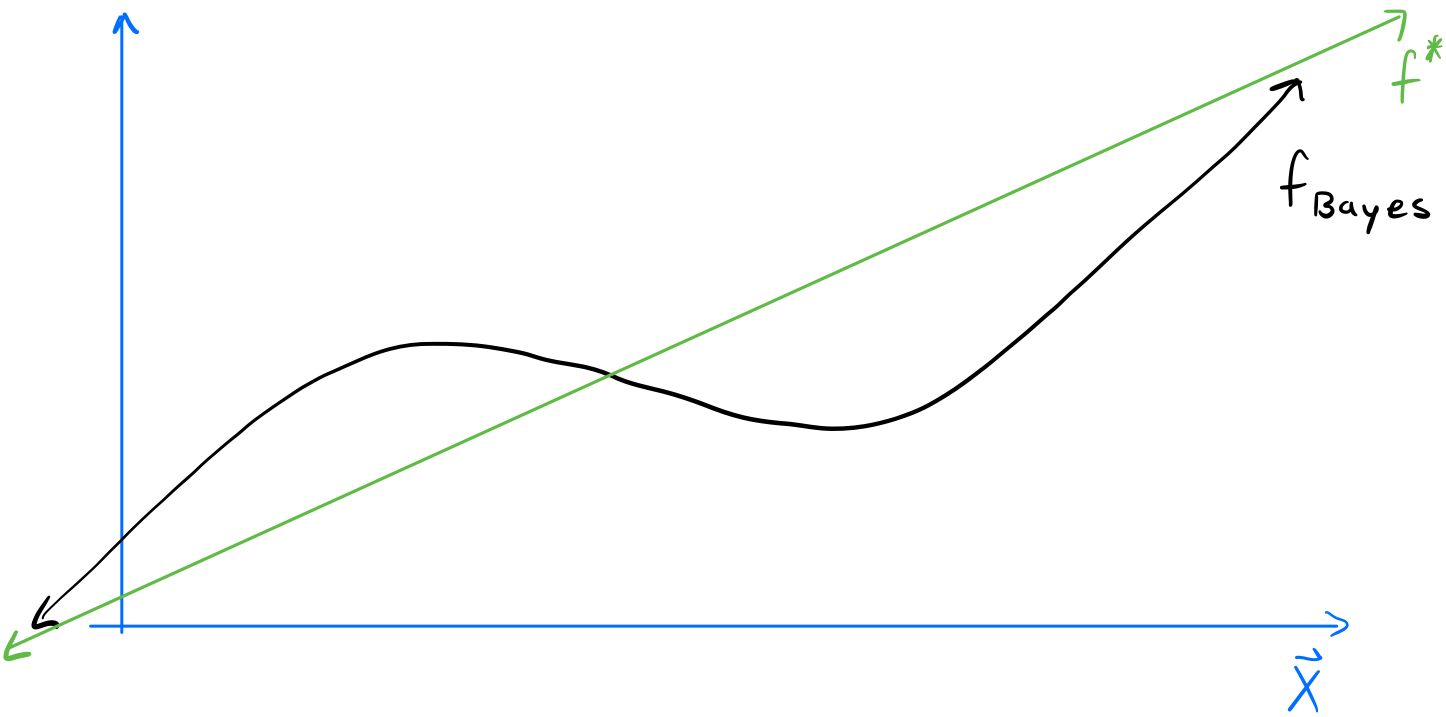


Suppose we knew $\mathbb{P}_{\vec{X},Y}$ but $\mathcal{A}:(\mathcal{X}\times\mathcal{Y})^n \to \mathcal{F}_1$

linear functions

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

best possible predictor in the function class



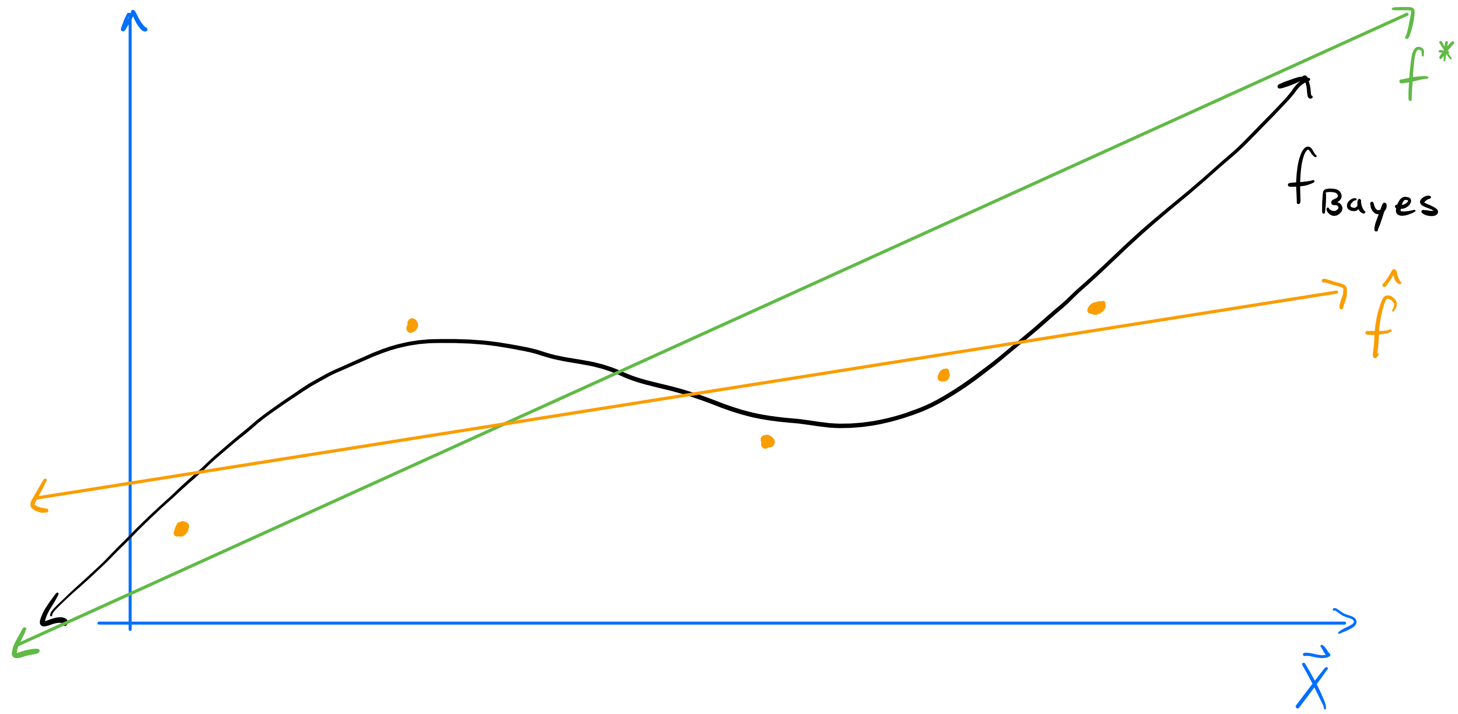
Suppose we didn't know $\mathbb{P}_{\vec{X},Y}$ but we had a dataset $D_1 = ((\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n))$ and $\mathcal{A}:(\mathcal{X}\times\mathcal{Y})^n \to \mathcal{F}_1$

our original setting

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

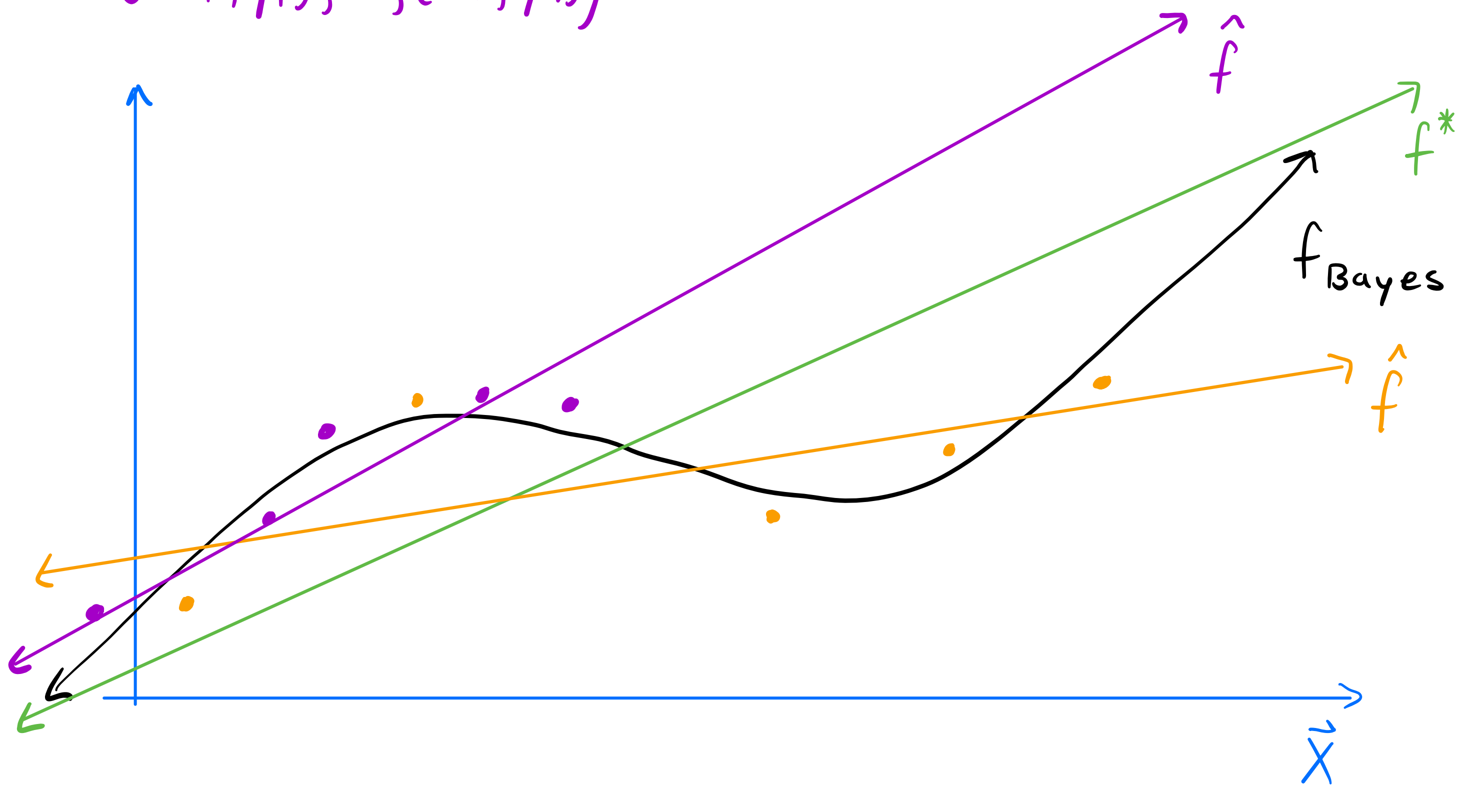
(estimate of the true expected loss)

$n = 5$

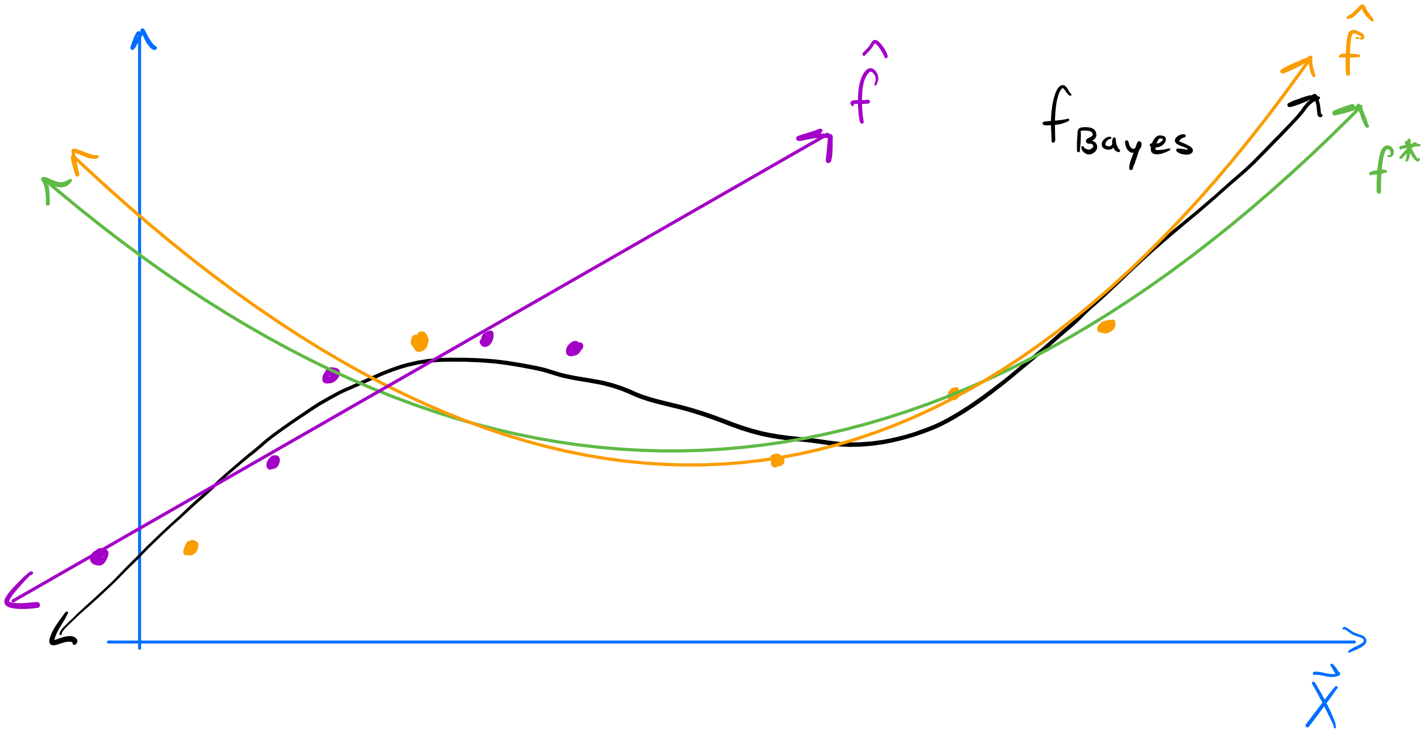


How about for a different dataset

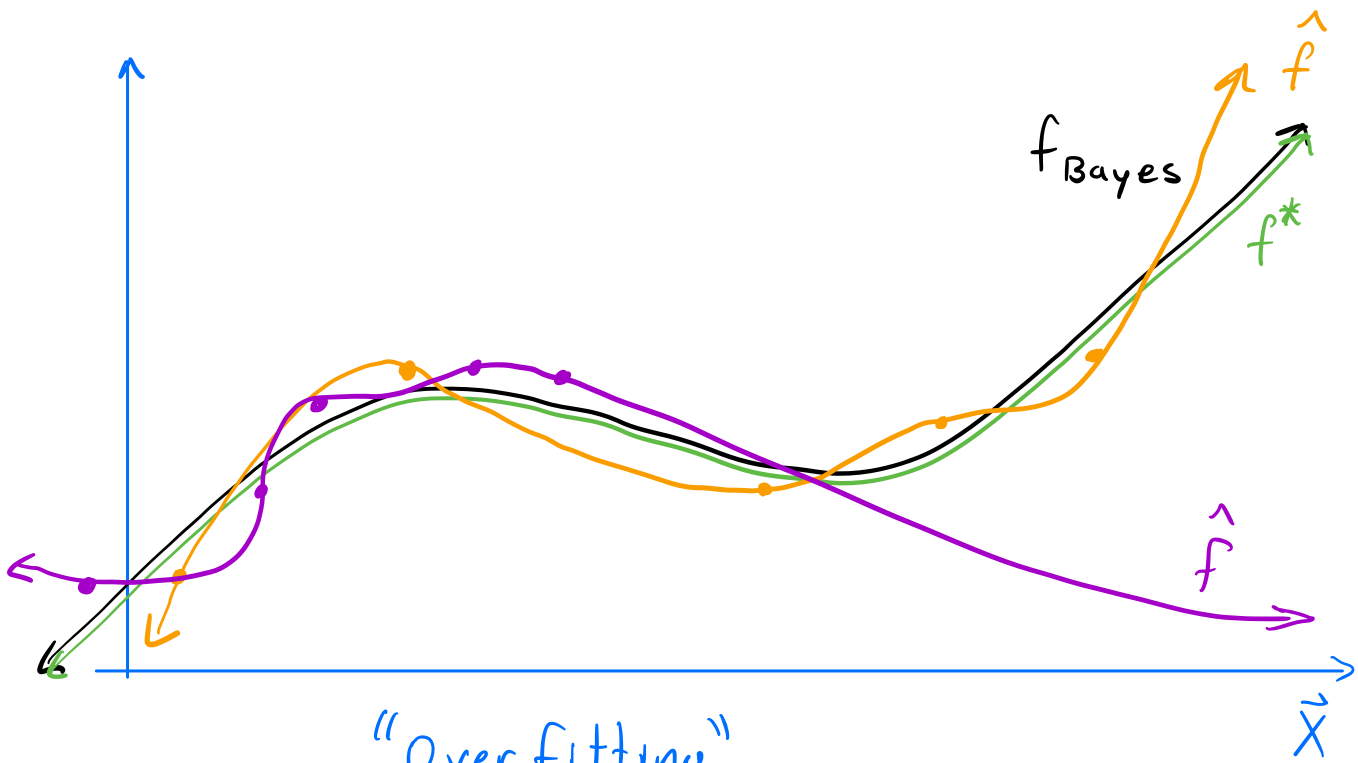
$D_2 = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$



How about for F_2



How about if F_{10}



"Overfitting"

$\Rightarrow \hat{f}$ and $\hat{f}$ very different for different samples of the dataset

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

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

$\mathcal{F}$ any function class

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

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

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

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

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

why EE $\uparrow$ if n decreases / $\mathcal{F}$ more complex

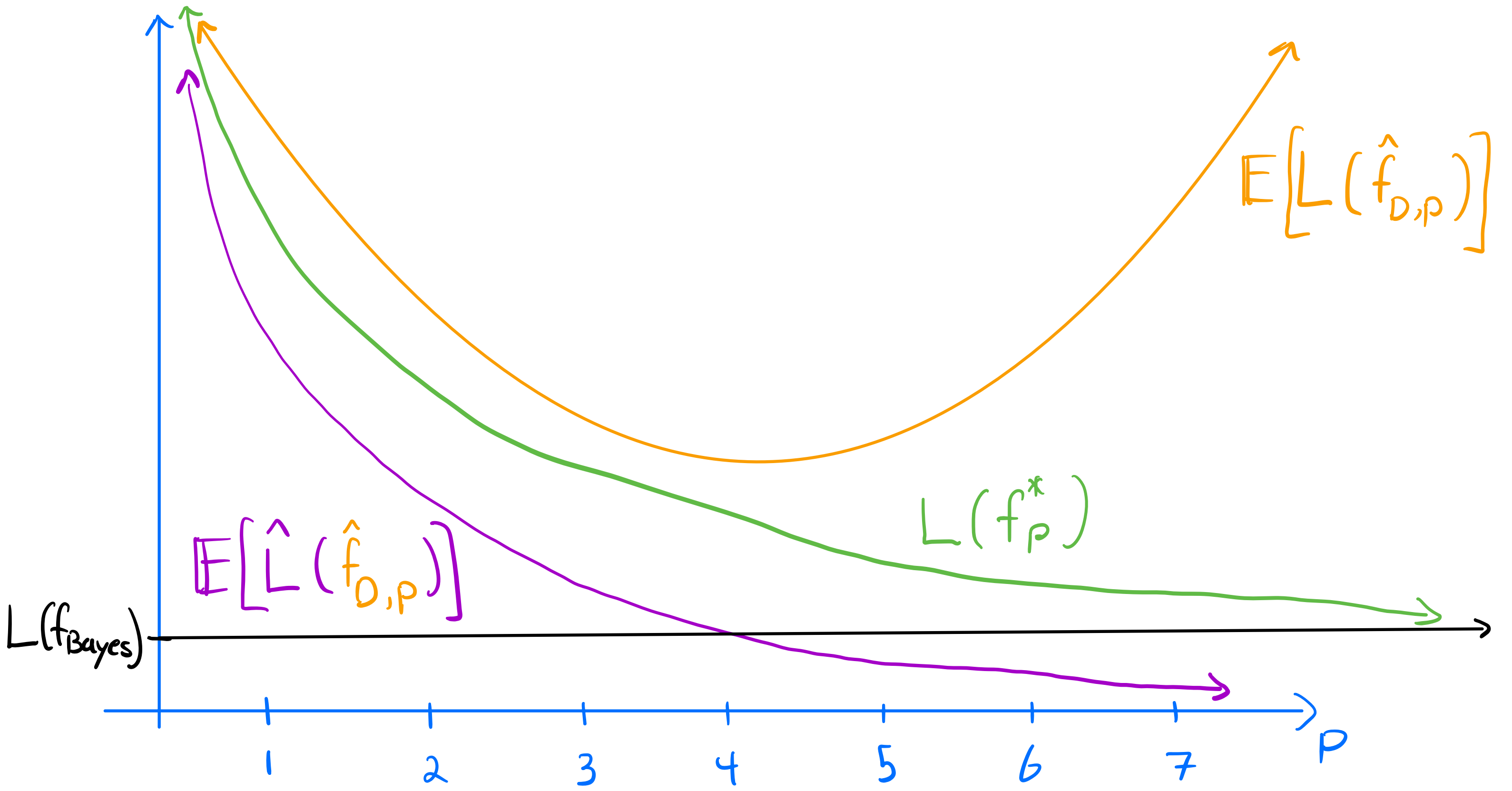
Understanding EE:

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

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

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

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

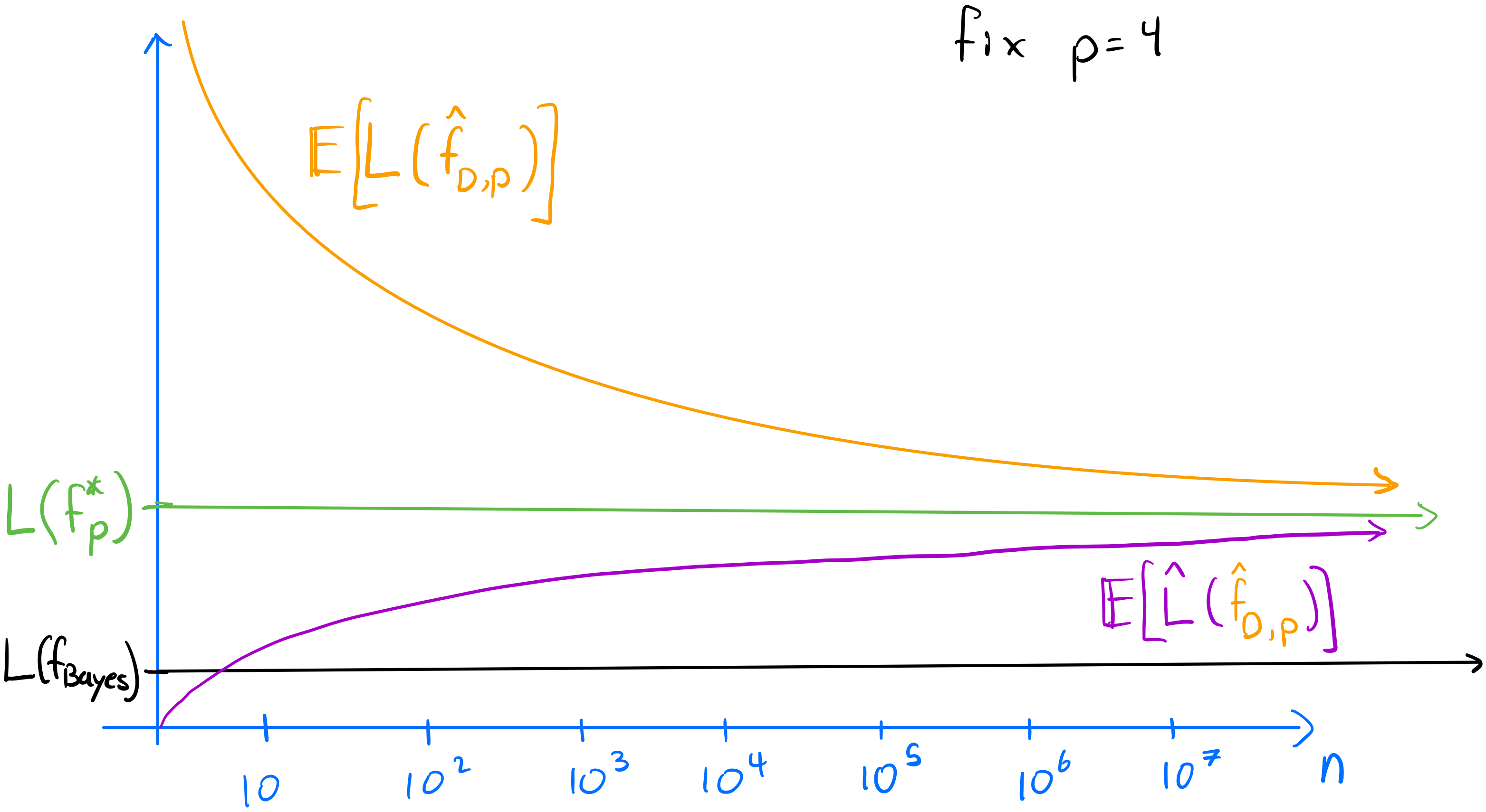


Underfitting: $\mathcal{F}$ is too simple (small) compared to n

- High AE, Low EE

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

- Low AE, High EE



fix p=4
E[L(f̂_{D,p})]
L(f*_p)
E[L̂(f̂_{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}

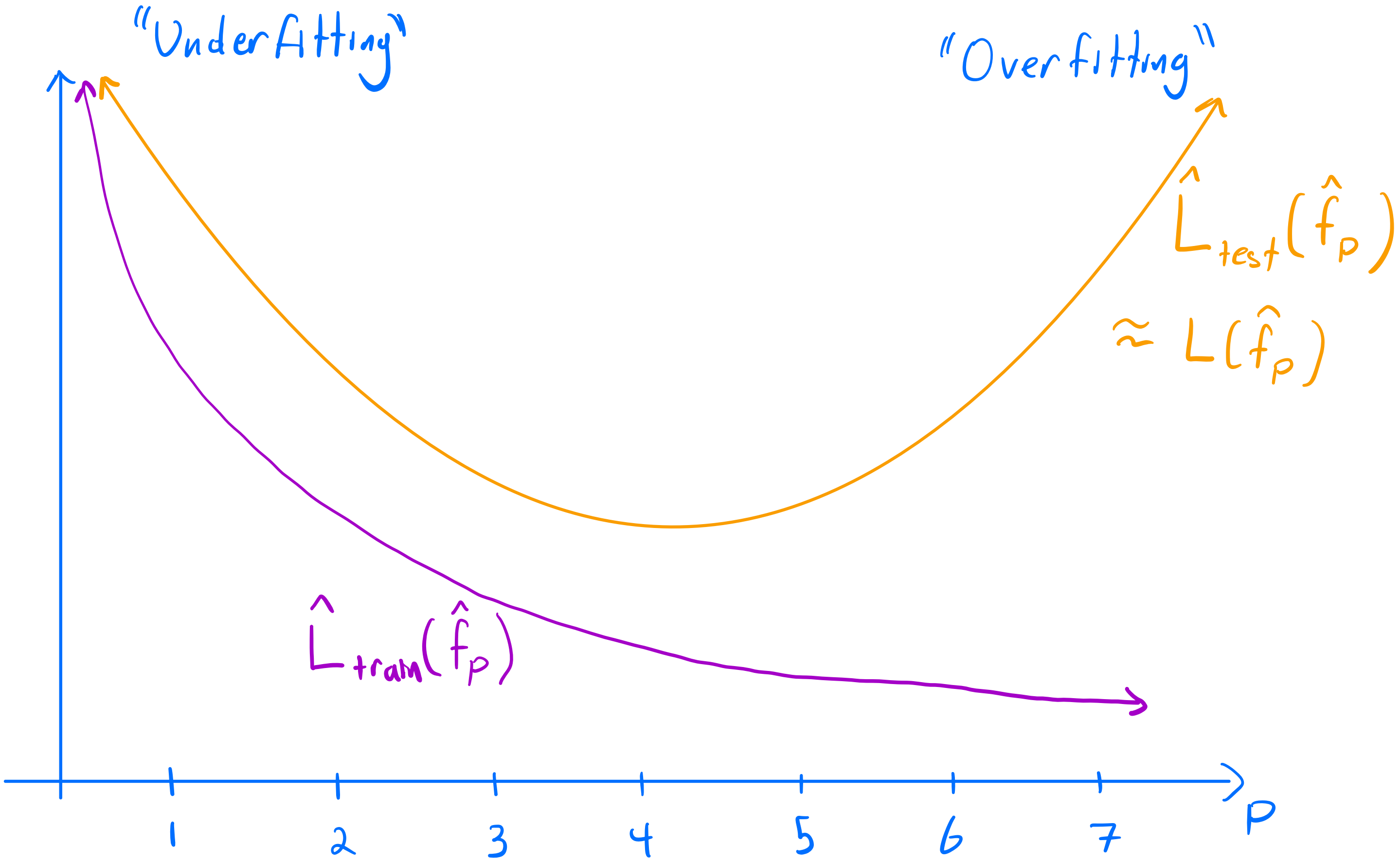
Since we can't gather new data we split D into D_{train}, D_{test}

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

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

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

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



Bias-Variance Tradeoff

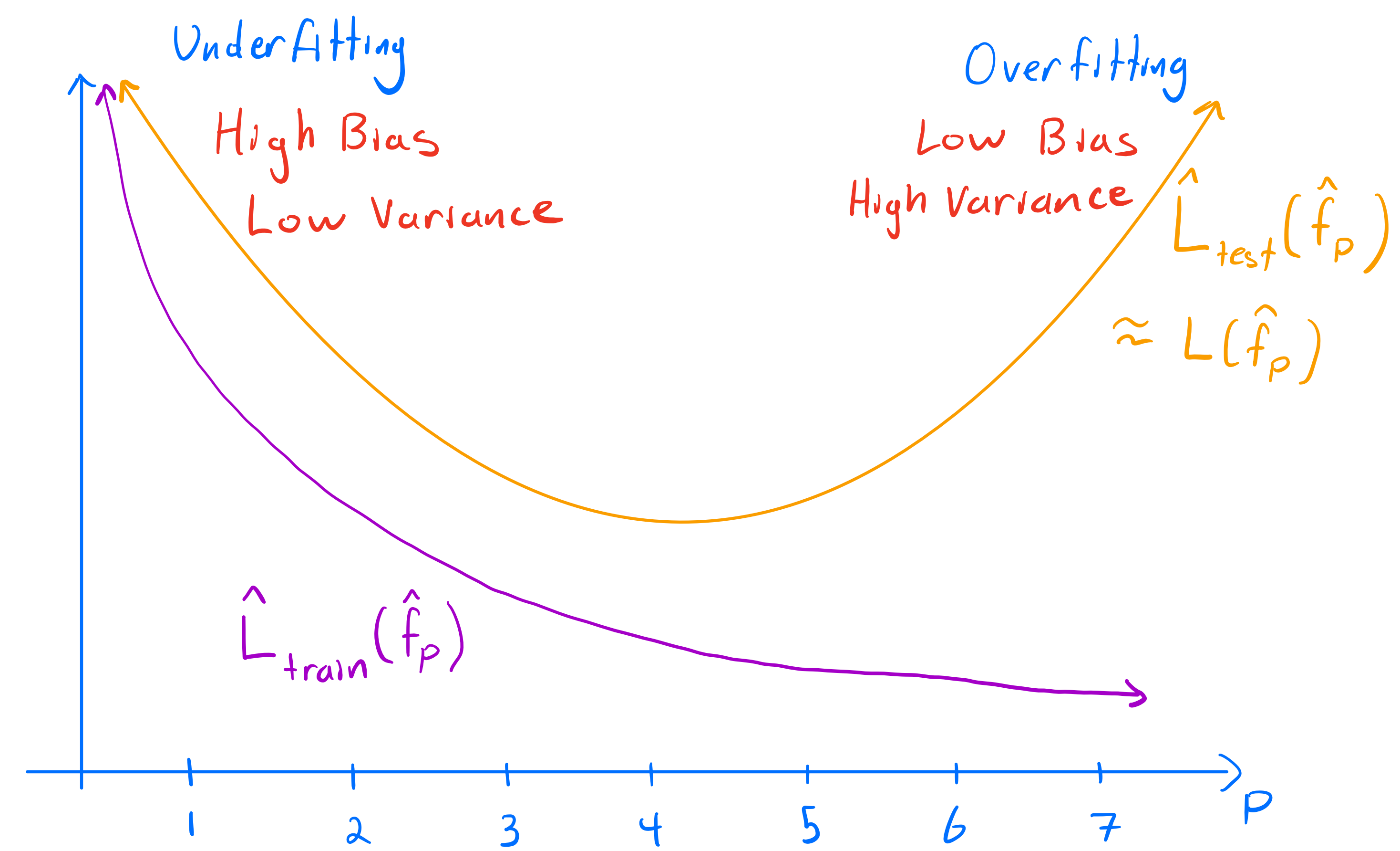
$$\mathbb{E}[L(\hat{f}_D)]$$

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

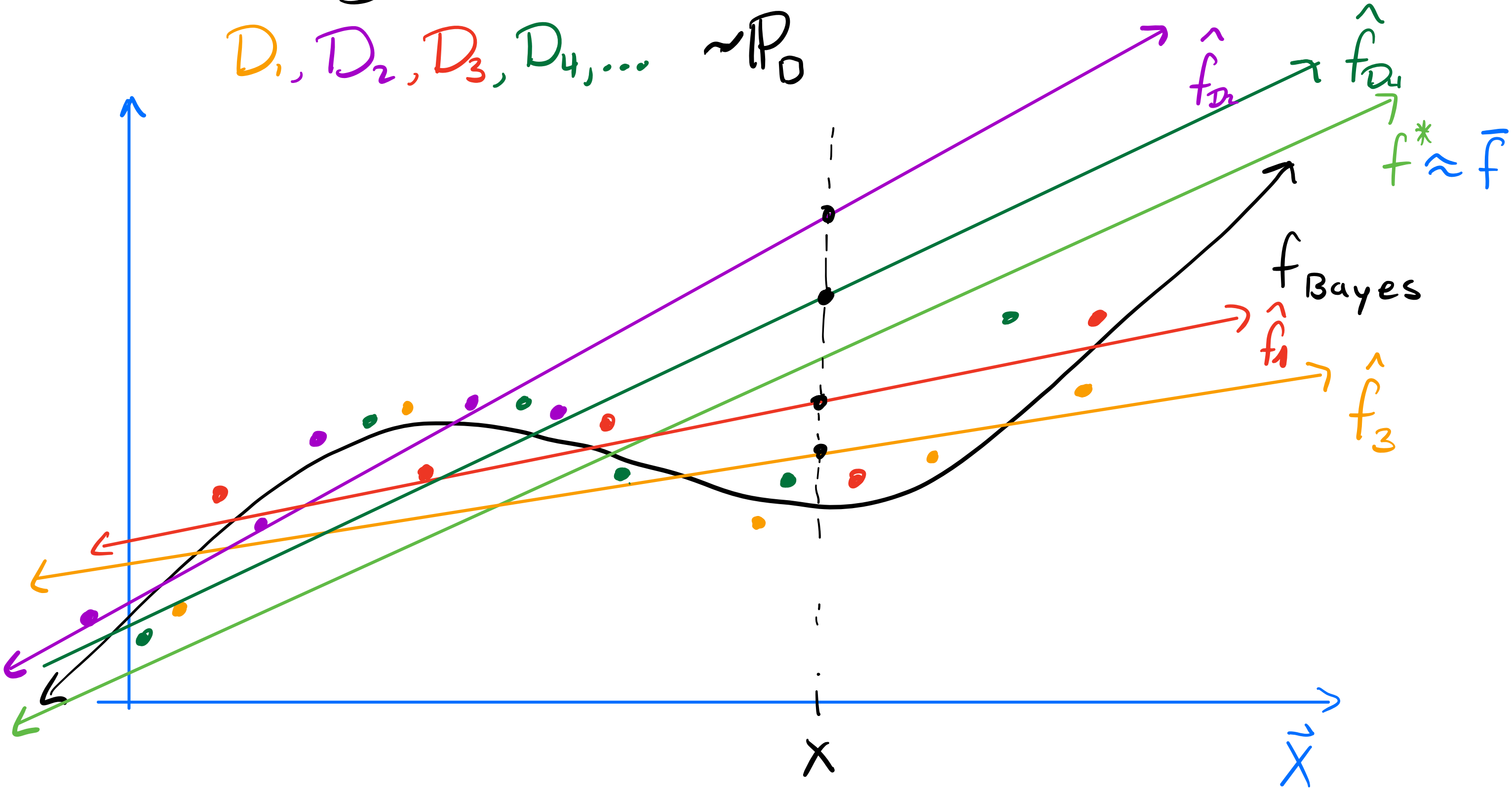
Bias $\downarrow$ if $\mathcal{F} \uparrow$

Variance $\downarrow$ if $n \uparrow$

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



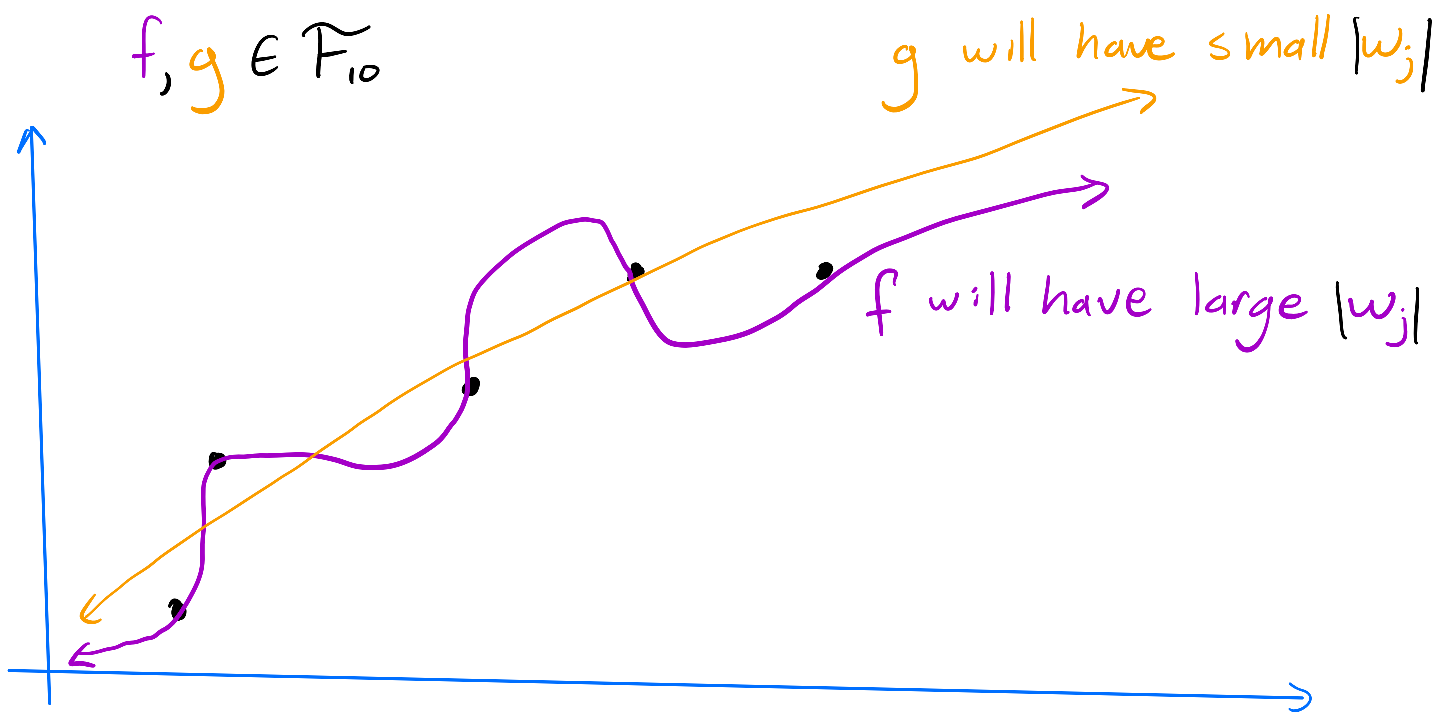
Visualizing $\bar{f}(x) = \mathbb{E}[\hat{f}_D(X) | X]$



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

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

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

"Ridge Regression"

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

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

- Decreases if λ decreases

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

- Increases if λ decreases
- Decreases if n increases

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

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

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

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

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

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