9/19/13

BIAS-VARIANCE

(1) How do we pick d in $^{(1D)}$ polynomial regression?

One Idea: $\hat{y}^{(0)} = w_0$ $\qquad x \in \mathbb{R}^1$

$$\hat{y}^{(1)} = w_0 + w_1 x^1$$

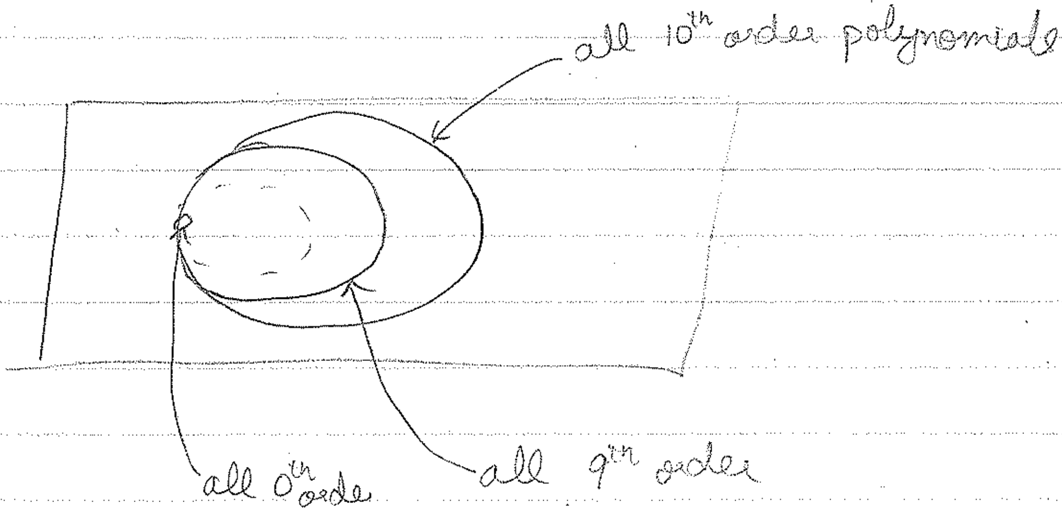
$$\vdots$$

$$\hat{y}^{(d)} = w_0 + w_1 x^1 + \cdots + w_d x^d$$

$$\vec{w}^{(d)} = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{\substack{\text{training} \\ \text{data}}} (y_i - \hat{y}_i^{(d)})^2$$

$$\hat{d} = \underset{d \in \{0, 1, \ldots, 100\}}{\operatorname{argmin}} \sum_{\substack{\text{training} \\ \text{data}}} (y_i - \hat{y}^{(d)})^2$$

Won't work too well. Model classes are nested,
$\hat{d} = 100$ will always gives lowest training error.



This is called the problem of "Model Selection"
→ How do I pick a model class to search in?

② Types of Error
→ Larger Model classes will always do better on training error, but that's not what we care about.

What we really care about → Expected Loss / Error
$X, Y \sim P(X, Y)$ [Unknown]

$$E_{P(X,Y)}\left[L(y, \hat{y}(x; w)) \right]$$

all parameters to hyperparameters

What we ideally want to do

$$\min_{w} \int_x \int_y L(y, \hat{y}(x; w))\, p(x, y)\, dx\, dy$$

Two problems
→ Integral hard to compute
→ $p(x, y)$ unknown

So lets approximate integral with samples $(x_i, y_i) \sim P(X, Y)$

$$E_{approx}(w) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{y}(x_i; w))$$

So here's how we optimize various quantities:

ALL DATA

~60% of N	~20%	~20%
TRAIN	VAL	TEST
$\underbrace{E_{train}}$	$\underbrace{E_{val}}$	$\underbrace{E_{test}}$

E_{train}: Used to fit model parameters

$\hat{w} = \underset{w}{\arg\min}\ E_{train}$

E_{val}: Used to choose model classes

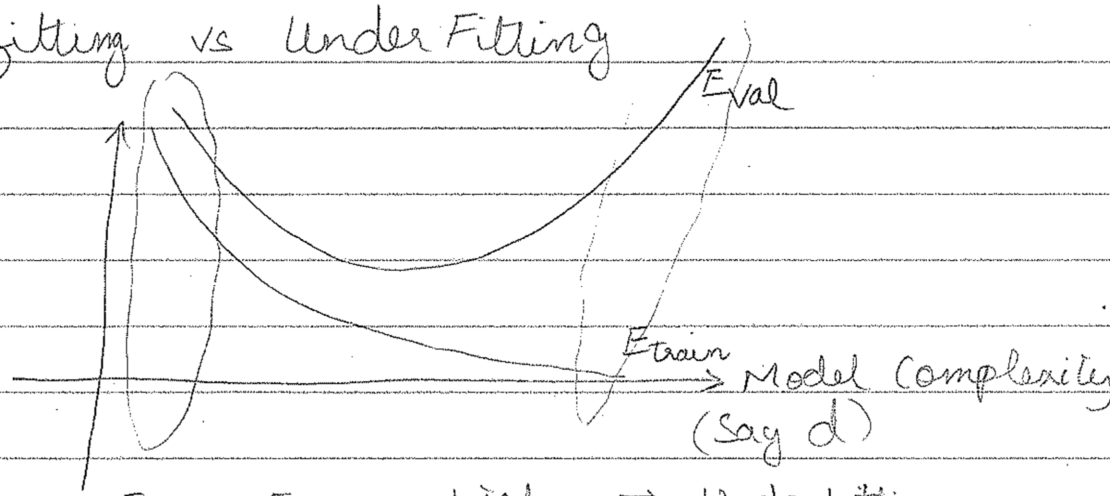
$\hat{d} = \underset{d \in \{0, \ldots, 100\}}{\arg\min}\ E_{val}$

E_{test}: Used to estimate Expected Error/Loss.

No tweaking, No learning on this, otherwise becomes a biased estimate.

If not enough data to do: (train, val) split we do cross-validation.

③ Overfitting vs Under Fitting



$E_{val} \approx E_{train} = \text{high} \Rightarrow$ Underfitting

$E_{val} \gg E_{train} \Rightarrow$ Overfitting

Overfitting $\Rightarrow$ Model Class too large, too expressive
Need more assumptions, Need smaller model class

Underfitting $\Rightarrow$ Model Class too small, too simple.
Need fewer assumptions

$\longleftrightarrow$

④ Bias - Variance

Eval is high for both overfitting & underfitting
Can we differentiate?

Yes

Bias   Variance

Back to basics: Coin Toss.

$D = \{X_1, \ldots, X_n\} \underset{\text{IID}}{\sim} Ber(\theta^*)$ (Say $\theta^* = 0.5$)

D	$\hat{\theta}_{MLE} = \frac{1}{N}\sum X_i$	$\hat{\theta}_{lazy} = X_1$	$\hat{\theta}_{silly} = 1$
{H, H, T}	2/3	1	1
{T, T, H}	1/3	0	1
{H, T, H}	2/3	1	1
⋮	⋮	⋮	⋮

$$E[\hat{\theta}_{MLE}] = \frac{1}{N}\sum_{i=1}^{N} E[X_i] = \frac{1}{N}\sum_{i=1}^{N}[1\cdot\theta^* + 0\cdot(1-\theta^*)]$$

$$= \frac{1}{N}\sum_{i=1}^{N}\theta^* = \theta^*$$

bias = $\underbrace{E[\hat{\theta}] - \theta^*}$

Difference between what your estimator reports on average & the Truth.

$\hat{\theta}_{MLE}$ is UNBIASED $\because$ $E[\hat{\theta}_{MLE}] = \theta^*$

What about others?

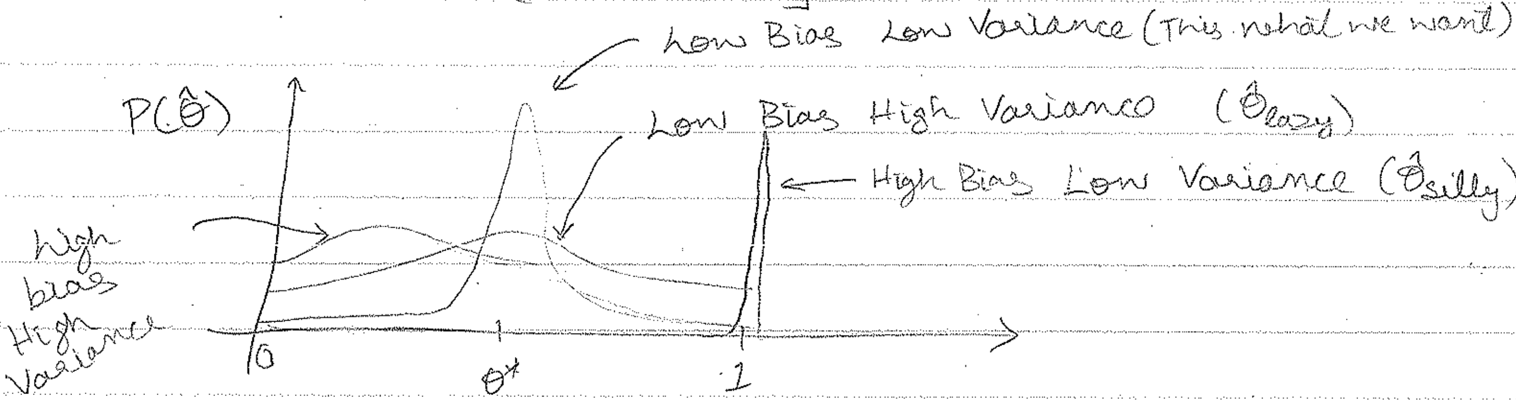
$E[\theta_{lazy}] = E[X_1] = \theta^*$ (Also Unbiased)

$E[\theta_{silly}] = E[1] = 1$ (Biased if $\theta^* \neq 1$)

High variance

No variance

Variance = $E[(\hat{\theta} - E[\hat{\theta}])^2]$



(5) Bias - Variance Decomposition for Squared - Loss

Say we want to estimate θ^*

Let $\bar{\theta} = E[\hat{\theta}]$ ← your estimator's avg value

Now $E[\text{Loss}] = E[(\hat{\theta} - \theta^*)^2]$

$= E[(\hat{\theta} - \bar{\theta} + \bar{\theta} - \theta^*)^2]$

$= E[(\hat{\theta}-\bar{\theta})^2 + (\bar{\theta}-\theta^*)^2 - 2(\hat{\theta}-\bar{\theta})(\bar{\theta}-\theta^*)]$

$= E[(\hat{\theta}-\bar{\theta})^2] + E[(\bar{\theta}-\theta^*)^2] - 2\underbrace{(\bar{\theta}-\theta^*)}_{\text{constants}} E[(\hat{\theta}-\bar{\theta})]$

$= \text{Var}(\hat{\theta}) + E[\text{bias}^2] - 2(\bar{\theta}-\theta^*)(\underbrace{E[\hat{\theta}] - \bar{\theta}}_{=0})$

$= \text{Var}(\hat{\theta}) + \text{bias}^2$

So $E[\text{Loss}] = \text{bias}^2 + \text{Var}(\hat{\theta})$

2 predictors (one with high bias, one with high loss can have the same L_2 loss)

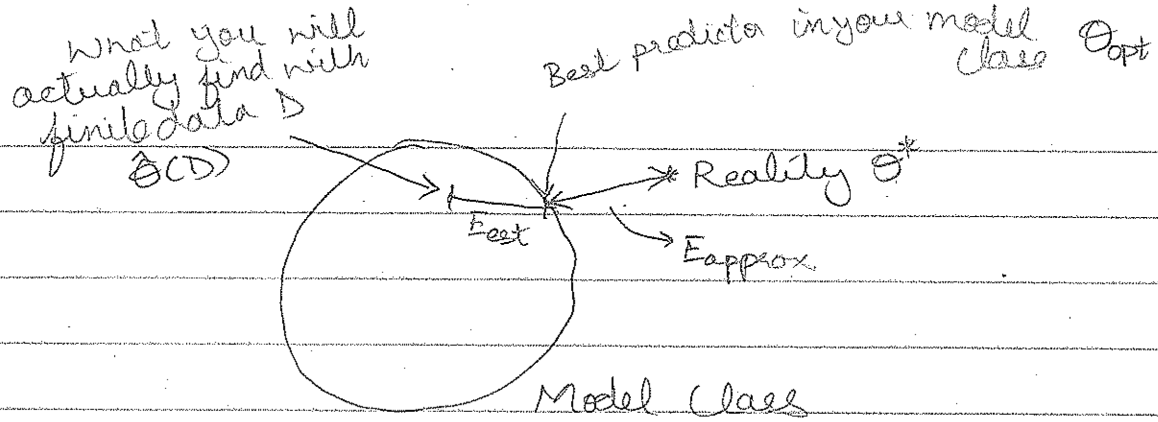
(6) General Error Decomposition

Expected Error = "Approximation Error" (or ≈ Bias)
→ you approximated reality with your model

+ "Estimation Error" (or ≈ Variance)
→ you estimated with finite data

+ "Optimization Error"
→ you didn't / couldn't maximize MLE exactly

+ Bayes Error
→ Lower Bound on error

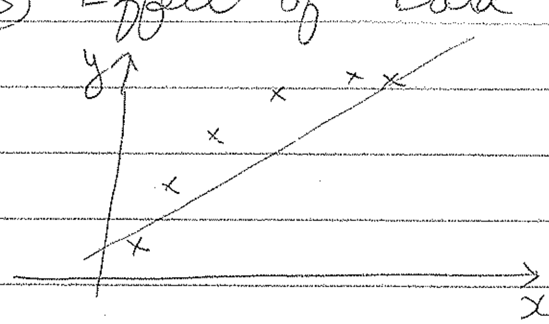


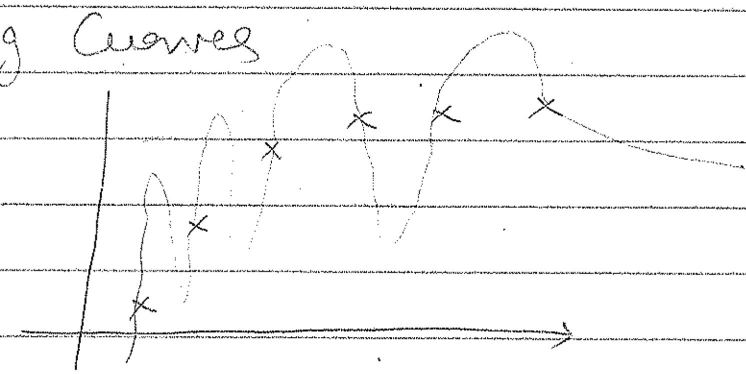
(7) Model Selection via Regularization
(Let's put in a preference for simpler models)

$$\min_{w} \sum_{train} L(y_i, \hat{y}(x_i; w)) + \lambda \|w\|_2^2$$

Can also be viewed as a Gaussian prior on w
(MAP estimator)

(8) Effect of More Data / Learning Curves





Simple Model Class
⇒ More dat won't help

Too Expressive Model Class
⇒ More data will help, but very slowly.