Lec 12 Math 320 3/19/19

Recall

$$y = g(\vec{x}) + \underbrace{h^*(\vec{x}) - g(\vec{x})}_{\text{estimation error}} + \underbrace{1^*(\vec{x}) - f(\vec{x})}_{\text{misspecification error}} + \underbrace{t(\vec{x}) - f(\vec{x})}_{\text{error due to pure noise}}$$

$$\xi(\vec{x})$$

$$e(\vec{x})$$

Previously, we measured $e_i = y_i - \hat{y}_i = y_i - g(\vec{x}_i)$

the $i \in \{1, \dots, n\}$ since $\mathcal{D}$ has n observations inside. These $e_1, \dots, e_n$ are "in-sample residuals"! Comparisons of, SSE, RMSE, R^2 are all in-sample error metrics. We discard these since these in-sample error metrics can be made arbitrarily more perfect simply by adding additional variables (i.e. raise p).

This is not only the case with $\mathcal{A}$ = OLS, this is a problem for most supervised learning alg's. You can see it most starkly with OLS!

The $e_1, \dots, e_n$ are not good estimates of the ϵ here!

What is the ϵ there? Out of sample.

Imagine

		time
$\mathbb{D}$	past	↓
$\mathbb{D}_*$	future	

we forgot to mention... we need to...

Assume $t, \vec{z}$ are the same across time and that the $\vec{x}$'s are measured the same way. This is called "stationarity" of the phenomenon. We need this otherwise... problem won't work!

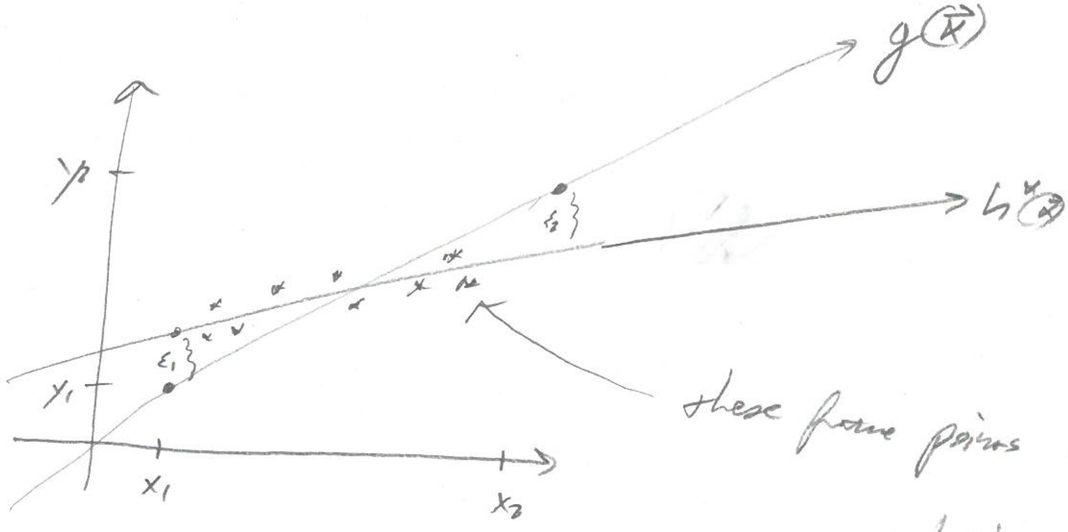
What are good estimates of e, the model's "true" error?

$e_* = y_* - \hat{y}_*$ for any $\vec{x}_* \in \mathbb{D}_*$. These are honest estimates? Why? They cannot be overfit. Overfitting is tantamount to memorizing the answers on the practice test. "Fitting" is learning the theory from the old exams to be able to answer new questions. "Underfitting" is not learning as much as you could've given studying the practice test.

Overfitting ..... extract some of the unknowable structure in ε and attributed it to model structure.

Overfitting in one dimension with intercept... (3

Recall $y = h^*(\vec{x}) + \varepsilon$, $n = 2$, $p = 1$ ⇒ perfect fit



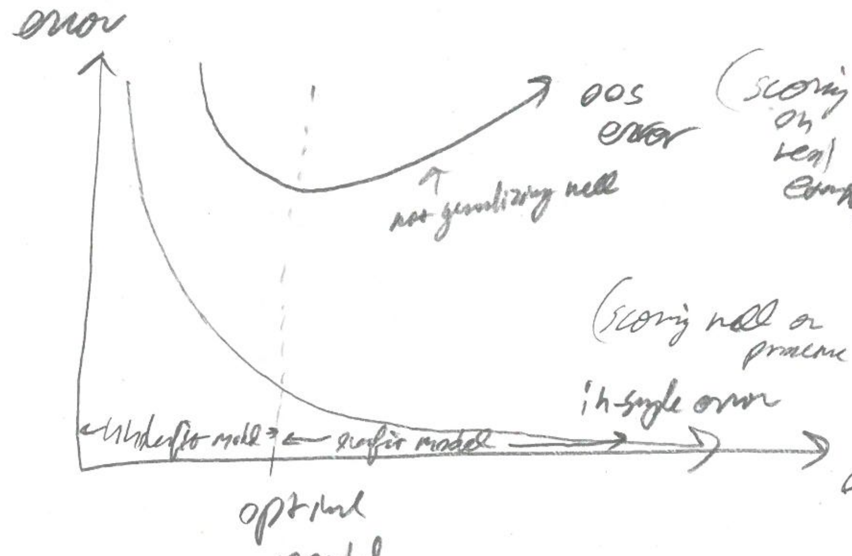
these frame pairs are further from g than h^*.
doesn't generalize well.

$$y = g + (h^* - g) + (f - h^*) + (t - f)$$

these three errors are all generalization errors
since they can be made $\to 0$ seemingly

Overfitting doesn't change h^* or f or t, only g.

⇒ Overfitting increases estimation error. (Demo)



(scoring well on real examples)

SSE, RMSE, R^2
in sample statistics

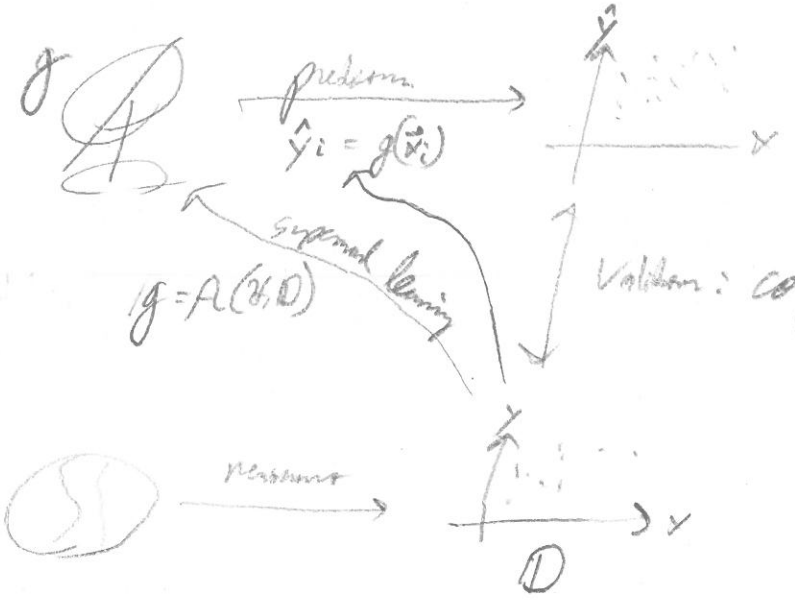
oos SSE, oos RMSE, oos R^2
oos statistics

(scoring well on previous examples you've seen before)

p59 Learning From Data

Recall

In-sample error $\to 0$. It's like memorizing the answers! Remember $y_1 \dots y_n$. KNN $K=1$!

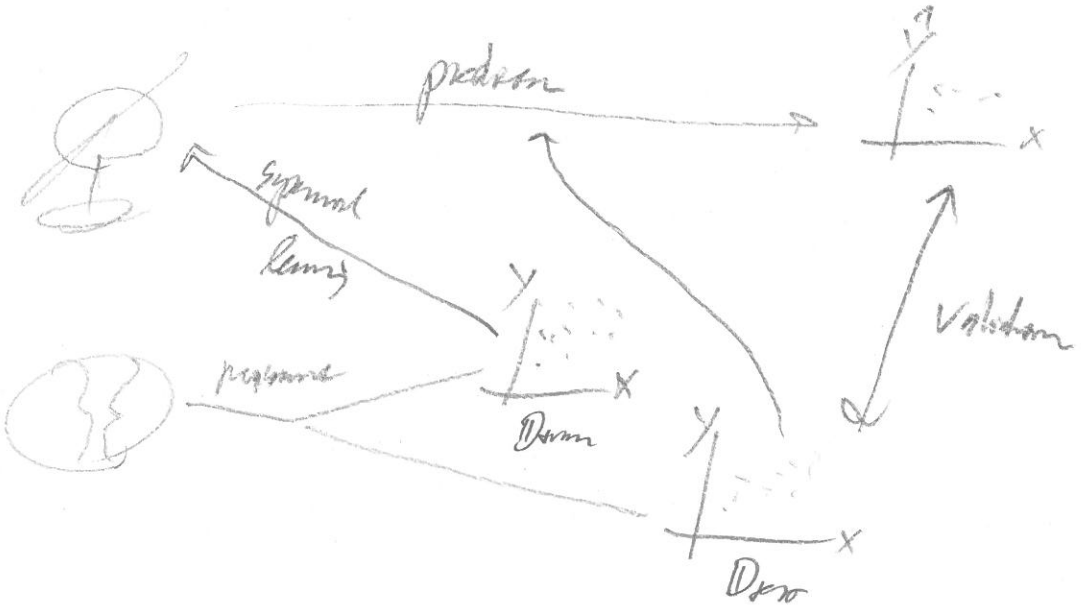


In-sample validation: use $\mathcal{D}$ to build g. Use $\vec{x}$ from $\mathcal{D}$ and compare to y. BAD! Can be unfair.

Solution: split $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$

$\mathcal{D}_{train}$ becomes the data used to build the model i.e. $g = A(\mathcal{H}, \mathcal{D}_{train})$

and $\mathcal{D}_{test}$ becomes the data used to validate

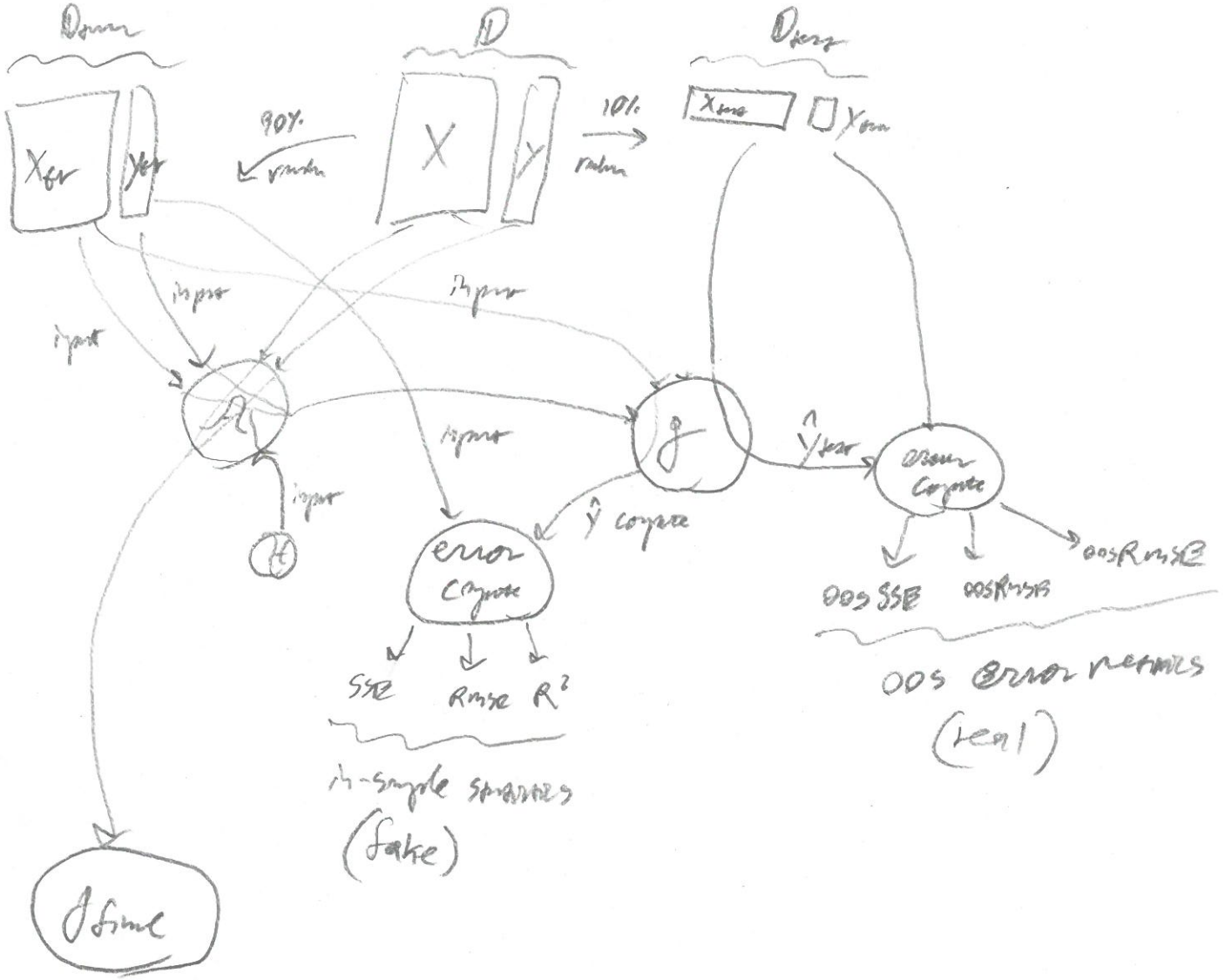


Final model: uses all $\mathcal{D}$.

typical split ratio: proportion left on for testing is 10%, 20%, 5%

Usual notation:

$K = \frac{1}{\rho}$ e.g. $\rho = 10\% \Rightarrow K = 10$. Run



$f \overset{?}{=} f_{final}$ No. f_{final} is better on avg than f. So OOS-error metrics are an upper bound / conservative estimate

Which error does f_{final} have less of? Estimation error! It is 11% higher!

Not huge gain... but why not take the cookie?