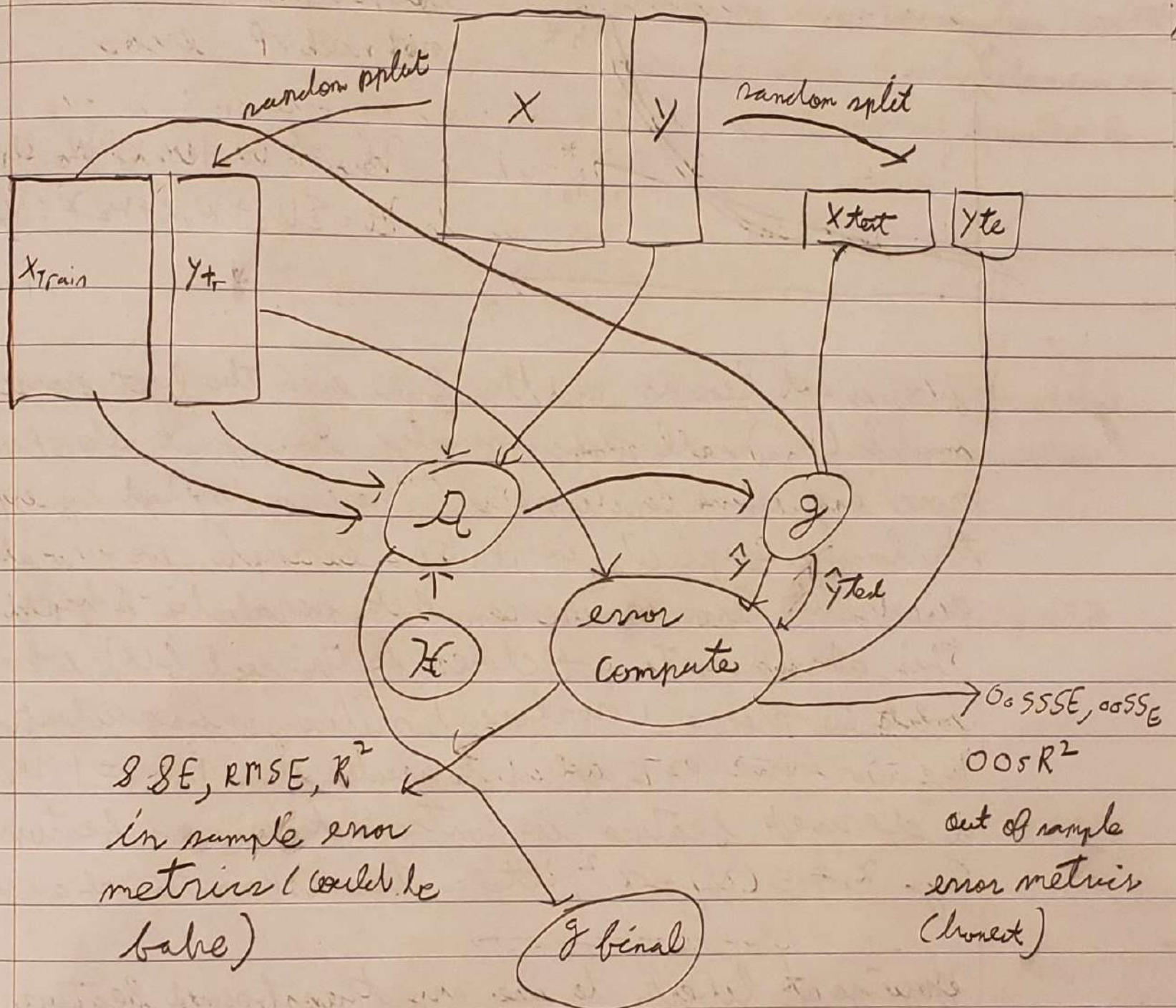


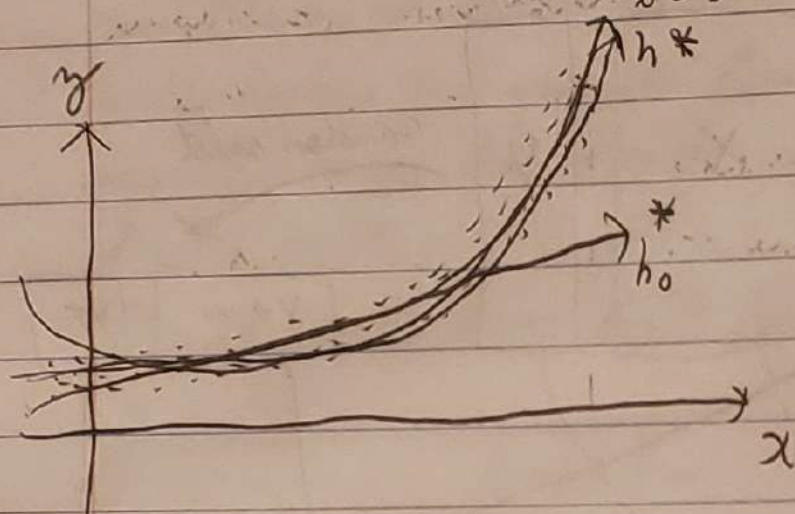
3/15/21

$K=10 \Rightarrow$ test set is 10% n



The g_{final} is the function used for future predictions. Its performance is at least as good as the oos metrics since you're running the same model fitting process but now n is slightly higher.

let $p=1$ feature, $y = g(x) + \underbrace{h^*(x) - g(x)}_{\text{high if } m \text{ misspecification}} + \underbrace{f(x) - h^*(x)}_{\text{not much } > p \text{ error}}$



$$\mathcal{H}_0 = \{w_0 + w_1 x; w_0, w_1 \in \mathbb{R}\}$$

$$\mathcal{H} = \{w_0 + w_1 x + w_2 x^2; w_0, w_1, w_2 \in \mathbb{R}\}$$

$f(x)$ is not linear and therefore even the best possible linear model (h_0^*) will perform poorly. So why not allow for a more expressive candidate set? We can do that by expanding the basis / complexity in \mathcal{H} . For example, we now allow for a quadratic term so we can fit parabolic-shaped curves. This allows us to get closer to the real $f(x)$ which may be very complex and non-linear, reducing misspecification error. We now have $p=2$ which is greater than $p_{\text{true}} = 1$. We call this a "derived feature" in contrast to a "raw feature" (original). Eg $x_2 = g(x_1) = x_1^2$. It's a transformation of a raw feature.

You're at liberty to use any transformed features you want. If they're useless, they appear as random noise and you can ignore them.

Using squares and cubes is a well-known modeling procedure called "polynomial regression".

Is polynomial regression "linear"? Yes and no. "Yes" in the sense, that you create a design matrix and use OLS and thus linear in the transformed features but "no" because the model is not linear in the raw features.

Advanced math note: polynomial regression is a principled approach because of the Weierstrass Approximation Thm (1885) which says that any continuous function f whose domain is x is $[a, b]$ can be approximated by a polynomial function P_d with arbitrary precision by picking d , the degree:

$$\forall \epsilon > 0 \quad \forall x \in [a, b] \quad \exists d \quad |f(x) - P_d(x)| < \epsilon$$

The Stone-Weierstrass Thm (1937) generalizes the above. One implication of this thm is that a mult. poly. function can approx. any contin. func. $f(x_1, \dots, x_p)$.

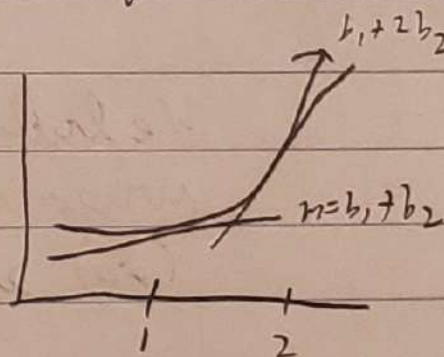
How do we do a polynomial regression of degree d , e.g. $d=2$

$$X_{\text{raw}} = \begin{matrix} & \vec{x} \cdot 1 \\ \begin{bmatrix} 1 & x_{1,1} \\ 1 & x_{1,2} \\ \vdots & \vdots \\ 1 & x_{1,n} \end{bmatrix} & \xrightarrow{\text{transform}} & X = \begin{matrix} \vec{x} \cdot 1 & \vec{x} \cdot 2 \\ \begin{bmatrix} 1 & x_{1,1} & x_{1,1}^2 \\ 1 & x_{1,2} & x_{1,2}^2 \\ \vdots & \vdots & \vdots \\ 1 & x_{1,n} & x_{1,n}^2 \end{bmatrix} \end{matrix} \\ P_{\text{raw}} = 1 & & P = 2 \end{matrix}$$

The transformed matrix X is still full rank since a polynomial function cannot be expressed with finite linear terms.

$$\vec{b} = (X^T X)^{-1} X^T \vec{y} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}$$

$$g(x) = \hat{y} = b_0 + b_1 x + b_2 x^2 = b_0 + (b_1 + b_2 x) x$$



Can you do polynomial of degree $d=3$? Yes, same way! Just make a new feature and cube x . How far can you go in OLS

$p = n - 1$ is $d = n - 1$. That would yield a perfect fit. Any higher d , and you can't invert $X^T X$ (eg. $n = 5$)

$$X = \begin{bmatrix} 1 & x_{11} & x_{11}^2 & x_{11}^3 & x_{11}^4 \\ 1 & x_{12} & & & \\ 1 & x_{13} & & & \\ 1 & x_{14} & & & \\ 1 & x_{15} & & & \end{bmatrix}$$

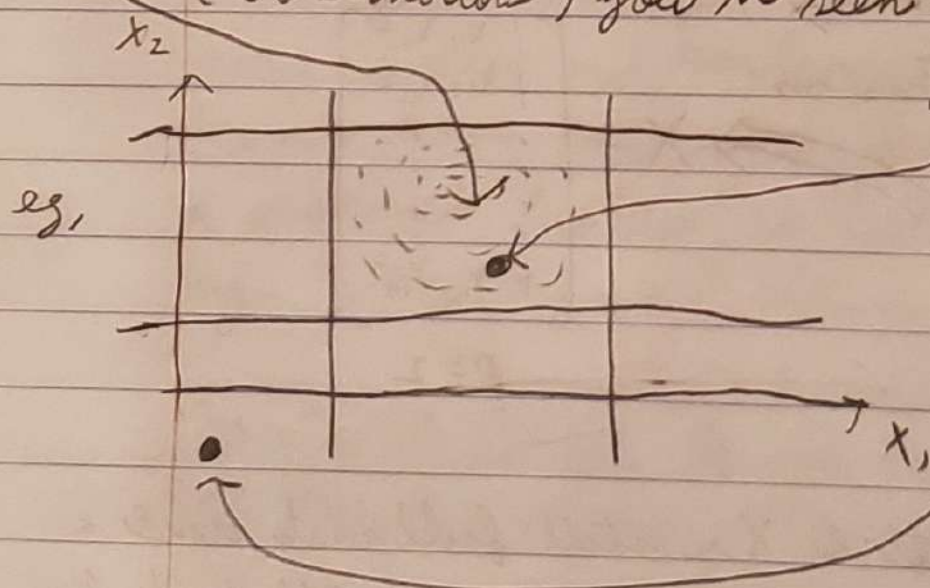
Is this full rank? This is a special matrix called a Vandermonde matrix and it's proven to be full rank if!

$$\det[X] = \prod_{i=1}^n \prod_{j=1}^n x_j - x_i \neq 0$$

Consider row features given by the columns of X . Define:

$$\text{Range}[X] = [x_{\cdot 1, \min}, x_{\cdot 1, \max}] \times [x_{\cdot 2, \min}, x_{\cdot 2, \max}] \times \dots \times [x_{\cdot p, \min}, x_{\cdot p, \max}]$$

This is a hyperrectangle representing the space of X -vectors (observation) you've seen in your n examples.



"Interpolation" is when you predict for X -vector inside the $\text{Range}[X]$.

"Extrapolation" is when you predict for X -vectors outside the $\text{Range}[X]$.

We build models for interpolation only. Bad things could happen when you extrapolate. Different model fitting procedures (A) extrapolate differently... beware

We expanded the complexity of our candidate set by using polynomials, but we found that high degree polynomials had unintended consequences (Runge's phenomenon), so there are other transformations of raw features that we can employ to expand it of course... there are tons of functions: Exponentials, logs, sines etc. Let's examine logs because they are very popular and very useful:

$$\ln(x+1) \approx x - \frac{x^2}{2} + \frac{x^3}{3} - \dots, \approx x \text{ if } x \approx 0$$

$$\Rightarrow \ln(x) = \ln((x+1) - 1) \approx x - 1 \text{ eg } \ln(1.02) = .019 \approx 1.02 - 1$$

Consider the following linear model:

$$y = b_0 + b_1 \ln(x) \text{ eg}$$

$$\Delta x = x_f - x_0 = 1.07 - 1.00$$

$$\Delta y = (b_0 + b_1 \ln(x_f)) - (b_0 + b_1 \ln(x_0)) = b_1 \ln\left(\frac{x_f}{x_0}\right) \approx b_1 \left(\frac{x_f}{x_0} - 1\right)$$

% Change in x
↓
 $\frac{x_f}{x_0} - 1$

This simple log model can be approx interpreted as proportional change in x yields a change in y (in y 's units) ie if x increases by 100%, y goes up by b_1 .

Likewise you can do $\ln(y) = b_0 + b_1 x$ and this is approx interpreted as unit change in x yields b_1 proportion change in y and $\ln(y) = b_0 + b_1 \ln(x)$ is approx interpreted as proportional change in x yields b_1 proportion change in y .