* Regression :

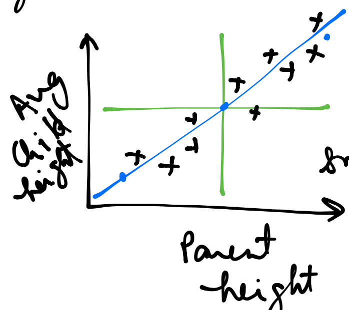
* Useful Pointers:

* historical facts
* model selection and under/over fitting
* crossvalidation
* linear, Polynomial regression
* best constant in terms of squared error: mean
* Representation for regression

PTO NEXT PAGE

* Regression : is continuous outputs , it is a function approximation

ex:

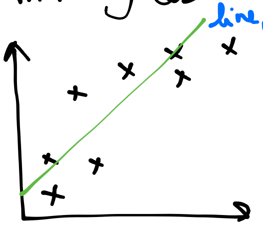



$m = \frac{2}{3} < 1$

Since m is less than 1, it makes it a regression to the mean

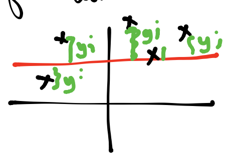
Regression: mathematical relationship to the mean or functional approximation to the point.

we may in many cases of regression be interpolating answers.




* not the best fit high bias.

* Finding the best constant function:




$f(x) = c$ mean

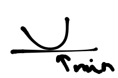
$$E(c) = \sum_{i=1} (y_i - c)^2$$

we can use calculus here

Loss Error

$$\frac{dE(c)}{dc} = \sum_{i=1}^{n} 2(y_i - c)\cdot(-1)$$ Chain rule

find min by setting it to 0




$$-1 \sum_{i=1}^{n} 2(y_i - c) = 0$$

$$\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} c = 0$$ ← its a constant

mean!

$$\Rightarrow n \cdot c = \sum_{i=1}^{n} y_i$$

i=1 i=1

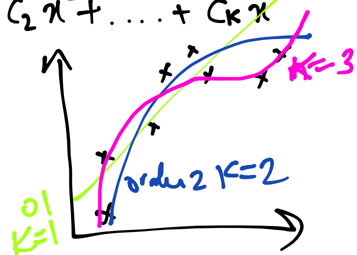
mean!

$$\Rightarrow n \cdot c = \sum_{i=1}^{n} y_i$$

$$c = \sum_{i=1}^{n'=1} y_i / n$$

## * Order of Polynomial:

$$f(x) = C_0 + C_1 x + C_2 x^2 + \dots + C_K x^K$$ ← degrees of freedom K




Best parabola to fit these curves

To minimize sum of squared error

* As we have more degrees of freedom we can fit the line better
* Degrees of freedom, K
* K=3, parabola is the best option for degrees of freedom and it does not overcon commit to the data, a good balance of bias and variance and capturing all the gener generalized trends in the data.

| X | Y |
|---|---|
| $x_1$ | $y_1$ |
| $x_2$ | $y_2$ |
| $x_3$ | $y_3$ |
| ⋮ | ⋮ |
| $x_n$ | $y_n$ |

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}}_{w} \approx \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}}_{y}$$

$$x\,w \approx y$$

X

$$x\,w \approx y$$

w = coefficient

$$xw = y$$

$x^T x w \approx X^T y$

$(X^TX)^{-1} X^TXw = (X^TX)^{-1} X^{X\text{-transpose}} y$

$w = (X^TX)^{-1} X^T y$

$X^T = X$ transpose

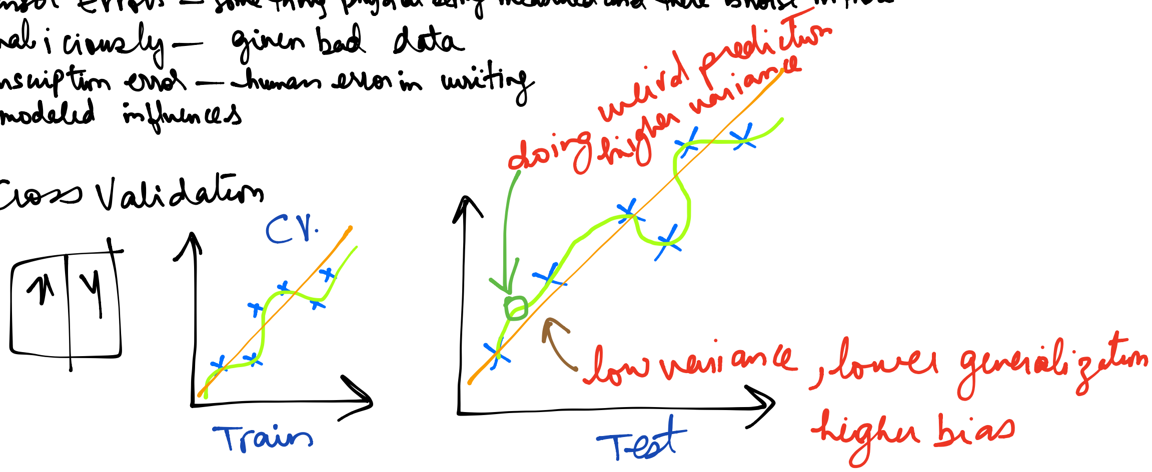
x Errors:

Training data has errors
not modeling $f$ but $f + \varepsilon$

* where do errors come from?
* Common errors in ML:

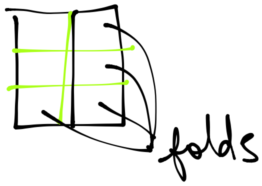
- ☐ Sensor Errors – something physical being measured and there is noise in there
- ☐ maliciously – given bad data
- ☐ transcription error – human error in writing
- ☐ unmodeled influences

* Cross Validation

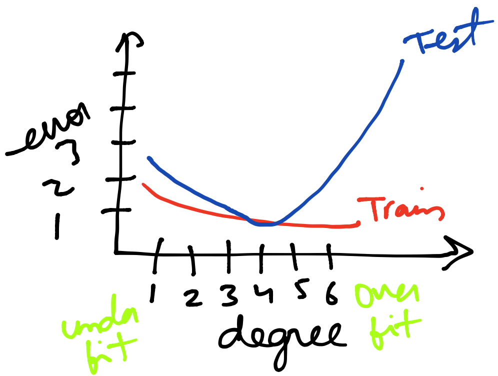



* the goal is to generalize the results
* The test is a sample of how real data will be used system
* we really rely on all the data test and training comming from the same source. It is a fundamental assumption for alot of algorithms including this one.

* In C.V we split training data and use hold some of it for a trial test set
* so we can actually learn the true structure which is being generalized
* we take our training data and split it into folds




* hold back folds for training and 1 for test/learner/trial.
* repeat that for all folds
* And take average




other inputs spaces:

→ Scalar inputs, continuous $x$

→ vector input, continuous $\vec{x}$
include more input features

* example house how far from zoo : size, distance from zoo

hyper planes