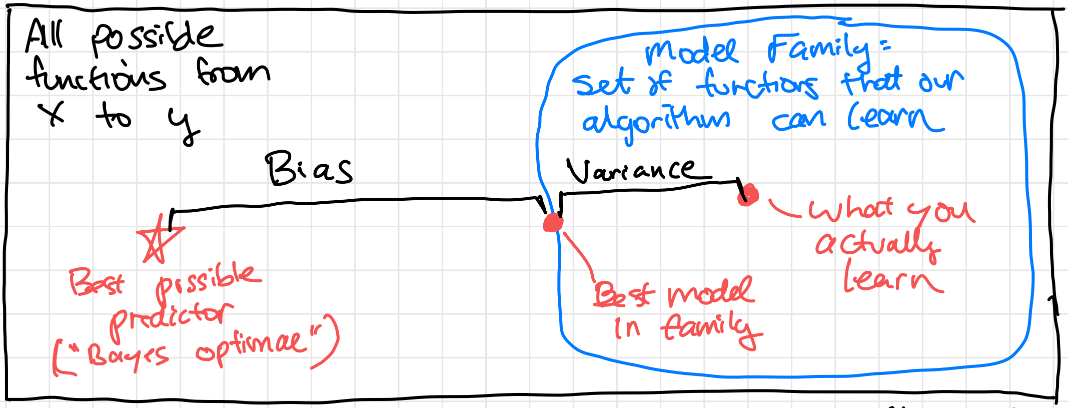
1/25/2024: Bias / variance, MAP, Normal Equations



Bias: Error because assumptions of ML method don't exactly match real world

Variance: Error because what you learn is not best possible model in model family
↳ Because training data is always incomplete, never covers all possible cases

Bias + Variance = total error of model

Reduce Bias

- Make fewer assumptions ⟺ Make model family bigger

Reduce Variance

Make it easier to find best model in family

1. Make model family smaller (Regularization does this)
2. Add training data

What is the probabilistic story behind regularization?

Idea: Think about learning as usage of Bayes Rule.

Bayesian probabilistic story:

1. Exists a prior distribution over w called $p(w)$
2. w gets sampled from $p(w)$
3. Dataset gets generated conditioned on w from distribution $p(D|w)$

Learning goal: Infer most likely value of w

ie maximize $p(w|D)$

unknown must be learned/inferred

observed

called MAP maximum a posteriori

By Bayes Rule: $p(w|D) = \dfrac{\boxed{p(w)}\ \boxed{p(D|w)}}{\boxed{p(D)}}$

New! Different choices of $p(w)$ give different types of regularization

Does not depend on w, can be ignored

likelihood of the data i.e. what we maximize for MLE

For example: Let's assume each w_j is Gaussian centered at 0

in particular: $p(w_j) = \dfrac{1}{\sigma\sqrt{2\pi}} e^{-w_j^2/2\boxed{\sigma^2}}$

Assume mean is 0

constant variance

Overall: $p(w)$ is just $\prod_{j=1}^{d} p(w_j)$

$$\max_w p(w \mid D) = \max_w p(w)\, p(D \mid w)$$

$$= \max_w \log p(w) + \log p(D \mid w)$$

$$= \max_w \sum_{j=1}^{d} \log p(w_j) + \log p(D \mid w)$$

$$= \max_w \cancel{\text{constant}} + \boxed{\sum_{j=1}^{d} -\frac{w_j^2}{2\sigma^2}} + \log p(D \mid w)$$

$$= \max_w -\frac{1}{2\sigma^2} \|w\|^2 + \log p(D \mid w)$$

$$= \min_w \underbrace{\boxed{\frac{1}{2\sigma^2} \|w\|^2}}_{\substack{L_2 \text{ regularization} \\ \text{with strength} \\ \lambda = \frac{1}{2\sigma^2}}} \underbrace{\boxed{-\log p(D \mid w)}}_{\text{MLE objective}}$$

If σ^2 small, λ large

If σ^2 large, λ small

Closed form Solution for Linear Regression

("Normal Equations")

$$L(w) = \frac{1}{n} \sum_{i=1}^{n} (w^T x^{(i)} - y^{(i)})^2$$

Solve for w where $\frac{\partial}{\partial w_j} = 0$ for all j

$$\nabla_w L(w) = \cancel{\frac{1}{n}} \sum_{i=1}^{n} \cancel{2} (w^T x^{(i)} - y^{(i)})\, x^{(i)} = 0$$

$$X^T X w \leftarrow \underbrace{\sum_{i=1}^{n} (w^T x^{(i)})\, x^{(i)}}_{} = \underbrace{\sum_{i=1}^{n} y^{(i)} x^{(i)}}_{} \rightarrow = X^T y$$

<u>define</u>

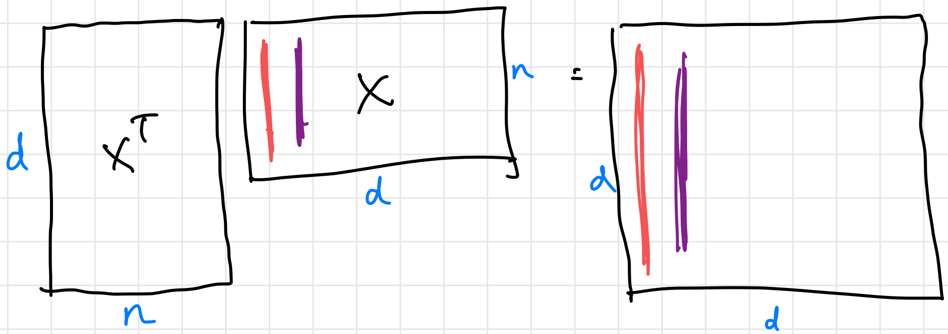
$$X = \underbrace{\begin{bmatrix} — x^{(1)} — \\ \vdots \\ — x^{(n)} — \end{bmatrix}}_{n \times d \text{ matrix}} \qquad y = \underbrace{\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}}_{n\text{-dim vector}}$$

$$\underbrace{\underset{d\times n}{X^T}\,\underset{n\times d}{X}}_{= d\times d}\ \underset{\text{d-dim vector}}{w} = \underset{d\times n}{X^T}\,\underset{\text{n-dim}}{y}$$

Solution: $\boxed{w = (X^T X)^{-1} X^T y}$ Closed-form Solution!

Question: What if $X^T X$ is not invertible?

Scenario 1: $\underset{\#\text{ train examples}}{n} < \underset{\#\text{ features}}{d}$ (we have too few examples compared to # of features)



Each column of $X^T X$ is result of $X^T \cdot$ some vector, which is a linear combo of columns of X^T

But X^T only has n columns, so all of $X^T X$'s colums lie in n-dimensional subspace

i.e. $\text{rank}(X^T X) \leq n < d$

So $X^T X$ is not invertible

Implication: $X^TXw = X^Ty$ has many solutions

there's many w's that perfectly fit training data

we don't know which one is actually best!

i.e. we have high variance

Rule of thumb: Want to have $n > d$

more training examples than features

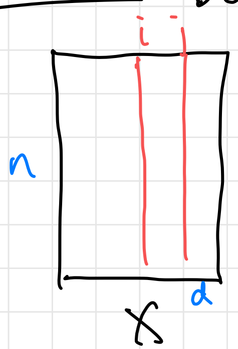
In practice: we can use pseudoinverse of X^TX

The pseudoinverse of a matrix A, denoted A^+:

- $A^+ = A^{-1}$ if A is invertible
- For any equation $Ax = b$,

  $x = A^+ b$ is a solution

Scenario 2: Duplicated features



Suppose features i & j are identical

then, X^TX is not invertible!

Intuitively: w is under-determined

$$w = [w_1, \ldots, w_i, \ldots, w_j, \ldots, w_d]$$

	w_i	w_j
	+100	−100
	−500	+500

all w's with identical performance

Another case of high variance!

Rule of thumb: Avoid (near-) duplicate features