Bias and variance: classical vs. modern elements

- Classical theory
  - Regularization
  - Parameter selection
- Comulative effect: Successive Haley
- Modern theory (Bons)

True function

\[ f_0(x) = \theta_2 x^2 + \theta_1 x + \theta_0 \]

We don't get to see \( f_0 \) directly — Only samples

What happens if we fit a line to these samples?
"UNDERFIT"

We informally call this **underfit** as the error is pretty high (we use this for det.)

What happens if we use degree 5 polynomial?

"OVERFIT"

This fits each sample well, but the function totally changes per sample. (High variance)

What if we use quadratics?

→ low error & low variance.

→ it fits "optimal complexity"
Underfit: Train risk matches test risk.
Overfit: We pick up on spurious correlations.

**NB:** this is **Classical Bias Variance**.
- helpful to understand many ML ideas.
- incomplete for modern models in important ways (more later)

**More Formal Bias-Variance**

Consider linear regression

\[
\text{Output} \quad y = \theta \cdot x + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma^2)
\]

**Procedure**

Fix \( x \in \mathbb{R}^d \), a test point (reason about learn here)

1. Draw \( n \) points \((x^{(1)}, y^{(1)}) \ldots (x^{(n)}, y^{(n)})\)
2. **Train a linear regressor** $h_g : \mathbb{R}^d \to \mathbb{R}

3. **Draw test sample** $(x, y)$ on which

   
   $h(x) + \epsilon = y$

   $\epsilon \sim \mathcal{N}(0, \sigma^2)$ noise

4. **Measure** $(h_s(x) - y)^2$

---

**We examine**

$$E[(h_s(x) - y)^2]$$

**$\epsilon$'s**

$\rightarrow$ two sources of randomness

**Goal:** decompose this error

$$E[(h_s(x) - (h_0(x) + \epsilon_0)^2)] =$$

$$E[\epsilon^2] + E[(h_{s\epsilon} - h_{s\epsilon})^2] + 2E[\epsilon(h_s(x) - h_0(x))]

\text{Depends on } \epsilon_0 \text{ and } \epsilon_1 \text{ s.d. independent of } x$$

**Avoidable error term**

$$h_{avg}(x) = \frac{E}{S} h_s(x)$$

"long run average of many $x"
\[ E_S \left[ \left( \theta(x) - h_{ang}(x) + h_{ang}(x) - h_5(x) \right)^2 \right] \]

\[ = E_S \left[ (\theta_0(x) - h_{ang}(x))^2 \right] + E \left[ (h_{ang} - h_5(x))^2 \right] + \delta \]

\[ = (\theta_0(x) - h_{ang}(x))^2 \quad \text{does not depend on } S \quad \text{VARIANCE OVER} \]

\text{CLASS OF HYPERPLIES} \quad \text{TRAINING SET}

\[ \rightarrow \text{BIAS} \quad \text{VARIANCE} \]

Recoap:
\[ E \left[ (y - h_5(x))^2 \right] = \sigma^2 + \text{BIAS} + \text{VARIANCE} \]

\text{Examples}

<table>
<thead>
<tr>
<th>Linear</th>
<th>Degree 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bias</td>
<td>large</td>
</tr>
<tr>
<td>Variance</td>
<td>lower</td>
</tr>
</tbody>
</table>

Bias: test right fit combines both.

Variance: test every point!

\text{Note:}
- Having different DEV / HOLDOUT set allows us to assess VARIANCE (and learn stability).
- If we use model class that is expressive may need to "trust points less" (reduce VARIANCE)
  \[ \rightarrow \text{Regularization} \]

Regularization is at heart of both classical...
and modern theory. Spend a little bit of time on this...

**Regularization**

Reduce variance to get robust model

$\Rightarrow$ can be explicit (change model)

$\Rightarrow$ implicit (procedure)

**Most Classical Linear Regression**

$$\text{argmin}_{\Theta \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^{n} (x_i^T \Theta - y_i)^2 + \frac{\lambda}{2} \| \Theta \|_2^2$$

$\Rightarrow$ penalized for really complex model (minimum norm solution)

$\lambda = 0 \Rightarrow$ ordinary least squares

$\lambda = 10^{-6} \Rightarrow \Theta = 0$ probably looks pretty good!

Set $\lambda$ to some value to balance loss & variance.

How will show you!

**Solution:** Fix $\lambda > 0$.

Take derivative w.r.t to $\Theta$:

$$x^T x \Theta - x^T y + \lambda \Theta = 0$$

$$(x^T x + \lambda I) \Theta = x^T y$$

**Undetermined case**
If $X^TX$ is not full rank, $\lambda = 0$ may not have a unique solution $f(x) \in \mathbb{R}^{m \times d}$, $n < d$.

If $X^TX$ is not full rank, $\exists V$ s.t.
$V \neq 0 \implies X^TXV = 0$

$8. \quad X^TX\theta = X^Ty \implies \text{then } X^T(x + \eta) = X^Ty \text{ as well} \implies \text{no unique solution}$

$\Rightarrow$ if $\lambda > 0$ then it does have a unique solution, since $X^TX + \lambda \mathbf{I}$ is full rank.

That is, eigenvalues of $X^TX$: $\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2 > 0$

Then $X^TX + \lambda \mathbf{I}$ has eigenvalues
$\sigma_1^2 + \lambda, \sigma_2^2 + \lambda, \ldots, \sigma_d^2 + \lambda > 0$

In this case, $\theta_x = (X^TX + \lambda \mathbf{I})^{-1}X^Ty$.

Back to variance

$$\mathbb{E}_S \left[ (f_S(x) - \mathbb{E}_S(f_S(x)))^2 \right]$$

Variance in $\theta_x$, the solution for fixed $\lambda$

$$\text{Var}_S = \mathbb{E}_S \left[ (\theta_x - \mathbb{E}_S(\theta_x))^2 \right]$$

$$\approx \mathbb{E} \left[ \|\theta_x - \mathbb{E}_S(\theta_x)\|^2 \right] \quad \text{(6)}$$
TO SIMPLIFY ANALYSIS,
AND ONLY RANDOMNESS IN DATA IS THE TOTAL
POINTS ERROR

\[ y = X\theta + \nu \quad \nu \in \mathcal{N}(0, \sigma^2 I_n) \]

Random Noise Per Point \( \nu \) fixed \( X \)

\[ \infty = \mathbb{E} \left[ \| (X^T X + I)^{-1} X^T \nu \|^2 \right] \]

then \( Av \sim \mathcal{N}(0, c^2 AA^T) \)

Hence, \[ c^2 \leq \frac{\sigma_{\max}}{(\sigma_{\min}, X)^2} \]

so as \( \lambda \) INCREASES, \( \nu \) DECREASES

BONUS OBSERVATION CAN SOMETIMES IMPPLICITLY
REGULARIZE AS WELL \( \left( \text{Surprisingly Important!} \right) \)

(\text{In modern theory})

THOUGHT EXPERIMENT, WE CAN GRADIENT DESCENT WITH \( \lambda = 0 \)
IN UNDERDETERMINED CASE

CLAIM IF WE INITIALIZE TO \( \theta_0 \), THEN OUR SOLUTION IS \( \theta_0 \).
\[
\Theta_0 = P_{null}(\Theta_0) + P_{span(x)}(\Theta_0) = \Theta_0
\]

Why? \[ \Theta^{(t+1)} = \Theta^{(t)} - \alpha \cdot x^T (x^T \Theta - y) \]
only changes in span(x).

Observation: We can regularize by initializing!
set \( \Theta^{(0)} = 0 \) has good properties.

\[ \Rightarrow \text{deep learning is underdetermined (often)} \]
and so initialization plays major role!

In fact, SGD plays a starring role in
mocked theory of bias variance

Belkin et. al 2018 “Double Descent”
First observed (widely) for deep nets, but also true for kernels.

SGD regularizes by picking min norm solutions.

Memorization and generalization!

Other methods of bias & variance:

+ Data Augmentation (see Sironi blog on AI)
+ Drop out "DATA ADAPTIVE"
+ Optimization Algorithms (Proximal Point methods)

Lots more to read.
Picking Hyper Parameters

Three sets of labeled data
- Train: Fit parameters
- Dev: "Fit" hyperparameters, e.g. k
- Test (blind)

Our first example

For degree d ∈ {0, 1, ..., k}

\[ \text{Train model}(d) \text{ on train set} \]

\[ \text{Score } \theta_d \text{ on dev set} \]

Pick best score, hope for best on test

If we have infinitely many models we can grid search, e.g.:

For each k ∈ {0, 10^{-4}, 10^{-3}, 10^{-2}, ...}

→ Same process
Why do we score on Dev, not Train?

**Improvements**

→ **Data Efficiency**: Make best use of data in Train/Dev
   "Classical Stats"

→ **Compute Efficiency**: Many related hyper parameters → many models
   "Modern ML situation"
   Combinatorial explosion.

**Data K-fold cross validation**

\[ k = 3 \text{ but } 5, 10, \ldots \text{ typical} \]

\[ S^{(1)} \quad S^{(2)} \quad S^{(3)} \quad \text{MAKE K=3 SETS} \]
2. \[ \frac{\text{TRAIN}}{S^{(1)}, S^{(2)}} \times \frac{\text{SCORE}}{S^{(3)}} \left\{ \frac{s^{(3)}}{s^{(2)}} \right\} \] 
3. Combine scores (average)

\[
\rightarrow \text{use this to pick best.}
\]

**Computational**

**Motivation:** Regularizer, dropout rate, stepsizes, dimensions — many layers!

**Practical trick:**
1. Tune 1 parameter at a time
2. Sweep over all parameters

\[
2(5 + 6 + 7) < 5 \cdot 6 \cdot 7
\]

**More Advanced Hyperband (random 15)**

Run all 5 \cdot 6 \cdot 7 models

\[
\rightarrow \text{but for just a few steps}
\]
Pick top half, run for $2k$ steps

$\Rightarrow$ Repeat ...

Each round, we use same number of resources but with fewer models.
Run $\log_2(5 \cdot 6 \cdot 7)$ total rounds.

Lots more to do here (clear across runs?)

---

**Recap:**

- **Bias + Variance → Classical**
- Regularization: Explicit and Implicit
- Tuning: Cross Validation & Hyperparameter Search