Three Learning Principles

- Occam’s Razor
- Sampling Bias
- Data Snooping
Occam’s Razor

• “It is futile to do with more things that which can be done with fewer"
Occam’s Razor

- "Nature operates in the shortest way possible"

- “An explanation of the data should be made as simple as possible, but no simpler”

- "When you hear hoofbeats, think of horses not zebras”

- The simplest model that fits the data is also the most plausible
Simple model

Complex model

Number of training points, $n$

$E_{\text{out}}$

$E_{\text{in}}$

Expected error
Simple Models

- Simple hypotheses, $h$
  - Low order polynomials
  - Linear models with small weights
  - Easily described in words (or bits)
- Or...
  - Simply hypothesis sets, $\mathcal{H}$
    - Low VC-dimension
    - Small number of hypotheses
    - Small number of free parameters
Simple Models

- A hypothesis set of simple hypotheses must be small.
- Suppose a hypothesis, $h$, can be described using $l$ bits.
- Then the hypothesis set $\mathcal{H}$ that contains all such hypotheses is of size $2^l$. 
Case Study #1

• Suppose I tell you that I’ve found a $10^{th}$-order polynomial that perfectly fits my dataset of 10 points.

• Should you believe that the true function is a $10^{th}$-order polynomial?
Case Study #1

- Suppose I tell you that I’ve found a line that perfectly fits my dataset of 10 points.

- Should you believe that the true function is a line?
Axiom of Non-falsifiability

- If an experiment has no chance of falsifying a hypothesis, then the result of that experiment provides no evidence one way or the other for the hypothesis.
Counterpoint: Hickam’s Dictum

• “A man can have as many diseases as he damn well pleases”
Sampling Bias

- If the data is sampled in a biased way, learning will produce a similarly biased outcome
Case Study #2

- A late election poll by the Chicago Daily Tribune had Dewey so far ahead of Truman that their they ran the following headline the day after election day:

A late election poll by the Chicago Daily Tribune had Dewey so far ahead of Truman that their they ran the following headline the day after election day:

\[ P\{|E_{\text{in}}(h) - E_{\text{out}}(h)| > \epsilon\} \leq 2e^{-2\epsilon^2 n} \]
Case Study #3

Source: [https://me.me/i/newsweek-madam-president-llan-clintons-ejourney-wiitelluune-dewey-defeats-truman-8726228](https://me.me/i/newsweek-madam-president-llan-clintons-ejourney-wiitelluune-dewey-defeats-truman-8726228)
Case Study #3

Case Study #3

Source: https://projects.fivethirtyeight.com/2016-election-forecast/
Case Study #3

- “the biggest difference between 1948 and 2016 may be our polling methods... the way in which participants were selected”

- “don’t expect 'Clinton Defeats Trump’ to become one of the great blunders of our political history"

Formal Setup

- **Unknown target function**
  \[ f: \mathcal{X} \rightarrow \mathcal{Y} \]

- **Hypothesis Set**
  \[ \mathcal{H} \]

- **Training data**
  \[ \mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \]

- **Probability Distribution**
  \[ \mathcal{P} \text{ on } \mathcal{X} \]

- **Learning Algorithm**
  \[ \mathcal{A} \]

- **Learned Hypothesis**
  \[ g: \mathcal{X} \rightarrow \mathcal{Y} \]
Training-Test Mismatch

- Suppose your training data comes from some distribution $P$ and you know that the true distribution over all inputs is some other distribution $P'$

- You can reweight or resample your training data to make it look like it came from the distribution $P'$
  - Requires that you know $P'$
  - Requires complete representation of the true distribution in the training distribution
Data Snooping

- If a data set has affected any step in the learning process, its ability to assess the outcome has been compromised

- Ask yourself: “If the data were different, could/would I have done something different?”
Given some dataset $\mathcal{D} = \{ (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \}$ where $\mathbf{x} = \mathbb{R}^d$ and $\mathbf{y} = \mathbb{R}$, you first normalize the outputs (can be helpful for some learning algorithms)

$\mathbf{y}'_i = \frac{y_i - \bar{y}}{\sqrt{V(y)}}$ where $\bar{y}$ is the mean of $\{ y_1, y_2, \ldots, y_n \}$ and $V(y)$ is the variance of $\{ y_1, y_2, \ldots, y_n \}$

Partition the dataset $\mathcal{D}' = \{ (x_1', y_1'), (x_2', y_2'), \ldots, (x_n', y_n') \}$ into test and training data $\mathcal{D}'_{test}$ and $\mathcal{D}'_{train}$

$\mathcal{D}'_{test}$ affected the training process via the normalization!
Case Study #4

- Given some dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ where $x = \mathbb{R}^d$ and $y = \mathbb{R}$, you first partition the dataset into test and training data $\mathcal{D}_{test}$ and $\mathcal{D}_{train}$.

- Then normalize the two datasets independently.

- $\mathcal{D}_{test}$ no longer affects $\mathcal{D}_{train}$ and therefore, no longer affects the training process.
Case Study #5

- Given some data you fit a linear model and find that the in-sample error is high.
- So you fit a quadratic model; turns out that makes the in-sample error even worse.
- You do some reading and find that others have tried SVMs on the same data with no success.
- Time to get serious: break out the neural networks!
• Trying different models on the same data set will eventually lead to “success”

• Account for data reuse by computing the combined VC dimension of all models (including what others tried)
<table>
<thead>
<tr>
<th>First Half of the Course: Foundations</th>
<th>Second Half of the Course: Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory, Proofs, Math, Probability, Boring Stuff, etc...</td>
<td>Random Forests! Support Vector Machines! Neural Networks! Yay!</td>
</tr>
</tbody>
</table>

Overview