Learning From Data
Lecture 4
Real Learning is Feasible

Real Learning vs. Verification
The Two Step Solution to Learning
Closer to Reality: Error and Noise

M. Magdon-Ismail
CSCI 4100/6100
RECAP: Verification

Hoeffding: \( E_{\text{out}}(g) \approx E_{\text{in}}(g) \) (with high probability)

\[
P[|E_{\text{in}}(h) - E_{\text{out}}(h)| > \epsilon] \leq 2e^{-2N\epsilon^2}.
\]

\( E_{\text{in}}(h) = \frac{2}{9} \)
Coin tossing example:

- If we toss one coin and get no **HEADS**, its very surprising.
  
  We expect it is biased: \( P[\text{heads}] \approx 0 \).

- Tossing 70 coins, and **find one** with no heads. Is it surprising?
  
  Do we expect \( P[\text{heads}] \approx 0 \) for the selected coin?
  
  Similar to the “birthday problem”: among 30 people, two will likely share the same birthday.

- This is called **selection bias**.
  
  Selection bias is a very serious trap. For example medical screening.

\[
P = \frac{1}{2^N}
\]

\[
P = 1 - \left(1 - \frac{1}{2^N}\right)^{70}
\]
Real Learning – Finite Learning Models

$E_{\text{out}}(h_1)$

$E_{\text{out}}(h_2)$

$E_{\text{out}}(h_3)$

$E_{\text{out}}(h_M)$

$E_{\text{in}}(h_1) = \frac{2}{9}$

$E_{\text{in}}(h_2) = 0$

$E_{\text{in}}(h_3) = \frac{5}{9}$

$E_{\text{in}}(h_M) = \frac{6}{9}$

Pick the hypothesis with minimum $E_{\text{in}}$; will $E_{\text{out}}$ be small?
Hoeffding says that $E_{\text{in}}(g) \approx E_{\text{out}}(g)$ for Finite $\mathcal{H}$

\[
\mathbb{P} \left[ |E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon \right] \leq 2|\mathcal{H}|e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.
\]
\[
\mathbb{P} \left[ |E_{\text{in}}(g) - E_{\text{out}}(g)| \leq \epsilon \right] \geq 1 - 2|\mathcal{H}|e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.
\]

We don’t care how $g$ was obtained, as long as it is from $\mathcal{H}$

**Proof:** Let $M = |\mathcal{H}|$.

The event “$|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon$” implies “$|E_{\text{in}}(h_1) - E_{\text{out}}(h_1)| > \epsilon$” OR . . . OR “$|E_{\text{in}}(h_M) - E_{\text{out}}(h_M)| > \epsilon$”

So, by the implication and union bounds:

\[
\mathbb{P} \left[ |E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon \right] \leq \mathbb{P} \left[ \bigvee_{m=1}^{M} |E_{\text{in}}(h_m) - E_{\text{out}}(h_m)| > \epsilon \right]
\]
\[
\leq \sum_{m=1}^{M} \mathbb{P} \left[ |E_{\text{in}}(h_m) - E_{\text{out}}(h_m)| > \epsilon \right],
\]
\[
\leq 2Me^{-2\epsilon^2 N}.
\]

(The last inequality is because we can apply the Hoeffding bound to each summand)
Interpreting the Hoeffding Bound for Finite $|\mathcal{H}|$

\[
\begin{align*}
\mathbb{P}\left[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon\right] &\leq 2|\mathcal{H}|e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0. \\
\mathbb{P}\left[|E_{\text{in}}(g) - E_{\text{out}}(g)| \leq \epsilon\right] &\geq 1 - 2|\mathcal{H}|e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.
\end{align*}
\]

**Theorem.** With probability at least $1 - \delta$,

\[
E_{\text{out}}(g) \leq E_{\text{in}}(g) + \sqrt{\frac{1}{2N} \log \frac{2|\mathcal{H}|}{\delta}}.
\]

We don’t care how $g$ was obtained, as long as $g \in \mathcal{H}$

**Proof:** Let $\delta = 2|\mathcal{H}|e^{-2\epsilon^2 N}$. Then

\[
\mathbb{P}\left[|E_{\text{in}}(g) - E_{\text{out}}(g)| \leq \epsilon\right] \geq 1 - \delta.
\]

In words, with probability at least $1 - \delta$,

\[
|E_{\text{in}}(g) - E_{\text{out}}(g)| \leq \epsilon.
\]

This implies

\[
E_{\text{out}}(g) \leq E_{\text{in}}(g) + \epsilon.
\]

From the definition of $\delta$, solve for $\epsilon$:

\[
\epsilon = \sqrt{\frac{1}{2N} \log \frac{2|\mathcal{H}|}{\delta}}.
\]
$E_{in}$ Reaches Outside to $E_{out}$ when $|\mathcal{H}|$ is Small

\[
E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{1}{2N} \log \frac{2|\mathcal{H}|}{\delta}}.
\]

If $N \gg \ln |\mathcal{H}|$, then $E_{out}(g) \approx E_{in}(g)$.

- Does not depend on $\mathcal{X}$, $P(x)$, $f$ or how $g$ is found.
- Only requires $P(x)$ to generate the data points independently \textit{and also} the test point.

What about $E_{out} \approx 0$?
The 2 Step Approach to Getting $E_{out} \approx 0$:

1. $E_{out}(g) \approx E_{in}(g)$.
2. $E_{in}(g) \approx 0$.

Together, these ensure $E_{out} \approx 0$.

How to verify (1) since we do not know $E_{out}$
- must ensure it theoretically - Hoeffding.

We can ensure (2) (for example PLA)
- modulo that we can guarantee (1)

There is a tradeoff:
- Small $|\mathcal{H}| \implies E_{in} \approx E_{out}$
- Large $|\mathcal{H}| \implies E_{in} \approx 0$ is more likely.

\[ \text{Error} = \sqrt{\frac{1}{2N} \log \frac{2|\mathcal{H}|}{\delta}} \]
Feasibility of Learning (Finite Models)

• No Free Lunch: can’t know anything outside $\mathcal{D}$, for sure.

• Can “learn” with high probability if $\mathcal{D}$ is i.i.d. from $P(x)$.
  $$E_{\text{out}} \approx E_{\text{in}} \quad (E_{\text{in}} \text{ can reach outside the data set to } E_{\text{out}}).$$

• We want $E_{\text{out}} \approx 0$.

• The two step solution. We trade $E_{\text{out}} \approx 0$ for 2 goals:
  (i) $E_{\text{out}} \approx E_{\text{in}}$;
  (ii) $E_{\text{in}} \approx 0$.

  We know $E_{\text{in}}$, not $E_{\text{out}}$, but we can ensure (i) if $|\mathcal{H}|$ is small.

  This is a big step!

• What about infinite $\mathcal{H}$ - the perceptron?
“Complex” Target Functions are Harder to Learn

What happened to the “difficulty” (complexity) of $f$?

- Simple $f \implies$ can use small $\mathcal{H}$ to get $E_{\text{in}} \approx 0$ (need smaller $N$).
- Complex $f \implies$ need large $\mathcal{H}$ to get $E_{\text{in}} \approx 0$ (need larger $N$).
Revising the Learning Problem – Adding in Probability

**UNKNOWN TARGET FUNCTION**

\( f : \mathcal{X} \mapsto \mathcal{Y} \)

**TRAINING EXAMPLES**

\((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\)

**LEARNING ALGORITHM**

\( \mathcal{A} \)

**HYPOTHESIS SET**

\( \mathcal{H} \)

**FINAL HYPOTHESIS**

\( g \)

**UNKNOWN INPUT DISTRIBUTION**

\( P(x) \)

\( y_n = f(x_n) \)

\( g(x) \approx f(x) \)
Error and Noise

Error Measure: How to quantify that $h \approx f$.

Noise: $y_n \neq f(x_n)$. 
Finger Print Recognition

Two types of error.

<table>
<thead>
<tr>
<th>$f$</th>
<th>$+1$</th>
<th>$-1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>$+1$</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Supermarket

<table>
<thead>
<tr>
<th>$h$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+1$</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

CIA

<table>
<thead>
<tr>
<th>$h$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+1$</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>100</td>
</tr>
</tbody>
</table>

In any application you need to think about how to penalize each type of error.

<table>
<thead>
<tr>
<th>$f$</th>
<th>$+1$</th>
<th>$-1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>$+1$</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Take Away

Error measure is specified by the user.

If not, choose one that is
- plausible (conceptually appealing)
- friendly (practically appealing)
Almost All Error Measures are Pointwise

Compare \( h \) and \( f \) on individual points \( \mathbf{x} \) using a pointwise error \( e(h(\mathbf{x}), f(\mathbf{x})) \):

- **Binary error:** \[ e(h(\mathbf{x}), f(\mathbf{x})) = \mathbb{I}[h(\mathbf{x}) \neq f(\mathbf{x})] \] (classification)
- **Squared error:** \[ e(h(\mathbf{x}), f(\mathbf{x})) = (h(\mathbf{x}) - f(\mathbf{x}))^2 \] (regression)

In-sample error:

\[ E_{\text{in}}(h) = \frac{1}{N} \sum_{n=1}^{N} e(h(\mathbf{x}_n), f(\mathbf{x}_n)). \]

Out-of-sample error:

\[ E_{\text{out}}(h) = \mathbb{E}_{\mathbf{x}}[e(h(\mathbf{x}), f(\mathbf{x}))]. \]
Noisy Targets

Consider two customers with the same credit data. They can have different behaviors.

The target ‘function’ is not a deterministic function but a stochastic function.

\[ f(x) = P(y|x) \]
Learning Setup with Error Measure and Noisy Targets

**UNKNOWN TARGET DISTRIBUTION**
(target function $f$ plus noise)

$P(y \mid x)$

$y_n \sim P(y \mid x_n)$

**TRAINING EXAMPLES**

$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$

**ERROR MEASURE**

$g(x) \approx f(x)$

**UNKNOWN INPUT DISTRIBUTION**

$P(x)$

**FINAL HYPOTHESIS**

$g$

**LEARNING ALGORITHM**

$A$

**HYPOTHESIS SET**

$\mathcal{H}$

Real Learning is Feasible: 16/16