A simple setting…

- **Classification**
  - \( N \) data points : i.i.d.
  - **Finite** number of possible hypothesis (e.g., dec. trees of depth \( d \))
- **A learner finds a hypothesis** \( h \) that is **consistent** with training data
  - Gets zero error in training – \( \text{error}_{\text{train}}(h) = 0 \)
- **What is the probability that** \( h \) **has more than** \( \varepsilon \) **true error?**
  - \( \text{error}_{\text{true}}(h) \geq \varepsilon \) for some \( \varepsilon > 0 \)
How likely is a bad hypothesis to get $N$ data points right?

- Hypothesis $h$ that is **consistent** with training data → got $N$ i.i.d. points right
  - $h$ “bad” if it gets all this data right, but has high true error
- Prob. $h$ with error $\text{true}(h) \geq \varepsilon$ gets one data point right
  - less than $1 - \varepsilon$ if $\text{error} \varepsilon = 0.25$
  - $75\%$ points are consistent $\geq 1 - \varepsilon$

- Prob. $h$ with error $\text{true}(h) \geq \varepsilon$ gets $N$ data points right
  - $\text{Prob. wins decreases exponentially in } N$

But there are many possible hypothesis that are consistent with training data

- Set of hypothesis consistent with training data, don’t want to make assumptions about which one was picked
How likely is learner to pick a bad hypothesis

- Prob. $h$ with $\text{error}_{\text{true}}(h) \geq \varepsilon$ gets $N$ data points right
  \[ \text{less than } (1 - \varepsilon)^N \]
- There are $k$ hypotheses consistent with data
  - How likely is learner to pick a bad one?

\[
P(\exists h \text{ consistent with data, } \text{error}_{\text{true}}(h) \geq \varepsilon) \equiv \text{deal with worst case}
= P(\text{error}_{\text{true}}(h_1) \geq \varepsilon \lor \text{error}_{\text{true}}(h_2) \geq \varepsilon \lor \ldots \lor \text{error}_{\text{true}}(h_k) \geq \varepsilon)
\]

Union bound

- $P(A \lor B \lor C \lor D \ldots) \leq P(A) + P(B) + P(C) + P(D) - \ldots$
How likely is learner to pick a bad hypothesis

- Prob. a particular $h$ with $\text{error}_{\text{true}}(h) \geq \epsilon$ gets $N$ data points right
  \[ \log \left( \frac{1}{1-\epsilon} \right)^N \]

- There are $k$ hypothesis consistent with data
  - How likely is it that learner will pick a bad one out of these $k$ choices?

Generalization error in finite hypothesis spaces [Haussler ’88]

- **Theorem**: Hypothesis space $H$ finite, dataset $D$ with $N$ i.i.d. samples, $0 < \epsilon < 1$; for any learned hypothesis $h$ that is consistent on the training data:
  \[ P(\text{error}_{\text{true}}(h) > \epsilon) \leq |H| e^{-N\epsilon} \]
  \[ \leq |H| (1-\epsilon)^N \leq |H| e^{-\epsilon \cdot N} \]
  \[ \Rightarrow \quad 0 \leq \epsilon \leq 1 \]
  \[ 1-\epsilon \leq e^{-\epsilon} \]
Using a PAC bound

Typically, 2 use cases:

1: Pick \( \varepsilon \) and \( \delta \), give you \( N \)

2: Pick \( N \) and \( \delta \), give you \( \varepsilon \)

Summary: Generalization error in finite hypothesis spaces [Haussler '88]

**Theorem**: Hypothesis space \( H \) finite, dataset \( D \) with \( N \) i.i.d. samples, \( 0 < \varepsilon < 1 \) : for any learned hypothesis \( h \) that is consistent on the training data:

\[
P(error_{true}(h) > \varepsilon) \leq |H|e^{-N\varepsilon}
\]

Even if \( h \) makes zero errors in training data, may make errors in test
Limitations of Haussler ‘88 bound

\[ P(\text{error}_{true}(h) > \varepsilon) \leq |H| e^{-N\varepsilon} \]

- Consistent classifier

\[ \text{error}_{\text{train}}(h) = 0 \rightarrow \text{highly unrealistic} \]

- Size of hypothesis space

\[ |h| \text{ is bad} \rightarrow |H| \text{ very very large} \]

\[ |H| \text{ infinite (e.g. SVM)} \]

What if our classifier does not have zero error on the training data?

- A learner with zero training errors may make mistakes in test set

- What about a learner with \( \text{error}_{\text{train}}(h) \) in training set?

\[ \text{What happens when } \text{error}_{\text{train}}(h) > 0 \text{?} \]

\[ \vdots \]

\[ \text{error}_{\text{train}}(h)? \]
Simpler question: What’s the expected error of a hypothesis?

- The error of a hypothesis is like estimating the parameter of a coin!

\[ \theta \sim \hat{\theta} = \frac{3}{5} \]

- Chernoff bound: for \( N \) i.i.d. coin flips, \( x^1, \ldots, x^N \), where \( x^i \in \{0, 1\} \). For \( 0 < \epsilon < 1 \):

\[
P \left( \theta - \frac{1}{N} \sum_{j=1}^{N} x^j > \epsilon \right) \leq e^{-2N\epsilon^2}
\]

Using Chernoff bound to estimate error of a single hypothesis

\[
P \left( \theta - \frac{1}{N} \sum_{j=1}^{N} x^j > \epsilon \right) \leq e^{-2N\epsilon^2}
\]

\[ \theta = \int p(x) \left( \frac{1}{N} \sum_{i=1}^{N} x^i \right) dx \]

\[ \hat{\theta} = \left\{ \frac{1}{N} \sum_{i=1}^{N} h(x^i) \right\} \]

\[
P \left( \text{error}_{\text{true}}(h) - \text{error}_{\text{train}}(h) > \epsilon \right) \leq e^{-2N\epsilon^2}
\]
But we are comparing many hypothesis: **Union bound**

For each hypothesis $h_i$:

$$P(error_{true}(h_i) - error_{train}(h_i) > \epsilon) \leq e^{-2N\epsilon^2}$$

What if I am comparing two hypothesis, $h_1$ and $h_2$?

$$P(error_{true}(h_1) < error_{true}(h_2) \text{, but } error_{train}(h_1) > error_{train}(h_2))$$

But want:

$$P(error_{true}(h_1) - error_{train}(h_1) \geq \epsilon) \text{ or } P(error_{true}(h_2) - error_{train}(h_2) \geq \epsilon)$$

$$\leq 2e^{-2N\epsilon^2}$$

**Generalization bound for $|H|$ hypothesis**

**Theorem:** Hypothesis space $H$ finite, dataset $D$ with $N$ i.i.d. samples, $0 < \epsilon < 1$: for any learned hypothesis $h$:

$$P(error_{true}(h_i) - error_{train}(h_i) > \epsilon) \leq e^{-2N\epsilon^2} \leq \delta$$

$$\forall h_i$$

$$P(error_{true}(h) - error_{train}(h) > \epsilon) \leq |H|e^{-2N\epsilon^2}$$

$$\epsilon > \sqrt{\frac{\ln |H| + \ln \frac{1}{\delta}}{2N}} \rightarrow o\left(\frac{1}{N}\right) \text{ rule}$$

with probability at least 1 - $\delta$:

$$|error_{true}(h) - error_{train}(h)| \leq \epsilon$$
PAC bound and Bias-Variance tradeoff

\[ P(\text{error}_{true}(h) - \text{error}_{train}(h) > \epsilon) \leq e^{-2N\epsilon^2} \]

or, after moving some terms around, with probability at least 1-\(\delta\):

\[ \text{error}_{true}(h) \leq \text{error}_{train}(h) + \sqrt{\frac{\ln |H| + \ln \frac{1}{\delta}}{2N}} \]

"Complex hypothesis space" | low |
| large \(\Rightarrow |H|\) is large |

"Simple \(\Rightarrow |H|\)" | low |
| low \(\Rightarrow |H|\) is small |

Important: PAC bound holds for all \(h\), but doesn’t guarantee that algorithm finds best \(h\)!!!
Boolean formulas with $m$ binary features

\[ N \geq \frac{\ln |H| + \ln \frac{1}{\delta}}{2e^2} \]

$H$ = any boolean formula: $|H| \geq 2$

- $H_1 = \text{all conjunction with root input}$
- $H_2 = \text{positive}$
- $H = 3^m \Rightarrow \ln |H| = m \ln 3$

Number of decision trees of depth $k$

Recursive solution

Given $m$ attributes

- $H_k = \text{Number of decision trees of depth } k$
- $H_0 = 2$
- $H_{k+1} = (\# \text{choices of root attribute}) \times$
  - (\# possible left subtrees) \times
  - (\# possible right subtrees)

\[ H_{k+1} \leq m \times H_k \times H_k \]

Write $L_k = \log_2 H_k$

- $L_0 = 1$
- $L_{k+1} = \log_2 m + 2L_k$
- So $L_k = (2^k - 1)(1 + \log_2 m) + 1$
PAC bound for decision trees of depth $k$

$$N \geq \frac{2^k \log m + \ln \frac{1}{\delta}}{\epsilon^2}$$

- Bad!!!
  - Number of points is exponential in depth!

- But, for $N$ data points, decision tree can’t get too big…
  - No reason to have more than $N$ leaves

Number of leaves never more than number data points

Number of Decision Trees with $k$ Leaves

- Number of decision trees of depth $k$ is really really big:
  - $\ln |H|$ is about $2^k \log m$

- Decision trees with up to $k$ leaves:
  - $|H|$ is about $m^k k^{2k}$ … only really big
    - A very loose bound
      $$\ln |H| \leq k \ln m + 2k \ln k$$
      much better!
PAC bound for decision trees with $k$ leaves – Bias-Variance revisited

$$\ln |H_{DTs \, k \, leaves}| \leq 2k(\ln m + \ln k)$$

$$error_{true}(h) \leq error_{train}(h) + \sqrt{\frac{2k(\ln m + \ln k) + \ln \frac{1}{\delta}}{2N}}$$

<table>
<thead>
<tr>
<th>max number of leaves $k$</th>
<th>&quot;bias&quot;</th>
<th>&quot;variance&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K &lt; N$</td>
<td>goes to 0</td>
<td></td>
</tr>
<tr>
<td>$K &lt;&lt; N$</td>
<td>potentially large</td>
<td></td>
</tr>
<tr>
<td></td>
<td>probably small</td>
<td></td>
</tr>
</tbody>
</table>

What did we learn from decision trees?

- Bias-Variance tradeoff formalized

$$error_{true}(h) \leq error_{train}(h) + \sqrt{\frac{2k(\ln m + \ln k) + \ln \frac{1}{\delta}}{2N}}$$

- Moral of the story:
Complexity of learning not measured in terms of size hypothesis space, but in maximum number of points that allows consistent classification
- $K < N$ – no bias, lots of variance
- Lower than $N$ – some bias, less variance
- $K = N$ – optimal learning
What about continuous hypothesis spaces?

\[
error_{\text{true}}(h) \leq error_{\text{train}}(h) + \sqrt{\frac{\ln |H| + \ln \frac{1}{\delta}}{2N}}
\]

- Continuous hypothesis space:
  - $|H| = \infty$
  - Infinite variance???

- As with decision trees, only care about the maximum number of points that can be classified exactly!
  - Called VC dimension… see readings for details

What you need to know

- Finite hypothesis space
  - Derive results
  - Counting number of hypothesis
  - Mistakes on Training data

- Complexity of the classifier depends on number of points that can be classified exactly
  - Finite case – decision trees
  - Infinite case – VC dimension

- Bias-Variance tradeoff in learning theory
- Remember: will your algorithm find best classifier?
Markov Decision Processes (MDPs)

Machine Learning – CSE546
Carlos Guestrin
University of Washington
December 2, 2013

©Carlos Guestrin 2005-2013

Reinforcement Learning

training by feedback
Learning to act

- Reinforcement learning
- An agent
  - Makes sensor observations
  - Must select action
  - Receives rewards
    - positive for "good" states
    - negative for "bad" states

[Ng et al. '05]

Markov Decision Process (MDP) Representation

- State space:
  - Joint state $x$ of entire system
- Action space:
  - Joint action $a = \{a_1, ..., a_n\}$ for all agents
- Reward function:
  - Total reward $R(x,a)$
    - sometimes reward can depend on action
- Transition model:
  - Dynamics of the entire system $P(x'|x,a)$
Discount Factors

People in economics and probabilistic decision-making do this all the time.
The “Discounted sum of future rewards” using discount factor $\gamma$ is

$$(\text{reward now}) + \gamma(\text{reward in 1 time step}) + \gamma^2(\text{reward in 2 time steps}) + \gamma^3(\text{reward in 3 time steps}) + \cdots$$

(\text{infinite sum})

---

Define:

$V_A = \text{Expected discounted future rewards starting in state A}$

$V_B = \text{Expected discounted future rewards starting in state B}$

$V_T = \ldots$ T

$V_S = \ldots$ S

$V_D = \ldots$ D

How do we compute $V_A, V_B, V_T, V_S, V_D$?
Policy

At state $x$, action $a$ for all agents

$\pi(x_0) = \text{both peasants get wood}$

$\pi(x_1) = \text{one peasant builds barrack, other gets gold}$

$\pi(x_2) = \text{peasants get gold, footmen attack}$

Value of Policy

Expected long-term reward starting from $x$

$V_{\pi}(x_0) = E_{\pi}[R(x_0) + \gamma R(x_1) + \gamma^2 R(x_2) + \gamma^3 R(x_3) + \gamma^4 R(x_4) + \ldots]$