Computational Learning Theory

- Computational learning theory
- Setting 1: learner poses queries to teacher
- Setting 2: teacher chooses examples
- Setting 3: randomly generated instances, labeled by teacher
- Probably approximately correct (PAC) learning
- Vapnik-Chervonenkis (VC) Dimension
- Mistake bounds
Computational Learning Theory: Issues

- What general laws constrain inductive learning?

- We seek theory to relate:
  - probability of successful learning,
  - number of training examples,
  - complexity of hypothesis space,
  - accuracy to which target concept is approximated,
  - manner in which training examples presented.
Prototypical Concept Learning Task

Given:

- Instances $X$: Possible days, each described by the attributes $Sky$, $AirTemp$, $Humidity$, $Wind$, $Water$, $Forecast$.
- Target function $c$: $EnjoySport : X \rightarrow \{0, 1\}$
- Hypotheses $H$: Conjunctions of literals. E.g.,
  \[ \langle ?, Cold, High, ?, ?, ? \rangle. \]
- Training examples $D$: Positive and negative examples of the target function
  \[ \langle x_1, c(x_1) \rangle, \ldots \langle x_m, c(x_m) \rangle \]

Determine:

- A hypothesis $h$ in $H$ such that $h(x) = c(x)$ for all $x$ in $D$?
- A hypothesis $h$ in $H$ such that $h(x) = c(x)$ for all $x$ in $X$?
Sample Complexity

- How many training examples are sufficient to learn the target concept?
  - If learner proposes instances as queries to teacher:
    - Learner proposes instance $x$, teacher provides $c(x)$.
  - If teacher (who knows $c$) provides training examples:
    - Teacher provides sequence of examples of form $\langle x, c(x) \rangle$.
  - If some random process (e.g., nature) proposes instances:
    - Instance $x$ generated randomly, teacher provides $c(x)$.
Sample Complexity: 1

- Assume $c$ is in learner’s hypothesis space $H$.

- Learner proposes instance $x$, teacher provides $c(x)$.
  - Optimal query strategy:
    - Pick instance $x$ such that half of the hypotheses in $VS$ classify $x$ positive, half classify $x$ negative.
    - How many queries are needed to learn $c$?
Sample Complexity: 1

- Assume $c$ is in learner’s hypothesis space $H$.

- Learner proposes instance $x$, teacher provides $c(x)$.

  - Optimal query strategy:
    - Pick instance $x$ such that half of the hypotheses in $V S$ classify $x$ positive, half classify $x$ negative.
    - How many queries are needed to learn $c$?
      - $\lceil \log_2 |H| \rceil$
Assume $c$ is in learner’s hypothesis space $H$.

Teacher (who knows $c$) provides training examples

- Optimal teaching strategy: depends on $H$ used by learner.
- Consider the case where $H$ are conjunctions of up to $n$ boolean atoms and their negation.
  - E.g., $(\text{AirTemp} = \text{Warm}) \land (\text{Wind} = \text{Strong})$, where $\text{AirTemp}, \text{Wind}, \ldots$ each have 2 possible values.
- How many examples are needed to learn $c$?
Sample Complexity: 2

- Assume $c$ is in learner’s hypothesis space $H$.
- Teacher (who knows $c$) provides training examples
  - Optimal teaching strategy: depends on $H$ used by learner.
  - Consider the case where $H$ are conjunctions of up to $n$ boolean atoms and their negation.
  - E.g., $(AirTemp = Warm) \land (Wind = Strong)$, where $AirTemp, Wind, \ldots$ each have 2 possible values.
  - How many examples are needed to learn $c$?
    - $n + 1$. 
Sample Complexity: 3

Given:

- set of instances $X$,
- set of hypotheses $H$,
- set of possible target concepts $C$,
- training instances generated by a fixed, unknown probability distribution $\mathcal{D}$ over $X$.

Learner observes a sequence $D$ of training examples of form $\langle x, c(x) \rangle$, for some target concept $c \in C$:

- instances $x$ are drawn from distribution $\mathcal{D}$
- teacher provides target value $c(x)$ for each

Learner must output a hypothesis $h$ estimating $c$.

$h$ is evaluated by its performance on subsequent instances drawn according to $\mathcal{D}$.

Note: randomly drawn instances, noise-free classifications.
True Error of a Hypothesis

Definition: The true error (denoted $error_D(h)$) of hypothesis $h$ with respect to target concept $c$ and distribution $D$ is the probability that $h$ will misclassify an instance drawn at random according to $D$.

$$error_D(h) \equiv \Pr_{x \in D}[c(x) \neq h(x)].$$
Two Notions of Error

- **Training error** of hypothesis $h$ with respect to target concept $c$:
  - How often $h(x) \neq c(x)$ over training instances.

- **True error** of hypothesis $h$ with respect to $c$:
  - How often $h(x) \neq c(x)$ over future random instances.

- True error is not directly observable by the learner.

- Our concern:
  - Can we bound the true error of $h$ given the training error of $h$?

- First consider the case when the training error of $h$ is zero (i.e., $h \in VS_{H,D}$).
Exhausting the Version Space

Hypothesis space $H$

\[ \forall h \in \text{VS}_{H,D} \] \[ \text{error}_D(h) < \epsilon \]

Definition: The version space $\text{VS}_{H,D}$ is said to be $\epsilon$-exhausted with respect to $c$ and $\mathcal{D}$, if every hypothesis $h$ in $\text{VS}_{H,D}$ has error less than $\epsilon$ with respect to $c$ and $\mathcal{D}$. 

\[ (r = \text{training error}, \text{error} = \text{true error}, \epsilon = 0.25) \]
How many examples will $\epsilon$-exhaust the VS?

- **Theorem [Haussler, 1988]:** If the hypothesis space $H$ is finite, and $D$ is a sequence of $m \geq 1$ independent random examples of some target concept $c$, then for any $0 \leq \epsilon \leq 1$, the probability that the version space with respect to $H$ and $D$ is not $\epsilon$-exhausted (with respect to $c$) is less than
  \[ |H|e^{-\epsilon m} \]

- This bounds the probability that any consistent learner will output a hypothesis $h$ with $error(h) \geq \epsilon$!

- If we want this probability to be below some $\delta$
  \[ |H|e^{-\epsilon m} \leq \delta \]

  then
  \[ m \geq \frac{1}{\epsilon}(\ln |H| + \ln(1/\delta)). \]

- Any consistent hypothesis will be probably (with probability $1 - \delta$) approximately (within error $\epsilon$) correct.
Learning Conjunctions of Boolean Literals

How many examples are sufficient to assure with probability at least \((1 - \delta)\) that every \(h \in V S_{H,D}\) satisfies \(error_D(h) \leq \epsilon\)?

Use our theorem:

\[
m \geq \frac{1}{\epsilon} (\ln |H| + \ln(1/\delta))
\]

Suppose \(H\) contains conjunctions of constraints on up to \(n\) boolean attributes (i.e., \(n\) boolean literals).

Then \(|H| = 3^n\), and

\[
m \geq \frac{1}{\epsilon} (\ln 3^n + \ln(1/\delta)) = \frac{1}{\epsilon} (n \ln 3 + \ln(1/\delta)).
\]
How About *EnjoySport*?

- \( m \geq \frac{1}{\epsilon} (\ln |H| + \ln(1/\delta)) \).
- If \( H \) is as given in *EnjoySport* then \(|H| = 973\), and

\[
m \geq \frac{1}{\epsilon} (\ln 973 + \ln(1/\delta))
\]

- If we want to assure that with probability 95% the version space contains only hypotheses with \( \text{error}_D(h) \leq .1 \), then it is sufficient to have \( m \) examples, where

\[
m \geq \frac{1}{.1} (\ln 973 + \ln(1/.05)) \\
= 10(\ln 973 + \ln 20) \\
= 10(6.88 + 3.00) \\
= 98.8.
\]
PAC Learning

Consider a class $C$ of possible target concepts defined over a set of instances $X$ of length $n$, and a learner $L$ using hypothesis space $H$.

Definition: $C$ is PAC-learnable by $L$ using $H$ if for all $c \in C$, distributions $\mathcal{D}$ over $X$, $\epsilon$ such that $0 < \epsilon < 1/2$, and $\delta$ such that $0 < \delta < 1/2$, learner $L$ will output a hypothesis $h \in H$ with probability at least $(1 - \delta)$ such that $\text{error}_\mathcal{D}(h) \leq \epsilon$, in time that is polynomial in $1/\epsilon$, $1/\delta$, $n$ and $\text{size}(c)$. 
Consider a class $C$ of possible target concepts defined over a set of instances $X$ of length $n$, and a learner $L$ using hypothesis space $H$.

Definition: $C$ is PAC-learnable by $L$ using $H$ if for all $c \in C$, distributions $D$ over $X$, $\epsilon$ such that $0 < \epsilon < 1/2$, and $\delta$ such that $0 < \delta < 1/2$, learner $L$ will output a hypothesis $h \in H$ with probability at least $(1 - \delta)$ such that $\text{error}_D(h) \leq \epsilon$, in time that is polynomial in $1/\epsilon$, $1/\delta$, $n$ and $\text{size}(c)$.

Theorem: The class $C$ of conjunctions of boolean literals is PAC-learnable by the FIND-S algorithm using $H = C$. 

PAC Learning

Consider a class $C$ of possible target concepts defined over a set of instances $X$ of length $n$, and a learner $L$ using hypothesis space $H$.

**Definition:** $C$ is PAC-learnable by $L$ using $H$ if for all $c \in C$, distributions $D$ over $X$, $\epsilon$ such that $0 < \epsilon < 1/2$, and $\delta$ such that $0 < \delta < 1/2$, learner $L$ will output a hypothesis $h \in H$ with probability at least $(1 - \delta)$ such that $\text{error}_D(h) \leq \epsilon$, in time that is polynomial in $1/\epsilon$, $1/\delta$, $n$ and $\text{size}(c)$.

**Theorem:** The class $C$ of conjunctions of boolean literals is PAC-learnable by the FIND-S algorithm using $H = C$.

**Problems:**
- What if $c \notin H$?
- What if $H$ is infinite?
- Bounds are weak.
Agnostic Learning

- So far we have assumed that \( c \in H \).
- Agnostic learning setting: don’t assume \( c \in H \).
- What do we want then?
  - The hypothesis \( h \) that makes fewest errors on training data.
- What is sample complexity in this case?

\[
m \geq \frac{1}{2\epsilon^2} (\ln |H| + \ln(1/\delta))
\]

derived from Hoeffding bounds:

\[
Pr[\text{error}_D(h) > \text{error}_D(h) + \epsilon] \leq e^{-2m\epsilon^2}.
\]
Shattering a Set of Instances

- **Definition:** A dichotomy of a set $S$ is a partition of $S$ into two disjoint subsets.
- **Definition:** A set of instances $S$ is shattered by hypothesis space $H$ if and only if for every dichotomy of $S$ there exists some hypothesis in $H$ consistent with this dichotomy.
- Three instances shattered:

  ![Diagram of shattered instances](image)

- If a set of instances is not shattered, then some concept cannot be represented by the hypothesis space.
- The ability to shatter a set of instances is related to the inductive bias.
Definition: The Vapnik-Chervonenkis dimension, $ VC(H) $, of hypothesis space $ H $ defined over instance space $ X $ is the size of the largest finite subset of $ X $ shattered by $ H $. If arbitrarily large finite sets of $ X $ can be shattered by $ H $, then $ VC(H) \equiv \infty $.

What is the VC dimension if $ H $ is finite?
Definition: The Vapnik-Chervonenkis dimension, $VC(H)$, of hypothesis space $H$ defined over instance space $X$ is the size of the largest finite subset of $X$ shattered by $H$. If arbitrarily large finite sets of $X$ can be shattered by $H$, then $VC(H) \equiv \infty$.

What is the VC dimension if $H$ is finite?

$VC(H) \leq \log_2 |H|$. 

The Vapnik-Chervonenkis Dimension
Definition: The Vapnik-Chervonenkis dimension, $VC(H)$, of hypothesis space $H$ defined over instance space $X$ is the size of the largest finite subset of $X$ shattered by $H$. If arbitrarily large finite sets of $X$ can be shattered by $H$, then $VC(H) \equiv \infty$.

- What is the VC dimension if $H$ is finite?
  
  $VC(H) \leq \log_2 |H|$.

- What is the VC dimension of linear decision surfaces?
The Vapnik-Chervonenkis Dimension

Definition: The Vapnik-Chervonenkis dimension, $V C(H)$, of hypothesis space $H$ defined over instance space $X$ is the size of the largest finite subset of $X$ shattered by $H$. If arbitrarily large finite sets of $X$ can be shattered by $H$, then $V C(H) \equiv \infty$.

What is the VC dimension if $H$ is finite?

$V C(H) \leq \log_2 |H|$.

What is the VC dimension of linear decision surfaces?

$V C(H) = 3$. 

Diagram:

(a) and (b) show examples of shattered sets.
Sample Complexity from VC Dimension

How many randomly drawn examples suffice to $\epsilon$-exhaust $V S_{H,D}$ with probability at least $(1 - \delta)$?

$$m \geq \frac{1}{\epsilon} (4 \log_2 (2/\delta) + 8 VC(H) \log_2 (13/\epsilon))$$
Mistake Bounds

► So far: how many examples are needed to learn?
► What about: how many mistakes before convergence?
► Let’s consider similar setting to PAC learning:
  ► Instances drawn at random from $X$ according to distribution $\mathcal{D}$.
  ► Learner must classify each instance before receiving correct classification from teacher.
  ► Can we bound the number of mistakes learner makes before converging?
Mistake Bounds: Find-S

- Consider Find-S when $H$ is conjunctions of boolean literals.
- **Find-S:**
  - **Initialize** $h$ to the most specific hypothesis $l_1 \land \neg l_1 \land l_2 \land \neg l_2 \ldots l_n \land \neg l_n$
  - **For each positive training instance $x$ do:**
    - Remove from $h$ any literal that is not satisfied by $x$.
  - **Output hypothesis $h$.**
- **How many mistakes before converging to correct $h$?**
Mistake Bounds: Find-S

- Consider Find-S when $H$ is conjunctions of boolean literals.

**FIND-S:**

- Initialize $h$ to the most specific hypothesis $l_1 \land \neg l_1 \land l_2 \land \neg l_2 \ldots l_n \land \neg l_n$
- For each positive training instance $x$ do:
  - Remove from $h$ any literal that is not satisfied by $x$.
- Output hypothesis $h$.

- How many mistakes before converging to correct $h$?
  - At most $n + 1$. 
Consider the Halving Algorithm:

- Learn concept using version space CANDIDATE-ELIMINATION algorithm.
- Classify new instances by majority vote of version space members.

How many mistakes before converging to correct $h$?

... in worst case?
Consider the Halving Algorithm:

▷ Learn concept using version space CANDIDATE-ELIMINATION algorithm.
▷ Classify new instances by majority vote of version space members.

How many mistakes before converging to correct \( h \)?

▷ ... in worst case?
  • at most \( \lceil \log_2 |H| \rceil \).
Consider the Halving Algorithm:

- Learn concept using version space CANDIDATE-ELIMINATION algorithm.
- Classify new instances by majority vote of version space members.

How many mistakes before converging to correct $h$?

- ... in worst case?
  - at most $\lceil \log_2 |H| \rceil$.
- ... in best case?
Mistake Bounds: Halving Algorithm

Consider the Halving Algorithm:

- Learn concept using version space CANDIDATE-ELIMINATION algorithm.
- Classify new instances by majority vote of version space members.

How many mistakes before converging to correct $h$?

- ... in worst case?
  - at most $\lfloor \log_2 |H| \rfloor$.
- ... in best case?
  - none.
Optimal Mistake Bounds

Let $M_A(C)$ be the maximum number of mistakes made by algorithm $A$ to learn concepts in $C$, where the maximum is taken over all possible $c \in C$, and all possible training sequences:

$$M_A(C) \equiv \max_{c \in C} M_A(c).$$

Definition: Let $C$ be an arbitrary non-empty concept class. The optimal mistake bound for $C$, denoted $Opt(C)$, is the minimum over all possible learning algorithms $A$ of $M_A(C)$.

$$Opt(C) \equiv \min_{A \in \text{learning algorithms}} M_A(C).$$

$VC(C) \leq Opt(C) \leq M_{Halving}(C) \leq \log_2(|C|).$