Computational Learning Theory  
(Part 1)

Mark Craven  
Computer Sciences 760  
Fall 2016  

www.biostat.wisc.edu/~craven/cs760/

Some of the slides in these lectures have been adapted/borrowed from materials developed 
by Tom Dietterich, Pedro Domingos, Tom Mitchell, David Page, and Jude Shavlik

Goals for the lecture

you should understand the following concepts  
• PAC learnability  
• consistent learners and version spaces  
• sample complexity  
• PAC learnability in the agnostic setting  
• the VC dimension  
• sample complexity using the VC dimension
PAC learning

- Overfitting happens because training error is a poor estimate of generalization error
  → Can we infer something about generalization error from training error?

- Overfitting happens when the learner doesn’t see enough training instances
  → Can we estimate how many instances are enough?
Learning setting #1

- set of instances $\mathcal{X}$
- set of hypotheses (models) $H$
- set of possible target concepts $C$
- unknown probability distribution $\mathcal{D}$ over instances

Learning setting #1

- learner is given a set $D$ of training instances $\langle x, c(x) \rangle$ for some target concept $c$ in $C$
  - each instance $x$ is drawn from distribution $\mathcal{D}$
  - class label $c(x)$ is provided for each $x$
- learner outputs hypothesis $h$ modeling $c$
**True error of a hypothesis**

The *true error* of hypothesis $h$ refers to how often $h$ is wrong on future instances drawn from $\mathcal{D}$

$$\text{error}_\mathcal{D}(h) \equiv P_{x \in \mathcal{D}} [c(x) \neq h(x)]$$

**Training error of a hypothesis**

The *training error* of hypothesis $h$ refers to how often $h$ is wrong on instances in the training set $\mathcal{D}$

$$\text{error}_\mathcal{D}(h) \equiv P_{x \in \mathcal{D}} [c(x) \neq h(x)] = \frac{\sum_{x \in \mathcal{D}} \delta(c(x) \neq h(x))}{|\mathcal{D}|}$$

Can we bound $\text{error}_\mathcal{D}(h)$ in terms of $\text{error}_\mathcal{D}(h)$?
Is approximately correct good enough?

To say that our learner $L$ has learned a concept, should we require $\text{error}_D(h) = 0$?

This is not realistic:
- unless we’ve seen every possible instance, there may be multiple hypotheses that are consistent with the training set
- there is some chance our training sample will be unrepresentative

Probably approximately correct learning?

Instead, we’ll require that
- the error of a learned hypothesis $h$ is bounded by some constant $\epsilon$
- the probability of the learner failing to learn an accurate hypothesis is bounded by a constant $\delta$
Probably Approximately Correct (PAC) learning [Valiant, CACM 1984]

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

- $C$ is PAC learnable by $L$ using $H$ if, for all $c \in C$
  - distributions $\mathcal{D}$ over $\mathcal{X}$
  - $\varepsilon$ such that $0 < \varepsilon < 0.5$
  - $\delta$ such that $0 < \delta < 0.5$

- learner $L$ will, with probability at least $(1 - \delta)$, output a hypothesis $h \in H$ such that $\text{error}_\mathcal{D}(h) \leq \varepsilon$ in time that is polynomial in
  - $1/\varepsilon$
  - $1/\delta$
  - $n$
  - $\text{size}(c)$

PAC learning and consistency

- Suppose we can find hypotheses that are consistent with $m$ training instances.
- We can analyze PAC learnability by determining whether
  1. $m$ grows polynomially in the relevant parameters
  2. the processing time per training example is polynomial
Version spaces

• A hypothesis $h$ is consistent with a set of training examples $D$ of target concept if and only if $h(x) = c(x)$ for each training example $\langle x, c(x) \rangle$ in $D$

$$consistent(h, D) \equiv \left( \forall \langle x, c(x) \rangle \in D \right) h(x) = c(x)$$

• The version space $VS_{H,D}$ with respect to hypothesis space $H$ and training set $D$, is the subset of hypotheses from $H$ consistent with all training examples in $D$

$$VS_{H,D} \equiv \{ h \in H \mid consistent(h, D) \}$$

Exhausting the version space

• The version space $VS_{H,D}$ is $\epsilon$-exhausted with respect to $c$ and $D$ if every hypothesis $h \in VS_{H,D}$ has true error $< \epsilon$

$$\left( \forall h \in VS_{H,D} \right) error_D(h) < \epsilon$$
Exhausting the version space

- Suppose that every $h$ in our version space $V S_{H,D}$ is consistent with $m$ training examples
- The probability that $V S_{H,D}$ is not $\epsilon$-exhausted (i.e. that it contains some hypotheses that are not accurate enough)

$$\leq |H|e^{-\epsilon m}$$

Proof:

- $(1 - \epsilon)^m$ probability that some hypothesis with error $\geq \epsilon$
- is consistent with $m$ training instances

$$k(1 - \epsilon)^m$$ there might be $k$ such hypotheses

$$|H|(1 - \epsilon)^m$$ $k$ is bounded by $|H|$,

$$\leq |H|e^{-\epsilon m} \quad (1 - \epsilon) \leq e^{-\epsilon} \text{ when } 0 \leq \epsilon \leq 1$$

Sample complexity for finite hypothesis spaces


- we want to reduce this probability below $\delta$

$$|H|e^{-\epsilon m} \leq \delta$$

- solving for $m$ we get

$$m \geq \frac{1}{\epsilon} \left( \ln |H| + \ln \left( \frac{1}{\delta} \right) \right)$$

log dependence on $H$ $\epsilon$ has stronger influence than $\delta$
PAC analysis example: learning conjunctions of Boolean literals

- each instance has $n$ Boolean features
- learned hypotheses are of the form $Y = X_1 \land X_2 \land \neg X_3$

How many training examples suffice to ensure that with prob $\geq 0.99$, a consistent learner will return a hypothesis with error $\leq 0.05$?

There are $3^n$ hypotheses (each variable can be present and unnegated, present and negated, or absent) in $H$

$$m \geq \frac{1}{.05} \left( \ln(3^n) + \ln\left(\frac{1}{.01}\right) \right)$$

For $n=10$, $m \geq 312$  
For $n=100$, $m \geq 2290$

PAC analysis example: learning conjunctions of Boolean literals

- we’ve shown that the sample complexity is polynomial in relevant parameters: $1/\epsilon$, $1/\delta$, $n$

- to prove that Boolean conjunctions are PAC learnable, need to also show that we can find a consistent hypothesis in polynomial time (the FIND-S algorithm in Mitchell, Chapter 2 does this)

FIND-S:
- initialize $h$ to the most specific hypothesis $x_1 \land \neg x_1 \land x_2 \land \neg x_2 \ldots x_n \land \neg x_n$
- for each positive training instance $x$
  - remove from $h$ any literal that is not satisfied by $x$
- output hypothesis $h$
PAC analysis example: learning decision trees of depth 2

- each instance has \( n \) Boolean features
- learned hypotheses are DTs of depth 2 using only 2 variables

\[
|H| = \binom{n}{2} \times 16 = \frac{n(n-1)}{2} \times 16 = 8n(n-1)
\]

\( \# \) possible split choices  \( \# \) possible leaf labelings

How many training examples suffice to ensure that with prob \( \geq 0.99 \), a consistent learner will return a hypothesis with error \( \leq 0.05 \)?

\[
m \geq \frac{1}{.05} \left( \ln(8n^2 - 8n) + \ln \left( \frac{1}{.01} \right) \right)
\]

for \( n=10 \), \( m \geq 224 \)  
for \( n=100 \), \( m \geq 318 \)
PAC analysis example:  
*K*-term DNF is not PAC learnable

- each instance has *n* Boolean features
- learned hypotheses are of the form \( Y = T_1 \lor T_2 \lor \ldots \lor T_k \) where each \( T_i \) is a conjunction of *n* Boolean features or their negations

\(|H| \leq 3^m\), so sample complexity is polynomial in the relevant parameters

\[ m \geq \frac{1}{\varepsilon} \left( nk \ln(3) + \ln \left( \frac{1}{\delta} \right) \right) \]

however, the computational complexity (time to find consistent \( h \)) is not polynomial in \( m \) (e.g. graph 3-coloring, an NP-complete problem, can be reduced to learning 3-term DNF)

What if the target concept is not in our hypothesis space?

- so far, we’ve been assuming that the target concept \( c \) is in our hypothesis space; this is not a very realistic assumption

- *agnostic learning* setting
  - *don’t assume* \( c \in H \)
  - learner returns hypothesis \( h \) that makes fewest errors on training data
Hoeffding bound

• we can approach the agnostic setting by using the Hoeffding bound
• let $Z_1 \ldots Z_m$ be a sequence of $m$ independent Bernoulli trials (e.g. coin flips), each with probability of success $E[Z_i] = p$
• let $S = Z_1 + \cdots + Z_m$

$$P[S > (p + \varepsilon)m] \leq e^{-2m\varepsilon^2}$$

Agnostic PAC learning

• applying the Hoeffding bound to characterize the error rate of a given hypothesis

$$P[error_D(h) > error_D(h) + \varepsilon] \leq e^{-2m\varepsilon^2}$$

• but our learner searches hypothesis space to find $h_{best}$

$$P[error_D(h_{best}) > error_D(h_{best}) + \varepsilon] \leq |H|e^{-2m\varepsilon^2}$$

• solving for the sample complexity when this probability is limited to $\delta$

$$m \geq \frac{1}{2\varepsilon^2} \left( \ln |H| + \ln \left( \frac{1}{\delta} \right) \right)$$
What if the hypothesis space is not finite?

• **Q:** If $H$ is infinite (e.g. the class of perceptrons), what measure of hypothesis-space complexity can we use in place of $|H|$?

• **A:** the largest subset of $\mathcal{X}$ for which $H$ can guarantee zero training error, regardless of the target function.

  this is known as the Vapnik-Chervonenkis dimension (VC-dimension)

Shattering and the VC dimension

• a set of instances $D$ is *shattered* by a hypothesis space $H$ iff for every dichotomy of $D$ there is a hypothesis in $H$ consistent with this dichotomy

• the *VC dimension* of $H$ is the size of the largest set of instances that is shattered by $H$
An infinite hypothesis space with a finite VC dimension

consider: $H$ is set of lines in 2D (i.e. perceptrons in 2D feature space)

can find an $h$ consistent with 1 instance no matter how it’s labeled

can find an $h$ consistent with 2 instances no matter labeling

An infinite hypothesis space with a finite VC dimension

consider: $H$ is set of lines in 2D

can find an $h$ consistent with 3 instances no matter labeling (assuming they’re not colinear)
cannot find an $h$ consistent with 4 instances for some labelings

can shatter 3 instances, but not 4 → the VC-dim($H$) = 3
more generally, the VC-dim of hyperplanes in $n$ dimensions = $n+1$
VC dimension for finite hypothesis spaces

for finite $H$, $VC\text{-dim}(H) \leq \log_2 |H|$

Proof:

suppose $VC\text{-dim}(H) = d$

for $d$ instances, $2^d$ different labelings possible

due to $H$ must be able to represent $2^d$ hypotheses

$2^d \leq |H|$

$d = VC\text{-dim}(H) \leq \log_2 |H|$

Sample complexity and the VC dimension

- using $VC\text{-dim}(H)$ as a measure of complexity of $H$, we can derive
  the following bound [Blumer et al., JACM 1989]

$$m \geq \frac{1}{\varepsilon} \left( 4 \log_2 \left( \frac{2}{\delta} \right) + 8 VC\text{-dim}(H) \log_2 \left( \frac{13}{\varepsilon} \right) \right)$$

$m$ grows log x linear in $\varepsilon$ (better than earlier bound)

can be used for both finite and infinite hypothesis spaces
**Lower bound on sample complexity**

[Ehrenfeucht et al., *Information & Computation* 1989]

- there exists a distribution $\mathcal{D}$ and target concept in $C$ such that if the number of training instances given to $L$

$$m < \max \left[ \frac{1}{\epsilon} \log \left( \frac{1}{\delta} \right), \frac{\text{VC-dim}(C) - 1}{32\epsilon} \right]$$

then with probability at least $\delta$, $L$ outputs $h$ such that $\text{err}_D(h) > \epsilon$

**Comments on PAC learning**

- PAC analysis formalizes the learning task and allows for non-perfect learning (indicated by $\epsilon$ and $\delta$)
- finding a consistent hypothesis is sometimes easier for larger concept classes
  - e.g. although $k$-term DNF is not PAC learnable, the more general class $k$-CNF is
- PAC analysis has been extended to explore a wide range of cases
  - noisy training data
  - learner allowed to ask queries
  - restricted distributions (e.g. uniform) over $\mathcal{D}$
  - etc.
- most analyses are worst case
- sample complexity bounds are generally not tight