Lecture 11: Computational Learning Theory (COLT)

Cognitive Systems II - Machine Learning

WS 2005/2006

Part II: Special Aspects of Concept Learning

COLT, Probably Approximately Correct (PAC) Learning
Motivation

- which concepts are learnable under which conditions?
- especially: which concepts are *effective* learnable
- providing learning algorithms
Goals

Give a rigorous, computationally detailed and plausible account of how to learning can be done. Translation:

- **Rigorous**: theorems, please.
- **Computationally detailed**: exhibit algorithms that learn.
- **Plausible**: with a feasible quantity of computational resources, and with reasonable information and interaction requirements.

Dana Angluin
PAC Learning Model

- PAC stands for *probably approximately correct*
- instances are generated at random from $X$ according to some probability distribution $\mathcal{D}$
  - generally $\mathcal{D}$ not known to the learner
  - generally $\mathcal{D}$ may be any distribution, *distribution free* learning
  - $\mathcal{D}$ is stationary
- a particular class $C$ of possible target concepts is fixed, $c : X \rightarrow \{0, 1\}$ for each $c \in C$, a hypothesis space $H$ is fixed, basically we assume $C \subseteq H$, a computational representation of $H$ is fixed, then the learnability of $C$ is investigated: *learnability of $C$ in terms of $H$*
PAC Learning Model Cont.

- true (prediction) error: \( \text{error}_D(h) = \Pr_{x \in D}(c(x) \neq h(x)) \)

- training error \( \text{error}_D(h) \): fraction of training examples misclassified by \( h \)

intuition: parameters \( \epsilon \) and \( \delta \) are chosen, then we require that the learner eventually conjectures a hypothesis \( h \in H \) which approximates \( c \) with \( \text{error}_D(h) < \epsilon \), the probability that this does not happen should be smaller than \( \delta \)

definition: a learning algorithm \textit{PAC-identifies} concepts from \( C \) in terms of \( H \) iff for every distribution \( D \) and every concept \( c \in C \), for all positive numbers \( \epsilon \) and \( \delta \) it eventually outputs a concept \( h \in H \) such that with probability at least \( 1 - \delta \), \( \text{error}_D(h) < \epsilon \)
polyomial time: efficiency of the learning algorithm is measured with respect to relevant parameters: length of $X$, size of target concept (note that this is dependent on the chosen computational representation), $1/\varepsilon$, and $1/\delta$

definition: $C$ is PAC-learnable in terms of $H$ provided there exists a polynomial-time learning algorithm that PAC-identifies $C$ in terms of $H$

note that the number of training examples is bound by the polynomial-time requirements: if any training example requires some minimum processing time, then for $C$ to meet the polynomial-time requirements (i.e. being PAC-learnable) the learning algorithm must learn from a polynomial number of training examples
PAC Learning Model and Sample Size

- for hypothesis space $H$, target concept $c$, probability $\mathcal{D}$, and training examples $\mathcal{D}$ the version space $V_{H,D}$ is said to be $\epsilon$-exhausted with respect to $c$ and $\mathcal{D}$, iff for all $h \in V_{H,D}$, $error_D(h) < \epsilon$

- theorem (Haussler 1988): let $m \geq 1$ be the number of training examples of $c$ drawn according to $\mathcal{D}$, if $H$ is finite, then for all $0 \leq \epsilon \leq 1$, the probability that $V_{H,D}$ is not $\epsilon$-exhausted is less than or equal to $|H|e^{-\epsilon m}$

- if we require that this probability of failure is below some $\delta$: $|H|e^{-\epsilon m} \leq \delta$ then rearranging terms to solve for $m$ yields the upper bound for $m$:

$$m \geq \frac{1}{\epsilon} (\ln |H| + \ln(\frac{1}{\delta}))$$
The given bound is a general bound on the number of training examples sufficient for *any consistent learner* to successfully learn any target concept in $H$ for any desired values of $\delta$ and $\epsilon$.

If $C \not\subseteq H$ then a consistent hypothesis cannot always be found. An *agnostic learner* makes no prior commitment about whether or not $C \subseteq H$ and simply outputs the hypothesis with *minimum* training error.

For an agnostic learner the sample size is bound to

$$m \geq \frac{1}{2\epsilon^2} \left( \ln |H| + \ln \left( \frac{1}{\delta} \right) \right)$$

where $\delta$ is the probability that $\text{error}_D(h) > \text{error}_D(h) + \epsilon$. 

Lecture 11: Computational Learning Theory (COLT) – p. 260
conjunctions of boolean literals are PAC-learnable, this can be shown by first showing that any consistent learner will require only a polynomial number of training examples to learn any \( c \in C \) and then suggesting a specific algorithm that uses polynomial time per training example.

- for \( n \) boolean variables, \(|H| = 3^n\), i.e. \( m \geq \frac{1}{\epsilon} (n \ln 3 + \ln(\frac{1}{\delta})) \)
- e.g. to learn concepts of up to 10 boolean literals with 95% accuracy, where \( m = \frac{1}{0.1} (10 \ln 3 + \ln(\frac{1}{0.05})) = 140 \)
- the computational effort depends on the specific learning algorithm, but e.g. the FIND-S algorithm outputs the most specific consistent hypothesis and updates the hypothesis for each training example using time linear in \( n \)
because the sample size for the conjunction of literals-class is polynomial in $n$, $1/\delta$, $1/\epsilon$ and independent of $size(c)$ and FIND-S requires time linear in $n$ and independent of $1/\delta$, $1/\epsilon$, and $size(c)$, this concept class is PAC-learnable (by FIND-S)

$k$-term DNF expressions are not PAC-learnable, they have polynomial sample size, but updating the hypothesis according to one example requires exponential time

surprisingly $k$-term CNF expressions are PAC-learnable, though this class is strictly larger than the class of $k$-term DNF expressions
beside \(|H|\) there exists another measure for the complexity of the hypothesis space, the *Vapnik-Chervonenkis dimension* of \(H\), written \(VC(H)\)

- we can state the sample size in terms of \(VC(H)\)
- that leads to tighter bounds and additionally it applies to infinite hypothesis spaces

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 subsets of \(X\) can be shattered by \(H\), then \(VC(H) = \infty\)

a set of instances \(S\) is *shattered* by hypothesis space \(H\) iff for every partition of \(S\) into two subsets with all positive and respectively all negative labeled instances there exists some hypothesis in \(H\) consistent with this partition