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*
- seminal paper: L. G. Valiant (1984). A theory of the learnable. *Communications of the ACM*, 27(11). 1134–1142
- 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: $error_{\mathcal{D}}(h) = \Pr_{x \in \mathcal{D}}(c(x) \neq h(x))$
- training error $error_{D}(h)$: fraction of training examples misclassified by h
- intuition: parameters ϵ and δ are chosen, then we require that the learner eventually conjectures a hypothesis $h \in H$ which approximates c with $error_{\mathcal{D}}(h) < \epsilon$, the probability that this does not happen should be smaller than δ
- definition: a learning algorithm *PAC-identifies* concepts from C in terms of H iff for every distribution $\mathcal{D}$ and every concept $c \in C$, for all positive numbers ϵ and δ it eventually outputs a concept $h \in H$ such that with probability at least $1 - \delta$, $error_{\mathcal{D}}(h) < \epsilon$

PAC Learning Model Cont.

- polynomial 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), $1/\epsilon$, 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. beeing 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 traning examples D the version space $VS_{H,D}$ is said to be ϵ-*exhausted* with respect to c and $\mathcal{D}$, iff for all $h \in VS_{H,D}$, $error_{\mathcal{D}}(h) < \epsilon$
- 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 $VS_{H,D}$ is not ϵ-exhausted is less than or equal to $|H|e^{-\epsilon m}$
- if we require that this probability of failure is below some δ: $|H|e^{-\epsilon m} \leq \delta$ then rearranging terms to solve for m yields the upper bound for m:

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

PAC Learning Model and Sample Size cont

- The given bound is a general bound on the number of training examples sufficient for *any consistent learner* to succesfully learn any target concept in H for any desired values of δ and ϵ
- 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

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

where δ is the probability that $error_{\mathcal{D}}(h) > error_D(h) + \epsilon$

PAC-learnable Concept Classes

- 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 traing 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 95to present m examples, 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

PAC-learnable Concept Classes Cont.

- because the sample size for the concjunction 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

Vapnik-Chervonenkis Dimension

- 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
- 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

Vapnik-Chervonenkis Dimension Cont.

- 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 $VC(H) = \infty$
- for all finite H, $VC(H) \leq \log_2 |H|$ because there are 2^d hypotheses required for shattering a set of $d = VC(H)$ instances. Hence $2^d \leq |H|$ and with $d = VC(H)$, $VC(H) \leq \log_2 |H|$
- for finite hypothesis spaces we gave an upper bound dependent on |H| for the number of examples which is sufficient to PAC-learn a target concept. for infinite hypothesis spaces such a bound can be given dependent on $VC(H)$:

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

VC Dimension, Examples

- example 1: suppose $X = \mathbb{R}$ and H all intervals on $\mathbb{R}$, that is, each h has the form $a < x < b$, where a and b are any real constants. Since every set of two real numbers can be shattered but not any set of three real numbers, $VC(H) = 2$

VC Dimension, Examples Cont.

- example 2: suppose $X = \mathbb{R} \times \mathbb{R}$ is the set of points on the x, y plane and H is the set of all linear decision surfaces, that is, all perceptrons defined for this instance space
  - for every set of two points and every classification of these points, a linear decision surface can be found, hence $VC(H) \geq 2$
  - if three colinear points are given, they cannot be shattered, but every set of three non-colinear points can be shattered. Since the definition of VC Dimension depends on *one* existing largest subset, $VC(H) \geq 3$
  - since no set of four points can be shattered, $VC(H) < 4$, that is, $VC(H) = 3$