Machine Learning: Lecture 8

Computational Learning

Theory

(Based on Chapter 7 of Mitchell T.., Machine Learning, 1997)

Overview

- Are there general laws that govern learning?
  - ***Sample Complexity:*** How many training examples are needed for a learner to converge (with high probability) to a successful hypothesis?
  - ***Computational Complexity:*** How much computational effort is needed for a learner to converge (with high probability) to a successful hypothesis?
  - ***Mistake Bound:*** How many training examples will the learner misclassify before converging to a successful hypothesis?
- These questions will be answered within two analytical frameworks:
  - The ***Probably Approximately Correct (PAC)*** framework
  - The ***Mistake Bound*** framework

Overview (Cont'd)

- Rather than answering these questions for individual learners, we will answer them for broad classes of learners. In particular we will consider:
  - The size or complexity of the hypothesis space considered by the learner.
  - The accuracy to which the target concept must be approximated.
  - The probability that the learner will output a successful hypothesis.
  - The manner in which training examples are presented to the learner.

The PAC Learning Model

- **Definition:** Consider a concept class ***C*** defined over a set of instances ***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 ***D*** over ***X***, ε such that $0 < \varepsilon < 1/2$, and δ such that $0 < \delta < 1/2$, learner ***L*** will, with probability at least $(1-\delta)$, output a hypothesis $h \in H$ such that $error_D(h) \leq \varepsilon$, in time that is ***polynomial*** in $1/\varepsilon$, $1/\delta$, n , and ***size(c).***

Sample Complexity for ***Finite*** Hypothesis Spaces

- Given any ***consistent*** learner, the number of examples sufficient to assure that any hypothesis will be probably (with probability ***(1- δ)***) approximately (within error ***ε***) correct is $m = 1/\varepsilon (\ln|H| + \ln(1/\delta))$
- If the learner is ***not consistent***, $m = 1/2\varepsilon^2 (\ln|H| + \ln(1/\delta))$
- Conjunctions of Boolean Literals are also PAC-Learnable and $m = 1/\varepsilon (n.\ln 3 + \ln(1/\delta))$
- k-term DNF expressions are not PAC learnable because even though they have polynomial sample complexity, their computational complexity is not polynomial.
- Surprisingly, however, k-term CNF is PAC learnable.

Sample Complexity for Infinite Hypothesis Spaces I: VC-Dimension

- The PAC Learning framework has 2 disadvantages:
  - It can lead to weak bounds
  - Sample Complexity bound cannot be established for infinite hypothesis spaces
- We introduce new ideas for dealing with these problems:
  - **Definition:** A set of instances S is shattered by hypothesis space H iff for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy.
  - **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 $VC(H)=\infty$

Sample Complexity for Infinite Hypothesis Spaces II

- ***Upper-Bound*** on sample complexity, using the ***VC-Dimension***: $m \geq 1/\varepsilon\,(4\log_2(2/\delta)+8VC(H)\log_2(13/\varepsilon))$
- ***Lower Bound*** on sample complexity, using the ***VC-Dimension***:

Consider any concept class ***C*** such that $VC(C) \geq 2$, any learner ***L***, and any $0 < \varepsilon < 1/8$, and $0 < \delta < 1/100$. Then there exists a distribution ***D*** and target concept in ***C*** such that if ***L*** observes fewer examples than

$max[1/\varepsilon \log(1/\delta),(VC(C)-1)/(32\varepsilon)]$

then with probability at least ***δ***, ***L*** outputs a hypothesis ***h*** having $error_D(h) > \varepsilon$.

VC-Dimension for Neural Networks

- Let G be a layered directed acyclic graph with n input nodes and $s \geq 2$ internal nodes, each having at most r inputs. Let C be a concept class over $\mathbf{R}^r$ of VC dimension d, corresponding to the set of functions that can be described by each of the s internal nodes. Let C_G be the G-composition of C, corresponding to the set of functions that can be represented by G. Then $VC(C_G) \leq 2ds \log(es)$, where e is the base of the natural logarithm.
- This theorem can help us bound the VC-Dimension of a neural network and thus, its sample complexity (See, [Mitchell, p.219])!

The *Mistake Bound* Model of Learning

- The ***Mistake Bound*** framework is different from the PAC framework as it considers learners that receive a sequence of training examples and that predict, upon receiving each example, what its target value is.
- The question asked in this setting is: ***"How many mistakes will the learner make in its predictions before it learns the target concept?"***
- This question is significant in practical settings where learning must be done while the system is in actual use.

Optimal Mistake Bounds

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

$$Opt(C) = \min_{A \in Learning\_Algorithm} M_A(C)$$

- For any concept class C, the optimal mistake bound is bound as follows:

$$VC(C) \leq Opt(C) \leq \log_2(|C|)$$

A Case Study: The Weighted-Majority Algorithm

a_i denotes the ith prediction algorithm in the pool A of algorithm. w_i denotes the weight associated with a_i.

- For all i initialize w_i <-- 1
- For each training example <x,$c(x)$>
  - Initialize q_0 and q_1 to 0
  - For each prediction algorithm ai
    - If $a_i(x)=0$ then q_0 <-- q_0+w_i
    - If $a_i(x)=1$ then q_1 <-- q_1+w_i
  - If $q_1 > q_0$ then predict $c(x)=1$
  - If $q_0 > q_1$ then predict $c(x) =0$
  - If $q_0=q_1$ then predict 0 or 1 at random for $c(x)$
  - For each prediction algorithm a_i in A do
    - If $a_i(x) \neq c(x)$ then w_i <-- βw_i

Relative Mistake Bound for the Weighted-Majority Algorithm

- Let ***D*** be any sequence of training examples, let ***A*** be any set of n prediction algorithms, and let ***k*** be the minimum number of mistakes made by any algorithm in ***A*** for the training sequence ***D***. Then the number of mistakes over ***D*** made by the ***Weighted-Majority*** algorithm using $\beta=1/2$ is at most $2.4(k + \log_2 n)$.
- This theorem can be generalized for any $0 \leq \beta \leq 1$ where the bound becomes

$$(k \log_2 1/\beta + \log_2 n)/\log_2(2/(1+\beta))$$