Computational Learning Theory

Learning Theory

- Theorems that characterize classes of learning problems or specific algorithms in terms of computational complexity or sample complexity (the number of training examples necessary or sufficient to learn hypotheses of a given accuracy)

- **Complexity of a learning problem depends on**
  - Size or expressiveness of the hypothesis space
  - Accuracy to which target concept must be approximated
  - Probability with which the learner must produce a successful hypothesis
  - Manner in which training examples are presented, e.g., randomly or by query to an oracle

Types of Results

- **Learning in the limit:** Is the learner guaranteed to converge to the correct hypothesis in the limit as the number of training examples increases to infinity?
- **Sample complexity:** How many training examples are needed for a learner to construct (with high probability) a highly accurate concept?
- **Computational complexity:** How much computational resources (time and space) are needed for a learner to construct (with high probability) a highly accurate concept?
- **Mistake bound:** Learning incrementally, how many training examples will the learner misclassify before constructing a highly accurate concept

Learning in the Limit vs. PAC Model

- **Learning in the limit model is too strong**
  - Requires learning correct exact concept
- **Learning in the limit model is too weak**
  - Allows unlimited data and computational resources
- **PAC Model** (Leslie Valiant got a Turing Award!)
  - Only requires a Probably Approximately Correct (PAC) concept: learn a decent approximation most of the time
  - Requires polynomial sample complexity and computational complexity

PAC Learning

- The only reasonable expectation of a learner is that with *high probability* it learns a *close approximation* to the target concept

- In the PAC model, we specify two parameters, ϵ and δ, and require that with probability at least $(1 - \delta)$ a system learn a concept with error at most ϵ

PAC Learning

- How to prove PAC learnability?
  - First, prove sample complexity of learning a target concept *h** using a hypothesis space *H* is polynomial
  - Second, prove that the learner can train on a polynomial-sized data set in polynomial time
- To be PAC-learnable
  - There must be a hypothesis in *H* with arbitrarily small error for every target concept *h**

Consistent Learners

- A learner using a hypothesis space *H* and training data *D* is said to be a consistent learner if it always outputs a hypothesis with zero error on *D* whenever *H* contains such a hypothesis

Sample Complexity Result

- Any consistent learner, given at least
  - $\left(\ln\frac{1}{\delta} + \ln|H|\right).\frac{1}{\epsilon}$ examples will produce a result that is PAC
- Just need to determine the size of a hypothesis space to instantiate this result for learning specific target concepts
- This gives a *sufficient* number of examples for PAC learning, but not a necessary number – meaning the bound is very loose in practice

Infinite Hypothesis Spaces

- The preceding analysis was restricted to finite hypothesis spaces
- Some infinite hypothesis spaces (such as those including real-valued thresholds or parameters) are more expressive than others
- Need some measure of the expressiveness of infinite hypothesis space
- The *Vapnik-Chervonenkis (VC)* dimension provides such a measure, denoted *VC(H)*

The VC Dimension

- A set of instances *S* is *shattered* by hypothesis space H <u>if and only if</u> for every dichotomy of *S* there exists some hypothesis in *H* consistent with this dichotomy

- 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 with VC dimension

- Using VC dimension as a measure of expressiveness, the following number of examples have been shown to be sufficient for PAC Learning (Blum et al., 1989)

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

- In general, this can provide a tighter upper bound on the number of examples needed for PAC learning

Summary of Learning Theory

- The PAC framework provides a theoretical mechanism for analyzing the effectiveness of learning algorithms
- The sample complexity for any consistent learner using some hypothesis space, H, can be determined from a measure of its expressiveness *|H|* or *VC(H)*
- If sample complexity is tractable, then the computational complexity of finding a consistent hypothesis in *H* governs its PAC learnability
- Constant factors are more important in sample complexity than in computational complexity, since our ability to gather data is generally not growing exponentially
- Experimental results suggest that theoretical sample complexity bounds over-estimate the number of training examples needed in practice since they are worst-case bounds!

More Readings

- Michael Kearns and Umesh Vazirani: *Introduction to Computational Learning Theory*, MIT Press, 1994.
  - https://mitpress.mit.edu/books/introduction-computational-learning-theory