AN INTRODUCTION
TO
COMPUTATIONAL LEARNING THEORY

SIGML S02E02

Purushottam Kar

An Introduction to Learning

- Learning as problem in
  - Function Approximation
  - Pattern Detection
- How can one acquire the concept of *leanness* [VK94] ?
  - Have someone explicitly encode it as a proposition for us

$$\left[\left(h_1 \leq \text{height} \leq \text{h}_2\right) \wedge \left(w_1 \leq \text{weight} \leq w_2\right)\right]$$

  - "Learn" it from the *teacher*'s behavior
- Learning with *small* errors in *almost all* situations
  - Learn *approximately* in a *probabilistic* sense
  - PAC Learning

PAC Learning

- Aim : to learn a class of *concepts* (read dichotomies) $\mathfrak{C}$ on some domain $\mathfrak{X}$
  - The class of human traits that can be described in terms of height and weight – the domain here is $\mathbb{R}^2$
- Given : an concept C from this class and its behavior on some *labeled* instances $x_1, x_2, \ldots, x_n$ sampled from $\mathfrak{D}_{\mathfrak{X}}$
  - The height and weight of some persons along with leanness
- Output : With high probability, a dichotomy $H \in \mathfrak{H}$ that almost matches the unknown concept

$$\Pr_{x_1, x_2, \ldots, x_n \in \mathfrak{D}} \left[\Pr_{x \in \mathfrak{D}} \left[H(x) \neq C(x) \right] > \varepsilon \right] < \delta$$

Some Points to Note

- The learnt dichotomy is *tested* on the same distribution as the one that generated the training samples
  - Can afford to make errors on low probability regions of $\mathfrak{X}$
- However the distribution itself is unknown
  - Require that the learning algorithm work for every $\mathfrak{D}_{\mathfrak{X}}$
- A concept class $\mathfrak{C}$ is said to be PAC-*learnable* if there exists an algorithm that, for *every* concept $C \in \mathfrak{C}$, when given $\operatorname{poly}(d, 1/\varepsilon, 1/\delta)$ examples from *any* distribution $\mathfrak{D}_{\mathfrak{X}}$ outputs a hypothesis $H \in \mathfrak{H}$ such that

$$\Pr_{x_1, x_2, \ldots, x_n \in \mathfrak{D}} \left[\Pr_{x \in \mathfrak{D}} \left[H(x) \neq C(x) \right] > \varepsilon \right] < \delta$$

Learnable classes

- When can a concept be learnt ?
  - Interpolating a linear polynomial requires at least 2 points
  - Interpolating a quadratic requires at least 3 points
  - Interpolating a cubic requires at least 4 points
  - …
- Intuitively : the more complex a concept, the larger the training set required to learn it
- Simple observation : The class of finite-degree polynomials is not learnable
- What about PAC-learnability, where errors are allowed

Vapnik-Chervonenkis dimension

- PAC-learnability admits a beautiful characterization in terms of the expressive power of the concept class
- The *VC Dimension* of a concept class $\mathfrak{C}$ is the size of the largest set S in $\mathfrak{X}$ such that the concepts in $\mathfrak{C}$ can together realize all possible binary partitions over S
  - Intervals over the real line : 2
  - Halfspaces in $\mathbb{R}^2$: 3 (not all 3-point sets are shattered)
  - Halfspaces in $\mathbb{R}^d$: $d+1$ (not all point sets are shattered)
  - Thresholded polynomials over reals : ∞
  - Convex d-polygons in the plane : $2d+1$

PAC-learning "leanness"

- Concept Class : axis aligned rectangles over $\mathbb{R}^2$
- VC dimension : 4
- Algorithm :
  - Sample $m = 4/\varepsilon \log(1/\delta)$ points IID
  - Return the smallest rectangle that contains all the + points
- The output rectangle will always be contained in the *rectangle of leanness*
- It is very unlikely that a sequence of samplings will trick us into learning a bad rectangle
- The key is to slip in a *hitting set* argument

PAC-learnable classes

- Let $\mathfrak{C}$ be a concept class of VC dimension d, then
  - An algorithm that takes $m = \mathcal{O}\left(1/\varepsilon \log\left(1/\delta\right) + d/\varepsilon \log\left(1/\varepsilon\right)\right)$ training samples and outputs a consistent concept from $\mathfrak{C}$ is able to meet the PAC requirement for any $\mathfrak{D}_{\mathfrak{X}}$
  - However there always exists a concept $C \in \mathfrak{C}$ and a distribution $\mathfrak{D}^*$ for which any algorithm would require at least $\Omega\left(d/\varepsilon\right)$ training samples
- Polynomials are not PAC-learnable
- Convex polygons are not PAC-learnable
- Convex bodies (polyhedrons) are not PAC-learnable
- What next … ?

Some Points to Note

- The VC dimension characterizes the *sample complexity* of learning algorithms that work for a given class
  - Silent on the time complexity of algorithms
  - Useful only in proving time lower bounds
- Only partial results known for time complexity
- [KO'DS08] For learning Geometric concepts (bodies) under the Gaussian distribution, the *Gaussian Surface Area* of the bodies is a near perfect indicator of computational complexity

Distribution Specific Learning

- Can we try to learn the concepts under certain "natural" distributions ?
- [GR09] : Convex bodies are hard to learn even under the uniform distribution
- More specifically, there are convex bodies which force every learning algorithm to draw at least $2^{\Omega\left(\sqrt{d/\varepsilon}\right)}$ samples from the uniform distribution
- [KO'DS08] Under the Gaussian distribution, learning is
  - possible in time $2^{\tilde{\mathcal{O}}\left(\sqrt{d}\right)}$
  - requires $2^{\Omega\left(\sqrt{d}\right)}$ samples

References

- [GR09] Navin Goyal and Luis Rademacher, **Learning Convex Bodies is Hard**, COLT, 2009.
- [KO'DS08] Adam Klivans, Ryan O'Donnel and Rocco Servedio, **Learning Geometric Concepts via Gaussian Surface Area**, FOCS, 2008.
- [VK94] Umesh Vazirani and Michael Kearns, **An Introduction to Computational Learning Theory**, The MIT Press, 1994.