Computational Learning Theory: Probably Approximately Correct (PAC) Learning

Machine Learning
Spring 2018

THE UNIVERSITY OF UTAH

The slides are mainly from Vivek Srikumar

This lecture: Computational Learning Theory

- The Theory of Generalization
- Probably Approximately Correct (PAC) learning
- Positive and negative learnability results
- Agnostic Learning
- Shattering and the VC dimension

Where are we?

- The Theory of Generalization
  - When can be trust the learning algorithm?
  - What functions can be learned?
  - Batch Learning
- Probably Approximately Correct (PAC) learning
- Positive and negative learnability results
- Agnostic Learning
- Shattering and the VC dimension

This section

1. Analyze a simple algorithm for learning conjunctions

2. Define the PAC model of learning

3. Make formal connections to the principle of Occam's razor

This section

✓ Analyze a simple algorithm for learning conjunctions

2. Define the PAC model of learning

3. Make formal connections to the principle of Occam's razor

Formulating the theory of prediction

All the notation we have so far on one slide

In the general case, we have

- X: instance space, Y: output space = {+1, -1}
- D: an unknown distribution over X
- f: an unknown target function X → Y, taken from a concept class C
- h: a hypothesis function X → Y that the learning algorithm selects from a hypothesis class H
- S: a set of m training examples drawn from D, labeled with f
- $err_D(h)$: The true error of any hypothesis h
- $err_S(h)$: The empirical error or training error or observed error of h

Theoretical questions

- Can we describe or bound the true error (err_D) given the empirical error (err_S)?
- Is a concept class C learnable?
- Is it possible to learn C using only the functions in H using the supervised protocol?
- How many examples does an algorithm need to guarantee good performance?

Requirements of Learning

- Cannot expect a learner to learn a concept exactly
  - There will generally be multiple concepts consistent with the available data (which represent a small fraction of the available instance space)
  - Unseen examples could *potentially* have any label
  - We “agree” to misclassify *uncommon* examples that do not show up in the training set

Requirements of Learning

- Cannot expect a learner to learn a concept exactly
  - There will generally be multiple concepts consistent with the available data (which represent a small fraction of the available instance space)
  - Unseen examples could *potentially* have any label
  - We “agree” to misclassify *uncommon* examples that do not show up in the training set

- Cannot always expect to learn a close approximation to the target concept

Requirements of Learning

- Cannot expect a learner to learn a concept exactly
  - There will generally be multiple concepts consistent with the available data (which represent a small fraction of the available instance space)
  - Unseen examples could *potentially* have any label
  - We “agree” to misclassify *uncommon* examples that do not show up in the training set

- Cannot always expect to learn a close approximation to the target concept
  - Sometimes (only in rare learning situations, we hope) the training set will not be representative (will contain uncommon examples)

Requirements of Learning

- Cannot expect a learner to learn a concept exactly
  - There will generally be multiple concepts consistent with the available data (which represent a small fraction of the available instance space)
  - Unseen examples could *potentially* have any label
  - We "agree" to misclassify *uncommon* examples that do not show up in the training set

- Cannot always expect to learn a close approximation to the target concept
  - Sometimes (only in rare learning situations, we hope) the training set will not be representative (will contain uncommon examples)

- The only realistic expectation of a good learner is that with high probability it will learn a close approximation to the target concept

Probably approximately correctness

- The only realistic expectation of a good learner is that with high probability it will learn a close approximation to the target concept

Probably approximately correctness

- The only realistic expectation of a good learner is that with high probability it will learn a close approximation to the target concept

- In Probably Approximately Correct (PAC) learning, one requires that
  - given small parameters ϵ and δ,
  - With probability at least 1 - δ, a learner produces a hypothesis with error at most ϵ

Probably approximately correctness

- The only realistic expectation of a good learner is that with high probability it will learn a close approximation to the target concept

- In Probably Approximately Correct (PAC) learning, one requires that
  - given small parameters ϵ and δ,
  - With probability at least 1 - δ, a learner produces a hypothesis with error at most ϵ

- The only reason we can hope for this is the *consistent distribution assumption*

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if

for all $f \in C$,

for all distribution D over X, and fixed $0 < \varepsilon, \delta < 1$,

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if

for all $f \in C$,

for all distribution D over X, and fixed $0 < \varepsilon, \delta < 1$,

given m examples sampled independently according to D, the algorithm L produces, with probability at least $(1-\delta)$, a hypothesis $h \in H$ that has error at most ε,

where m is *polynomial* in $1/\varepsilon$, $1/\delta$, n and size(H)

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if

for all $f \in C$,

for all distribution D over X, and fixed $0 < \varepsilon, \delta < 1$,

given m examples sampled independently according to D, the algorithm L produces, with probability at least $(1-\delta)$, a hypothesis $h \in H$ that has error at most ε,

where m is *polynomial* in $1/\varepsilon$, $1/\delta$, n and size(H)

recall that $Err_D(h) = Pr_D[f(x) \neq h(x)]$

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if

for all $f \in C$,

for all distribution D over X, and fixed $0 < \varepsilon, \delta < 1$,

given m examples sampled independently according to D, the algorithm L produces, with probability at least $(1 - \delta)$, a hypothesis $h \in H$ that has error at most ε,

where m is *polynomial* in $1/\varepsilon$, $1/\delta$, n and size(H)

recall that $Err_D(h) = Pr_D[f(x) \neq h(x)]$

The concept class C is *efficiently learnable* if L can produce the hypothesis in time that is polynomial in $1/\varepsilon$, $1/\delta$, n and size(H)

PAC Learnability

- We impose two limitations

PAC Learnability

- We impose two limitations
- Polynomial *sample complexity* (information theoretic constraint)
  - Is there enough information in the sample to distinguish a hypothesis h that approximate f ?

PAC Learnability

- We impose two limitations
- Polynomial *sample complexity* (information theoretic constraint)
  - Is there enough information in the sample to distinguish a hypothesis h that approximate f ?
- Polynomial *time complexity* (computational complexity)
  - Is there an efficient algorithm that can process the sample and produce a good hypothesis h ?

PAC Learnability

- We impose two limitations
- Polynomial *sample complexity* (information theoretic constraint)
  - Is there enough information in the sample to distinguish a hypothesis h that approximate f ?
- Polynomial *time complexity* (computational complexity)
  - Is there an efficient algorithm that can process the sample and produce a good hypothesis h ?

To be PAC learnable, there must be a hypothesis $h \in H$ with arbitrary small error for every $f \in C$. We assume $H \supseteq C$. (*Properly* PAC learnable if H=C)

PAC Learnability

- We impose two limitations
- Polynomial *sample complexity* (information theoretic constraint)
  - Is there enough information in the sample to distinguish a hypothesis h that approximate f ?
- Polynomial *time complexity* (computational complexity)
  - Is there an efficient algorithm that can process the sample and produce a good hypothesis h ?

To be PAC learnable, there must be a hypothesis $h \in H$ with arbitrary small error for every $f \in C$. We assume $H \supseteq C$. (*Properly* PAC learnable if H=C)

Worst Case definition: the algorithm must meet its accuracy

- for every distribution (The distribution free assumption)
- for every target function f in the class C