Computational Learning Theory: Probably Approximately Correct (PAC) Learning

Machine Learning

THE UNIVERSITY OF UTAH

Slides based on material from Dan Roth, Avrim Blum, Tom Mitchell and others

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

This section

1. Define the PAC model of learning
2. Make formal connections to the principle of Occam's razor

This section

1. Define the PAC model of learning

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

Recall: The setup

- Instance Space: X, the set of examples
- Concept Space: C, the set of possible target functions: $f \in C$ is the hidden target function
  - Eg: all n-conjunctions; all n-dimensional linear functions, …
- Hypothesis Space: H, the set of possible hypotheses
  - This is the set that the learning algorithm explores
- Training instances: $S \times \{-1,1\}$: positive and negative examples of the target concept. (S is a finite subset of X)
  - *Training instances are generated by a fixed unknown probability distribution D over X*
- What we want: A hypothesis $\mathrm{h} \in H$ such that $h(x) = f(x)$
  - Evaluate h on subsequent examples $x \in X$ drawn according to D

Formulating the theory of prediction

All the notation we have seen 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 $\mathrm{X} \rightarrow Y$, taken from a concept class C
h	a hypothesis function $\mathrm{X} \rightarrow Y$ that the learning algorithm selects from a hypothesis class H
S	a set of m training examples drawn from D, labeled with f
$\mathrm{err}_{\mathrm{D}}(h)$	The true error of a hypothesis h
$\mathrm{err}_{\mathrm{S}}(h)$	The empirical error or training error or observed error of h

Theoretical questions

- Can we describe or bound the true error ($\mathrm{err}_{\mathrm{D}}$) given the empirical error ($\mathrm{err}_{\mathrm{S}}$)?
- 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?

Expectations of learning

We 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
- Let us "agree" to misclassify uncommon examples that do not show up in the training set

Expectations of learning

We 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
- Let us “agree” to misclassify uncommon examples that do not show up in the training set

We cannot always expect to learn a close approximation to the target concept

Sometimes (hopefully only rarely) the training set will not be representative (will contain uncommon examples)

Expectations of learning

We 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

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

We cannot always expect to learn a close approximation to the target concept

Sometimes (hopefully only rarely) the training set will not be representative (will contain uncommon examples)

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 $1 - \delta$, 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 $1 - \delta$, 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 $0 < \epsilon, \delta < 1$,

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 $0 < \epsilon, \delta < 1$,

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

where m is *polynomial* in $1/\epsilon$, $1/\delta$, 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 $0 < \epsilon, \delta < 1$,

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

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

Given a small enough number of examples

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 $0 < \epsilon, \delta < 1$,
given m examples sampled independently according to D, with $\boxed{\text{probability at least } (1 - \delta)}$, the algorithm L produces a hypothesis $h \in H$ that has error at most ϵ,
where m is *polynomial* in $1/\epsilon$, $1/\delta$, n and $size(H)$.

Given a small enough number of examples

with high probability

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 $0 < \epsilon, \delta < 1$,

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

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

Given a small enough number of examples

with high probability

the learner will produce a "good enough" classifier.

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 $0 < \epsilon, \delta < 1$,

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

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

recall that $Err_D(h) = Pr_{x \sim D}[f(x) \neq h(x)]$

Given a small enough number of examples

with high probability

the learner will produce a "good enough" classifier.

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 $0 < \epsilon, \delta < 1$,

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

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

The concept class C is *efficiently learnable* if L can produce the hypothesis in time that is polynomial in $1/\epsilon$, $1/\delta$, 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 approximates 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 approximates 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 approximates 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 approximates 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