15251 Final Project - Learning

mconn

December 11, 2020

Contents

1 Introduction

I'll make this very short because there's the whole rest of the paper. This project was about Learning and specifically **PAC-learning**. What excited me about it was a simple aside in a book

> ...rectangles are efficiently pac-learnable, the proof is left as an exercise to the reader.

Of course! Classic math books. But as I delved into it more I saw all these interesting connections to AI, to philosophy, even to data compression! These following words are largely my own, unless they are not. I'm sure I've made gaffs, errors, and I've missed some of the latest research. Anticipating that, let me offer this as a *carte-blanche errata*.

The format of my paper was inspired by 15-251 Chapters but obviously diverges quite a bit. My goal is for the reader to leave with both an interesting concept and a healthy amount of math underpinning it.

2 Notes on Grue

> There's a joke about a planet full of people who believe in anti-*induction*: if the sun has risen every day in the past, then today, we should expect that it won't. As a result, these people are all starving and living in poverty. Someone visits the planet and tells them, "Hey, why are you still using this anti-induction philosophy? You're living in horrible poverty!"
>
> "Well, it never worked before ..."
>
> -Scott Aronson[1]

> What's a grue? I don't know, what's a grue with you?
>
> - Me to myself at 2 AM

> Now let me introduce another predicate less familiar than "green". It is the predicate "grue" and it applies to all things examined before t just in case they are green but to other things just in case they are blue. Then at time t we have, for each evidence statement asserting that a given emerald is green, a parallel evidence statement asserting that that emerald is grue. And the statements that emerald a is grue, that emerald b is grue, and so on, will each confirm the general hypothesis that all emeralds are grue.
>
> -Nelson Goodman[4]

> Induction is the glory of science and the scandal of philosophy
>
> -C. D. Broad

2.1 Hume's Problem of Induction - aka Old Induction

2.1.1 A Refresher on Mathematical Induction

If you're reading this you're most likely familiar with the mathematical concept of induction. I'll rehash it briefly here. Assume we have a proposition $P : \mathbb{N} \to \{0, 1\}$. Then if we show the following

$$P(1)\text{is true}$$
$$P(n) \to P(n+1)$$

Then we can conclude $\forall n, P(n)$ is true. Why is this the case? It follows from the Well-Ordering Principle. I'll include the proof here becomes it's elegant and I feel we often take induction for granted.

Theorem 1 (Well-Ordering Principle)**.** *Every non-empty set of positive integers has a least element.*

Proof of Induction by Contradiction.

Assume the following

$$P(1)\text{is true}$$
$$P(n) \to P(n+1)$$

however we are assuming that $\forall n P(n)$ doesn't follow. Formally

$$P(1) \wedge (P(n) \to P(n+1)) \implies \exists n, !P(n)$$

Let S denote the set of all positive integers where $P(n)$ is false. We do not know the cardinality of S however we do know there is a least element by **WOP**.

Without loss of generality denote l to be the minimum element in S. Then we know that $l \neq 1$ because of our initial assumption of $P(1)$. From this it follows there $\exists l - 1$ for which $P(l - 1)$ is true. This is because we assumed l was our minimum element in the set of elements for which P is false. $l - 1$ is positive because $l > 1$ and $l - 1 \notin S$ because of our assumption that l is minimal.

Then by the second part of our assumption we have

$$P(l-1) \implies P(l) \therefore P(l) \text{ is true.}$$

We have reached a contradiction.

We have proved that $S = \emptyset$. If S is empty then $\forall n, P(N)$ follows from our initial assumptions. □

So we have shown the validity of mathematical induction. But why is this important? I believe we often take it to be trivially true in number theory that we can prove truthfulness over infinite sets. In a few easy steps we've shown it's possible. However, our reality isn't as easily represented by $\mathbb{N}$

2.1.2 Hume

During his life Hume was concerned with *Epistemology*, that is, the philosophical field studying where knowledge comes from.

In trying to grapple with where knowledge comes from Hume starts off with a fairly safe assumption - our knowledge about unknown things comes from the synthesis of simpler experiences and ideas. The famous example is that we see gunpowder and we then infer the effect of gunpowder to be explosion. We relate our past experience (knowing gunpowder can explode), with our present experience (seeing the gunpowder), to conclude that it can explode in the future. Hume reasons that we can't make this conclusion *a priori*, ie without previously knowing that gunpowder will explode. This conclusion, this leap from the past and the present to the future is what Hume is concerned about. Hopefully here you can begin to see why this is deemed a problem of induction. In Hume's words this problem is the following:

> I have found that such an object has always been attended with such an effect, and I foresee, that other objects, which are, in appearance, similar, will be attended with similar effects. [5]

What exactly can we base this reasoning on? Well, we assume that nature will act consistently. Could you imagine how bizarre it'd be if nature did not?[1] I'd argue this is a fairly safe assumption. In fact, I'd go so far as to say that all of science is hinged upon this assumption. Even Quantum Mechanics a field where at a glance you see nature behaving 'non-deterministically' what you're really seeing is nature behaving probably. Even though nature acts probably, it still acts consistently. Coincidentally Hume would argue for the same assumption. He referred to this assumption as "Uniformity Principle". However, Hume then takes this argument and turns it on its head by concluding there's no clear reasoning behind such an assumption.

For Hume there were two kinds of reasoning that could possibly be used to justify such a principle: *Demonstrative Reasoning* and *Moral Reasoning*.

Demonstrative Reasoning is like what mathematical induction is for us. This is reasoning that deals with things that are apparently true or demonstratively true, like proving a mathematical statement.[2] However, the problem with using something like mathematical reasoning (in Humes formulation) is that its unwavering. Once we've proven something true in mathematics it usually stays true (assume for simplicity that this is true). But using a similar style of reasoning in the real world would prove problematic. There are cases where we believe something to be true but then new experiences invalidate our past belief of truthfulness. Because of this, clearly something like *Demonstrative Reasoning* can not be the basis of which we derive new knowledge. If it were, we'd be incapable of invalidating previously found truths. If this were the case, well, no good would come out of that.

Moral Reasoning is the second type of reasoning. This was also referred to as "probable" reasoning. Imagine a chain of assumptions where each preposition is probable. However, this style

[1] If we accepted *proof reductio ad absurdum* as a valid proof technique we could end the discussion here.

[2] I'm weary to make such a connection but I feel its the only clear analogy I can draw.

of reasoning depends on the Uniformity Principle. That is, if we assume from a chain of reasoning that the future will be conformable to the past, then we're using the Uniformity Principle to prove itself. Even if we've drawn a strong conclusion about our previous experiences, to Hume, there is no good reason to apply that conclusion to the future.

Hume in turn presents his own possibility:

> When the mind, therefore, passes from the idea or impression of one object to the idea or belief of another, it is not determin'd by reason, but by certain principles, which associate together the ideas of these objects, and unite them in the imagination.

That is to say, there is no reasoning just heuristics. A natural question then would be, what are the heuristics?

We will end the section on Hume here. I included it to shed some historical light on the possibly difficulty of reasoning/learning. If I misrepresented Hume that is my own fault. He's a rich and diverse author and there is much to delve into and explore with him.

2.2 Grue and Induction

As we've seen in the previous section there is some difficulty with justifying induction about our experiences. Nelson Goodman saw Hume's conclusion which can be summed up (poorly) as, "There is no reasoning for induction, just heuristics", and sought to further examine these heuristics. Goodman came up with the following "New Riddle of Induction" to demonstrate his progress:

> Consider the following two (supposedly true) statements:
> (B1) This piece of copper conducts electricity.
> (B2) This man in the room is a third son.
> B1 is a confirmation instance of the following regularity statement:
> (L1) All pieces of copper conduct electricity.
> But does B2 confirm anything like L2?
> (L2) All men in this room are third sons.

[3]

The conclusion in $L2$ follows from the same reasoning as that in $L1$. However, it feels obvious that $L2$ is not-true while $L1$ is plausibly true. What gives? Goodman classified the first statement, $L1$ as lawlike, while the second, $L2$, as accidentally true. Making the distinction between these statements is important if we're looking for universal truths. However, it's clear that using basic semantic transformations ie a function f : Evidence $\rightarrow$ Proposition isn't good enough.

Consider in addition the quote at the beginning of this section by Goodman (note, it's not immediately obvious but the phrase "but to other things" means after t). Consider that you're someone who studies gems. We want to make a statement about all the green gems we've since seen and the ones we will see. Say we've had a hard day examining gems in the field. We've kept observing green gems and so we form the following two predictions:

1. The next gem we'll observe after t will be green.

2. The next gem we'll observe after t will be grue.

And this right here shows the crux of the issue. Both our predictions are confirmed by our past evidence and yet these predictions are mutually incompatible after t.

Now, you might take offense to the time predicate. An obvious critique is how arbitrary it is. However, without delving too much into this consider that all observations have time and spatial predicates. Because of the Uniformity Principle we assume that these restrictions don't apply ie - discovering some phenomenon in X at time y should hopefully also be true at Z at time $y + 1$. But how do we know? How can we be certain. According to Hume, there is no good reasoning for accepting such a principle. Even without such a 'ridiculous' predicate this extreme example highlights a clear problem in inductive reasoning.

Its important to highlight that in our quest for science we hope to find statements like being *green* and to avoid statements like being *grue*. So then the question is, how do we distinguish these?

To reframe this in another sense. Say we're hiking a mountain during a blizzard. Since we're hiking a mountain we're of course looking for the absolute highest point, the summit. The reader will realize we're finding a relative maximum. That's easy, as we're hiking we find Δf and if $\Delta f = \hat{0}$ we're at a critical point. We'll inspect f'' and see just what kind of point it is. In hiking terms this means arriving at a plateau and looking around you to see if the mountain slopes down, or it slopes up. However, how can we be absolutely certain we're at the maxima. After all, there's a fog and we can't see the summit. All we can hope to do is to keep recording our results and keep finding maximums and hope if we do this enough we'll eventually reach the summit. However, with an unknown mountain with an unknown amount of plateaus. There's no guaranteed bound for if or when we'll ever get there. Our hope is that we can be probably correct, ie we've correctly assessed we're at a local maxima and approximately correct, ie this local maxima is actually the universal maxima.

Perhaps, knowledge is the mountain and we're the hiker.

Conclusion

I wanted to include this section to highlight the difficulties of inductive reasoning. Mainly, as I've said previously, I feel we take for granted the ease we have of proving something in mathematics. By no means is math easy. However, applied generally induction becomes something that can be shown in a page to an argument drawn out over centuries - with no end in sight!

The following section is concerned with a 'Theory of knowledge-finding' that tries to take into account the difficulties stated above. Namely, that we might not have priors about the information we're learning, we might now even know if we're generally right. In a way, I feel **PAC** learning is a formalization of Goodman and Hume's concerns.

Post Note

If you're curious how Goodman resolves his induction problem he presents a few rules where you choose the most 'entrenched' predicate. In the example above, 'green' would be chosen over 'grue' because it's been used more frequently to support more hypothesises. Goodman uses the term 'projected' to refer to a predicate supporting a hypothesis. His rules for choosing predicates are as follows:

> A hypothesis is projectible iff it is supported, unviolated, and unexhausted, and all hypotheses conflicting with it are overridden.
>
> A hypothesis is unprojectible iff it is unsupported, exhausted, violated, or overridden.
>
> A hypothesis is nonprojectible iff it and a conflicting hypothesis are supported, unviolated, unexhausted, and not overridden.

3 A Theory of the Learnable

This entire chapter is concerned with what's called **Computational Learning Theory**. **CLT** is not well defined. Of the papers and textbooks I've read concerning it many different notations are used for similar things. I hope this does not trip up the reader and that the general idea is well conveyed.

The origins of the field can be attributed to this paper[8] whose abstract I've copied here

> Humans appear to be able to learn new concepts without needing to be programmed explicitly in any conventional sense. In this paper we regard learning as the phenomenon of knowledge acquisition in the absence of explicit programming. We give a precise methodology for studying this phenomenon from a computational viewpoint. It consists of choosing an appropriate information gathering mechanism, the learning protocol, and exploring the class of concepts that can be learned using it in a reasonable (polynomial) number of steps. Although inherent algorithmic complexity appears to set serious limits to the range of concepts that can be learned, we show that there are some important nontrivial classes of propositional concepts that can be learned in a realistic sense.
>
> - L. G. Valiant

A helpful translation of Valiants abstract is the following - by all accounts (Hume and Goodman's) learning shouldn't be possible! It's 'impossible' to fully confirm a theory. And yet, humans seem to do such things everyday without any explicit programming. A funny analogy is that a toddler can expertly learn to walk but it takes teams of PhDs to make a robot do the same thing half as well.

One of the key observations reflecting on our own thinking is that we never actually consider every hypothesis on its own. A given problem might have a hypothesis space thats exponential. Despite this we still manage to learn. The secret to our success? Who knows exactly. Obviously we employ many heuristics to cut down the search tree. Occam's razor cuts down the tree by quite a bit and then priors help even further. Somehow we implicitly take a search problem thats potentially NP (or harder) and turn it into P. But how do we formalize Occam's Razor and priors? This is the interest, among other things, of **CLT**.

3.1 Math

Definition 1 (Sample Space)**.** *We denote X as the set of all possible examples or instance. Some X is referred to as the input space or sample space.*

An important note. We assume that our examples are identically and independently distributed by some fixed but unknown distribution $\mathbf{D}$. This is incredibly important as will be seen later.

Definition 2 (Labels)**.** *The set of all possible labels or target values is denoted by y. In the special case $|y| = 2$ then we have $y = \{0, 1\}$ which is when we're referring to binary classification.*

Definition 3 (Concept)**.** *A concept c is a function $c : X \to Y$*

Definition 4 (Concept Class)**.** *Denoted $\mathbf{C}$ is the set of concepts which we want to learn.*

Definition 5 (Hypothesis and Hypothesis Space)**.** *A hypothesis is guess as to a mapping from samples to labels, in this way $h \sim c$. The hypothesis can have various relationships to the concept class, subset, equal, superset, etc. These various notions lead to different notions of PAC learnability.*

3.1.1 Problem Statement

The learner considers a fixed set of possible concepts called $\mathbb{H}$ (note $\mathbb{H} \subset \mathbf{C}$ but this is not strict). It then receives a sample of data drawn from $\mathbf{D}$, $S = (x_1, \ldots, x_m)$ as well as the labels $(c(x_1), \ldots, c(x_m)$ which are based on some fixed $c \in \mathbf{C}$. The problem is to use S to find $h_s \in \mathbb{H}$ with small error.

TL;DR 1. *Given some labels and input data we want to find a reasonable hypothesis for such data.*

Definition 6 (Generalization Error)**.**

$$R(h) = Pr_{x \sim D}(h(x) \neq c(x)] = E_{x \sim D}[1_{h(x) \neq c(x)}]$$

Where 1_ω is the indication random variable for even ω

The generalization error is not accessible to the learner because C, D are unknown.

Definition 7 (Empirical Error)**.**

$$\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} 1_{h(x_i) \neq c(x_i)}$$

Note the use of R We can think of R as $Error$ or $Risk$. We avoid using E as not to confuse the notation with expected value.

The empirical error is available to the learner.

Corollary 1.

$$\forall h \in \mathbf{H}, E_{S\sim D^m}[\hat{R}_S(h)] = R(h)$$

Therefore by linearity of expectation we can write (recall that samples are drawn i.i.d):

$$\underset{S\sim D^m}{E}[\hat{R}_S(h)] = \frac{1}{m}\sum_{i=1}^{m} E_{S\sim D^m}[1_{h(x_i)\neq c(x_i)}]\frac{1}{m}\sum_{i=1}^{m} E_{S\sim D^m}[1_{h(x)\neq c(x)}]$$

$$E_{S\sim D^m}[\hat{R}_S(h)] = E_{S\sim D^m}[1_{h(x_i)\neq c(x_i)}] = E_{x\sim D}[1_{h(x)\neq c(x)}] = R(h)$$

What's nice about the above corollary is that it relates our two measures of error, one known and the other unknown, with each other.

Definition 8 (PAC-learnable)**.** *A concept class C is said to be PAC-learnable if $\exists$ an algorithm $\mathbf{A}$ and a polynomial function $f(\cdot,\cdot,\cdot,\cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all distributions $\mathbf{D}$ on X and for any target concept $c \in \mathbf{C}$ the following holds for any sample size $m \geq f(\frac{1}{\epsilon}, \frac{1}{\delta}, n, |c|)$:*

$$Pr_{S\sim D^m}[R(h_s) \leq \epsilon] \geq 1 - \delta$$

If $\mathbf{A}$ runs in $f(\frac{1}{\epsilon}, \frac{1}{\delta}, n, |c|)$ then C is said to be efficiently PAC-learnable. When an algorithm $\mathbf{A}$ exists, it is called a PAC-learning algorithm for $\mathbf{C}$.

[6]

In a way the whole problem is going to be how to fit our empirical error to our generalization error.

Theorem 2. *In order to satisfy the requirement that the output hypothesis h agrees with $1-\epsilon$ of the future data drawn from D, with probability $1-\delta$ over the choice of samples, it suffices to find any hypothesis h that agrees with*

$$m \geq \frac{1}{\epsilon}\log(\frac{|C|}{\delta})$$

[8]

Proof. Denote a hypothetical hypothesis h. Then h is *bad* if it disagrees with f for more than an ϵ fraction of the data. Given m samples we have the following bound (note the samples are independent)

$$Pr[h(x_1) = f(x_1), \dots, h(x_m) = f(x_m)] < (1-\epsilon)^m$$

Recall the union bound. If $A_1, A_2, \dots, A_n$ are events then:

$$Pr[A_1 \cup A_2 \cup \dots \cup A_n] \leq Pr[A_1] + Pr[A_2] + \dots + Pr[A_n]$$

Now we want to find the probability there *exists* a bad hypothesis $h \in C$ that agrees with all the sample data. Since our sample space has size $|C|$ we have

$$Pr[h\ \textit{is bad and agrees with f for all our samples}] < |C|(1-\epsilon)^m$$

We then use the following helpful bound $1 - x \leq e^{-x}, \forall x \in \mathbb{R}$. By setting $Pr \leq \delta$ we get the desired result. □

Definition 9 (VCDimension)**.** *Let s be a sample. Suppose we represent s as a vector of the form $s = \{x_0, x_1, \ldots, x_m\}$. Then the VCDim of s represented as $\prod_C(S) = \{(h(x_1), \ldots, h(x_m)|h \in C\}$*

This definition is somewhat confusing so I'll try and shed some light on it. Firstly, it's assuming that $H = C$, that is our hypothesises are drawn from our concept class. The VCDimension of a sample is all possible valid settings of that sample. An example would be if we had two points on a line and we wanted to try and classify the points as positive or negative. Then the possible settings in the VCDimension would be, they're all in, $\{+, +\}$, one is one isn't, $\{+, -\}$, $\{-, +\}$, and finally neither of them are in, $\{-, -\}$.

To give some exposition to the axis-aligned Rectangles problems that's coming up that problem has a VCDim of 4. Imagine you had 4 points and you wanted to label them in a rectangle or not. Well this is trivial for 4 points. You can just move the rectangle in various ways to either contain or not contain them. Now, imagine you had 5 points. Then there's always some point that will be interior. All possible configurations of contained or not contained by the rectangle are not reachable.

Definition 10 (Shattering)**.** *A set of samples S is shattered by C if $|\prod_C(s)| = 2^m$*

To show this, show $\exists s$ of size d that can be shattered and $\exists s', |s'| = d + 1$ such that s' can't be shattered.

Simply put, this means all possible settings of vectors (for binary classification) are possible in our concept class.

The VCDimension is important because it allows us to upper bound how many samples we might need as you'll see in the next theorem. Practically VCDim is often used to describe the complexity of a model. Shattering, means that all possible configurations of points are reachable in the model/hypothesis.

Corollary 2. *One can perform PAC learning $\iff$ $VCDim(c)$ is finite.*

Corollary 3. *A concept class C with $VCDim(c) = \infty$ is not PAC-learnable.*

Theorem 3 (Blumer, Ehrenfeucht, Haussler, and Warmuth 1989)**.** *In order to produce a hypothesis h that will explain a $1 - \epsilon$ fraction of future data drawn from the disteribution D, with probability $1 - \delta$, it suffices to output any $h \in C$ that agrees with*

$$m \geq O(\frac{1}{\epsilon}(VCdim(C)\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})))$$

[2]

TL;DR 2. *Unfortunately the proof for the above theorem is a bit involved. However, I'd like to share some exposition on why the above might be the case.*

If the VCDimension is finite then after seeing a number of samples greater than the VCDim the variance of the data should grow roughly bounded by the VCDim. After m observations we'll have seen less than 2^m possible outcomes in our vectors. If this were not the case, than $VCDim(C) \geq m$. To describe the m outcomes takes at most m bits. So, this bound relates the VCDim, with our sample numbers.

The VCDimension is one way of rephrasing the bias-variance tradeoff. In Machine Learning and Statististics this is the conflict of trying to minimize bias and variance however minimizing one will maximize the other. Similarly the VCDimension relates the complexity of the model with the samples needed. As our model gets more complex, possibly in an effort to reduce bias, then you'll end up with more variance. This will require more samples to then correct for.

3.1.2 Examples

This is the classic PAC example and I'd be remiss not to include it. Unfortunately the proof is rather geometric so I'll provide the problem here and leave the problem to the reader ;)

Exercise 1 (Axis-Aligned Rectangles)**.** *Consider the cartesian plane $X = R^2$ and the concept class C which is axis-aligned rectangles lying in R^2. The learning problem is that each $c \in C$ represents some positively labelled and some negatively labelled points such that the positive points are in a rectangle. The goal is to show that finding a rectangle that encapsulates the positively labelled points is PAC-learnable.*

Exercise 2 (Hempel's Paradox)**.**

> *(1) All ravens are black. In the form of an implication, this can be expressed as: If something is a raven, then it is black.*

Via contrapositive we have the following:

> *If something is not black, then it is not a raven.*

Part of the paradox is that because these statements are logically equivalent then they should be equally effective in the process of learning something to be a raven. However, if you ask someone to go out and learn about ravens they (usually) never begin by studying those things which are not black.

Now consider the following evidence:

> *The raven at the zoo is black.*

This would support the hypothesis that all ravens are black. But what about,

> *My white mug is not black, and by virtue of being a mug, is not a raven.*

Since we have the logical equivalence of the first two statements then this last piece of evidence also supports the initial hypothesis. However, this seems quite unfair because we've contributed evidence about ravens without having to encounter one.

The takeaway from Hempel's Paradox is that positive observations/classifications should affect the hypothesis, contradicting observations, like a white raven, should hinder it, and observations about those things which are not ravens should not have any influence. While not included in this paper this thinking can be exploited when working with learning boolean literals. Namely, a positive setting of a boolean variable in some true statement means that variable must be positive or should be positive. However, a false statement doesn't provide as much information because it's not known which part of the boolean statement leads to it being false.

3.2 Extensions

3.2.1 Learning Reductions

In **CLT** there's been a number of interesting results. Obviously I can't share them all here but I'd like to share a few. One of them is the borrowing of Reductions from Complexity Theory. Much like we show the hardness of a problem by reducing it to an already known hard problem, we can perform learning reductions to show that if B is *PAC-learnable* and $A \leq B$ then A is also *PAC-learnable*. A word of warning however, learning reduction transformations **do not** need to be efficiently computable. They need to be length preserving. This has the nice property of preserving the sample complexity (how many sample we need) but still means that something that is computably hard might stay computably hard.

Definition 11 (Reducibility)**.** *Let R and R' be representations of a problem. Let Σ, Σ' refer to the languages used for these representations. Then R reduces to R', $R \leq R'$ iff there are function $f : \Sigma^* \times \mathbb{N} \times \mathbb{N} \to \Sigma^*$(this is a transformation of of samples) and $g : R \times \mathbb{N} \times \mathbb{N} \to R'$(this is a transformation of representations) and polynomials t, q such that $\forall s, n \in \mathbb{N}, r \in R, r' \in R$ and $w \in \Sigma, w' \in \Sigma'$.* [3] *Note that c is our concept mapping.*

1. $w \in c(r) \iff f(w, s, n) \in c'(g(r, s, n))$
2. f *is computable in time* $t(|w|, s, n)$
3. $|g(r, s, n)| \leq q(|r|, s, n)$

Now an interesting aspect of Learning Reductions is that just because a problem is intractable in a certain representation doesn't necessarily mean it's intractable in all representations. I credit my TA Eddie with the following insight. This in a way is a rehashing of the fundamentals of Deep Learning. That is, we have some problem in some high dimensional representation and it is current

[3] s, n are usually unnecessary parameters. Their usage in the definition is to bound the length of the representation. These would be necessary if in a proof you had to show that the representation complexity is similar.

state its intractable. The network in turn then tries lower the dimensionality of the problem and sift the data so in the end we have a probability distribution for classification. At its most basic form fully connected layers try to fit hyperplanes to labels while activation functions push the labels apart. Things like stride, maxpooling, and other techniques can be thought of as lowering the dimensionality of the data. Hopefully when we use such thecniques we're throwing away excess data. To put it brutishly, deep learning is a form of representational learning reductions.

An interesting result along these lines is the following:

For each constant $k \geq 2$ the class of $k - term$ **DNF** (disjunctive normal formulas) is not *PAC-learnable* (assuming **RP** $\neq$ **NP**). This means, that $k - term$ **DNF**s are not learnable in their given representation. That said, $k - term$ **CNF**s are *PAC-learnable*.[7]

If you transform a $k - term$ **DNF** to a $k - term$ **CNF** ($k - term$ **DNF** $\subset$ $k - term$ **CNF**) then you can efficiently learn $k - term$ **DNF**s.

Here's an example reduction.

Theorem 4 ($R_{DNF} \leq R_{DFA}$)**.** [4]

Proof. Let $r \in R_{DNF}$ encode a DNF expression of n variables. Then a possible setting of the variables that would satisfy the DNF will be a word with length $n.$ Denote s to be an upper bound on the number of terms in a representation r. Then for all possible assignments of the variables w, $f(w, s, n) = w^s$ (this is just the word w, s times). Since f has a finite length bound in s one can construct a DFA with $O(sn)$ states such that r is true $\iff$ our DFA accepts w^s. Our DFA has the following behaviour. For each $w \in w^s$ our DFA uses $O(n)$ states to check if the term is satisfied. In this case a term is of the following form $(x_1 \wedge x_2 \cdots \wedge x_n)$. If it isn't, then our DFA moves on to the next copy of the word and checks the next set of states to test whether the next term is satisfied. If a single term is satisfied, then our DFA accepts. If no terms are satisfied then our DFA rejects. □

There does however exist some problems that are definitely intractable. They can be thought of as similar to NP-complete problems. These include

1. Convex Polytope Intersection
2. Horn clause Consistency
3. Augmented CFG emptiness

The 'proof' that these problems are most likely not learnable is that if we can effectively learn these, then we can reverse certain one-way functions that are assumed to be non-reversible.

The important take away from this problem is that because something is not immediately *PAC-learnable* in its current form doesn't mean that is definitely the case. Representation matters. This is of course unless its *learnability* leads to reversing one-way functions ...

[4] R_{DNF} refers to representations of problems in the disjunctive normal form. R_{DFA} refers to representations as DFAs

3.2.2 Compression

This leads into an interesting tangent that I'd like to mention. There exists the Hutter Prize for €500,00(which is still ongoing) which is the the following - efficiently compress the first 1,000,000,000 words of English Wikipedia. Why this is such an exciting challenge is because its based on the assumption that artificial intelligence $\sim$ compression. The idea being that intelligence is in a way just a form of really good data compression. If anything I think Learning Reductions support this idea incredibly well. Something that is efficiently learnable might not be immediately efficiently learnable in the form its initially represented. Intelligence is the way we bridge the gap.

What the absolute best solution is is still to be known. This is because Kolmogorov Complexity is not computable!

PAC vs Bayes

It's important to note that PAC is a distribution-free model as opposed to Bayes. What's exciting about that is that we don't need to have any priors about the data we're learning. This unto itself opens up a whole can of philosophical worms about knowledge and priors. All I want to say is that when we think about learning, especially learning something specific, we often have some assumptions about that thing. When we're working with learning algorithms there is no way for us to introduce priors. [5] We have a prior that the sun rises during the day and the moon comes up at night. When we're thinking about the world all of these things tacitly affect our thinking. How do you pass this on to a machine?

In addition, there is a tradeoff of distributions - that is Bayes is a distribution over hypothesises while PAC is a distribution over the sample space. When might we prefer one over the other?

4 Final Thoughts

> The greatness of a piece of writing is not what contents are in it but instead, what quotes it chooses to present at the beginning of chapters.
> - Me

Note the above quote is especially applicable to 251 assignments . . .

I was going to write here about neuromorphic computing and how being probably right is like having an incomplete axiomatic system. There's a lot more here to explore. Nonetheless I hope the reader came away with some interesting ideas. The highlights in my opinion are the following:

1. Representation matters for efficient learning.

2. The dimensionality of the hypothesis space relates to the complexity of the learning.

[5] I say no way but in machine learning previously trained weights can often be used as a starting point for image classification problems. This is based on the assumption that most image classification problems, although the domain may differ, use common techniques like segmentation, filtering, etc.

5 Check Your Understanding

1. What is the relationship between the empirical error and the Generalization error? How are they different?
2. What affect does the VCDim have on learnability?
3. How are learning reductions different from complexity reductions?

References

[1] Scott Aaronson. *Quantum Computing Since Democritus*. Cambridge University Press, 2018.

[2] Anselm Blumer et al. “Learnability and the Vapnik-Chervonenkis dimension”. In: *Journal of the ACM (JACM)* 36.4 (1989), pp. 929–965.

[3] Daniel Cohnitz and Marcus Rossberg. “Nelson Goodman”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2020. Metaphysics Research Lab, Stanford University, 2020.

[4] Nelson Goodman. *Fact, Fiction and Forecast*. Harvard University Press, 1983.

[5] Leah Henderson. “The Problem of Induction”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2020. Metaphysics Research Lab, Stanford University, 2020.

[6] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2018.

[7] Leonard Pitt and Manfred K Warmuth. “Prediction-preserving reducibility”. In: *Journal of Computer and System Sciences* 41.3 (1990), pp. 430–467.

[8] Leslie G Valiant. “A theory of the learnable”. In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142.