Principles of learning systems

Zach Furman

A businessman may observe stock prices over time and then attempt to infer a model of this process in order to predict the market. A parent may return home from work to discover a chair propped against the refrigerator with the cookie jar on top a little emptier. Whether we are a detective trying to catch a thief, a scientist trying to discover a new physical law, or a businessman attempting to understand a recent change in demand, we are all in the process of collecting information and trying to infer the underlying causes.

- Shane Legg

Overview

- The goal: “Make artificial general intelligence safe and aligned”

Overview

- The goal: “Make artificial general intelligence safe and aligned”
- Problem: this seems like a massively underspecified problem, even without the issue of what we mean by “safe” and "aligned". How can we make a safety plan for something we don’t know how to build?

Overview

- The goal: “Make artificial general intelligence safe and aligned”
- Problem: this seems like a massively underspecified problem, even without the issue of what we mean by “safe” and "aligned". How can we make a safety plan for something we don’t know how to build?
- Assumption: what if AGI is a *learning* system? What can we say about it then?

Outline

- What is a learning machine? Why care about learning machines? What's our scope?
- The approximation problem
- The generalization problem
- A solution: Solomonoff induction
- The optimization problem
- Outlook and implications

Background

- GOFAI vs connectionism
- Bitter lesson

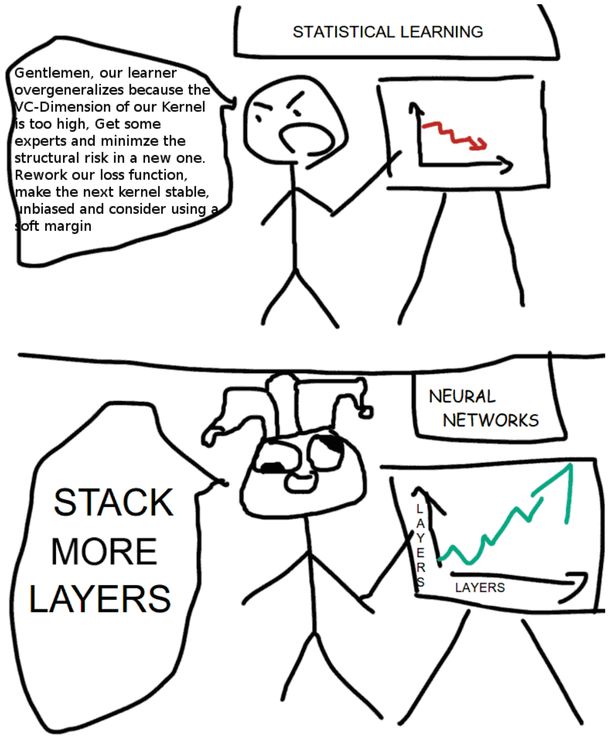


Image source: https://x.com/ilyasut/status/1492969018759602176

Scope

- Today we study learning machines at a theoretical level: what are their limits, what principles constrain how they must work, what can we expect from them?
- We focus on the kinds of learning systems which can plausibly lead to *general* intelligence - general-purpose, universally applicable, etc
- We draw on the fields of *algorithmic information theory* and *statistical learning theory*

Scope

- We focus specifically on learning machines for *prediction* (supervised/self-supervised learning). Goal-directed learning (RL, agents) comes later in the course
    - But note that good prediction and world models are a prerequisite for achieving goals effectively
    - Alignment issues largely come from goal-directed learning; our analysis here should be viewed in the spirit of characterizing AGI rather than characterizing alignment (yet)
- Especially keep in mind the scenario of *sequential prediction*

What is a learning machine?

- Roughly: a system that takes in data about the world and uses it to make predictions.
- First, pause: why is this even possible? If I've seen the sun rise for the past 100 days, what justifies predicting that it will rise for the 101st?
  - We must assume that the world obeys some consistent rules: it is generated from (or can be well-predicted by) a computable data-generating process. That is, we must be able to model the world and select between different models of the world (a *hypothesis class*) given observations
- Hypothesis class can be discrete (Solomonoff induction) or continuous (statistics, deep learning)
- Hypotheses can be chosen by many mechanisms: Bayes, MLE, SGD, etc

The approximation problem

- The choice of hypothesis class is crucial: nothing guarantees that there exists *any* hypothesis in the class which is a good approximation of the true hypothesis
  - Linear regression can only learn linear data
- The statistician's view: each learning machine is a specific tool for a specific task, requiring a human to pick the right tool
- But we want to focus on AGI - can we build a *universal* learning machine?

The approximation problem

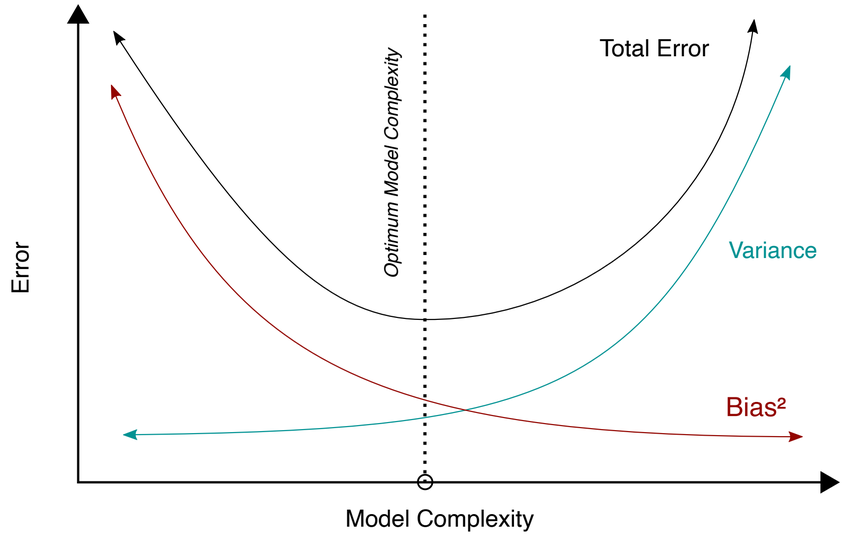
- Can we build a universal hypothesis class?
  - Yes! Consider: every hypothesis we could ever in principle consider must be *computable*. There are a countable number of Turing machines, and they can form a hypothesis class in themselves
  - Church-Turing thesis and the physical Church-Turing thesis
  - Aside: “universal approximation” bad

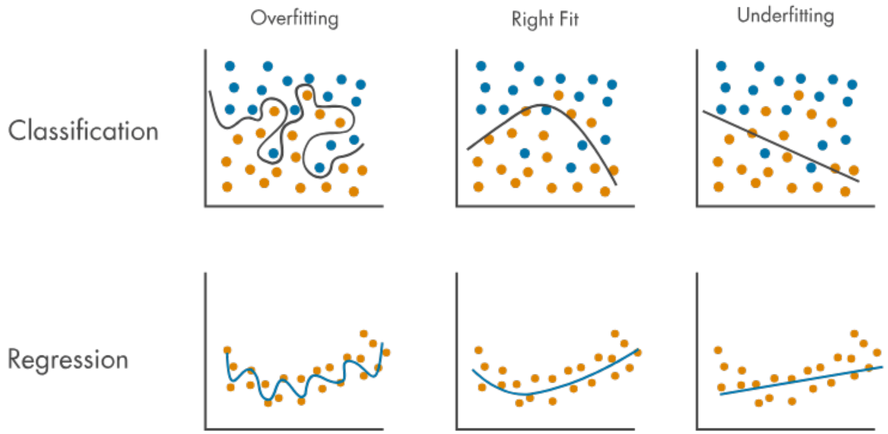
The generalization problem

- Unfortunately, making the hypothesis class bigger doesn't (a priori) come for free
- It takes log(n) bits to specify a hypothesis in a finite hypothesis class of size n, assuming I have a uniform prior over hypotheses
    - Therefore, for a given amount of information from observations, a larger hypothesis class will mean a larger chance that I pick a bad hypothesis and generalize poorly (come to regret my choice) with more observations
- Even worse, a larger hypothesis class can reduce the amount of information from observations, because I need to make finer-grained distinctions in order to tell different hypotheses apart (it's easier to decide "snowing vs sunny" rather than "10 snowflakes a second vs 11 vs 12, etc")
- (Note this notion of "generalization" refers to in-distribution generalization, not out-of-distribution generalization)

The generalization problem

- The (classical) statistician's view: this is the "bias-variance [or approximation-generalization] tradeoff," an unavoidable tradeoff between the expressivity of your hypothesis class and its ability to generalize to more data
- *Is this tradeoff really unavoidable?* If it was, it would suggest that truly universal learning machines are impossible
- Spoiler: it is avoidable, at least in principle. What if we didn't use a *uniform* prior over hypotheses?





Occam’s Razor

“Plurality should not be posited without necessity”
- William of Ockham

Continue the sequence: 1, 2, 4, 6, 8, 16

Occam’s Razor

“Plurality should not be posited without necessity”
- William of Ockham

Continue the sequence: 1, 2, 4, 6, 8, 16, **31**

$$\frac{1}{24}(n^4 - 6n^3 + 23n^2 - 18n + 24)$$

Occam's Razor

"Plurality should not be posited without necessity"
- William of Ockham

Continue the sequence: 1, 2, 4, 6, 8, 16, ~~31~~ **32**

$$\cancel{\frac{1}{24}(n^4 - 6n^3 + 23n^2 - 18n + 24)}$$

$$2^n$$

We should prefer this one

Kolmogorov complexity

- How can we make this notion of simplicity precise and objective?
- Intuitively, among competing hypotheses, the one with the fewest assumptions should be selected; we should prefer a hypothesis with shorter description length to a longer one which has more “burdensome details.”
- Kolmogorov complexity: the complexity of a string is the length of the shortest possible Turing machine (more precisely, UTM code) which can generate it
- Note that the definition of Kolmogorov complexity depends on a choice of UTM, but one can prove that this can at most lead to a constant difference in complexity

Short description (**simple**)

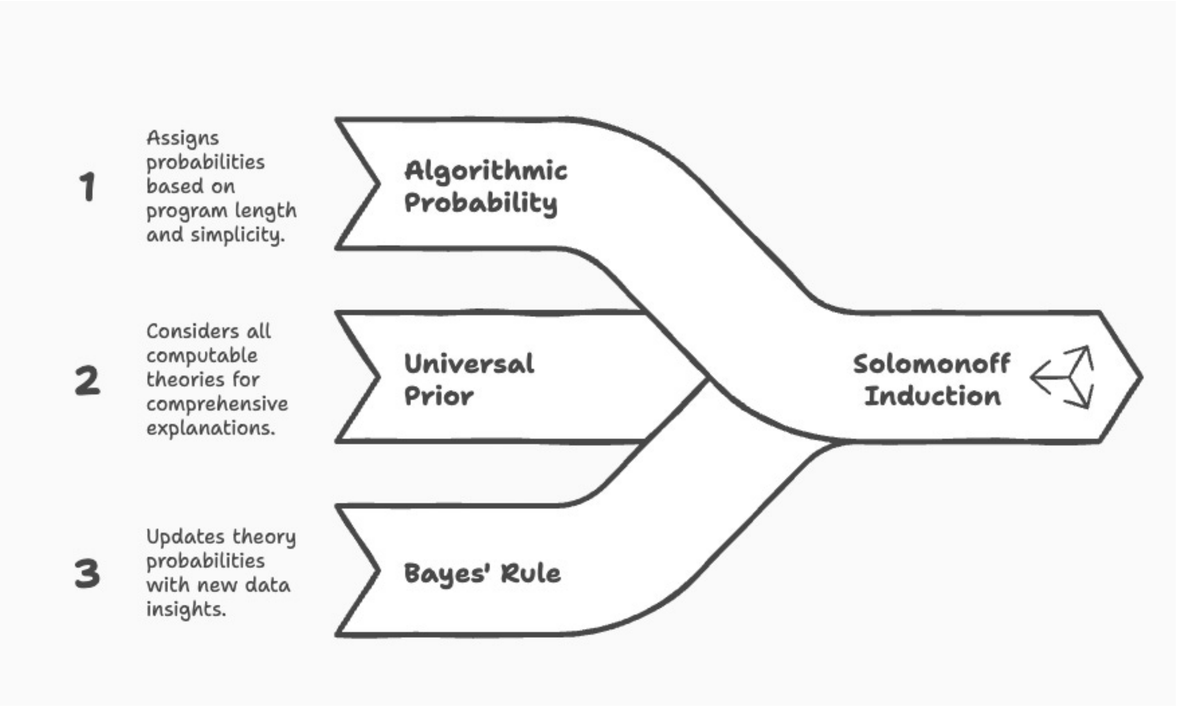
```
0101010101010101010101010101…
                ↓
for i in 1 to 50: print
            “01”
```

Long description (**complex**)

```
1101001011101011001110 10…
                ↓
            print
“1101001011101011001110 10…”
```

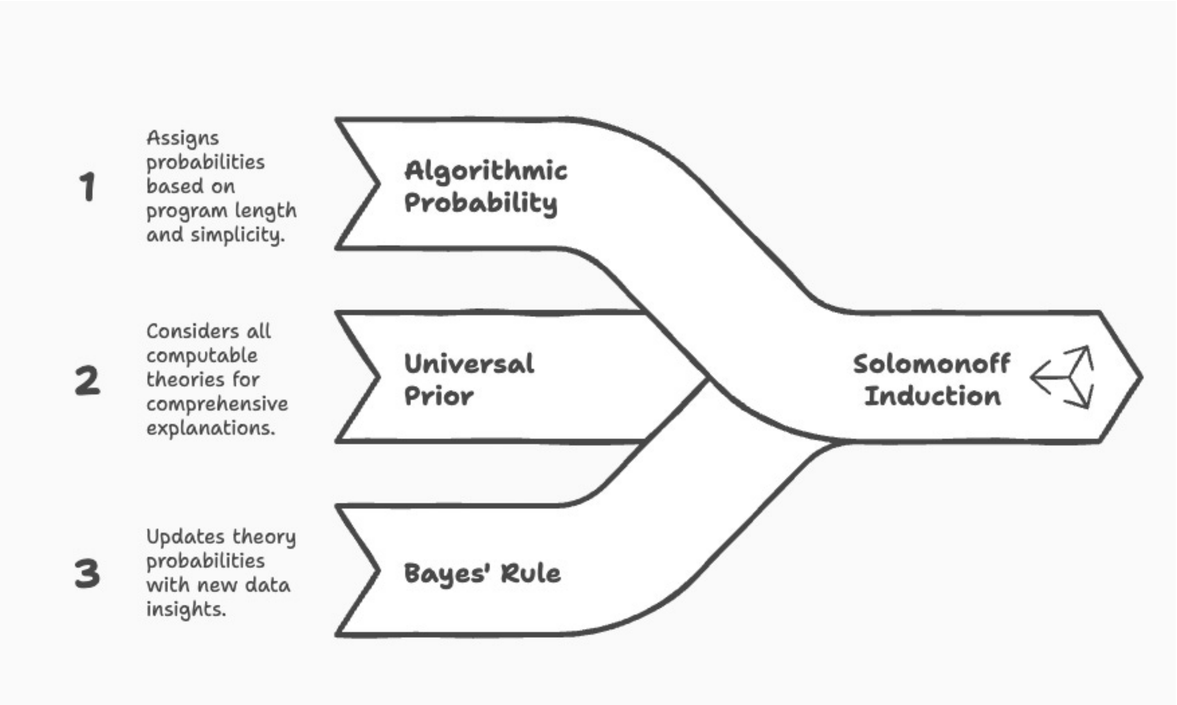
Solomonoff induction

Definition: Consider the hypothesis class consisting of all Turing machines. Put a prior semimeasure $\mu(p)$ on this space (the *Solomonoff prior*), equal to $2^{-|p|}$ where $|p|$ is its length. *Solomonoff induction* is defined by predicting the probability of future observations conditional on past observations according to Bayes rule with prior $\mu(p)$.



Solomonoff induction avoids problems

- We avoid the approximation problem: our hypothesis class contains all computable hypotheses
- We avoid the generalization problem: we can enlarge our hypothesis class arbitrarily large without significantly hurting generalization by making the prior probability of new hypotheses much lower than existing hypotheses



Solomonoff induction is optimal (*)

- For any computable predictor q, Solomonoff induction's total prediction error exceeds that of q by a most a constant proportional to $K(q)$
  - If *any* computable method predicts the sequence well, Solomonoff induction will learn to predict nearly as well, with overhead depending only on how complex that method is
- However, note this argument only works well if we assume the true hypothesis has relatively low Kolmogorov complexity, which is a bit circular. One can alternatively justify the method by assuming the true hypothesis was generated by a uniform distribution over UTM codes

The optimization problem

- Unfortunately, Solomonoff induction is incomputable
- This is somewhat of a technical issue; more important is that even approximations like speed induction (penalizing runtime) take exponential time (Filan, Leike, & Hutter 2016)
- At a low level, the problem is that Bayesian inference is too slow - it must be replaced with something more tractable
    - But this is easier said than done, and it can introduce new problems. Local search methods like gradient descent are much faster, but we have no *a priori* reason to think they won't get stuck in local minima

The optimization problem

- At a more abstract level, the problem is that *any* method which promises uniform (over the hypothesis class) efficiency must necessarily run slower than polynomial time, under standard cryptographic assumptions
- A *pseudorandom generator (PRG)* is a deterministic, polynomial-time function $G : \{0,1\}^{\ell} \rightarrow \{0,1\}^{m}$ that stretches a short, truly random seed of length ℓ into a longer string of length m, in such a way that the output is (up to some arbitrarily small ϵ) indistinguishable from uniform randomness by any polynomial-time function
  - Note: PRGs exist iff one-way functions exist. Existence of PRGs would imply P =/= NP, but the converse is not necessarily true

The optimization problem

- Call a learning method *uniformly efficient* if: 1. when the true hypothesis is realizable by the method, it will eventually be found, and 2. it will be found in polynomial time. Then if PRGs exist, no uniformly efficient learning method over Turing machines can exist.
    - Sketch: if such a method did exist, then one could distinguish PRG output from true randomness in polynomial time by applying the learning method and observing the final output of the learning method. This contradicts the definition of a PRG, hence such a learning method could not exist.
- If we ask for an *efficient* and *universal* learning algorithm, we must fail to (quickly) learn *even hypotheses that lie in the hypothesis class of our model*
- Note that these arguments also apply to the difficulty of training neural networks, since they can realize pseudorandom generators

Outlook and implications

- We discussed three problems that learning systems must overcome in order to perform well: approximation, generalization, and optimization
- Solomonoff induction is able to optimally solve the first two problems, at the cost of being egregiously unable to solve the third problem. Nevertheless it provides a basis for the limitations of learning machines, and a prior for thinking about what universal learning algorithms might conceivably look like
- It remains open why deep neural networks empirically seem to overcome these three issues (is the mechanism for approximation/generalization similar to Solomonoff induction, for instance?) We will discuss this more on day A.3

Questions?