Optimal Inductive Inference & its Approximations

Nikos Nikolaou

Part I: Solomonoff Induction

Foreword

- ‘*... Solomonoff induction makes use of concepts and results from* ***computer science****,* ***statistics****,* ***information theory****, and* ***philosophy*** *[...] Unfortunately this means that* ***a high level of technical knowledge from these various disciplines is necessary*** *to fully understand its technical content. This has* ***restricted a deep understanding of the concept*** *to a fairly small proportion of academia which has hindered its discussion and hence progress*’

-Marcus Hutter

Introduction

Types of Reasoning

Deductive

- Drawing valid conclusions from assumed/given premise (reasoning about the <u>known</u>)

- Mathematical Proofs
- Formal Systems (Logic)

Inductive

- Drawing ‘the best’ conclusion from a set of observations (reasoning about the <u>unknown</u>)

- **Learning rules from examples**
- Scientific Method

Transductive

- Drawing ‘the best’ conclusion from observed, specific (training) cases to specific (test) cases

- Learning properties of objects from examples

Induction

- Given data O
- Discover process H that generated O

(Can then use H to make predictions O’)

Learning / Statistical Inference

- Given data O
- Find hypothesis (model) H that explains O

(Can then use H to make new predictions O')

Solomonoff Induction

- A recipe for performing inference (induction)
- Basic Ingredients:
  – Epicurean Principle
  – Occam’s Razor
  – Bayes Theorem
  – Universal Turing Machines
  – Algorithmic Information Theory

The Ingredients

Running Example: The Case of the Missing Cookie

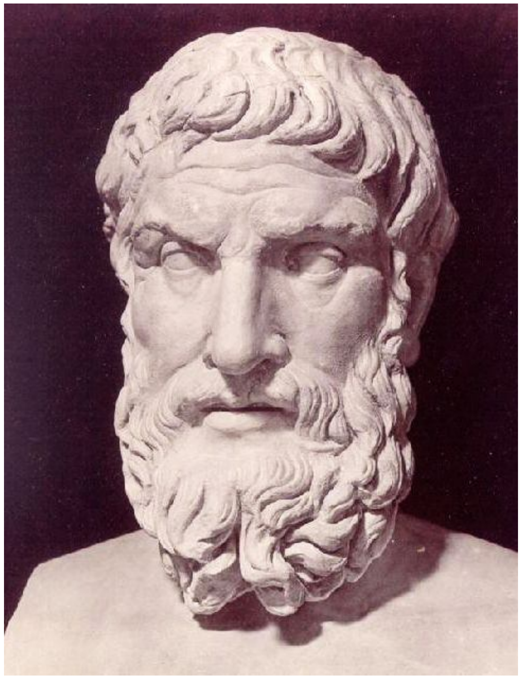
- You just baked cookies & left them out to cool
- Your 8yr old child was in the kitchen with you
- You turn your back for a few seconds & then this is what you see:

- What happened?

The Epicurean Principle

- 'If several theories are consistent with the observed data, retain them all'.

Consider all hypotheses that explain the data



Epicurus (Ἐπίκουρος)
(c. 341–270 BC)

Epicurus on ‘the Missing Cookie’

- Hypotheses consistent with your data:
  - The child ate it
  - You ate it & forgot it
  - Someone else came in, ate it & left unnoticed
  - The missing cookie was never there to start with
  - Your entire ‘life’ is a figment of your imagination, in fact you have been in a coma for the last 10 years
  - Aliens, obviously

  ⋮

Occam's (Ockham's) Razor

- ‘Among competing hypotheses that predict equally well, the one with the fewest assumptions should be selected’.

Explanatory power being equal, favor simpler hypotheses

William of Ockham
(c. 1287–1347)

Ockham on ‘the Missing Cookie’

– The child ate it

– ~~You ate it & forgot it~~

– ~~Someone else came in, ate it & left unnoticed~~

– ~~The missing cooki~~

– ~~Your entire ‘life’ is~~
~~fact you have bee~~

– ~~Aliens, obviously~~

⋮

Bayes' Theorem

- $$P(H|O) = \frac{P(O|H)P(H)}{P(O)}$$

$$\underbrace{P(H|O)}_{\text{posterior}} = \frac{\overbrace{P(O|H)}^{\text{likelihood}}\overbrace{P(H)}^{\text{prior}}}{P(O)}$$

Transform prior distribution
to posterior based on evidence

Thomas Bayes
(c. 1701 – 1761)

Bayes on ‘the Missing Cookie’

– The child ate it
– You ate it & forgot it
– Someone else came in, ate it & left unnoticed
– The missing cookie was never there to start with
– Your entire ‘life’ is a figment of your imagination, in fact you have been in a coma for the last 10 years
– Aliens, obviously

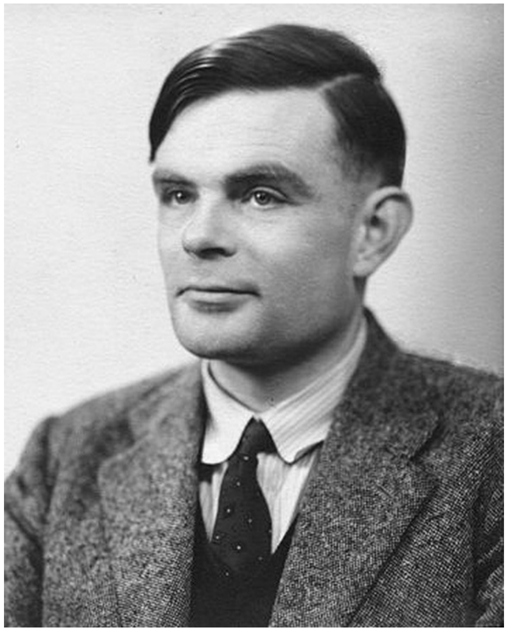
⋮

Evidence supports **all hypotheses** H_i, but **priors** P(H_i) **differ**, so P(H_i|O) differ

Universal Turing Machine

- A universal model of computation

A way to formalize the concept of 'algorithm'



Alan Mathison Turing
(1912 – 1954)

Information Theory

- A quantitative study of information

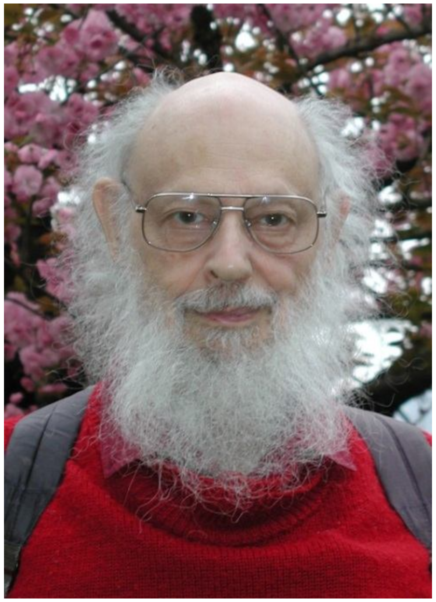
A way to formalize the concept of 'information'

Claude Elwood Shannon
(1916 – 2001)

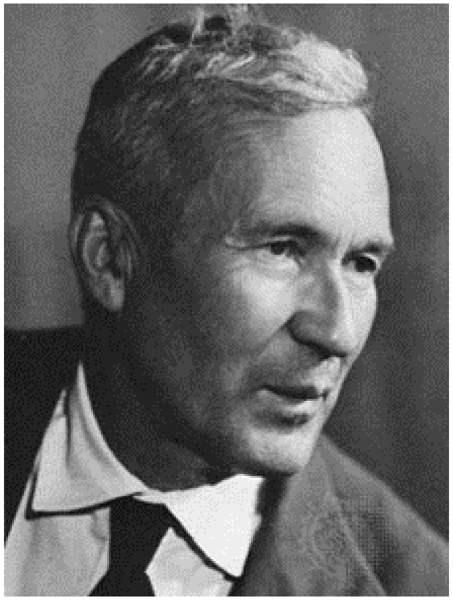
Algorithmic Information Theory

- Relate computation, information & randomness

A formalization of the concept of ‘complexity’



Ray Solomonoff
(1926 –2009)



Andrey Nikolaevich Kolmogorov
(1903 –1987)

Gregory John Chaitin

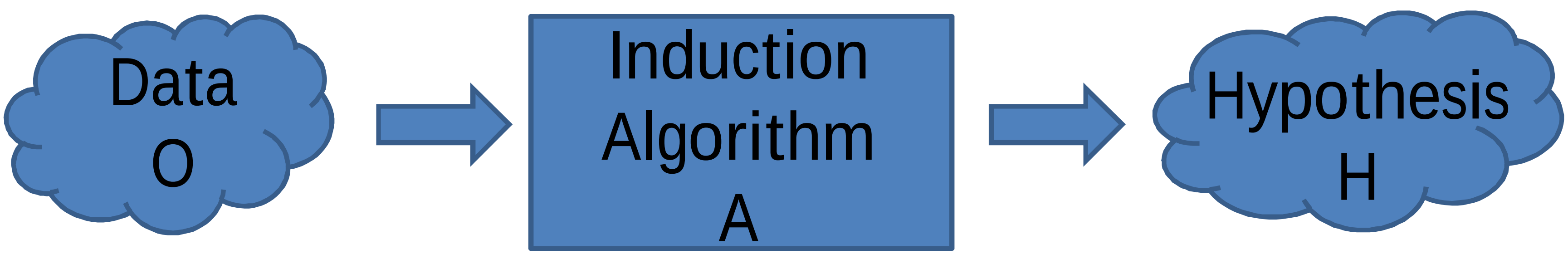
Solomonoff Induction

The Problem

- Given data O
- Discover process H that generated O

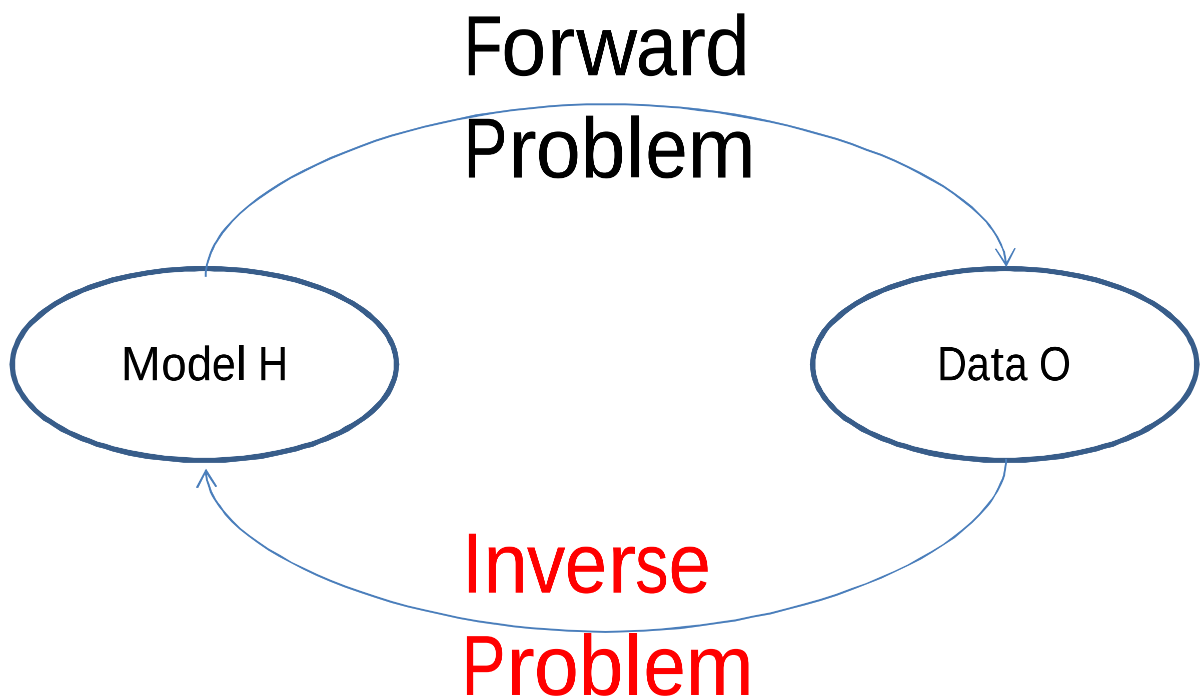
Induction

- Need an induction algorithm A :



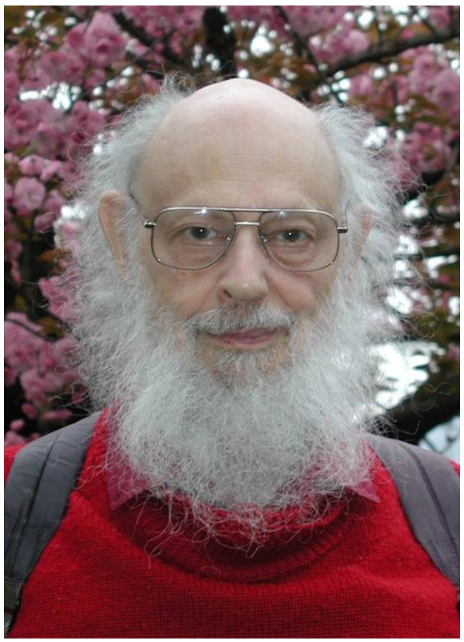
Spoiler: Induction is Ill-posed

- ‘Inverse problem’: Inferring model (hypothesis) from data (set of observations)
- Data can be consistent with multiple hypotheses



Solomonoff Induction

Solomonoff combined the Epicurean Principle & Occam's Razor in a probabilistic way according to

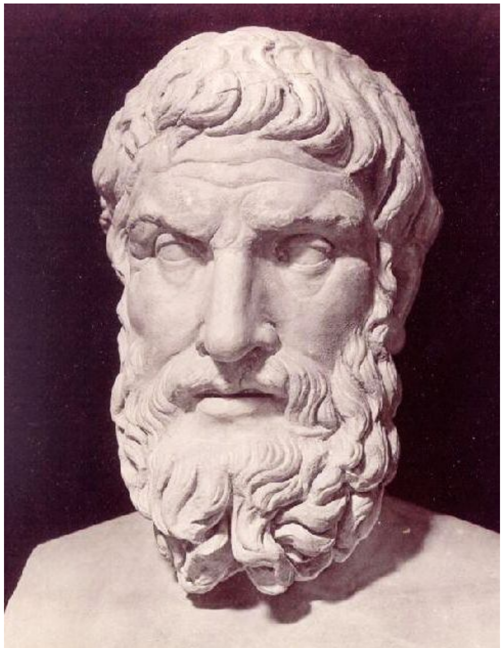


Bayes Theorem, used Turing Machines to represent hypotheses & Algorithmic Information Theory to quantify their complexity.

Let' s follow his reasoning...

Epicurean Principle

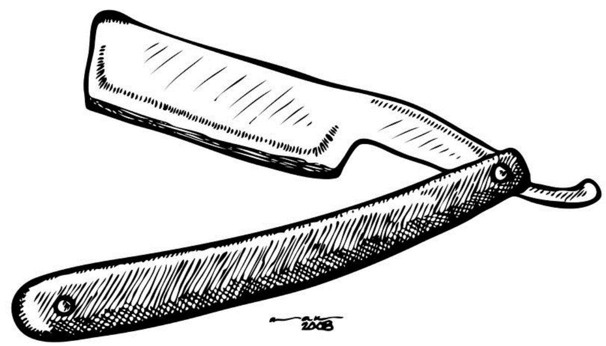
For starters, **all hypotheses** that are **consistent** with the data **must be examined** as possibilities.



Once you eliminate the impossible...

Occam’s Razor

But we should **drop complex hypotheses** once we find simpler equally explanatory ones.



Bayes' Theorem

We could instead assign a **prior probability** to each hypothesis, deeming more complex ones less likely.

$$P(H_i|O) = \frac{P(O|H_i)P(H_i)}{P(O)},$$

with $P(H_i)$ **lower for 'more complex'** hypotheses H_i (as we will see)

The Problem of Priors

- Why not calculate priors $P(H_i)$ based on data?
  - If we have data, can compute them
  - If we don' t, we can' t; so assign them based on the principle that **'simpler' hypotheses are more likely** (we will see how this is justified)

- Next goal: **Define 'simple' / 'complex'**... but **first** need to **choose a 'language'** to represent O & H_i

Representing Data

- Represent information in **binary**
  - 2-letter alphabet {0, 1} the smallest one that can communicate a difference
  - can encode all information as binary strings (?)

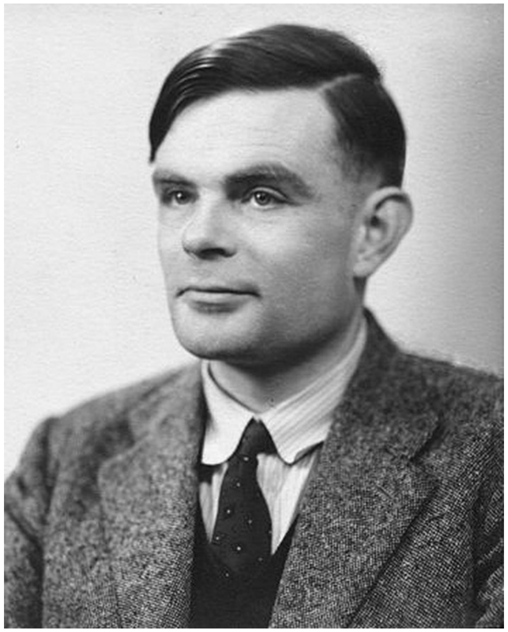
- Data O: a binary string

1101…1001

Representing Hypotheses

H_i: a **process** that generates data, an **algorithm**.

Turing proposed a **universal algorithm model**, the **Turing Machine (TM)**.



Church-Turing Thesis: TMs truly capture the idea of ‘algorithm’

All attempts to formalize the intuitive idea of ‘algorithm’ or ‘process’ have proven to be at most as powerful as TMs

(3-Tape) Turing Machine

- Input sequence : 1101…1001
- Work sequence: 0110…0101
- Output Sequence: 1011…1000
- Equivalent to ‘standard’ (single tape) TMs; more intuitive for what we want to show here

(3-Tape) Turing Machine

- Every TM has a finite number of states (‘rules’)
- Starts at a state:
  - Input sequence : 0101...0111
  - Work sequence: 0000...0000
  - Output Sequence: 0000...0000

(3-Tape) Turing Machine

- Rules for 1st state: read input & work sequences; depending on the values perform certain actions:
  1. Feed the input tape (optional)
  2. Write 0 or 1 on the work tape
  3. Move the work tape left or right
  4. Write 0 or 1 on output tape
  5. Feed the output tape (optional)
- After that, rules specify next state and so on...

(3-Tape) Turing Machine

- A TM has a **finite number of states (‘rules’)**
- **Rules are fixed**; only what is written on the tapes (‘memory’) & current state are changing
- Yet with such simple, finite rules we **can simulate every algorithm**

Universal Turing Machine (1)

- Turing showed that a specific set of ‘rules’ (UTM) could simulate all other sets of ‘rules’ (TMs)
- Can simulate another TM by giving the UTM a ‘**compiler**’ binary sequence
- Such a sequence exists for every TM
- UTM Input sequence : $\underbrace{\boxed{10\ldots1}}_{\text{Compiler}}\underbrace{\boxed{11011\ldots1001}}_{\text{TM Input}}$

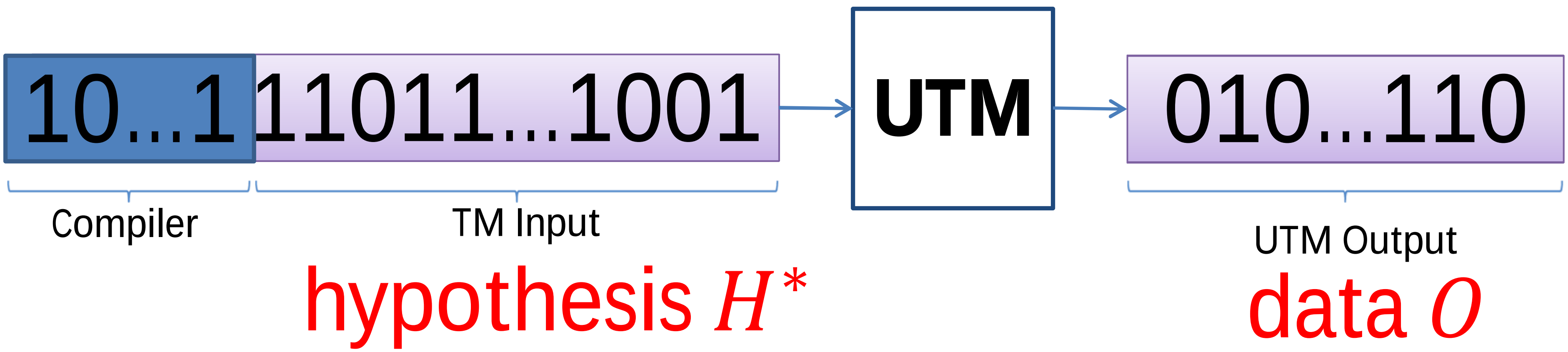
Universal Turing Machine (2)

- Hypotheses are processes, i.e. algorithms*
- Algorithms are represented by TMs
- TMs are represented as binary input sequences to the UTM, so...
- Hypotheses H_i: are represented as binary input sequences of UTMs

*This is the only assumption of Solomonoff Induction

Solomonoff Induction

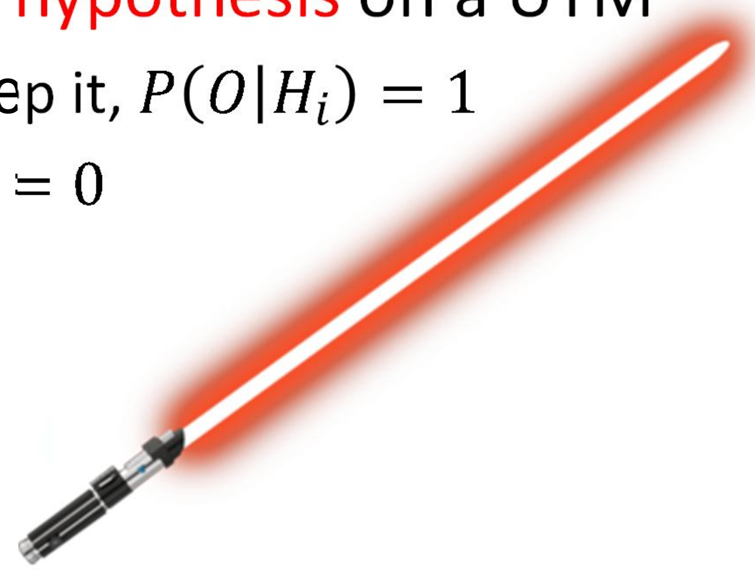
- So, a UTM will output the data O if you give it a correct hypothesis H^* as input



- The set of all possible inputs to the UTM is the set of all possible hypotheses $\{H_i\}$

Solomonoff’ s Lightsaber

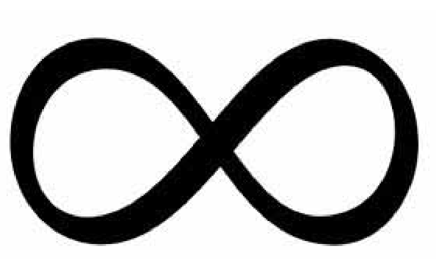
- Given data O
- Can find **all** potential hypotheses H_i that explain O by
  - Running every possible hypothesis on a UTM
    - If output matches O, keep it, $P(O|H_i) = 1$
    - Else discard it, $P(O|H_i) = 0$

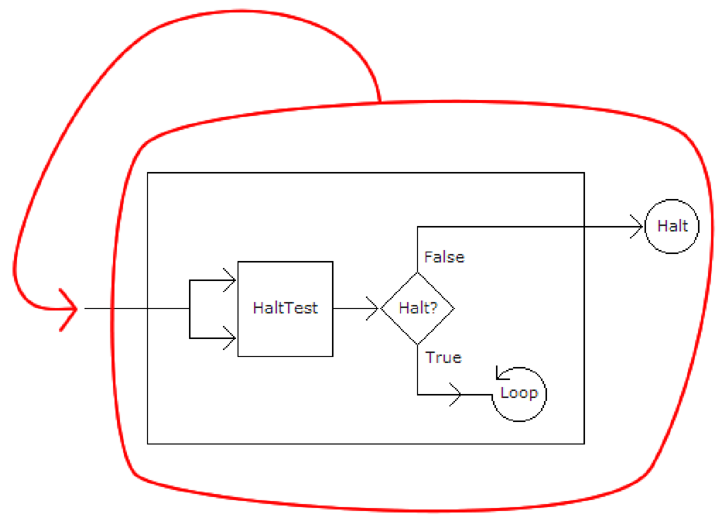


Nice... but Intractable

- Solomonoff Induction is **intractable**...
  - '... **every possible hypothesis** ...': they are **infinite**
  - **Halting problem**: some hypotheses would **run forever** w/o producing the output & we **can't prove they won't terminate**

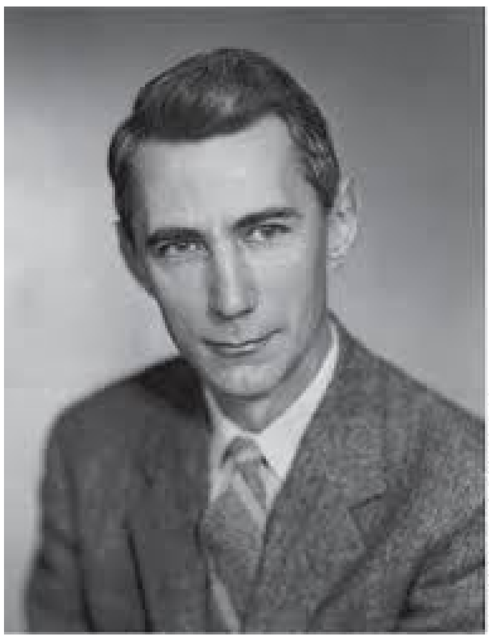
- The problem of induction is ill-posed...





Defining Simplicity / Complexity (1)

Entropy: A measure for quantifying **uncertainty** / unpredictability / surprise / (lack of) information



A message M with low entropy -> M is predictable -> M has low **complexity** -> is **easy to compress**

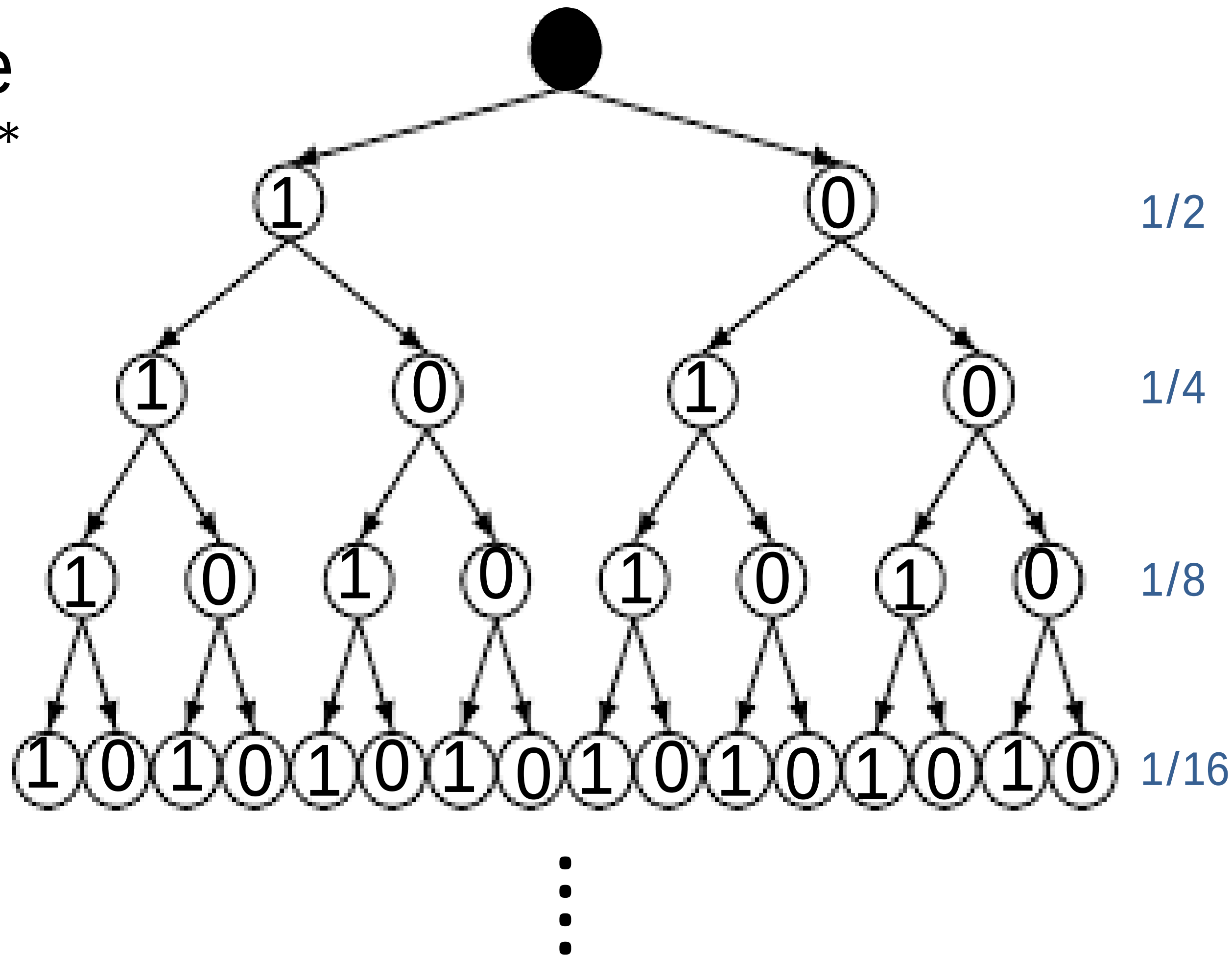
e.g. 0101010101 vs. 1001110100

5x‘01’

Here we will discuss the related notion of **Algorithmic Entropy**...

Defining Simplicity / Complexity (2)

- Assume* true hypothesis H^* produced by fair coin-flips
- As length of sequence grows, its probability diminishes

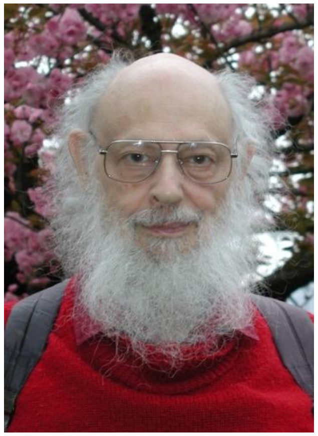
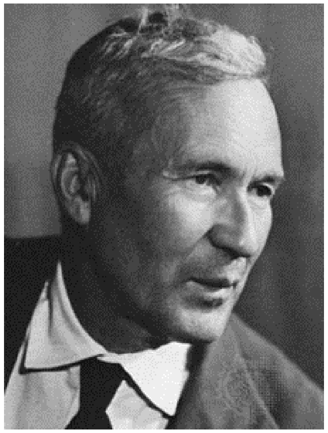


Defining Simplicity / Complexity (3)

- A binary sequence that is one bit shorter is twice as likely to be the true hypothesis H^*
  - **Shorter sequences (hypotheses) more likely**
- Kolmogorov Complexity (Algorithmic Entropy):

$$K(H_i) = \{\text{Length of shortest description of } H_i\},$$

Remember, 'description of H_i' : binary input to UTM

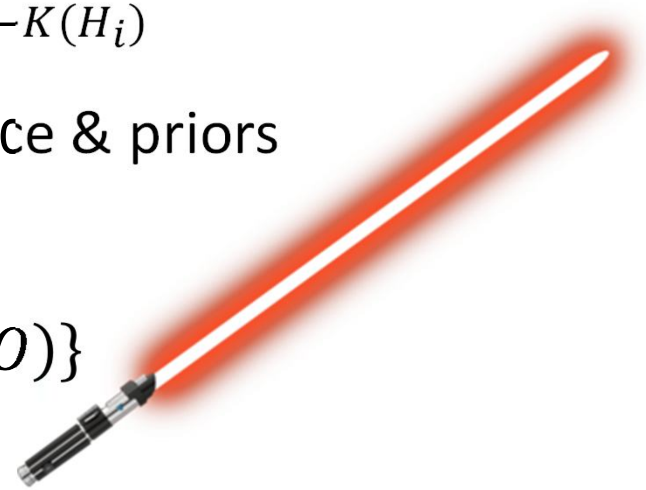




Back to the Priors

- Quantified simplicity by Kolmogorov Complexity:
  $K(H_i)$ = {Length of shortest description of H_i}
- A hypothesis that is one bit shorter is twice as likely to be the true hypothesis H^*
- So priors must be:
  $$P(H_i) = 2^{-K(H_i)}$$
- Priors of hypotheses H_i reflect principle that 'simpler' hypotheses are more likely

Putting it All Together

- **Given observations O, find hypothesis H^* that produced them**
- Represent O as binary sequence
- Represent hypotheses H_i as binary input sequences of a UTM
- Set $P(O|H_i) = 1$ if H_i consistent with data, i.e. if fed as input to the UTM, will output O, $P(O|H_i) = 0$ for the rest
- Find Kolmogorov Complexity of hypotheses:
  $$K(H_i) = \{\text{Length of shortest description of } H_i\}$$
- Prior of each hypothesis is $P(H_i) = 2^{-K(H_i)}$
- Use Bayes Theorem to combine evidence & priors
  $$P(H_i|O) = \frac{P(O|H_i)P(H_i)}{P(O)}$$
- Select H^*: $P(H^*|O) = \underset{H_i}{argmax}\{P(H_i|O)\}$



Optimal Induction is Intractable

- Solomonoff solved the problem of formalizing optimal inductive inference…
- … but the problem is shown to be intractable
- So we can at best approximate it…

Approximations

- Give **higher prior** to hypotheses H_i that can be **quickly computed** (‘**Levin Complexity**’ rather than ‘Kolmogorov Complexity’)

Leonid Anatolievich Levin

Jürgen Schmidhuber

- Randomly generate a set of hypotheses to test using **Monte Carlo techniques**
- **Restrict hypothesis space**

Implementations

- **Universal artificial intelligence (AIXI)**

- Solomonoff Induction + Decision Theory

Marcus Hutter

Criticisms

- Which UTM? (Infinitely many…)
  - Length of each H_i as a binary sequence will depend on this choice thus the priors assigned to each H_i …
  - … But only up to a constant factor (compiler to translate from UTM to UTM’), i.e. independent of H_i
- True hypothesis H^* might be intractable
  - No algorithm can find H^*… can at best converge to it
- Can everything be represented in binary?

End of Part I

Preview of Part II

- Philosophical problems with induction
- Optimal induction intractable, yet **learning feasible,** even **efficient**...
- We can have **guarantees on induction**!
- By making **assumptions** & settling for **approximations**
- How we do so in ML (**learning theory** elements)

Thank you

Part II: Efficient Inductive Reasoning

Review of Part I

- Solomonoff Induction: formalization of optimal inductive inference…

- … but we saw that the problem is intractable

- So we can at best approximate it

- First let's see why it is intractable, then how to approximate…

Induction in Philosophy

Problem of Induction (1)

When drawing general conclusions from a set of observations, we **either see all* observations, or some** of them**

Sextus Empiricus (Σέξτος Ἐμπειρικός)
(c. 160 – 210 AD)

***all (infinite): not possible**

****some: conclusions are not certain some other observation could falsify them** ‘black swans’)

Problem of Induction (2)

‘What is the foundation of all conclusions from experience?’

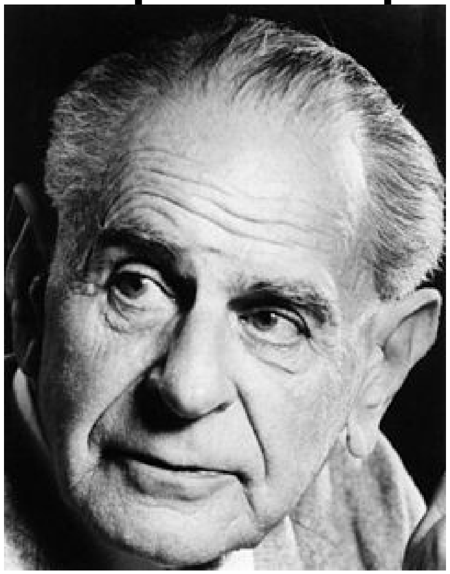
David Hume
(1711 – 1776)

We cannot hold that nature will continue to be uniform because it has been in the past.

(e.g. in machine learning:
no dataset shift, stationarity)

Problem of Induction (3)

A scientific idea can never be **proven** true; no matter how many observations seem to agree with it, it may still be wrong. On the other hand, a single counter-example can prove a theory forever false.

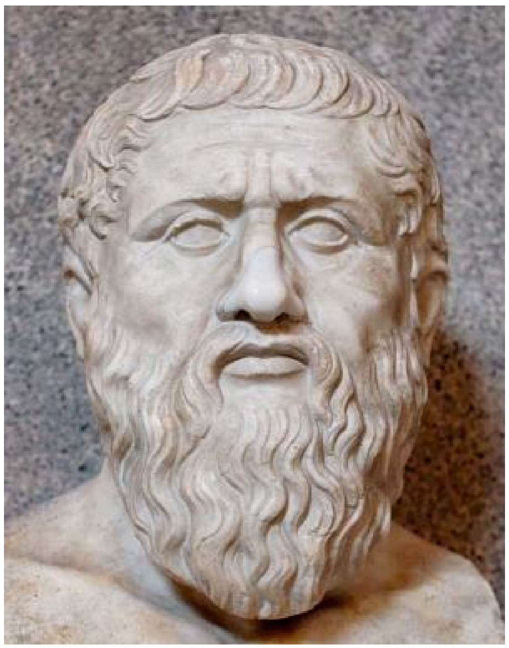


Sir Karl Raimund Popper
(1902 – 1994)

Observations are always in some sense incomplete (rem. 'black swans') & many hypotheses can be consistent with them (ill-posed)

Justified True Belief

Subject S knows that a proposition P is true iff:



Plato (Πλάτων)
(c. 427 – 348 BCE)

- P is true
- S believes that P is true, and
- S is justified in believing that P is true

Induction cannot be!
Yet, we use it all the time... successfully!

Induction in Science

The Scientific Method

1. Make observation O
2. Form hypothesis H that explains O

Induction

3. Conduct experiment E to test H
4. If results of E disconfirm H, return to (2) & form a hypothesis H' not yet used
   If results of E confirm H, provisionally accept H.

Deduction

Science is Based on Induction

- The scientific method heavily relies on inductive inference
- Note: also exhibits elements of what we call active learning in machine learning terminology

Induction & Learning

Learning vs. Optimization

- Learning means generalizing to unseen instances
- Not just **optimal fit on training data**...
- ... this is just **memorization**
- Induction is reasoning about the unknown, not the **known**

Memorization vs. Learning

Input	Output
1	2
4	8
5	10
6	12
9	18
11	22
17	34
20	40
22	44

- A **lookup table** tells us nothing about the output of input 2
- Learning the underlying rule $Output \ = 2 * Input$, does
- Can we guarantee that we can learn something from the training data?

Settling for Approximations

- Make **assumptions** about the data
- **Restrict hypothesis space** (drop Epicurean principle)
- Find a '**good enough**' hypothesis

Assumptions About the Data

- Assume **training set drawn from same distribution as test set** (**stationarity / no dataset shift** / ‘**uniformity of nature**’)
- Assume **independent & identically distributed (i.i.d.) data**: same probability distribution for each feature & all are mutually independent
- Similar datapoints should have similar properties (‘**smoothness**’)

Assumptions About Hypotheses

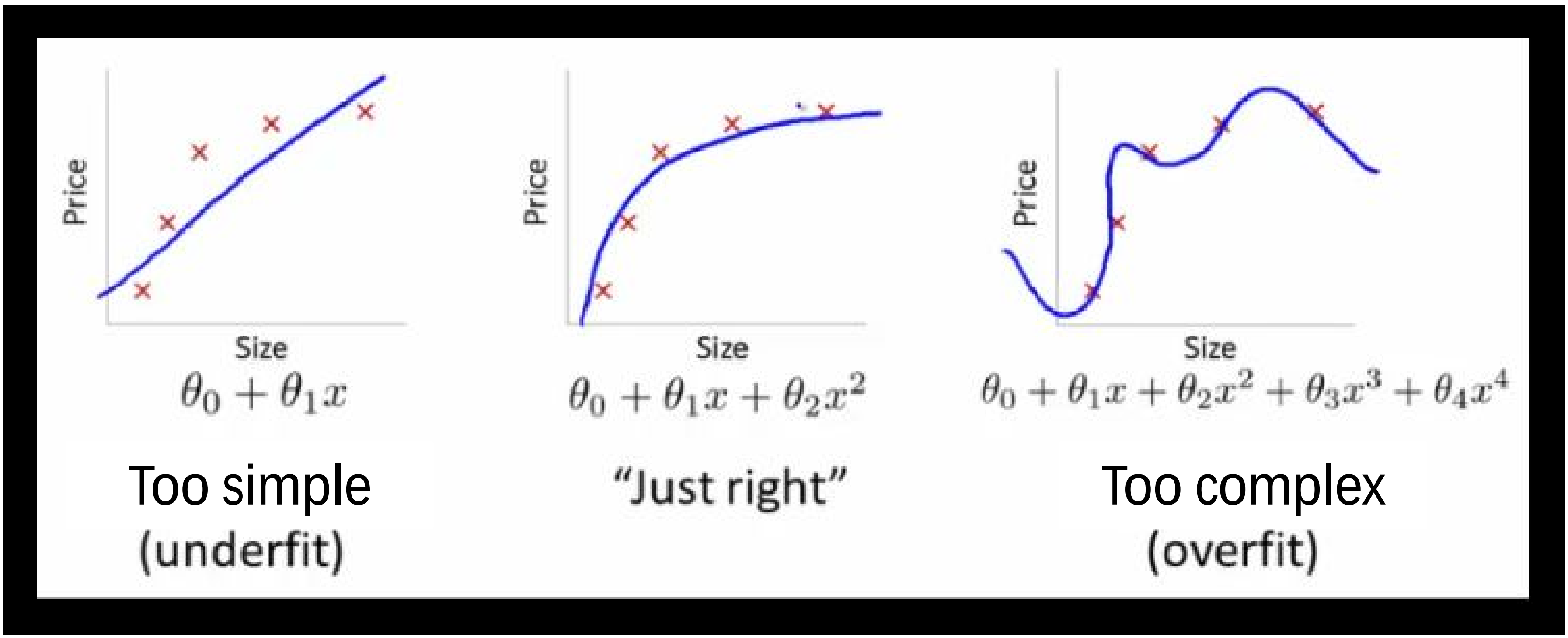
- Ignore / penalize complex hypotheses:
- Regularization (imposing more **constraints**)
  - Train s.t. both fit is optimized & model is simple
- Model selection (post-training)
  - Favor both goodness-of-fit & simplicity when comparing models

Overfitting vs. underfitting

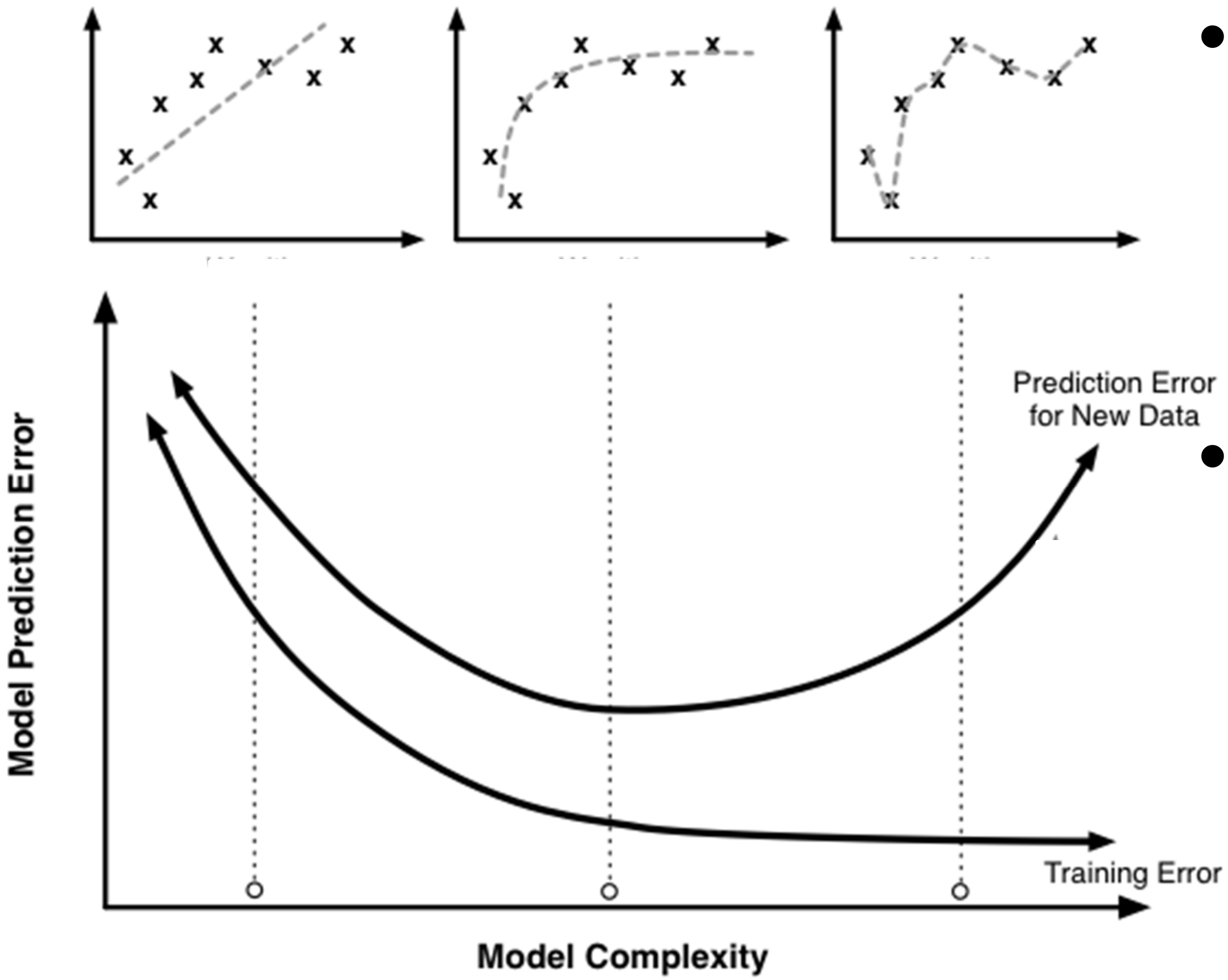
- Too **simple** models **underfit**, too **complex overfit**

Fail to capture pattern in training data

Memorize training dataset (including noise), fail to generalize on unseen data



Detecting overfitting



- **Good fit on training set** is **necessary** (no underfitting),
- ...but **not sufficient for learning** (good fit on test data)

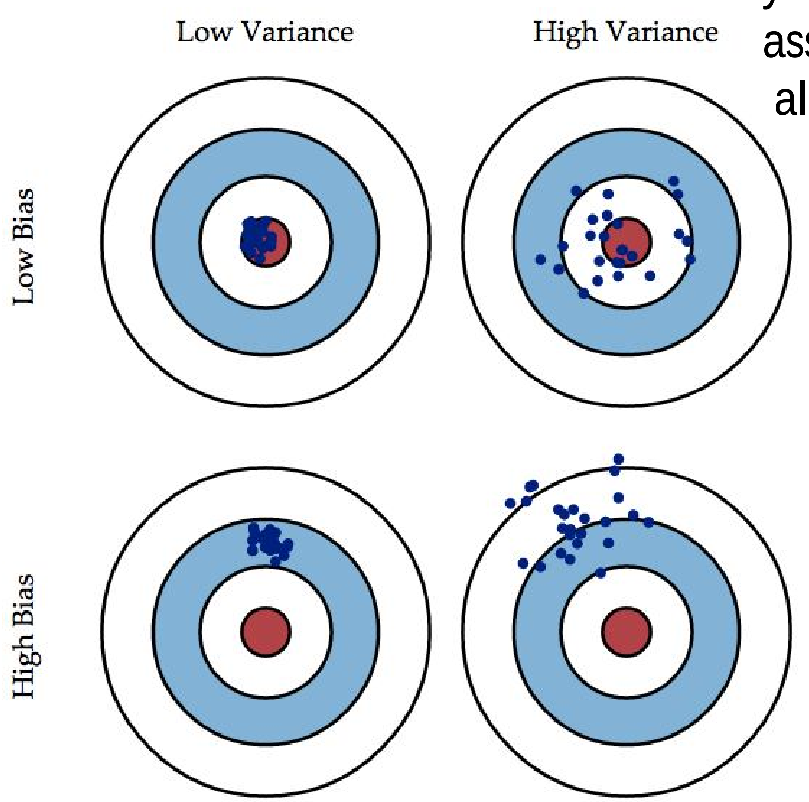
Bias vs. Variance

- Under certain loss functions can decompose **expected error** of a supervised learning algorithm into:

Error = **(Statistical) Bias** + **Variance** + **Noise**

Systematic error due to assumptions built into the algorithm; How far on average predictions are from truth; **Can reduce (increase complexity)**

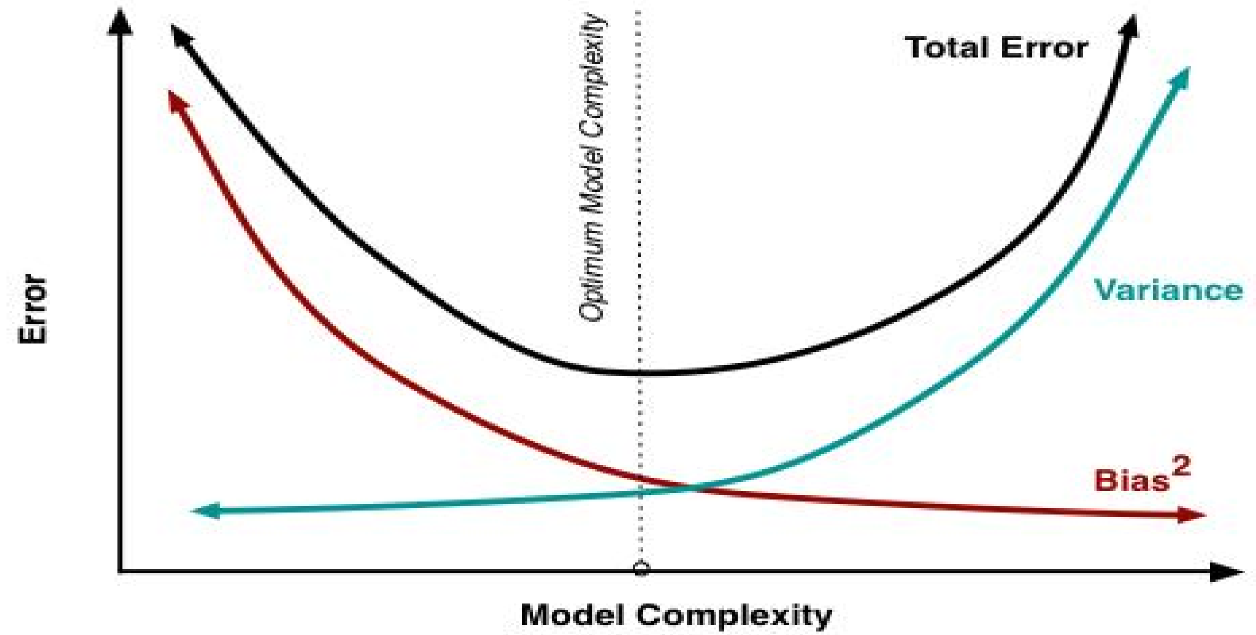
How ambiguous the problem is; **Cannot reduce** w/o re-annotating / asking for more features



Error due to sensitivity to small fluctuations in the training set; How different on average are individual predictions on the same input produced by versions of the predictor trained on slightly different training sets; **Can reduce (decrease complexity)**

Complexity & Bias-Variance

- As complexity increases, bias decreases & variance increases ; need to find ‘sweetspot’



- Most learning algorithms have hyperparameters to control the tradeoff; find optimal tuning via cross-validation

Inductive Bias

- **Inductive bias** of a learner: the set of assumptions it uses to predict outputs given inputs that it has not encountered
- **Without any such assumptions, learning cannot be solved exactly**
- e.g. **Linear regression**:
  Only look for **lines**
  assuming a **specific type of noise in the data,** etc.

Tom Michael Mitchell

- **Don't confuse with statistical bias** which is always bad

No Free Lunch Theorems

- **If we make no prior assumption** about the nature of the learning task*****, **no learning method can be said to be superior overall (or better than random guessing...)**

David H. Wolpert

- *i.e. across **all possible 'true' hypotheses**
- But **not all of them equally likely or interesting**!

Embracing Uncertainty (1)

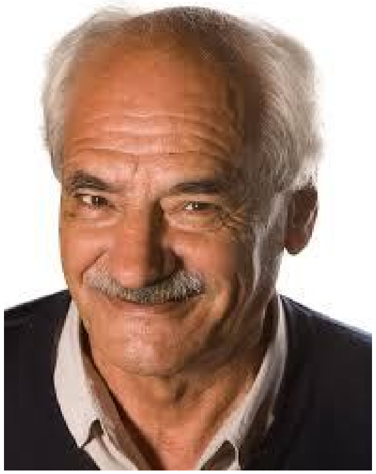
- Can have -probabilistic- guarantees on induction!
- **PAC-learning**: If we restrict the hypothesis space to be finite & use enough training examples, we can be fairly confident (**probably**) that we find a hypothesis that is not that bad (**approximately correct**), in **polynomial time**[Turing Award 2010]

Leslie Gabriel Valiant

Embracing Uncertainty (2)

- **VC-theory**: Similar guarantees but need not restrict the hypothesis space to a finite one.
- **Complexity** of hypotheses used in both theories: **Cardinality of hypothesis space** in PAC, **VC-dimension** in VC
- Guarantees pessimistic; in practice can do better …perhaps also in theory?

Vladimir Naumovich Vapnik



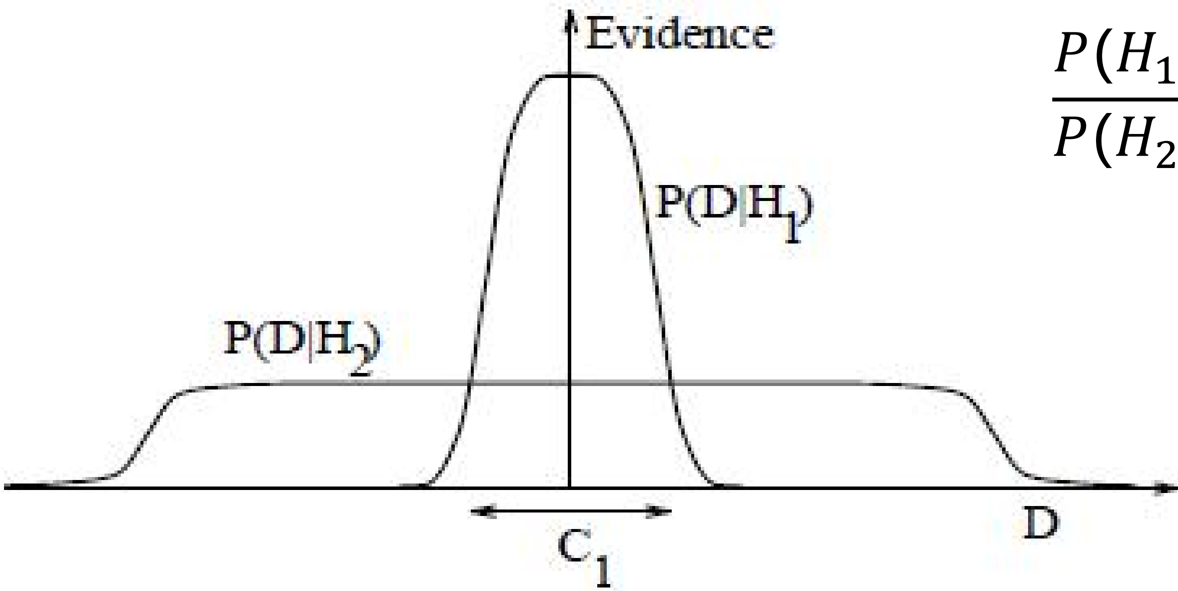
Alexey Yakovlevich Chervonenkis
(1938 –2014)

Occam’s Razor Everywhere! (1)

- Kolmogorov Complexity & MDL [Part I]
  - Hypotheses of smaller descr. length -> higher prior
- PAC-learning
  - Tighter generalization bounds for more constrained hypothesis spaces given the same amount of data
- VC-theory
  - As above, for hypotheses of lower VC dimension
- Logic
  - Conjunctions with more conjuncts ‘easier’ to falsify

Occam's Razor Everywhere! (2)

- (Not so) Bayesian Learning



$$\frac{P(H_1|D)}{P(H_2|D)} = \frac{P(D|H_1)}{P(D|H_2)}\frac{P(H_1)}{P(H_2)}$$

More complex hypothesis H_2 consistent with more outcomes

So $P(D|H_2)$ mass spread thinner than $P(D|H_1)$

When D in region C_1 , $P(D|H_1)$ > $P(D|H_2)$

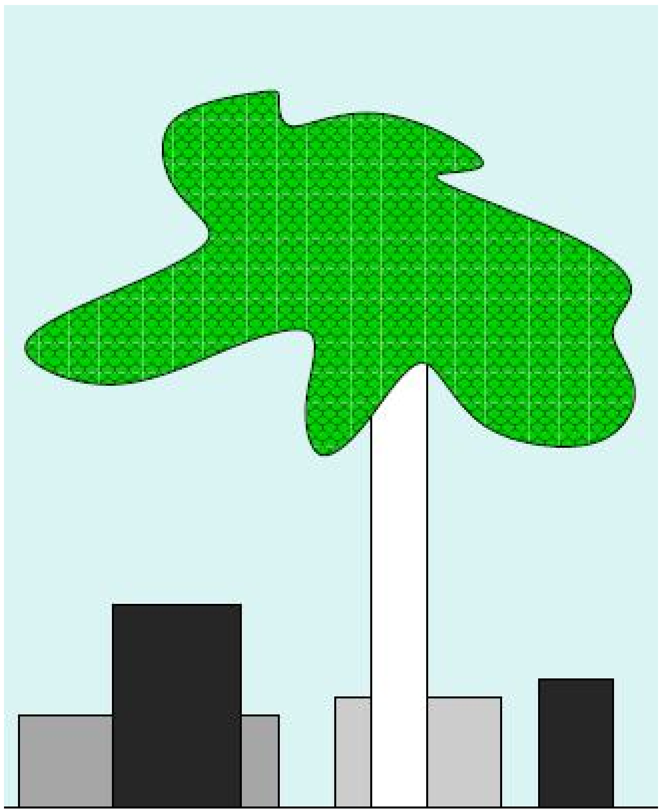
Assumptions Everywhere!

- Both Bayesian & frequentist inference do
- Both parametric & non-parametric methods do

- Most learning theory based on assumptions…
- … some are reasonable, some not so much…

Occam's Razor in Human Inference (1)

- How many boxes do are there?



Occam's Razor in Human Inference (2)

- Are you sure?

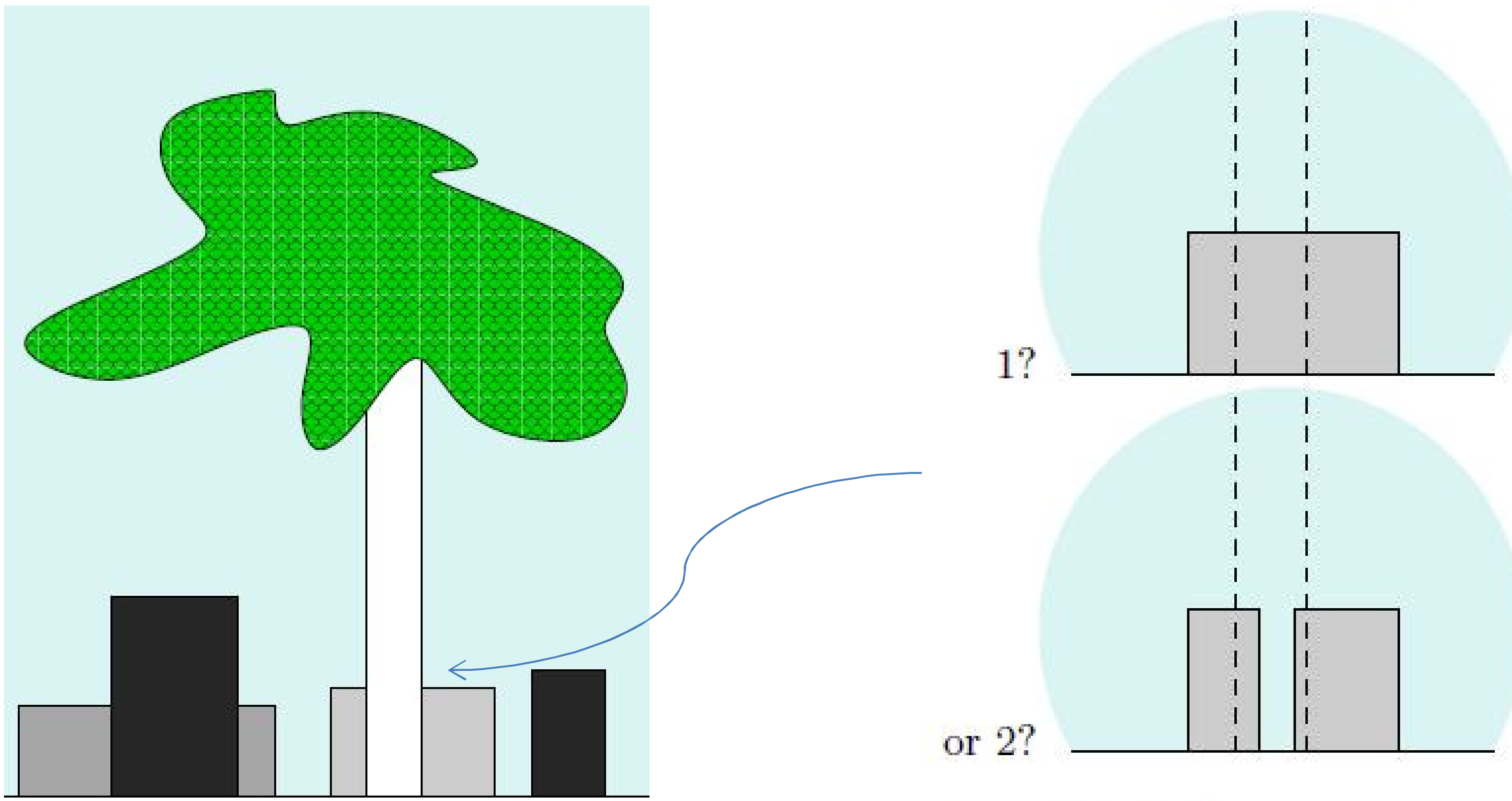


Figure 28.2. How many boxes are behind the tree?

Inductive Bias in Human Inference (1)

- Think of ‘I.Q. tests’
- Which is the next number in the sequence

0, 1, 3, 6, 10, 15, ?

Inductive Bias in Human Inference (2)

- We could have chosen **infinite** other hypotheses but we **all** thought of this one:

$$H\colon x_{n+1} = x_n + n$$

0, 1, 3, 6, 10, 15, 21

1 2 3 4 5 6

- ...because of our **built-in inductive bias**

We Machine Learners Must…

- Be aware that **induction is an ill-posed problem** & its **optimal solution intractable**
- Be aware of the **limits of our predictions** (**confidence**, **approximations**)
- Be aware of our **assumptions** (**inductive bias**) and **how realistic** they are **in the problem** at hand

- Not be discouraged by all these; **inductive reasoning is –apparently– a solved problem in nature** (at least most of the time, approximately & under certain assumptions)!

End of Part II

Thanks again!