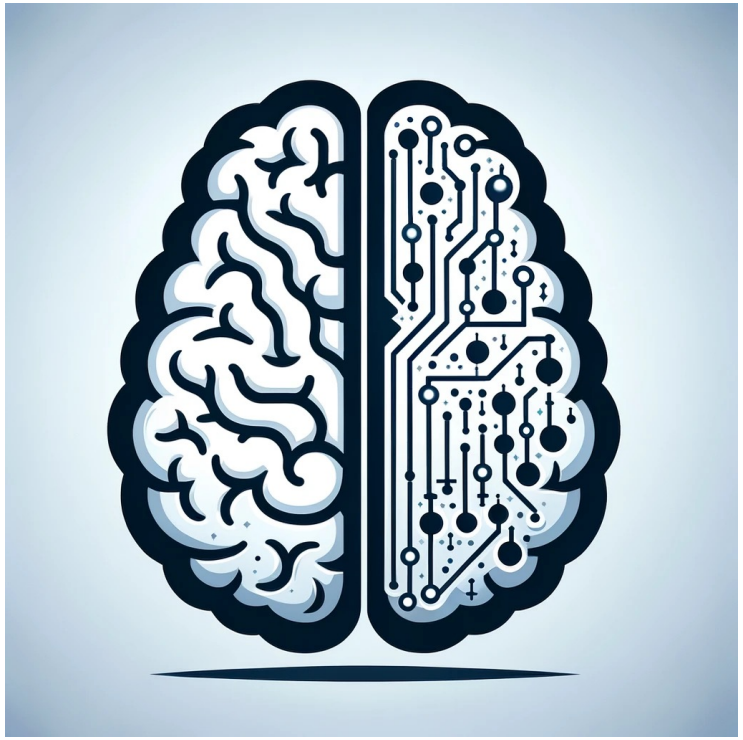
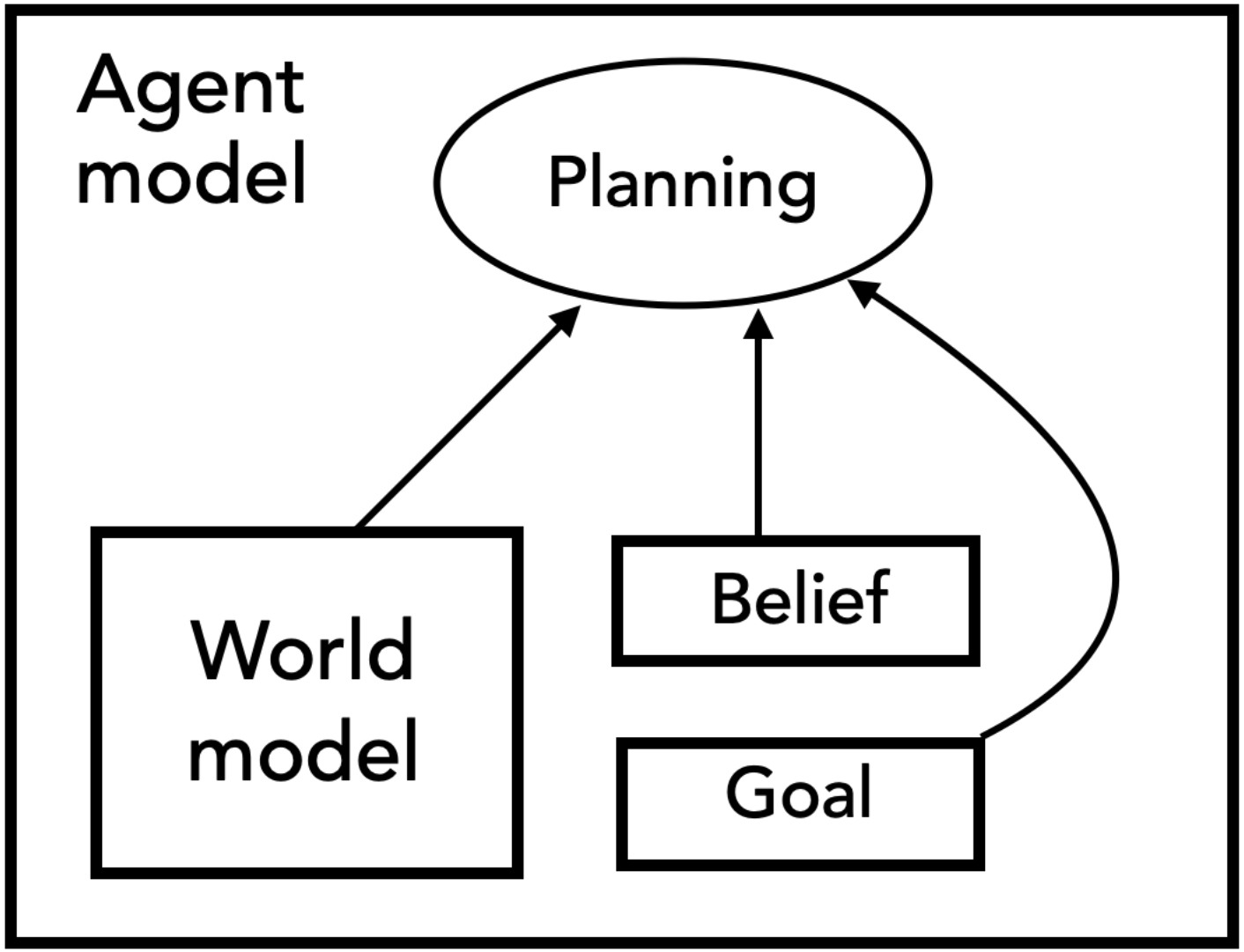
EN 601.473/601.673: Cognitive Artificial Intelligence (CogAI)



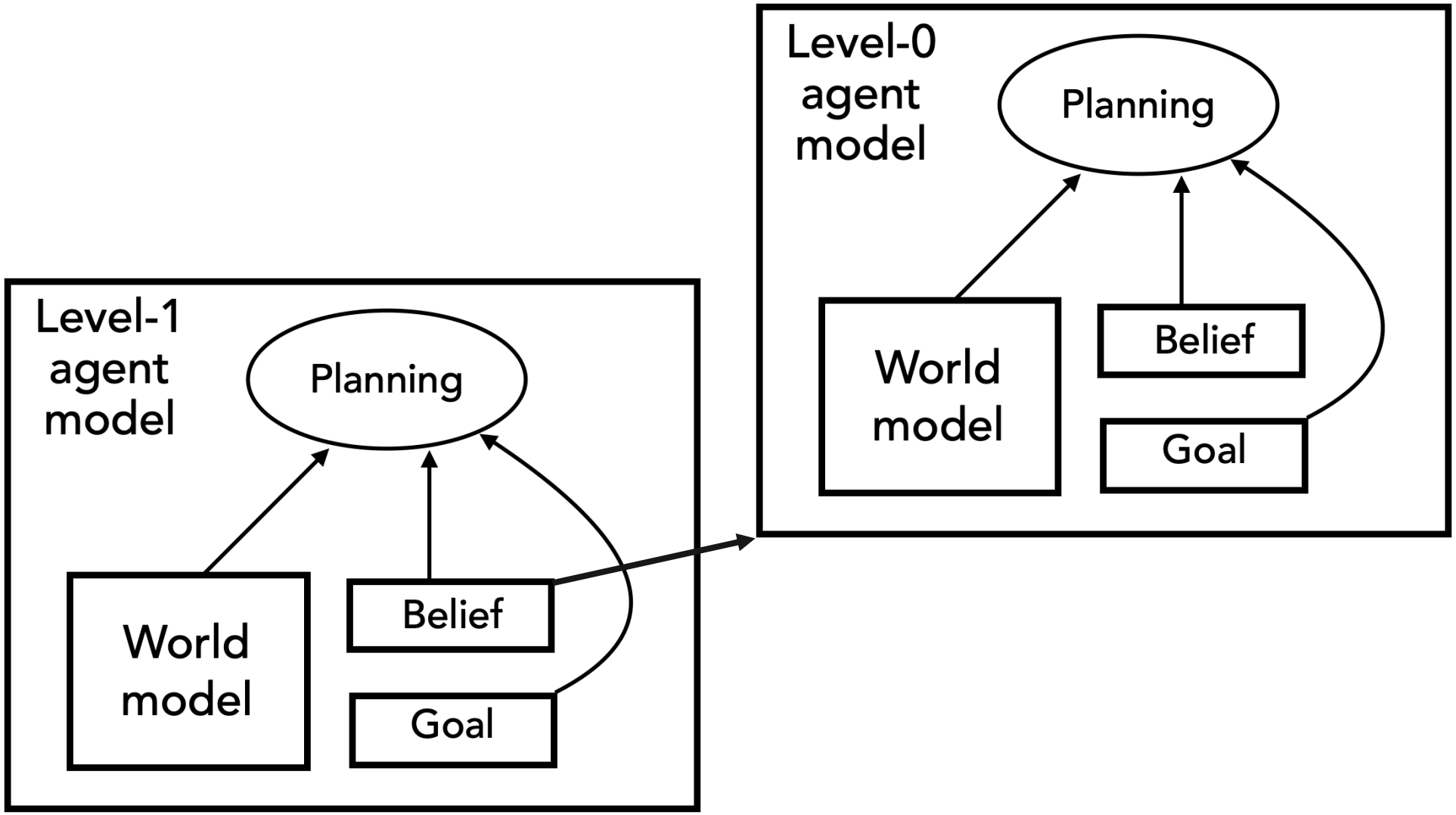
Lecture 3: Bayesian cognition & Bayesian modeling

Tianmin Shu

Recap: world models and agent models



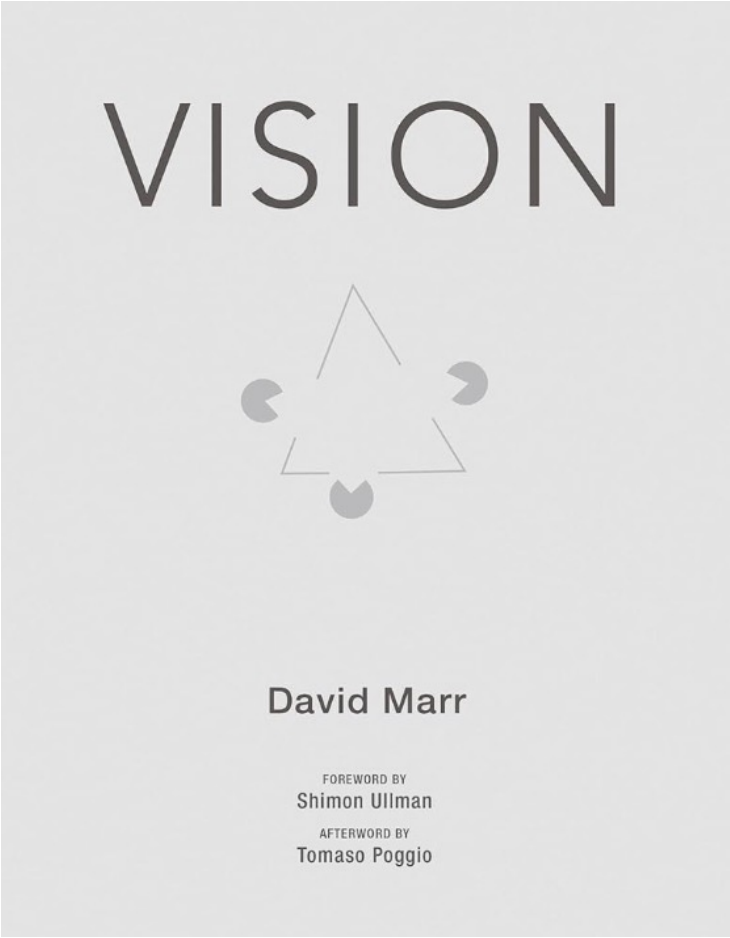
Recap: world models and agent models



Recap: Reverse engineer a cognitive system

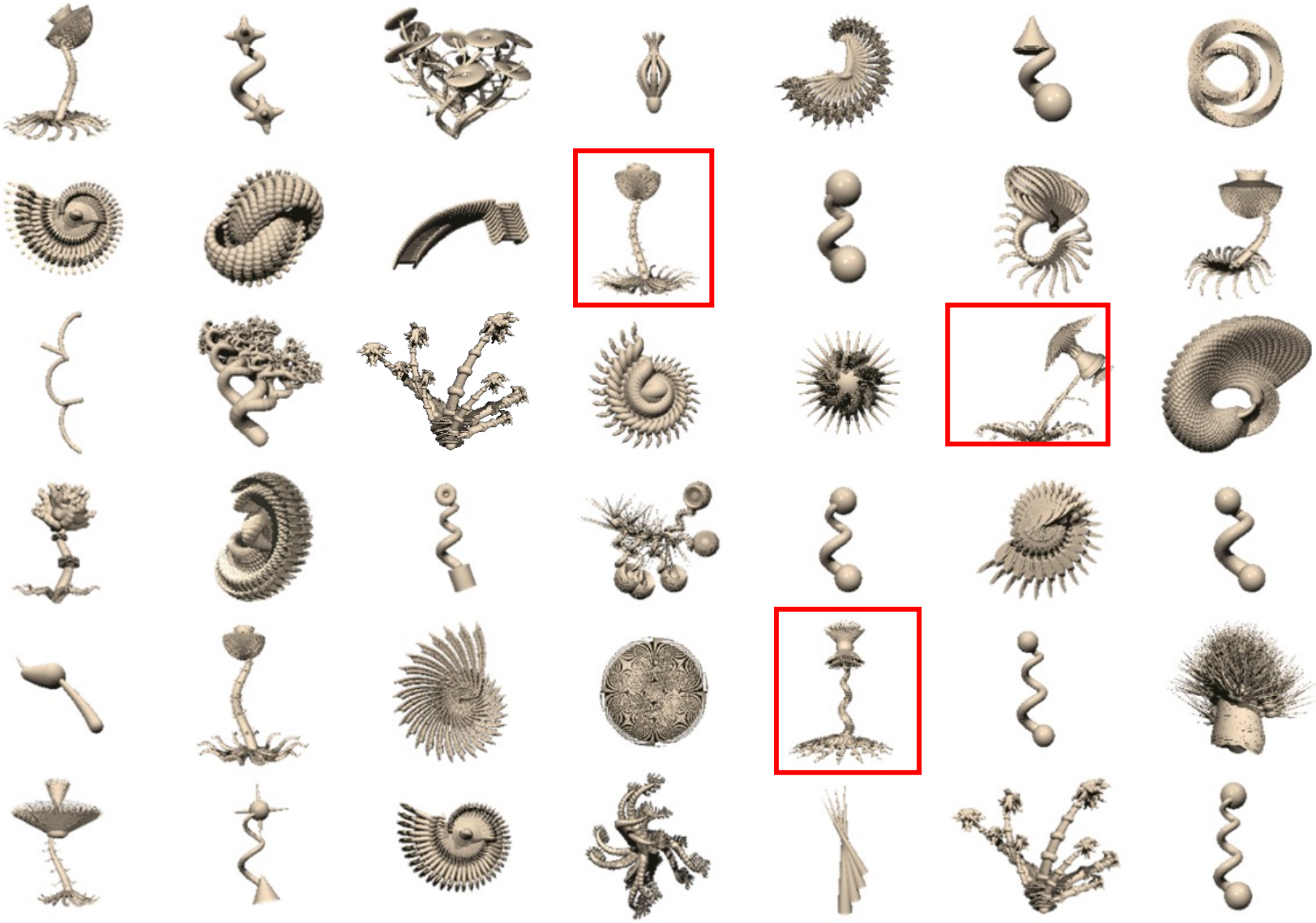
- From the “problem of induction” to the problem of inductive inference
- Three levels of analysis
- A toolkit for solving these problems

Three levels of analysis for reverse engineering a cognitive system



- **Level 1: Computational theory**
  - What are the inputs and outputs to the computation, what is its goal, and what is the logic by which it is carried out?
- **Level 2: Representation and algorithm**
  - How is information represented and processed to achieve the computational goal?
- **Level 3: Hardware implementation**
  - How is the computation realized in physical or biological hardware?

Learning words for objects



"tufa"

A toolkit for solving the problem of induction

- 1. How does abstract knowledge guide learning and inference given sparse data?

  Bayesian inference in probabilistic generative models.

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{\sum_{h_i \in H} P(d \mid h_i)P(h_i)}$$

Word learning as Bayesian inference

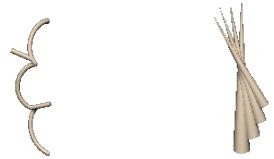
Word Learning as Bayesian Inference

Fei Xu
University of British Columbia

Joshua B. Tenenbaum
Massachusetts Institute of Technology

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{\sum_{h_i \in H} P(d \mid h_i)P(h_i)}$$

h: knowledge, i.e., word meaning / categorization
d: examples/images of an object category



What is the right prior?
What is the right hypothesis space?
How do learners acquire that background knowledge?

A toolkit for solving the problem of induction

- 1. How does abstract knowledge guide learning and inference sparse data?

  Bayesian inference in probabilistic generative models.

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{\sum_{h_i \in H} P(d \mid h_i)P(h_i)}$$

- 2. What form does that knowledge take, across different domains and tasks? Probabilities defined richly structured symbolic representations: spaces, graphs, grammars, logical predicates, schemas...
- 3. How is the knowledge itself constructed, from some combination of innate specifications and experience?

  Hierarchical models, with inference at multiple levels.

  In machine learning terms: learning models as probabilistic inference, “learning to learn”, transfer learning, learning representations and learning inductive biases

Word learning as Bayesian inference

Plants?
Fungi?

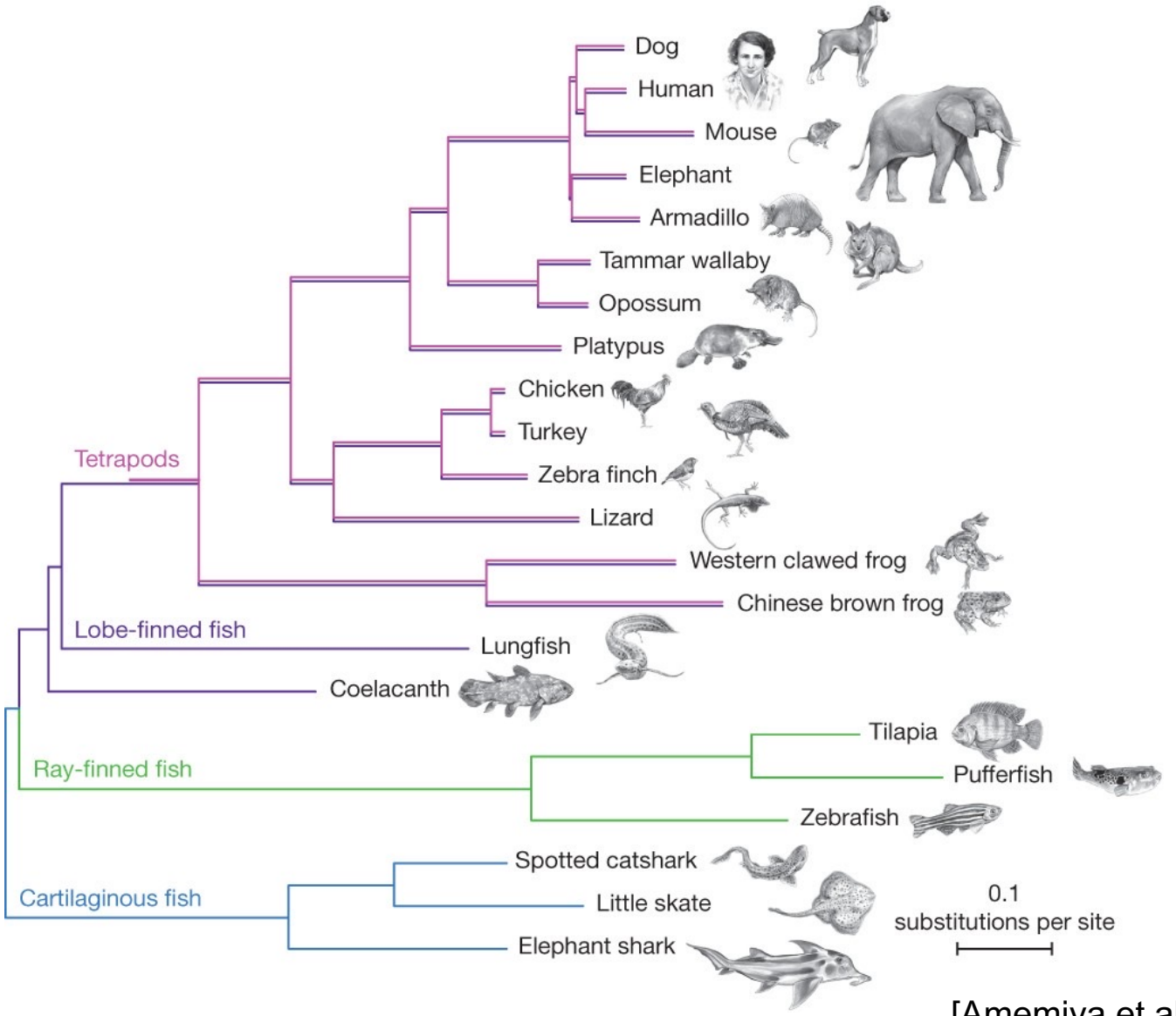
Bacteria?
Worms?

Shellfish?

Informed by our prior knowledge of biology

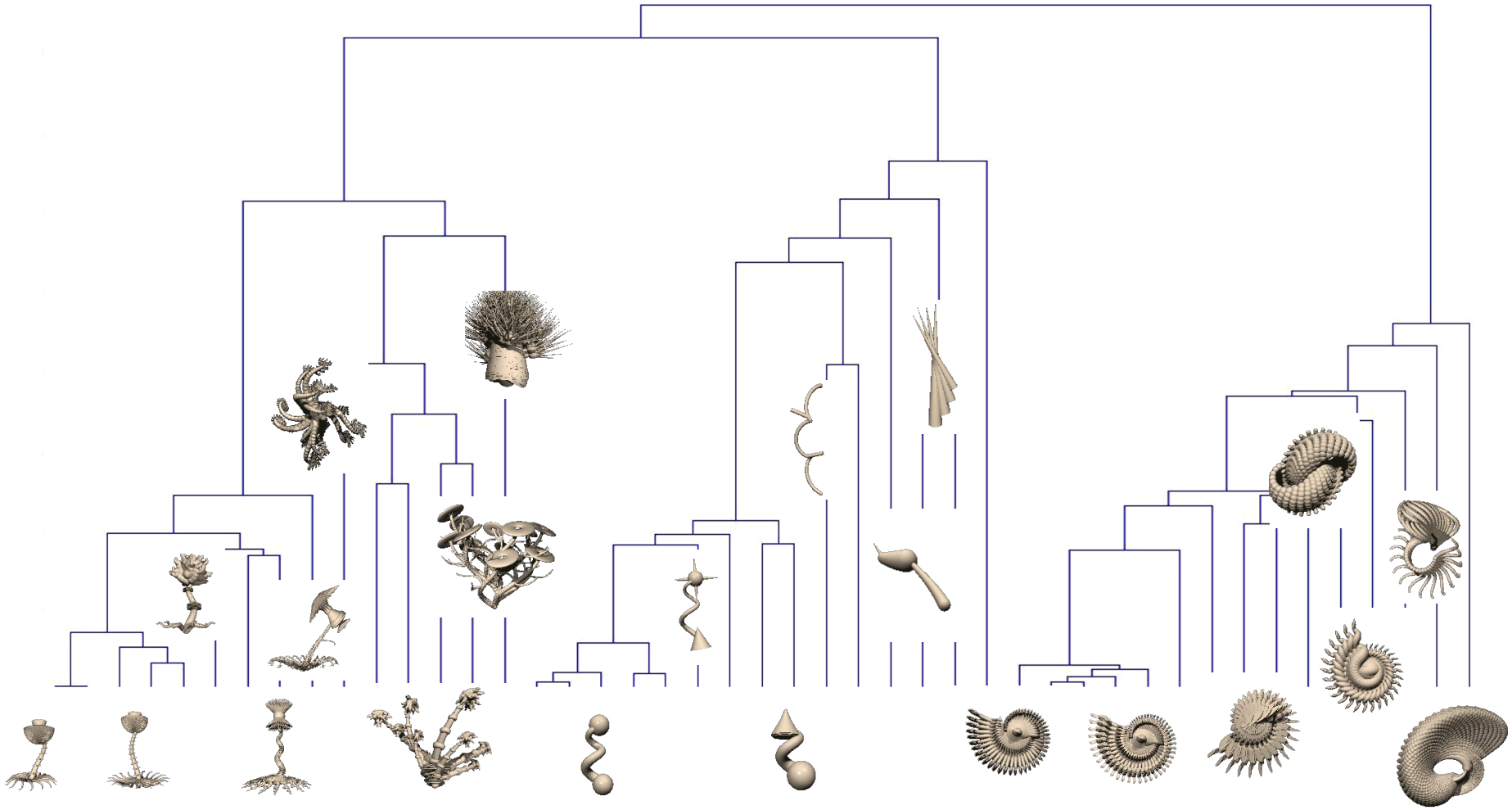
Word learning as Bayesian inference

From biology: **tree** structure to represent evolutionary or genetic lineage

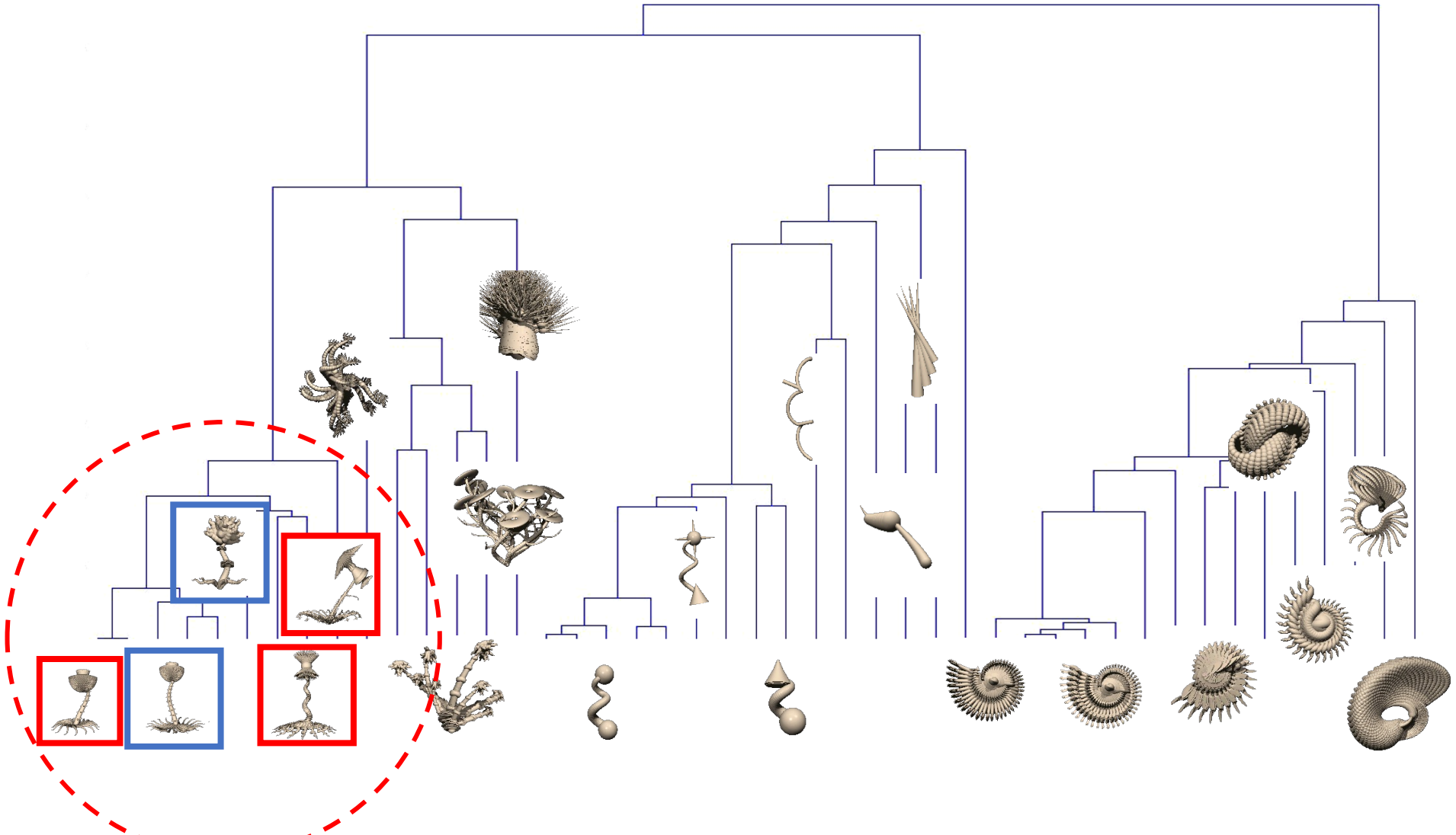


[Amemiya et al. 2013: 312, fig. 1]

Word learning as Bayesian inference



Word learning as Bayesian inference



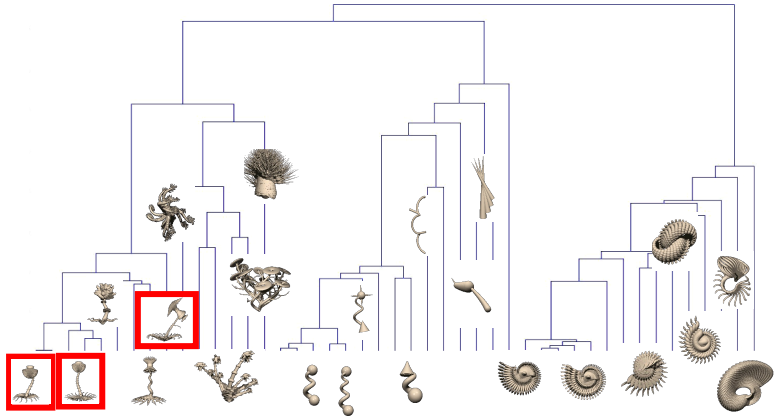
Origins of the hypothesis space

$P(\text{form})$

Hierarchically construct a hypothesis space

F: form

Tree

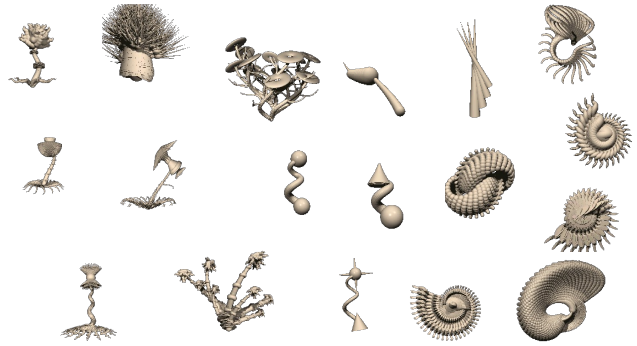


$P(\text{structure} \mid \text{form})$

S: structure

$P(\text{data} \mid \text{structure})$

D: data



A toolkit for solving the problem of induction

- 1. How does abstract knowledge guide learning and inference sparse data?

  Bayesian inference in probabilistic generative models.

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{\sum_{h_i \in H} P(d \mid h_i)P(h_i)}$$

- 2. What form does that knowledge take, across different domains and tasks? Probabilities defined richly structured symbolic representations: spaces, graphs, grammars, logical predicates, schemas...
- 3. How is that knowledge itself constructed, from some combination of innate specifications and experience?

  Hierarchical models, with inference at multiple levels.

  In machine learning terms: learning models as probabilistic inference, “learning to learn” / meta-learning, transfer learning, learning representations and learning inductive biases

A toolkit for solving the problem of induction

- 4. How can learning and inference proceed efficiently and accurately, even with very complex hypothesis spaces?

  Sampling-based algorithms for approximate inference, e.g., MCMC, sequential Monte Carlo (particle filtering), fast initialization with bottom-up recognition models (neural networks).

- 5. How can probabilistic inference be used to drive action?

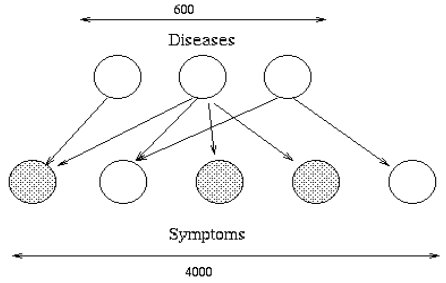
Utility-based frameworks for decision and planning under uncertainty and risk.

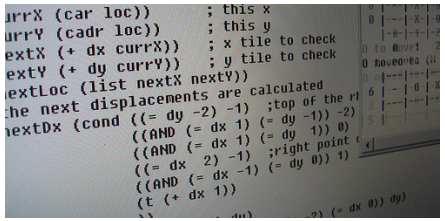
- 6. How could these computations be implemented in hardware?
  Brain, GPU

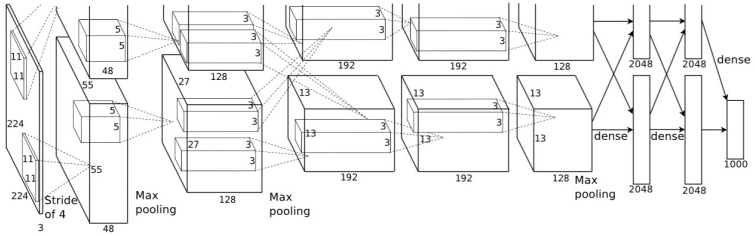
Probabilistic programming languages (PPLs)

A unifying framework integrating our best computational ideas on intelligence, across multiple eras of cognitive science and AI:

- **Probabilistic (Bayesian) inference** for reasoning about likely unobserved causes from observed effects, and decision making under uncertainty.
- **Symbolic programs** for representing and reasoning with abstract knowledge
- **Neural networks** for amortized inference, data-driven priors, and differentiable generative models.







Look ahead

- Bayesian cognition and Bayesian modeling
- Bayesian concept learning
  - As an example of Bayesian modeling
  - Foundation for world modeling and agent modeling
- Bayesian networks
  - Complex Bayesian modeling
  - Efficient inference algorithms
- Probabilistic programming languages
- Physical reasoning
  - Intuitive physics in humans and machines
  - Model-based physical scene understanding
- Social reasoning
  - Intuitive psychology in humans and machines
  - Single/multi-agent decision making
  - Inverse decision making
  - Moral judgment (a guest lecture)

Readings for this & next week

- Papers under week 2 (Bayesian cognition & Bayesian concept learning)
  - Computational cognitive science & modern cognitive AI
- Probmods.org, Chapters 2 & 3 (WebPPL basics)

Outline

- Basic Bayesian cognition and Bayesian modeling
  - Theoretical foundations (modeling human inductive reasoning)
  - Flipping coins
  - Bayesian concept learning

Inductive reasoning

- Learn abstract knowledge from little data and generalize the knowledge beyond the given data
- If our inferences go beyond the data given, then something must be making up the difference…
- What is it? **constraints** (in psychology), **inductive biases** (machine learning and AI), **priors** (stats), etc.
- Key questions: What does this prior knowledge look like? How do we combine prior knowledge with data to make inferences? What are the models and algorithms?

Bayesian cognition's answer: Bayesian modeling

- An approaching for understanding inductive problems, and it typically takes a strong "top-down" strategy

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{\sum_{h_i \in H} P(d \mid h_i)P(h_i)}$$

- **Level 1: Computational theory**
  - What are the inputs and outputs to the computation, what is its goal, and what is the logic by which it is carried out?
- **Level 2: Representation and algorithm**
  - How is information represented and processed to achieve the computational goal?
- **Level 3: Hardware implementation**
  - How is the computation realized in physical or biological hardware?

Bayesian inference for evaluating hypotheses given data

- Bayes' Rule

Posterior Likelihood Prior

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{\sum_{h_i \in H} P(d \mid h_i)P(h_i)}$$

- An example
  - Data *d*: John is coughing
  - Some hypotheses *h*:
    - 1. John has a cold
    - 2. John has lung cancer
    - 3. John has a stomach flu
- Which hypotheses should we believe, and with what **certainty**?

Bayesian inference for evaluating hypotheses given data

Posterior Likelihood Prior

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{\sum_{h_i \in H} P(d \mid h_i)P(h_i)}$$

Data

$d =$ John is coughing

Hypotheses:

$h_1 =$ John has a cold

$h_2 =$ John has lung cancer

$h_3 =$ John has a stomach flu

$P(h_1) = 0.75$ $P(d|h_1) = 1$

$P(h_2) = 0.05$ $P(d|h_2) = 1$

$P(h_3) = 0.2$ $P(d|h_3) = 0.2$

$$P(h_1|d) = \frac{0.75 \times 1}{0.75 \times 1 + 0.05 \times 1 + 0.2 \times 0.2} = 0.89$$

$$P(h_2|d) = 0.06$$

$$P(h_3|d) = 0.05$$

Where does Bayes' rule come from?

Product rule: $P(a,b) = P(a|b)P(b)$

$$P(h,d) = P(h|d)P(d)$$
$$P(h,d) = P(d|h)P(h)$$

Product rule applied twice

$$P(h|d)P(d) = P(d|h)P(h)$$

Equating the two right hand sides

$$P(h|d) = \frac{P(d|h)P(h)}{P(d)}$$

Divided by *P(d)*

$$P(h|d) = \frac{P(d|h)P(h)}{\sum_{h_i} P(d|h_i)P(h_i)}$$

$$P(d) = \sum_{h_i} P(d|h_i)P(h_i)$$

Why Bayesian modeling?

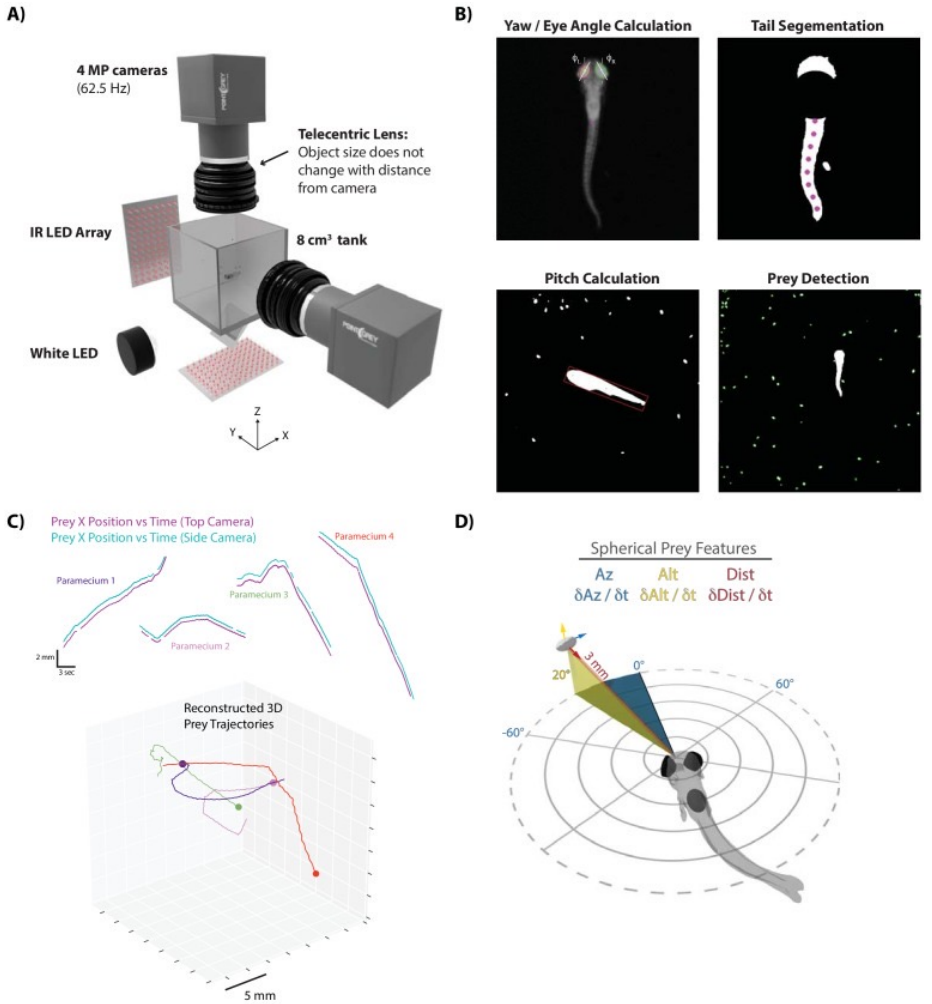
- Fundamentally, cognition is about making good guesses, and good bets -- i.e., forming **beliefs** about the world and other agents and making decisions based on the **beliefs**

- Way before thinking, before learning, before language, brains were making guesses and bets...

Elements of a stochastic 3D prediction engine in larval zebrafish prey capture

Andrew D Bolton[1†*], Martin Haesemeyer[1], Josua Jordi[1], Ulrich Schaechtle[2], Feras A Saad[2], Vikash K Mansinghka[2], Joshua B Tenenbaum[2], Florian Engert[1]

eLIFE
elifesciences.org

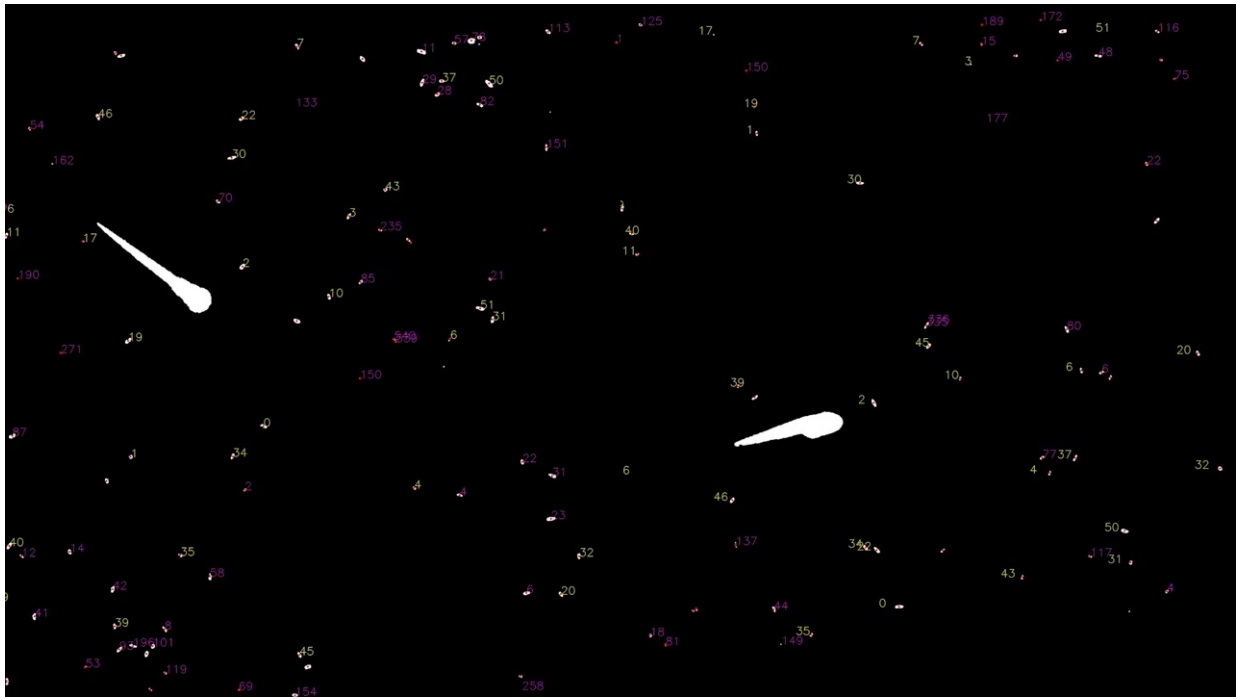


Elements of a stochastic 3D prediction engine in larval zebrafish prey capture

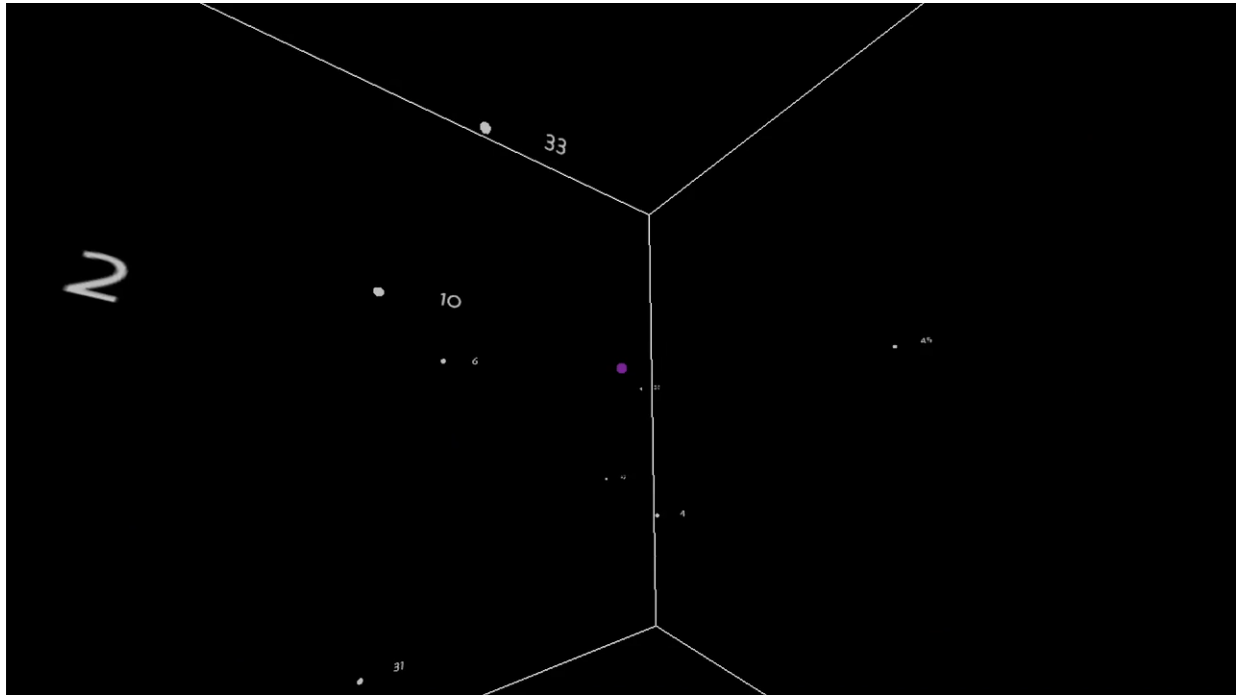
Andrew D Bolton[1†*], Martin Haesemeyer[1], Josua Jordi[1], Ulrich Schaechtle[2], Feras A Saad[2], Vikash K Mansinghka[2], Joshua B Tenenbaum[2], Florian Engert[1]

eLIFE
elifesciences.org

1. Based on perception, form **belief** about: preys' 3D locations, velocities, etc.
2. Then decide which prey to pursue & how



2D reconstruction
(overhead perspective)



3D reconstruction
(zebrafish perspective)

Why Bayesian modeling?

- Fundamentally, cognition is about making good guesses, and good bets -- i.e., forming **beliefs** about the world and other agents and making decisions based on the **beliefs**

- Way before thinking, before learning, before language, brains were making guesses and bets...

- Bayes' rule provides the gold standard for what it means to have rational, coherence beliefs (under uncertainty)
  - Evolutionary considerations:
    - If your beliefs violate probability theory, then you can always be out-gambled by someone whose beliefs do so accord.
    - If you are betting your life on your inferences, you better bet well.

Betting against an agent who has wrong beliefs

- Axioms of probability:
- $P(a \vee b) = P(a) + P(b) - P(a \wedge b)$
- Is this a valid belief?
- $P(a) = 0.4, P(b) = 0.3, P(a \vee b) = 0.8, P(a \wedge b) = 0$

Betting against an agent who has wrong beliefs

If Agent 1 expresses a set of degrees of belief that violate the axioms of probability theory then there is a combination of bets by Agent 2 that guarantees that Agent 1 will lose money every time.

Agent 1		Agent 2		Outcomes and payoffs to Agent 1			
Proposition	Belief	Bet	Stakes	a, b	$a, \neg b$	$\neg a, b$	$\neg a, \neg b$
a	0.4	a	4 to 6	–6	–6	4	4
b	0.3	b	3 to 7	–7	3	–7	3
$a \vee b$	0.8	$\neg(a \vee b)$	2 to 8	2	2	2	–8
				–11	–1	–1	–1

Why Bayesian modeling?

- But don't we know that people aren't good at even simple Bayesian inference problems?
  - Simple text problems versus common sense
  - People may not be able to solve textbook physics problems, but we have intuitive understanding about the physics
- And don't we know that people's beliefs are often irrational or incoherent?
  - A bigger worry. But we will focus on the commonsense domains at the core of human and machine intelligence
  - In other words, we aren't saying "people are always Bayesian...," or even that "people typically approximate Bayes pretty well"
- Bayesian modeling can potentially help us identity the source of irrationality or incoherence. E.g., false belief.

Why Bayesian modeling?

- Despite all its limitations, human thought remains the gold standard for forming “good” beliefs
  - accurate, reasonable, calibrated, useful, general purpose, flexible, data-efficient...
- Bayes provides a theory of learning (where our beliefs come from)
  - A common currency for combining prior knowledge and data of experience
- Bayes provides a theory of decision-making and planning (how beliefs inform our actions)
  - A way to balance *expected* outcomes and *valuable* outcomes
  - We will discuss in single/multi-agent decision making
- That doesn’t mean Bayes is all we need -- very much not!
  - Other complementary tools for knowledge representation, learning, decision-making and planning (e.g., neural networks)
  - Resource-rational, tractable approximate computation

Inductive reasoning

- Learn abstract knowledge from little data and generalize the knowledge beyond the given data
- If our inferences go beyond the data given, then something must be making up the difference…
- What is it? **constraints** (in psychology), **inductive biases** (machine learning and AI), **priors** (stats), etc.
- Key questions: What does this prior knowledge look like? How do we combine prior knowledge with data to make inferences? What are the models and algorithms?

A toolkit for solving them

- 1. How does abstract knowledge guide learning and inference sparse data?

  Bayesian inference in probabilistic generative models.

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{\sum_{h_i \in H} P(d \mid h_i)P(h_i)}$$

- 2. What form does that knowledge take, across different domains and tasks?

  Probabilities defined richly structured symbolic representations: spaces, graphs, grammars, logical predicates, schemas...

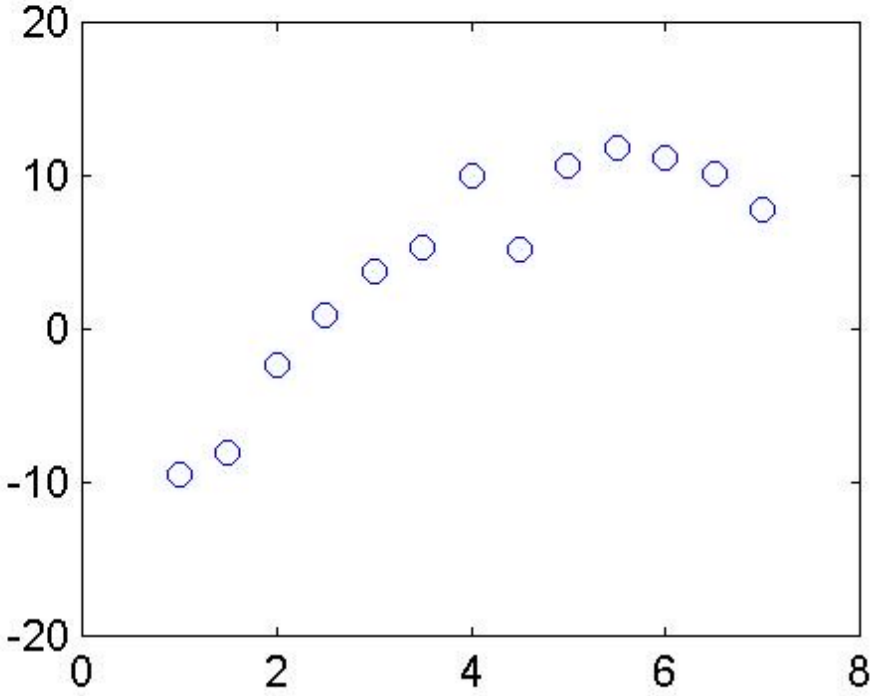
- 3. How is that knowledge itself constructed, from some combination of innate specifications and experience?

  Hierarchical models, with inference at multiple levels.

  In machine learning terms: learning models as probabilistic inference, “learning to learn” / meta-learning, transfer learning, learning representations and learning inductive biases

Sources on inductive constraints

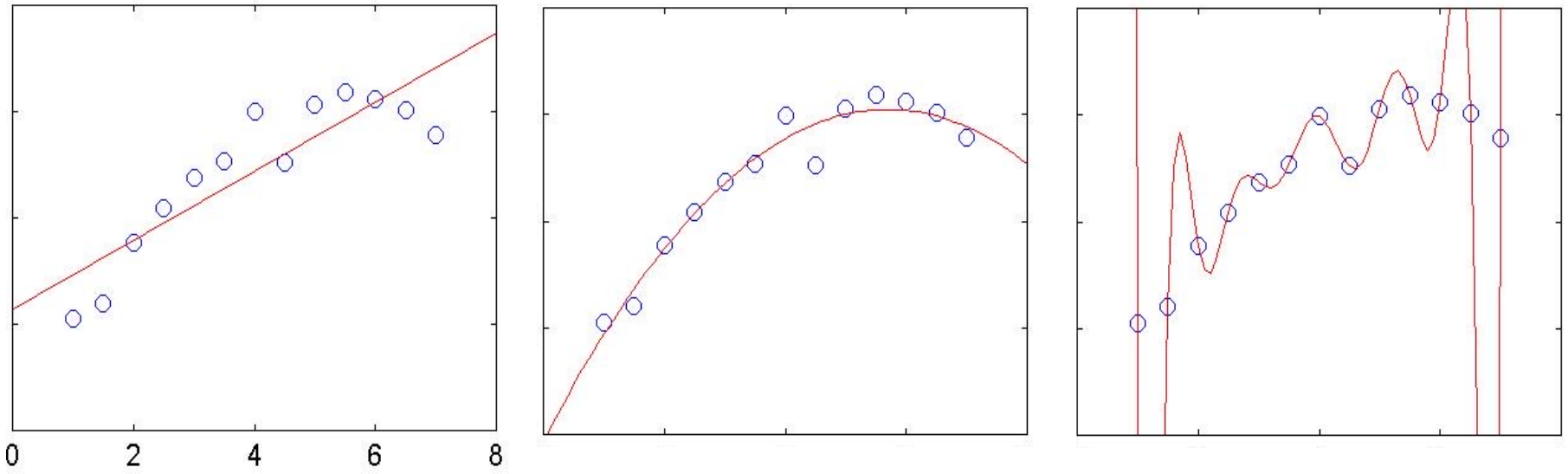
- Any successful generalization from limited data requires some form of simplicity constraint (Occam's razor)
- But what do we mean by simple (in computational terms)?
- Hypotheses with fewer arbitrary free parameters are simpler
  - Shorter description lengths, smoother, etc.



Sources on inductive constraints

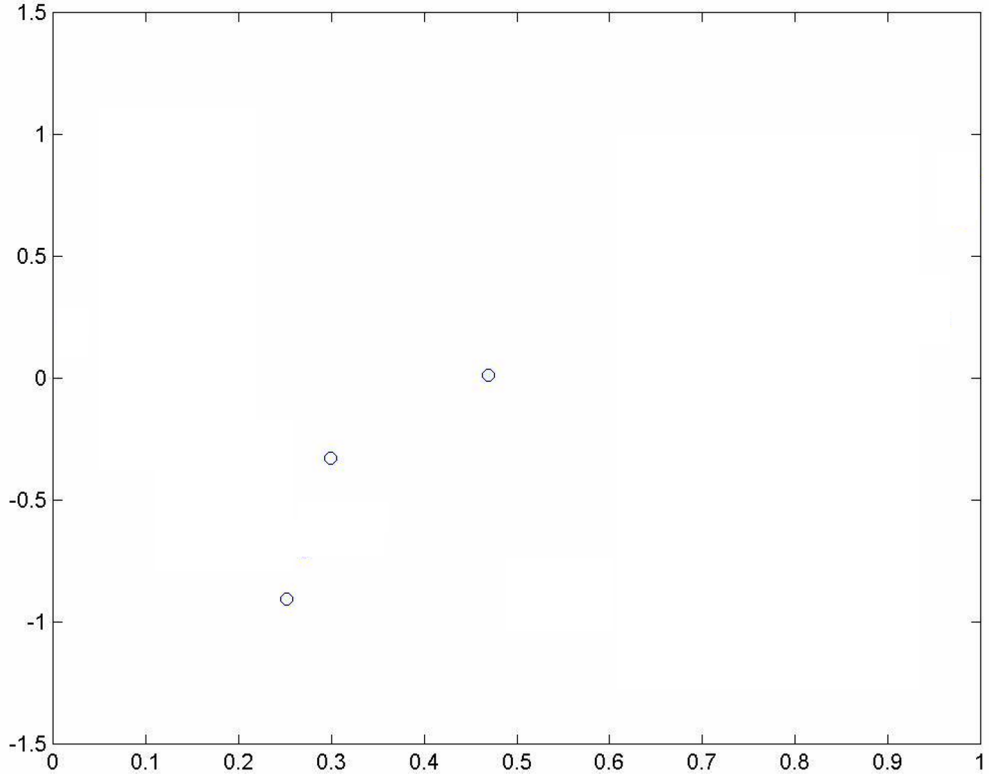
- Any successful generalization from limited data requires some form of simplicity constraint (Occam's razor)
- But what do we mean by simple (in computational terms)?
- Hypotheses with fewer arbitrary free parameters are simpler
  - Shorter description lengths, smoother, etc.

Which curve is best supported by the data?



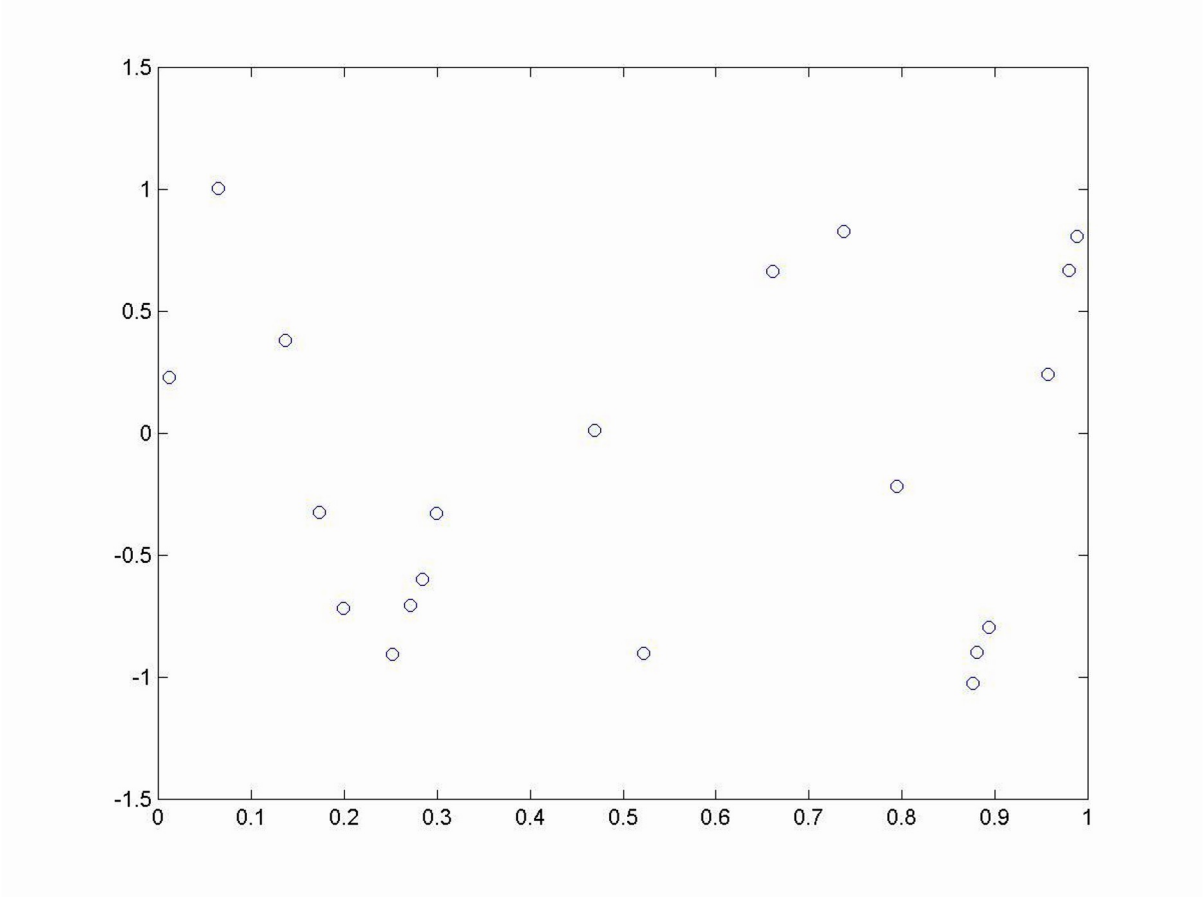
Simplicity is not simple

- What is the relation between y and x?



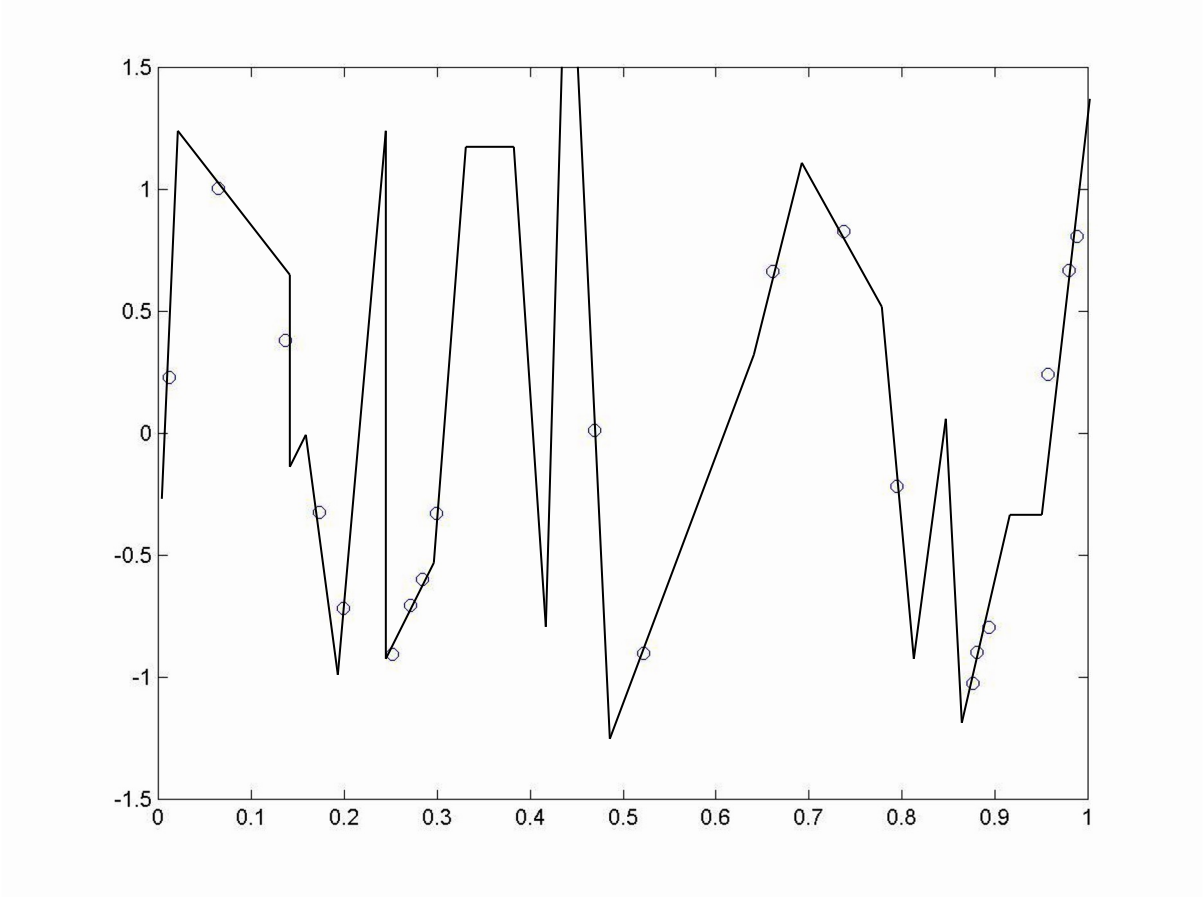
Simplicity is not simple

- What is the relation between y and x?



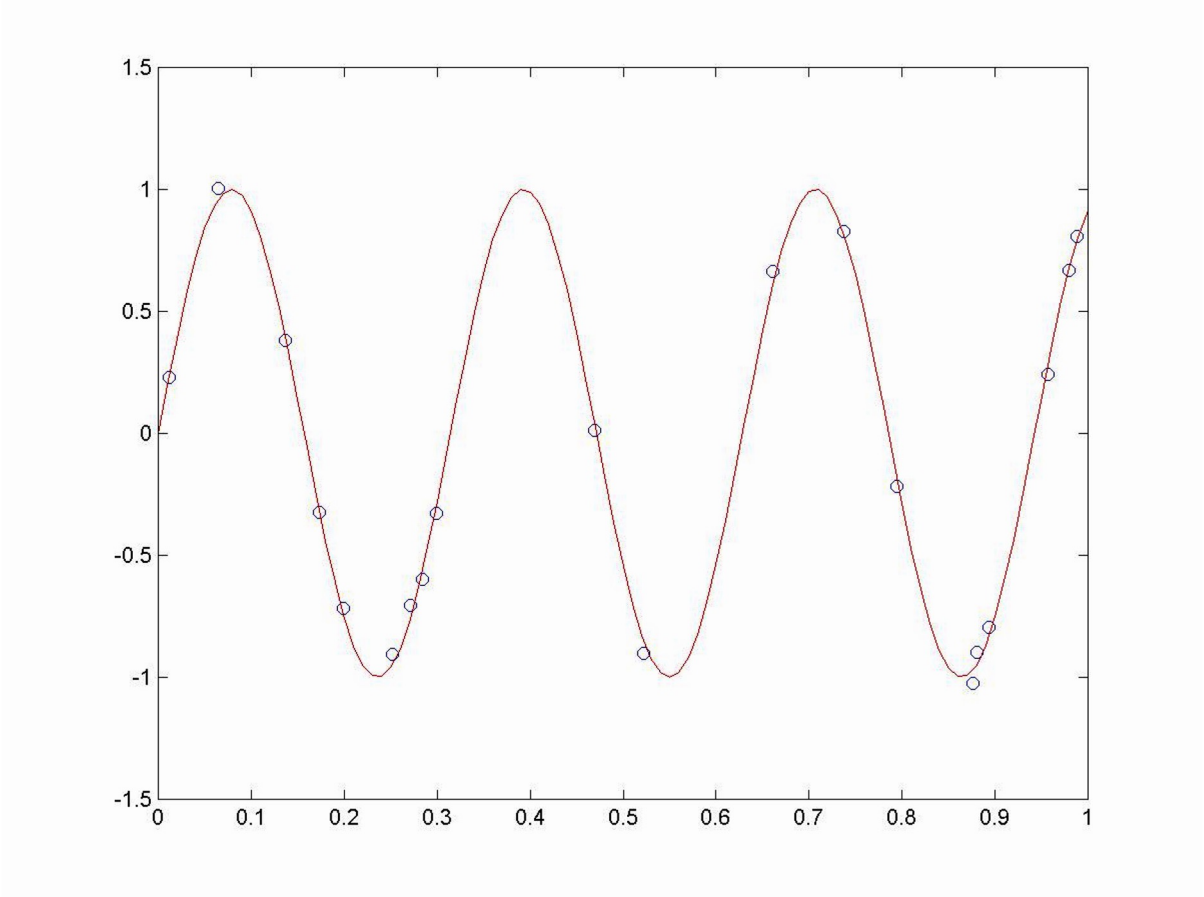
Simplicity is not simple

- What is the relation between y and x?



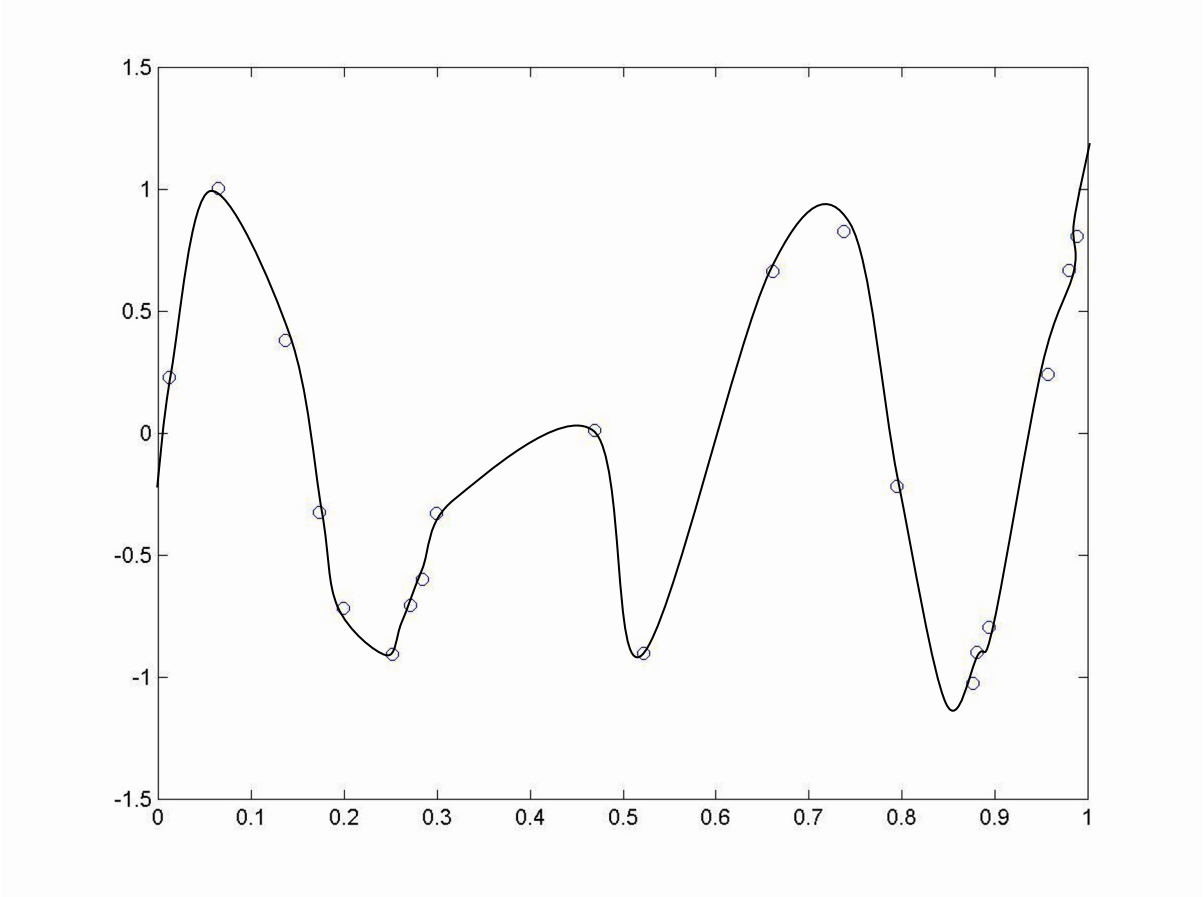
Simplicity is not simple

- What is the relation between y and x?



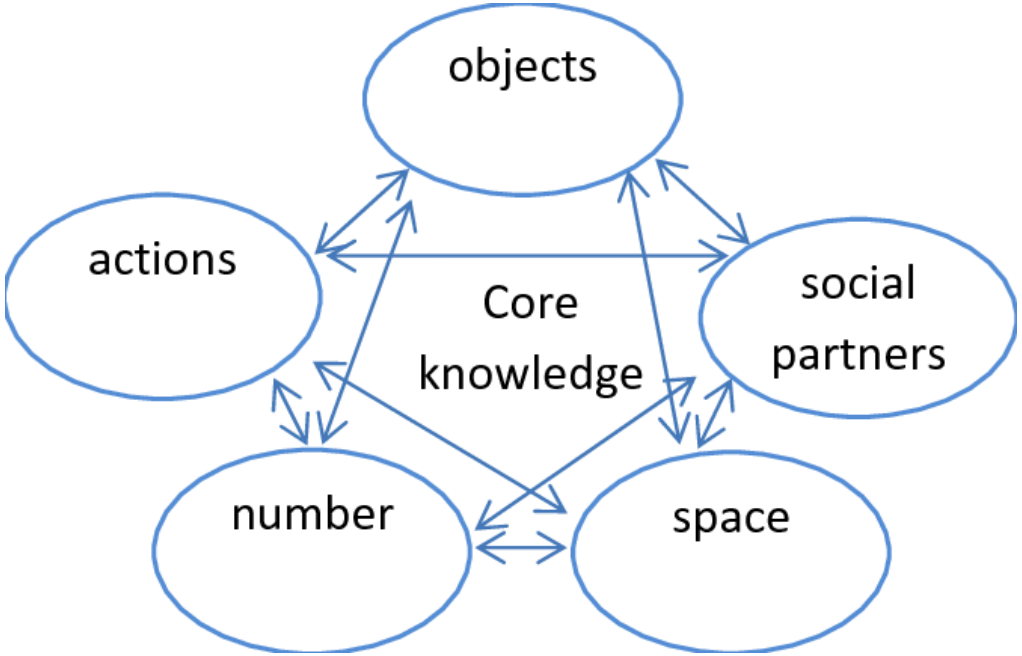
Simplicity is not simple

- What is the relation between y and x?



Sources of inductive constraints

- Occam's razor
  - Hypotheses with fewer arbitrary free parameters (shorter description lengths, smoother) are simpler
- Innate domain-specific knowledge
  - Just the way our brains…
  - Spelke, Core Knowledge System in human cognition



Sources of inductive constraints

- Occam's razor
  - Hypotheses with fewer arbitrary free parameters (shorter description lengths, smoother) are simpler
- Innate domain-specific knowledge
  - Just the way our brains…
  - Spelke, Core Knowledge System in human cognition
- Learned from experience
  - Philosophy of science (Goodman) & Developmental Psychology (Piaget, Carey, Gopnik): Overhypotheses

Goodman's marbles

- Goodman (1955) introduces over hypotheses with an example based on bags of colored marbles
- Suppose we have a stack of bags filled with colored marbles
- Empty several bags and find some contain all black marbles and the rest all white marbles
- Choose a new bag and draw a single black marble
- How many black marbles do you think are in the bag?

Goodman's marbles

- How many black marbles do you think are in the bag?
  - *H*: All of the marbles in the bag
  - *H* is an example of a hypothesis
- Why do you think this?
  - *O*: Each bag in the stack contains marbles that are all the same in color
  - *O* is an example of an overhypothesis
- An overhypothesis is "any abstract knowledge that sets up a hypothesis space at a less abstract level"
- O is an overhypothesis since it sets up a space of hypotheses about the marbles in the bag:
  - All black
  - All white
  - All green
  - etc

Sources of inductive constraints

- Occam's razor
  - Hypotheses with fewer arbitrary free parameters (shorter description lengths, smoother) are simpler
- Innate domain-specific knowledge
  - Just the way our brains…
  - Spelke, Core Knowledge System in human cognition
- Learned from experience
  - Philosophy of science (Goodman) & Developmental Psychology (Piaget, Carey, Gopnik): Overhypotheses
  - Statistics and ML: Hierarchical Bayes or Hierarchical Bayesian Models
    - A two-level inference problem: P(H|O)P(O)

Outline

- Basic Bayesian cognition and Bayesian modeling
  - Theoretical foundations
  - Flipping coins: a generative process → likelihood, prior
  - Bayesian concept learning

Flipping coins as a simple example of Bayesian inference

- You flip a coin 5 times and get the following sequence:

HHTHT

- You flip another coin 5 times and get the following sequence:

HHHHH

- Which coin is more likely to be a fair coin?
- What process produced these sequences (a generative model, likelihood)?
- What are the plausible alternative hypotheses and their priors?

Comparing two simple hypotheses

- Contrast simple hypotheses:
  - H_1: “fair coin”
  - H_2:“always heads”
- Bayes’ rule:

$$P(H \mid D) = \frac{P(H)P(D \mid H)}{P(D)}$$

Comparing two simple hypotheses

- How many heads would you need to see in a row to actually become *suspicious* (~50%) that the coin might be a trick “always heads” coin?

Don’t do a calculation – ask your intuition!

- HHHHH 5
- HHHHHHHHHH 10
- HHHHHHHHHHHHHHH 15
- HHHHHHHHHHHHHHHHHHHH 20

- … To actually be *confident* (~95%) that it was a trick coin?

Comparing two simple hypotheses

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

D: HHTHT

H_1, H_2: "fair coin", "always heads"

$P(D|H_1) = 1/2^5$ $\quad P(H_1) =$?

$P(D|H_2) = 0$ $\quad P(H_2) =$?

Comparing two simple hypotheses

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

D: HHTHT

H_1, H_2: "fair coin", "always heads"

$P(D|H_1) = 1/2^5$ $\quad P(H_1) = 999/1000$

$P(D|H_2) = 0$ $\quad P(H_2) = 1/1000$

$$\frac{P(H_1 \mid D)}{P(H_2 \mid D)} = \frac{1/32}{0} \times \frac{999}{1} = \text{infinity}$$

Comparing two simple hypotheses

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

D: HHHHH

H_1, H_2: "fair coin", "always heads"

$P(D|H_1) = 1/2^5$ $\quad P(H_1) = 999/1000$

$P(D|H_2) = 1$ $\quad P(H_2) = 1/1000$

$$\frac{P(H_1 \mid D)}{P(H_2 \mid D)} = \frac{1}{32} \times \frac{999}{1} \approx 30$$

Comparing two simple hypotheses

- How many heads would you need to see in a row to actually become *suspicious* (~50%) that the coin might be a trick “always heads” coin?

  Don’t do a calculation – ask your intuition!

- … To actually be *confident* (~95%) that it was a trick coin?

Comparing two simple hypotheses

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

D: HHHHHHHHHH

H_1, H_2: "fair coin", "always heads"

$P(D|H_1) = 1/2^{10}$ $\quad P(H_1) = 999/1000$

$P(D|H_2) = 1$ $\quad P(H_2) = 1/1000$

$$\frac{P(H_1 \mid D)}{P(H_2 \mid D)} = \frac{1}{1024} \times \frac{999}{1} \approx 1$$

Comparing two simple hypotheses

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

D: HHHHHHHHHHHHHHH

H_1, H_2: "fair coin", "always heads"

$P(D|H_1) = 1/2^{15}$ $\quad P(H_1) = 999/1000$

$P(D|H_2) = 1$ $\quad P(H_2) = 1/1000$

$$\frac{P(H_1 \mid D)}{P(H_2 \mid D)} = \frac{1}{\sim 32000} \times \frac{999}{1} \approx \sim 0.03$$

An alternative analysis

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

D: HHTHT

H_1, H_2: "fair coin", "coin that always comes up HHTHT"

$P(D|H_1) = 1/2^5$ $\quad P(H_1) = 999/1000$

$P(D|H_2) = 1$ $\quad P(H_2) = 1/1000$

$$\frac{P(H_1 \mid D)}{P(H_2 \mid D)} = \frac{1}{32} \times \frac{999}{1} \approx 30$$

An alternative analysis

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

D: HHHHH

H_1, H_2: "fair coin", "coin that always comes up HHTHT"

$P(D|H_1) = 1/2^5$ $\quad P(H_1) = 999/1000$

$P(D|H_2) = 0$ $\quad P(H_2) = 1/1000$

$$\frac{P(H_1 \mid D)}{P(H_2 \mid D)} = \frac{1/32}{0} \times \frac{999}{1} = \text{infinity}$$

The role of priors

- What looks random, and what doesn't, depends very much on our hypothesis space and prior probabilities.

- The fact that `HHTHT` looks representative of a fair coin, and `HHHHH` does not, reflects our intuitive theories of physics, design, psychology, ...
  - Easy to imagine how a trick all-heads coin could be made: high prior probability.
  - Hard to imagine how a trick "`HHTHT`" coin could be made: low prior probability.

Low prior or zero prior?

- Is there any evidence you could see that would make you suspect you had a trick coin that always comes up “HHTHT”?

- How about…

HHTHT

Low prior or zero prior?

- Is there any evidence you could see that would make you suspect you had a trick coin that always comes up “HHTHT”?

- How about…

HHTHTHHTHT

Low prior or zero prior?

- Is there any evidence you could see that would make you suspect you had a trick coin that always comes up “HHTHT”?

- How about…

HHTHTHHTHTHHTHT

Low prior or zero prior?

- Is there any evidence you could see that would make you suspect you had a trick coin that always comes up “HHTHT”?

- How about…

HHTHTHHTHTHHTHTHHTHT

Low prior or zero prior?

- Is there any evidence you could see that would make you suspect you had a trick coin that always comes up “HHTHT”?

- How about…

HHTHTHHTHTHHTHTHHTHTHHTHT

Low prior or zero prior?

- A random coin you picked up on the side of the street
- What if a talented engineer trained by Hopkins gave you the coin?
- What if a talented engineer who likes to prank people gave you the coin?

Outline

- Basic Bayesian cognition and Bayesian modeling
  - Theoretical foundations
  - Flipping coins: a ***generative*** process → likelihood, prior
  - Bayesian concept learning

Outline

- Basic Bayesian cognition and Bayesian modeling
  - Theoretical foundations
  - Flipping coins
  - Bayesian concept learning

Bayesian concept learning

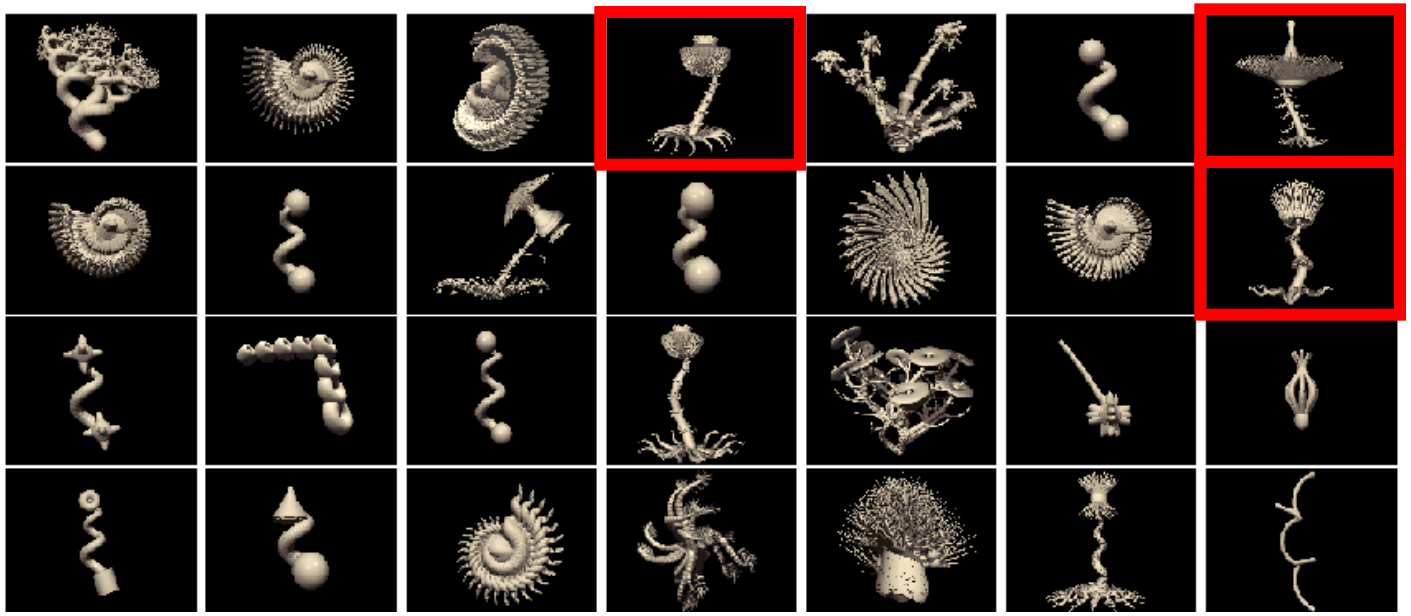
"horse"

"horse"

"horse"

For Cog Sci: simplified version that gets to the core of human cognition

"tufa"

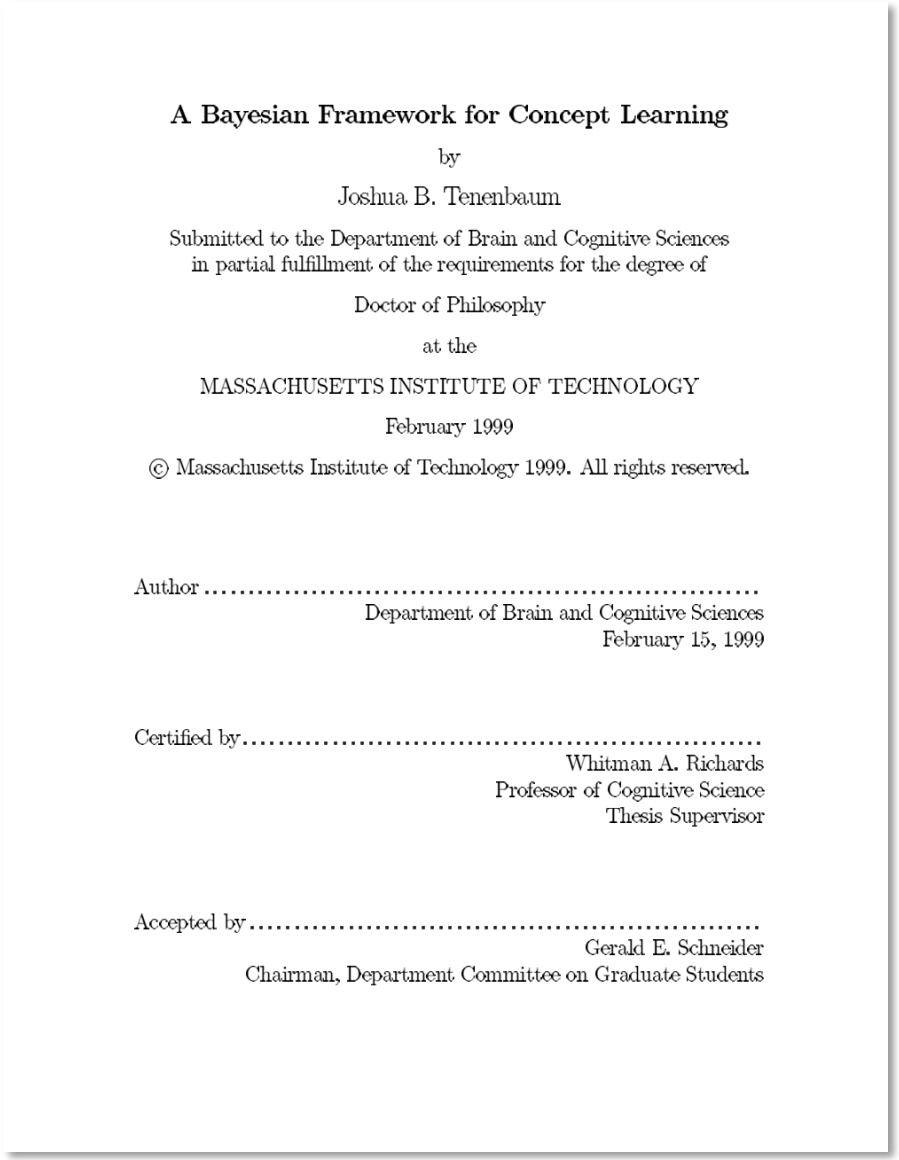


For AI: principled approach for solving more complex problems

"tufa"

"tufa"

A minimum domain for Bayesian concept learning: The number game



A Bayesian Framework for Concept Learning

by

Joshua B. Tenenbaum

Submitted to the Department of Brain and Cognitive Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

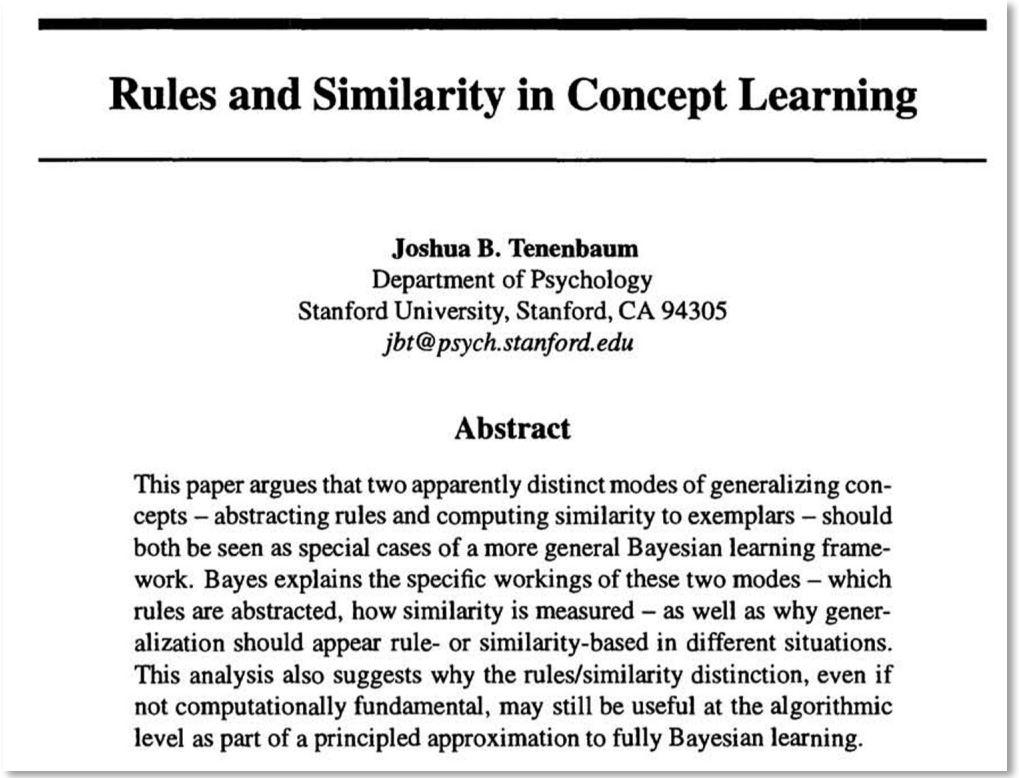
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1999

Author .....
Department of Brain and Cognitive Sciences
February 15, 1999

Certified by.....
Whitman A. Richards
Professor of Cognitive Science
Thesis Supervisor

Accepted by.....
Gerald E. Schneider
Chairman, Department Committee on Graduate Students



Rules and Similarity in Concept Learning

Joshua B. Tenenbaum
Department of Psychology
Stanford University, Stanford, CA 94305
jbt@psych.stanford.edu

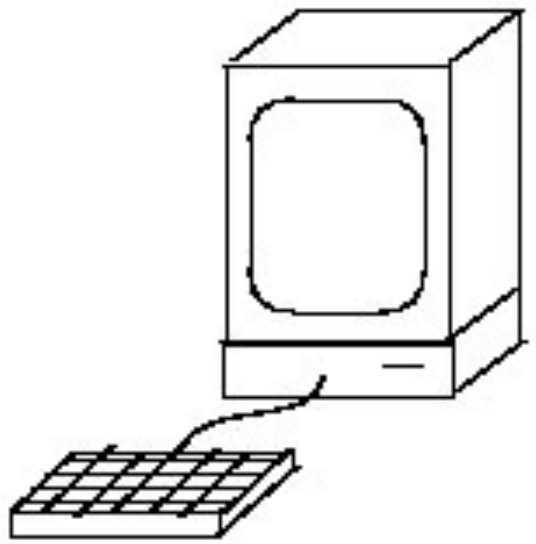
Abstract

This paper argues that two apparently distinct modes of generalizing concepts – abstracting rules and computing similarity to exemplars – should both be seen as special cases of a more general Bayesian learning framework. Bayes explains the specific workings of these two modes – which rules are abstracted, how similarity is measured – as well as why generalization should appear rule- or similarity-based in different situations. This analysis also suggests why the rules/similarity distinction, even if not computationally fundamental, may still be useful at the algorithmic level as part of a principled approximation to fully Bayesian learning.

NeurIPS 1999

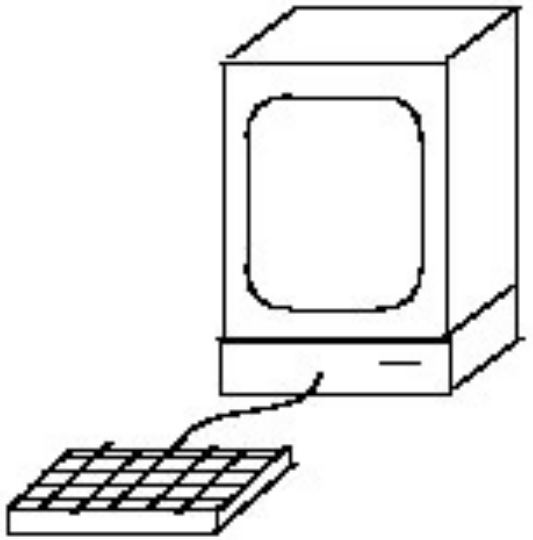
Too old? Still relevant in 2024?

The number game



- Program input: number between 1 and 100
- Program output: “yes” or “no”

The number game



- Your task:
  - Observe one or more positive (“yes”) examples.
  - Judge whether other numbers are “yes” or “no”.

The number game

One positive example: 60

What other numbers do you think are likely to be accepted?

50, 20, 40, 6

The number game

Four positive examples: 60, 80, 10, 30

20, 40, 50, 70, 100, 90

The number game

Four positive examples: 60, 52, 57, 55

59, 50, 56, 58

The number game

Results from a human experiment

Examples of “yes” numbers	Generalization judgments (N = 20)