Computational Cognitive Modeling

Bayesian modeling

Brenden Lake & Todd Gureckis

email address for instructors:
instructors-ccm-spring2020@nyuccl.org

course website:
https://brendenlake.github.io/CCM-site/

The problem of induction

Original thought experiment due to W. V. Quine (1960).

The problem of induction

A mug?

A mug *on* a table?

A white mug on a white marble table?

Coffee?

3 pm?

Handle?

Marble?

Smell?

A beverage?

White objects?

A mug filled with coffee?

“gavagai”

Location?

Ceramics?

Original thought experiment due to W. V. Quine (1960).

The problem of induction

now you get more data…

“gavagai”

The problem of induction

now you get more data…

A mug?

A mug *on* a table?

Coffee?

A white mug on a white marble table?

3 pm?

Marble?

Handle?

Smell?

White objects?

A beverage?

“gavagai”

A mug filled with coffee?

Location?

Ceramics?

How do we learn so much from so little?

- If our inferences go beyond the data given, then something must be making up the difference…

- What is it? **constraints** (as described by psychologists and linguists), **inductive biases** (machine learning and AI researchers), **priors** (statisticians), 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?

For more discussion see, Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022), 1279-1285.

Bayesian modeling is an approach for understanding inductive problems, and it typically takes a strong "top-down" strategy

Three levels of description *(David Marr, 1982)*

Computational

Why do things work the way they do?
What is the goal of the computation?
What are the unifying principles?

$$P(h_i|D) = \frac{P(D|h_i)P(h_i)}{\sum_j P(D|h_j)P(h_j)}$$

Algorithmic

What representations can implement such computations?
How does the choice of representations determine the algorithm?

Implementational

How can such a system be built in hardware?
How can neurons carry out the computations?

Key principles of Bayesian models of cognition

- Start by analyzing the computational problem at hand, and describe it as a problem of **Bayesian inference**

- A successful "computational level" account **provides strong constraints when developing an "algorithmic" and "implementational" level accounts**

- Bayesian inference provides a flexible framework for testing different types of representation, without having to worry about defining special algorithms for inference and learning

Bayesian inference for evaluating hypotheses in light of data

Data (D): John is coughing

Hypotheses:

$h_1 =$ John has a cold

$h_2 =$ John has emphysema

$h_3 =$ John has a stomach flu

"Bayes' rule"

posterior: $P(h_i|D)$ — likelihood: $P(D|h_i)$ — prior: $P(h_i)$

$$P(h_i|D) = \frac{P(D|h_i)P(h_i)}{\sum_j P(D|h_j)P(h_j)}$$

Which hypotheses should we believe, and with what certainty?

We want to calculate the posterior probabilities: $P(h_1|D)$, $P(h_2|D)$, and $P(h_3|D)$

Review Russell & Norvig reading for basics on probability. This example is from Josh Tenenbaum.

Bayesian inference

posterior — likelihood — prior

$$P(h_i|D) = \frac{P(D|h_i)P(h_i)}{\sum_j P(D|h_j)P(h_j)}$$

Data (D): John is coughing

Hypotheses:

$h_1 =$ John has a cold

$h_2 =$ John has emphysema

$h_3 =$ John has a stomach flu

$P(h_1) = .75 \quad P(D|h_1) = 1$

$P(h_2) = .05 \quad P(D|h_2) = 1$

$P(h_3) = .2 \quad P(D|h_3) = .2$

Prior favors h_1 and h_3, over h_2

Likelihood favors h_1 and h_2, over h_3

Posterior favors h_1, over h_2 and h_3

$$P(h_1|D) = .89 = \frac{.75(1)}{.75(1) + .05(1) + .2(.2)}$$

$P(h_2|D) = .06$

$P(h_3|D) = .05$

Where does Bayes' rule come from?

Definition of conditional probability:

$$P(a|b) = \frac{P(a,b)}{P(b)}$$

"product rule"

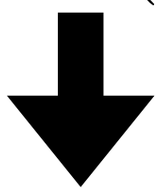
$$P(a,b) = P(a|b)P(b)$$

Derivation

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

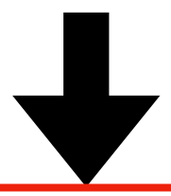
product rule applied twice

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



$$P(h|D)P(D) = P(D|h)P(h)$$

Equating the two right hand sides



$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} = \frac{P(D|h)P(h)}{\sum_{h'} P(D|h')P(h')}$$

Divide by $P(D)$

Why is this a reasonable way to represent beliefs?

- If your beliefs are inconsistent with axioms of probability, someone can take advantage of you in gambling (see example from Russel and Norvig reading)

Agent 1		Agent 2		Outcome for Agent 1			
Proposition	Belief	Bet	Stakes	$a \wedge b$	$a \wedge \neg b$	$\neg a \wedge b$	$\neg a \wedge \neg b$
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

Figure 13.2 Because Agent 1 has inconsistent beliefs, Agent 2 is able to devise a set of bets that guarantees a loss for Agent 1, no matter what the outcome of a and b.

- Also, Bayes' rule provides a very general account of learning, where prior knowledge can be combined with data to update beliefs

Bayesian concept learning with the number game

Rules and Similarity in Concept Learning

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

In *Advances in neural information processing systems (1999)*

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.

(You will work with this model in homework 3)

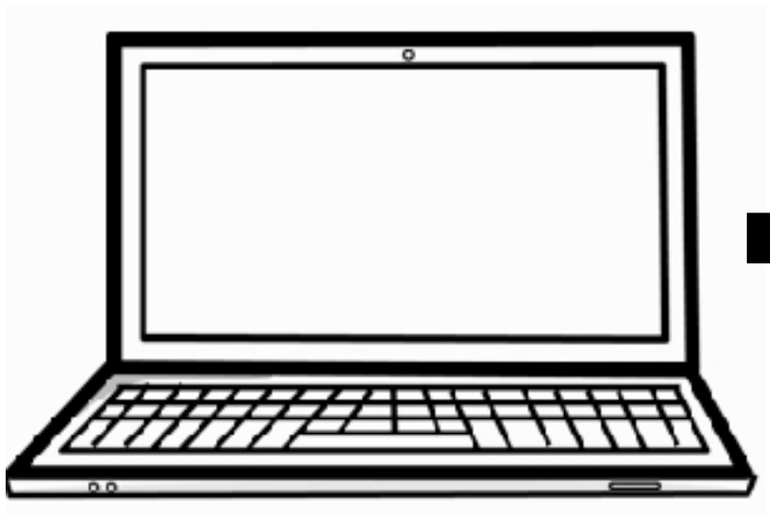
1 Introduction

In domains ranging from reasoning to language acquisition, a broad view is emerging of cognition as a hybrid of two distinct modes of computation, one based on applying abstract rules and the other based on assessing similarity to stored exemplars [7]. Much support for this view comes from the study of concepts and categorization. In generalizing concepts, people's judgments often seem to reflect both rule-based and similarity-based computations [9], and different brain systems are thought to be involved in each case [8]. Recent psychological models of classification typically incorporate some combination of rule-based and similarity-based modules [1,4]. In contrast to this currently popular modularity position, I will argue here that rules and similarity are best seen as two ends of a continuum of possible concept representations. In [11,12], I introduced a general theoretical framework to account

The number game

There is an unknown computer program that generates numbers in the range 1 to 100. You are provided with a small set of random examples from this program.

1 random “yes” example



60

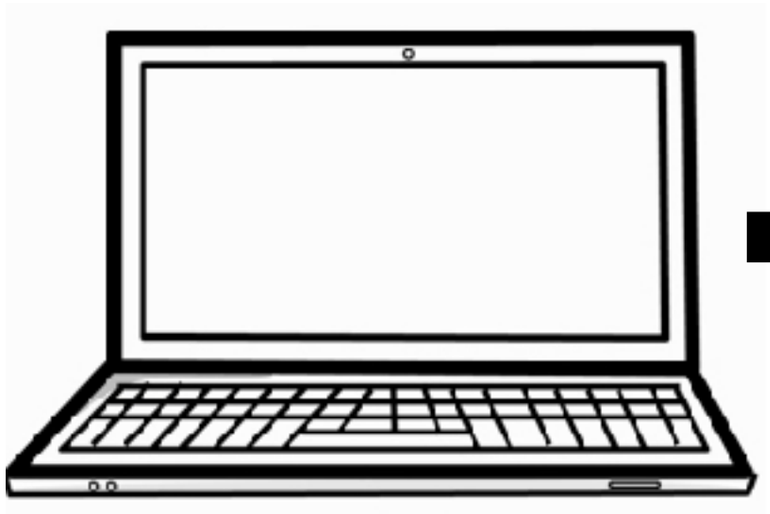
Which numbers will be accepted by the same computer program?

51? 58? 20?

The number game

There is an unknown computer program that generates numbers in the range 1 to 100. You are provided with a small set of random examples from this program.

4 random “yes” examples



60 80 10 30

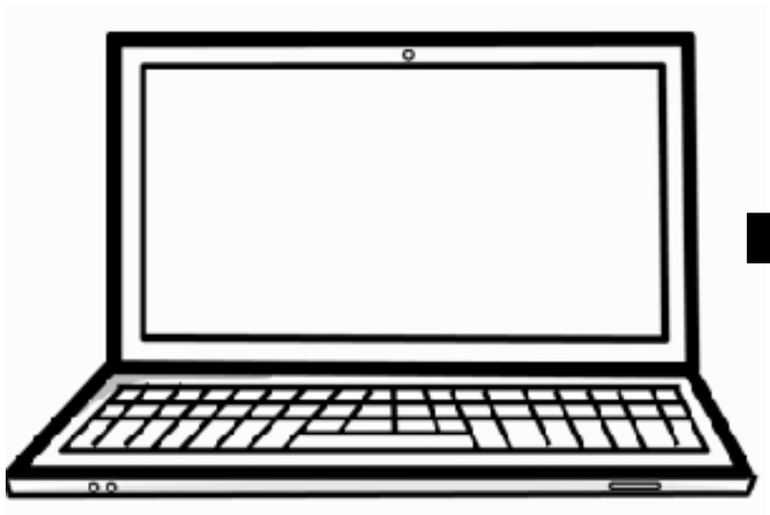
Which numbers will be accepted by the same computer program?

51? 58? 20?

The number game

There is an unknown computer program that generates numbers in the range 1 to 100. You are provided with a small set of random examples from this program.

4 random “yes” examples



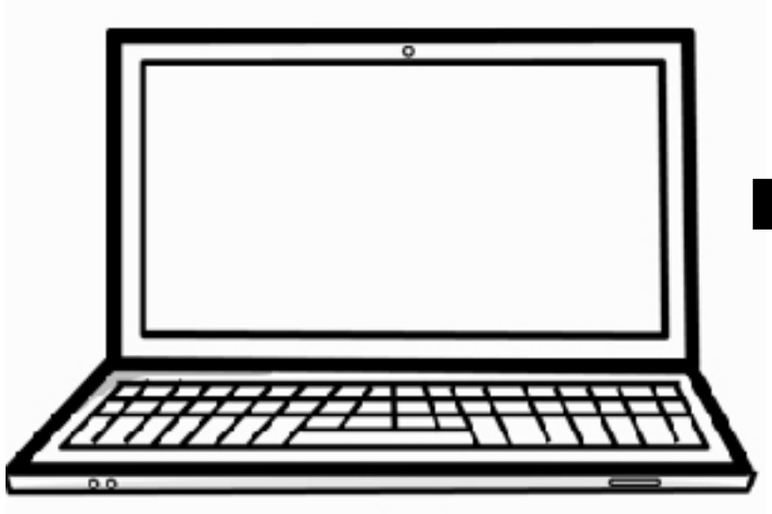
60 52 57 55

Which numbers will be accepted by the same computer program?

51? 58? 20?

A Bayesian model of the number game

random “yes” examples of an **unknown concept** ***C***



Observations

$$X = \{x^{(1)}, \ldots, x^{(n)}\}$$

$$= \{60, 52, 57, 55\}$$

Predictions:

Which numbers ***y*** will be accepted by the same computer program ***C***?

$$P(y \in C \mid X)$$

51? 58? 20?

$P(51 \in C \mid X)$ $P(58 \in C \mid X)$ $P(20 \in C \mid X)$

A Bayesian model of the number game

We have observations:

$$X = \{x^{(1)}, ..., x^{(n)}\}$$

We have a space of hypotheses, which are sets of numbers $h \in H$

and prior $P(h)$ (more details next slide)

- *mathematical hypotheses:* odd numbers (h = [1, 3, 5, …, 99]), even numbers (h = [2, 4, 6, … ,100]), square numbers (h = [1, 4, 9, 16]), cube numbers, primes, multiples of n, etc.

- *interval hypotheses:* continuous intervals of the number line

Likelihood $P(X|h)$

$$P(X|h) = \prod_{i=1}^{n} P(x^{(i)}|h)$$

(assumption that examples are independent)

$$P(x^{(i)}|h) = \frac{1}{|h|} \text{ if } x^{(i)} \in h$$

$$= 0 \text{ otherwise}$$

|*h*| is the "size" of *h*

Bayes' rule for computing posterior beliefs:

$$P(h|X) = \frac{P(X|h)P(h)}{\sum_{h' \in H} P(X|h')P(h')}$$

A Bayesian model of the number game

The hypothesis space and prior

Mathematical hypotheses

- odd numbers
- even numbers
- square numbers
- cube numbers
- primes
- multiples of n, such that 3 ≤ n ≤ 12
- powers of n, such that 2 ≤ n ≤ 10
- numbers ending in n, such that 0 ≤ n ≤ 9

(Mathematical hypotheses are equally likely in the prior)

$$P(h)$$

Interval hypotheses

- Intervals between n and m, such that 1 ≤ n ≤ 100; n ≤ m ≤ 100

(Interval hypotheses reweighted to favor intermediate sizes) $P(h)$

λ is free parameter that trades off “math” vs. “interval” hypotheses

A Bayesian model of the number game

We have observations:

$$X = \{x^{(1)}, ..., x^{(n)}\}$$

We want to make predictions for new numbers y:

$$P(y \in C \mid X)$$

Bayes' rule for computing posterior beliefs:

$$P(h|X) = \frac{P(X|h)P(h)}{\sum_{h' \in H} P(X|h')P(h')}$$

Posterior predictions about new example ***y:***

$$P(y \in C \mid X) = \sum_{h \in H} P(y \in C \mid h)P(h|X)$$

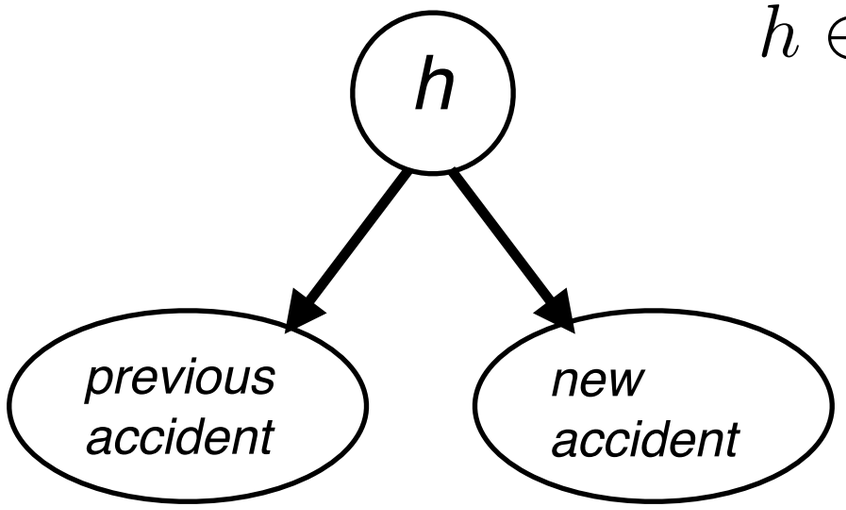
first term is 1 or 0 based on membership

Bayesian hypothesis averaging : when making Bayesian predictions, one must average over all possible hypotheses, weighted by their posterior belief

Examples: Bayesian hypothesis averaging

Say you are an insurance company, and you want to predict which customers are more likely to get in a car accident.

$$P(\text{new accident}|\text{previous accident}) = \sum_h P(\text{new accident}|h)P(h|\text{previous accident})$$



$$h \in \{\text{good driver}, \text{bad driver}\}$$

also known as “marginalization” of a variable (*h*)

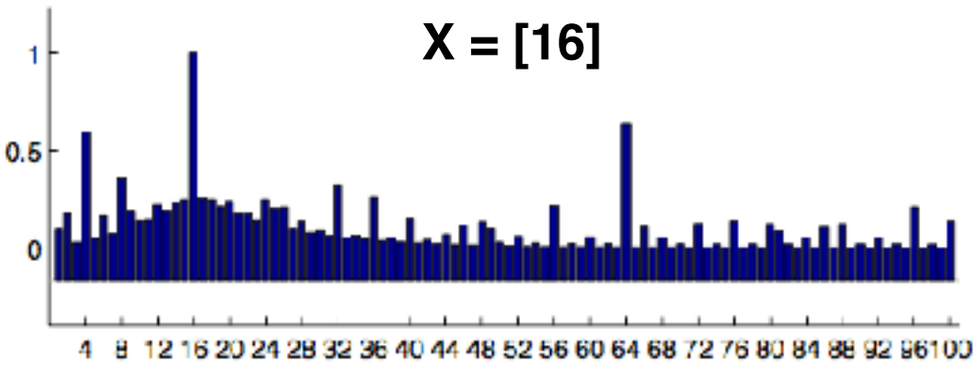
Another example (the previous evidence does not necessarily need to be relevant)

$$P(\text{new accident}|\text{born in Feb.}) = \sum_h P(\text{new accident}|h)P(h|\text{born in Feb.})$$

$$h \in \{\text{good driver}, \text{bad driver}\}$$

The **size principle:** hypotheses with smaller extensions are more likely than hypotheses with larger extensions

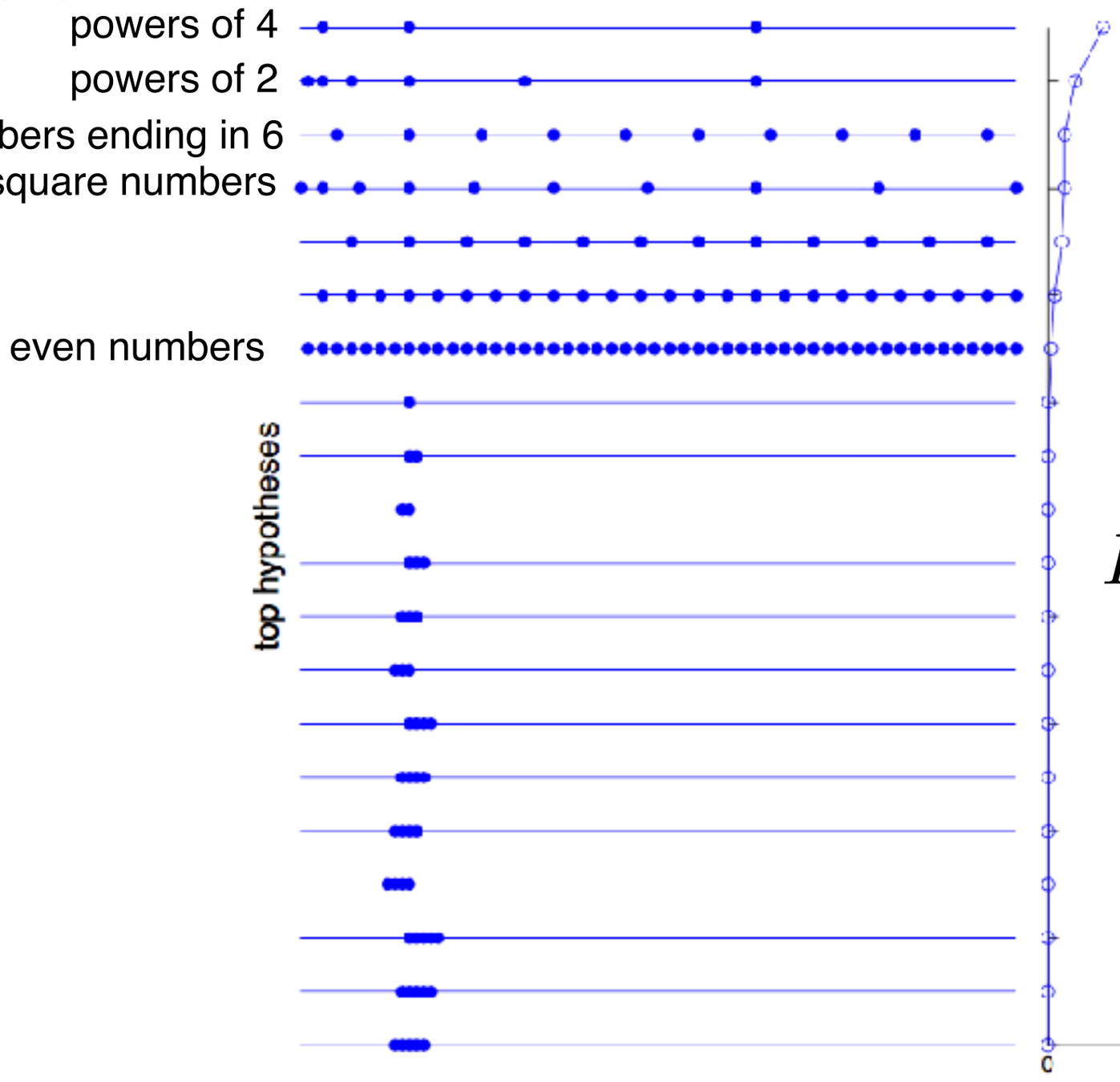
$P(y \in C|X)$



Likelihood

$$P(x^{(i)}|h) = \frac{1}{|h|} \text{ if } x^{(i)} \in h$$
$$= 0 \text{ otherwise}$$

Most likely hypotheses

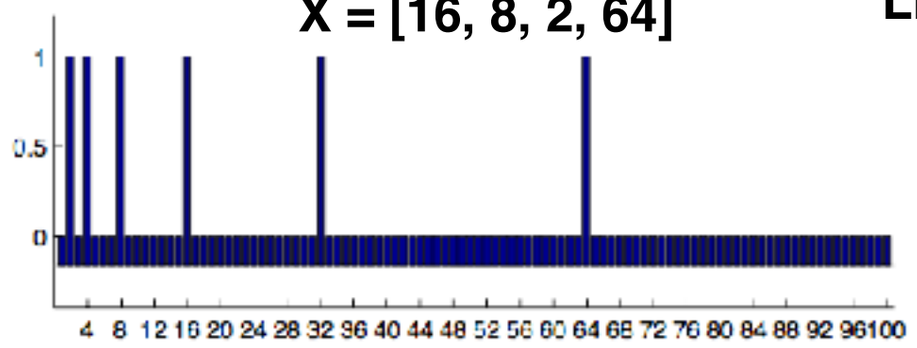


$P(h|X)$

The **size principle:** hypotheses with smaller extensions are more likely than hypotheses with larger extensions

$P(y \in C|X)$

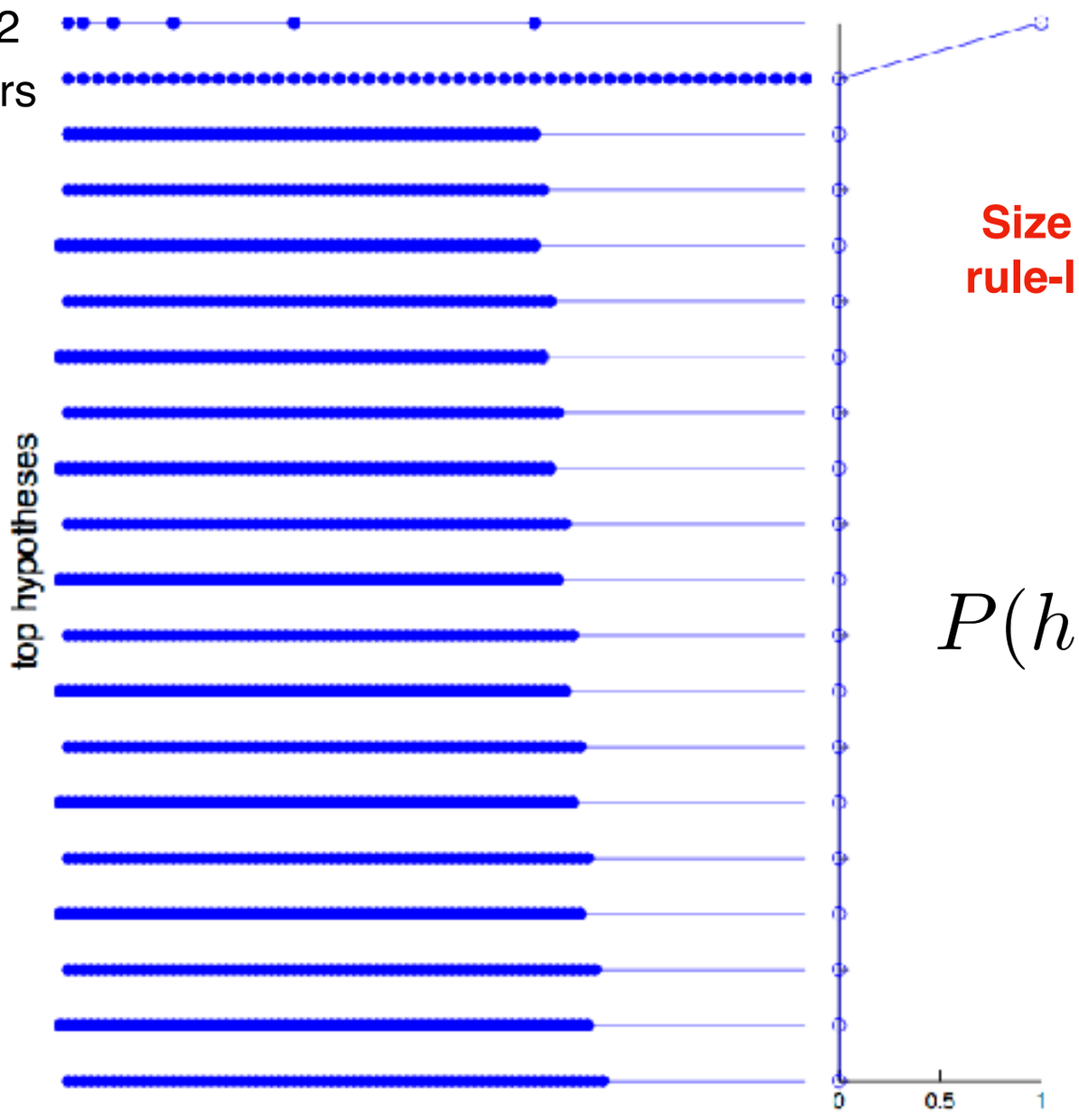
X = [16, 8, 2, 64]



Most likely hypotheses

powers of 2

even numbers



Likelihood **compounding evidence**

$$P(X|h) = \prod_{i=1}^{n} P(x^{(i)}|h)$$

$$P(x^{(i)}|h) = \frac{1}{|h|} \text{ if } x^{(i)} \in h$$
$$= 0 \text{ otherwise}$$

Size principle leads to very sharp, rule-like generalizations from just a few examples

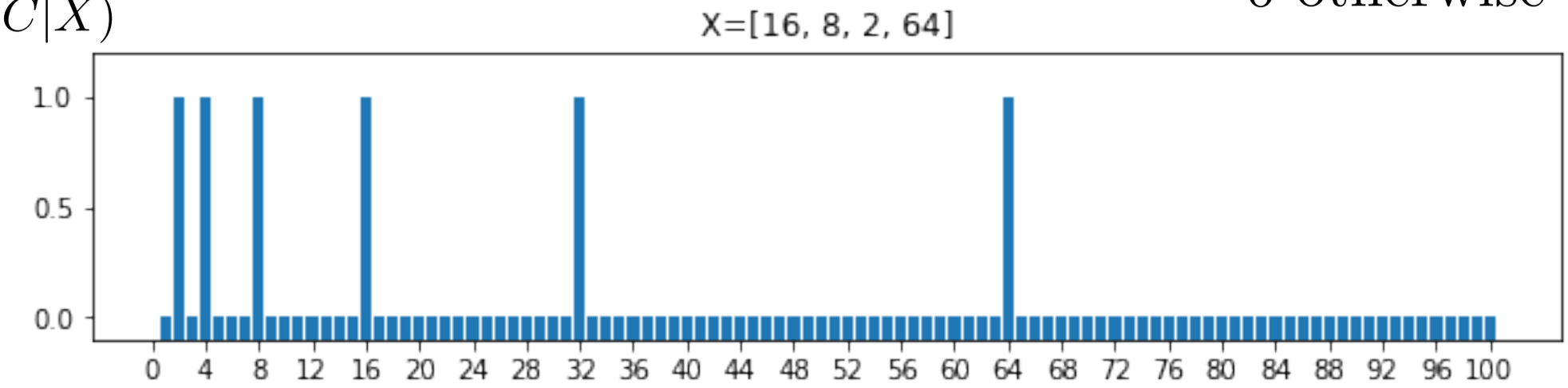
$P(h|X)$

How the size principle influences generalization

With size principle (strong sampling):

$$P(x^{(i)}|h) = \frac{1}{|h|} \text{ if } x^{(i)} \in h$$
$$= 0 \text{ otherwise}$$

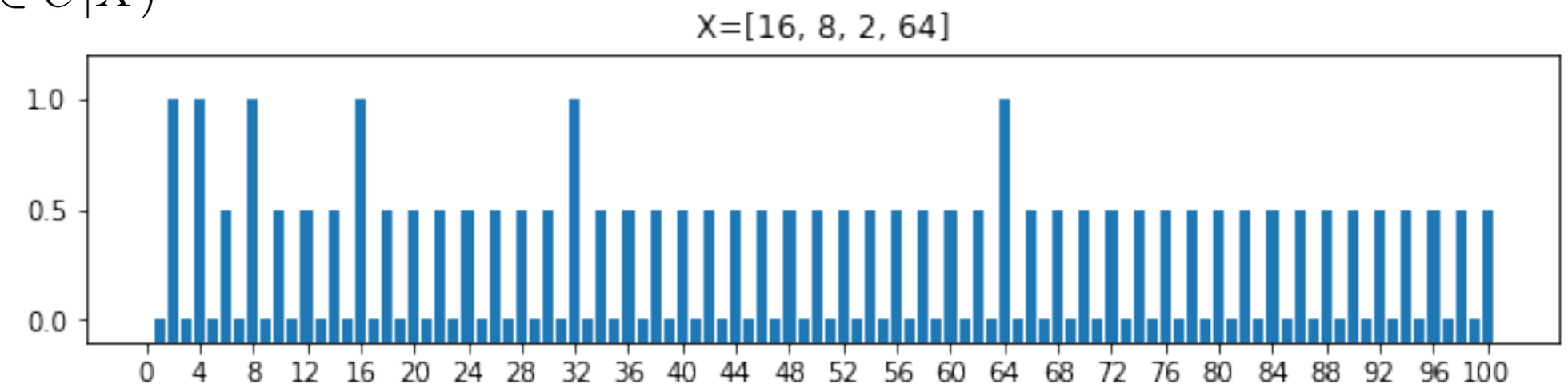
$P(y \in C|X)$



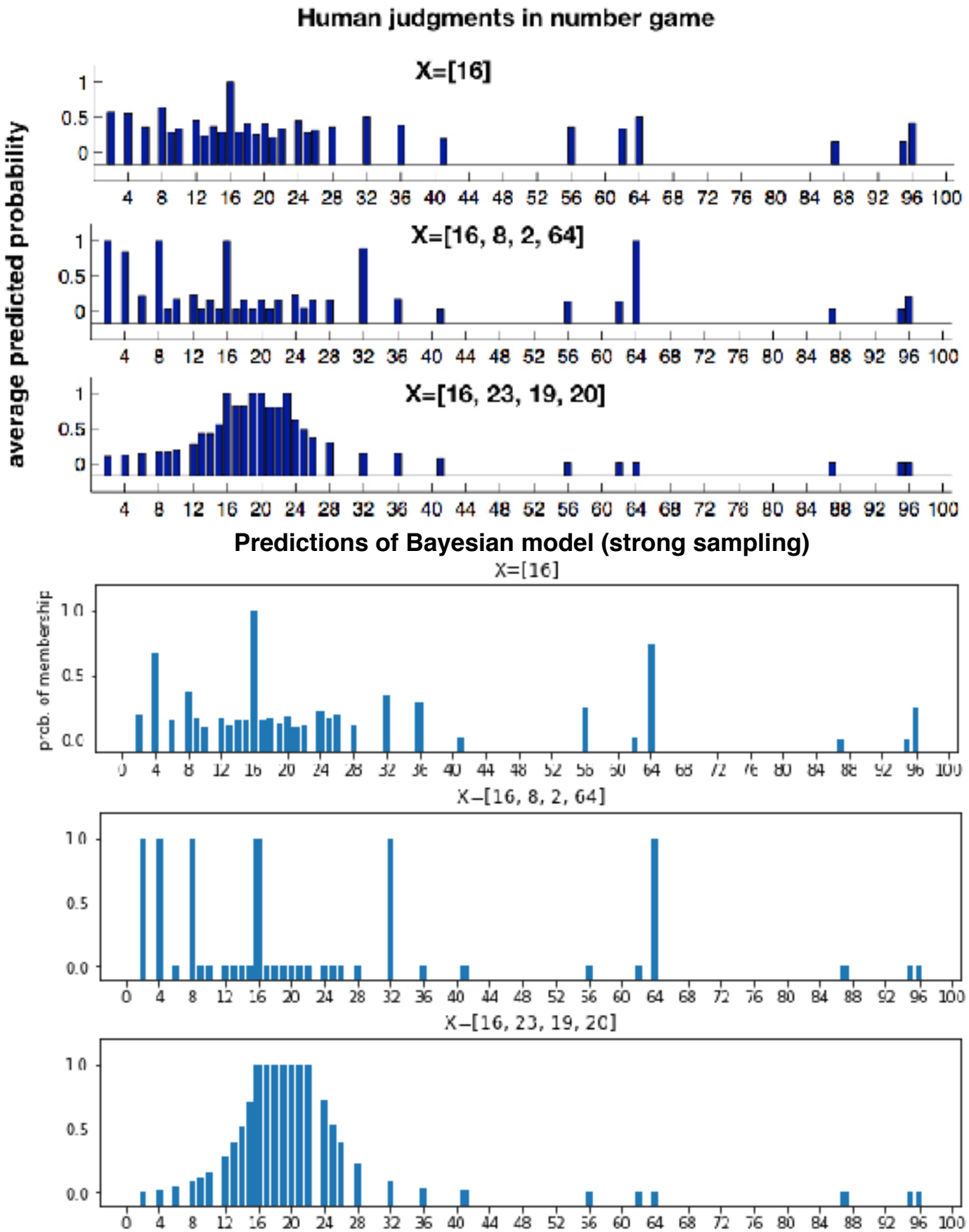
Without size principle (weak sampling):

$$P(X|h) = 1 \text{ if } x^{(i)} \in h \text{ for all } i$$
$$= 0 \text{ otherwise}$$

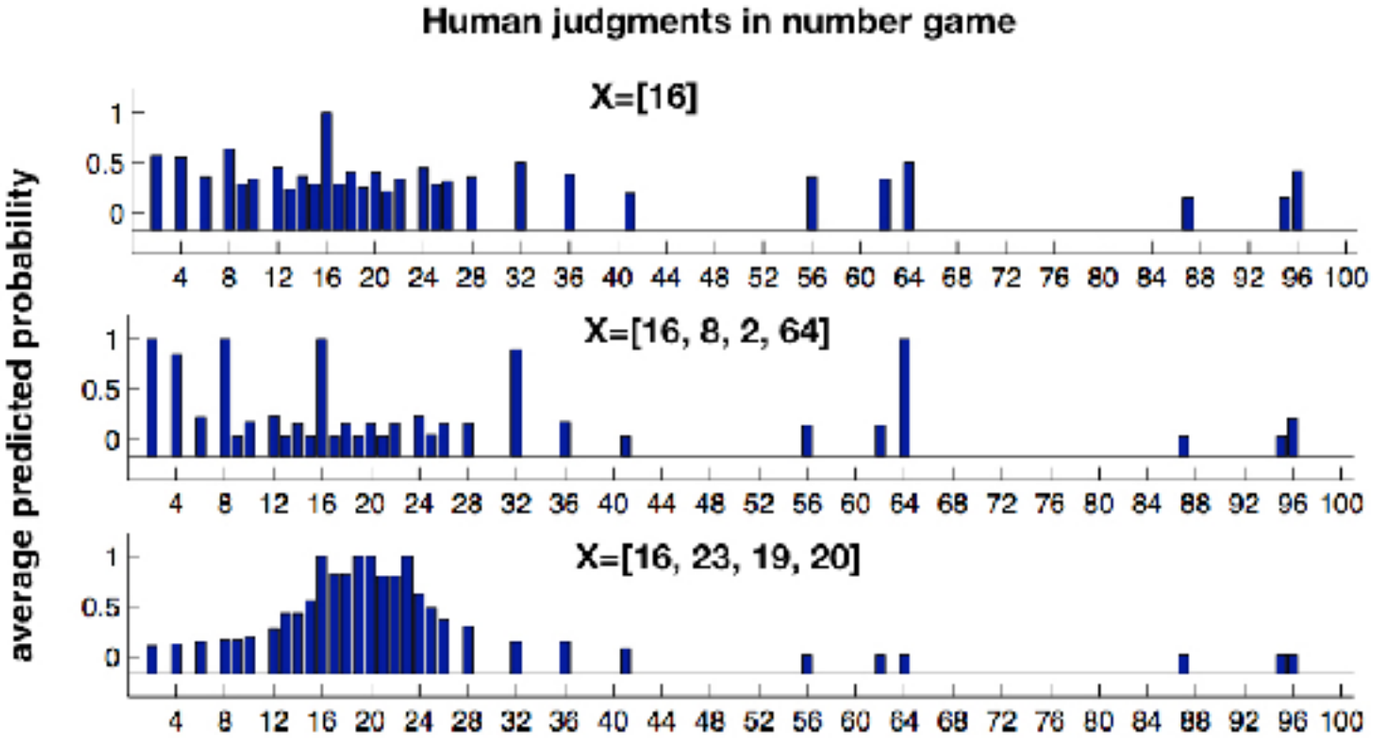
$P(y \in C|X)$

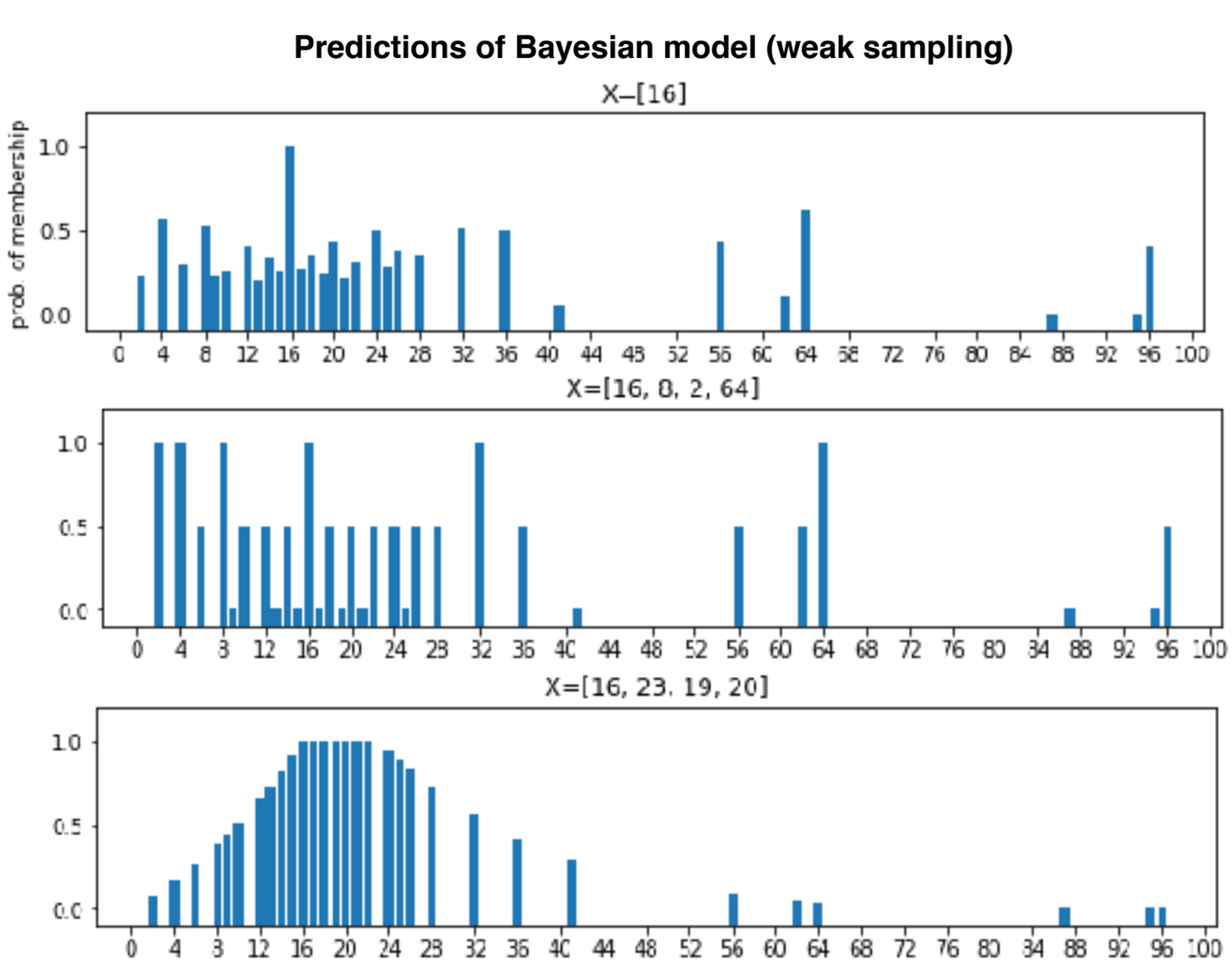


Human vs. model predictions in the number game



Human vs. model predictions in the number game (weak sampling)





(weak sampling does not capture the sharpness of people's generalization curves)

Conclusions from Bayesian concept learning and the number game

• People can make meaningful predictions from very sparse evidence, aided by strong assumptions for how the data is generated (strong sampling)

• People display a mixture of both “rule-like” and “similarity-like” generalizations, depending on what the data entails — where most previous psychological theories posited two different mechanisms, one for rules and one for similarity

• A Bayesian account of concept learning displays both of these characteristics, and can make quantitative predictions regarding how people generalize to new examples.

• Discussion point: Where does the hypothesis space come from?
(see final project idea on “Bayesian modeling / Probabilistic programming - Number game”)

Word learning as Bayesian inference

(Xu and Tenenbaum, 2007, *Psychological Review*)

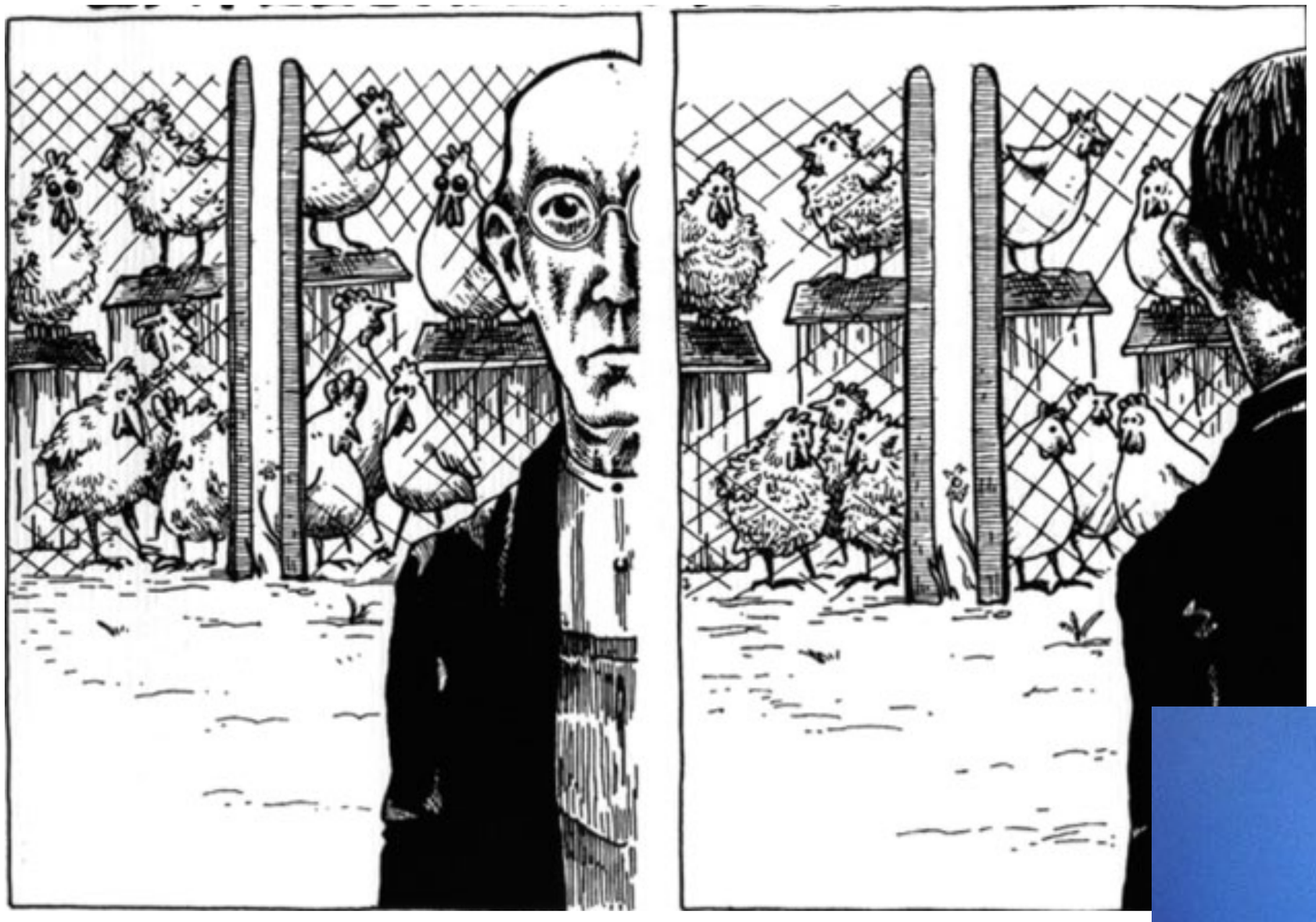
Prompt: "This is a dax"

Children's generalizations

Bayesian concept learning with tree-structured hypothesis space

Slide credit: Josh Tenenbaum

Categorical perception: A link between categorization and perception/discrimination



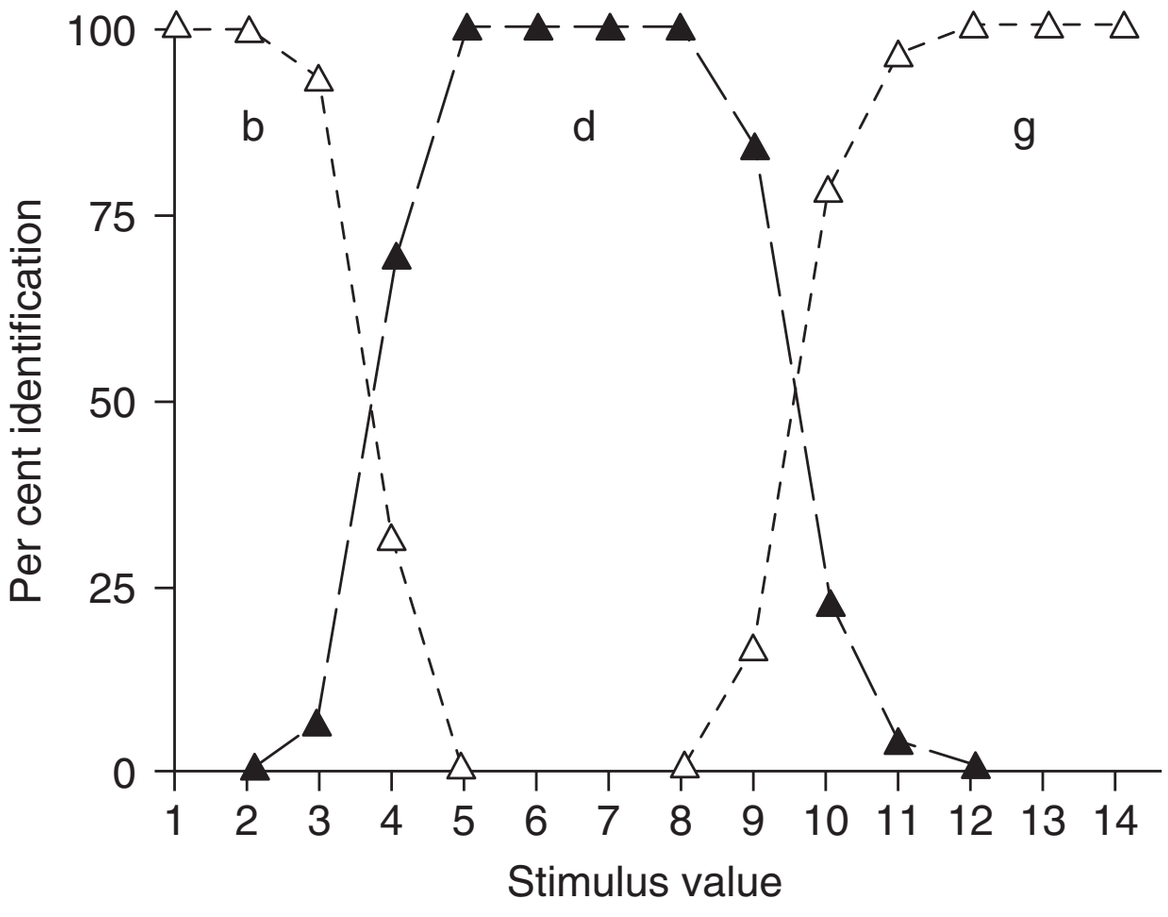
From Goldstone and Hendrickson (2009)

Categorical perception in speech

A link between categorization and discrimination

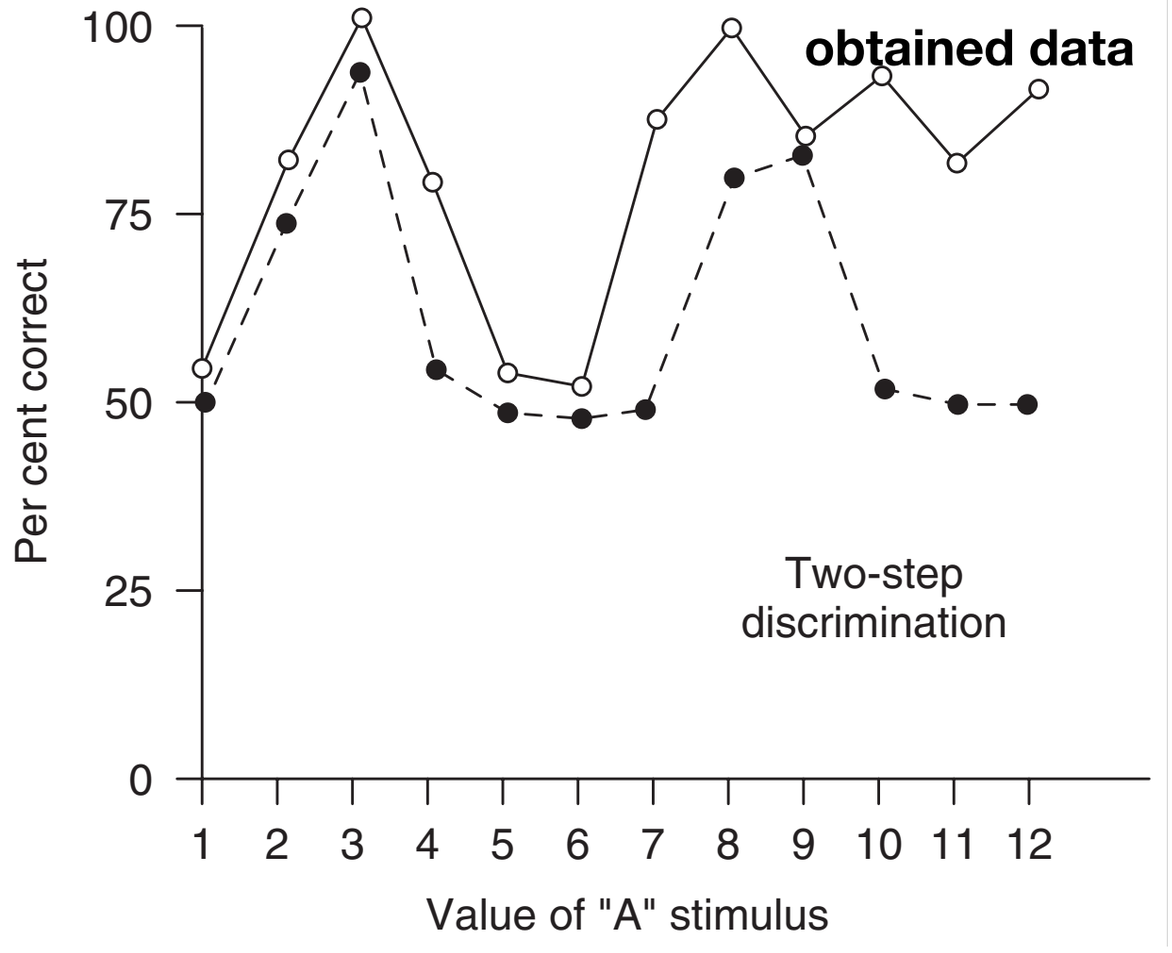
Identification (labeling) task

("ba" vs. "da" vs. "ga")



Discrimination task

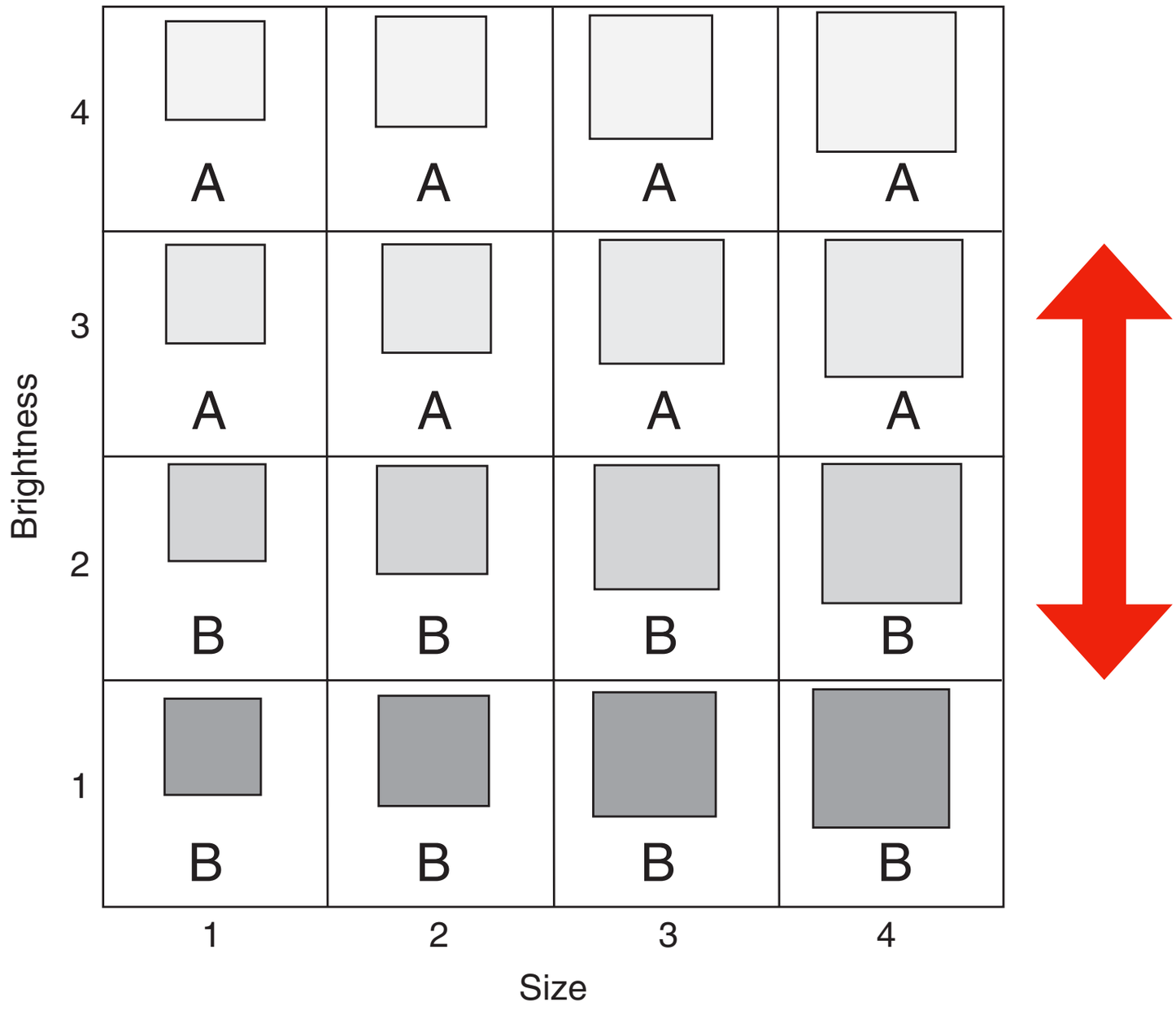
(ABX; which is X identical to, A or B?)



From Liberman, A. M., Harris, K. S., Hoffman, H. S., & Griffith, B. C. (1957). The discrimination of speech sounds within and across phoneme boundaries. Journal of experimental psychology, 54(5), 358.

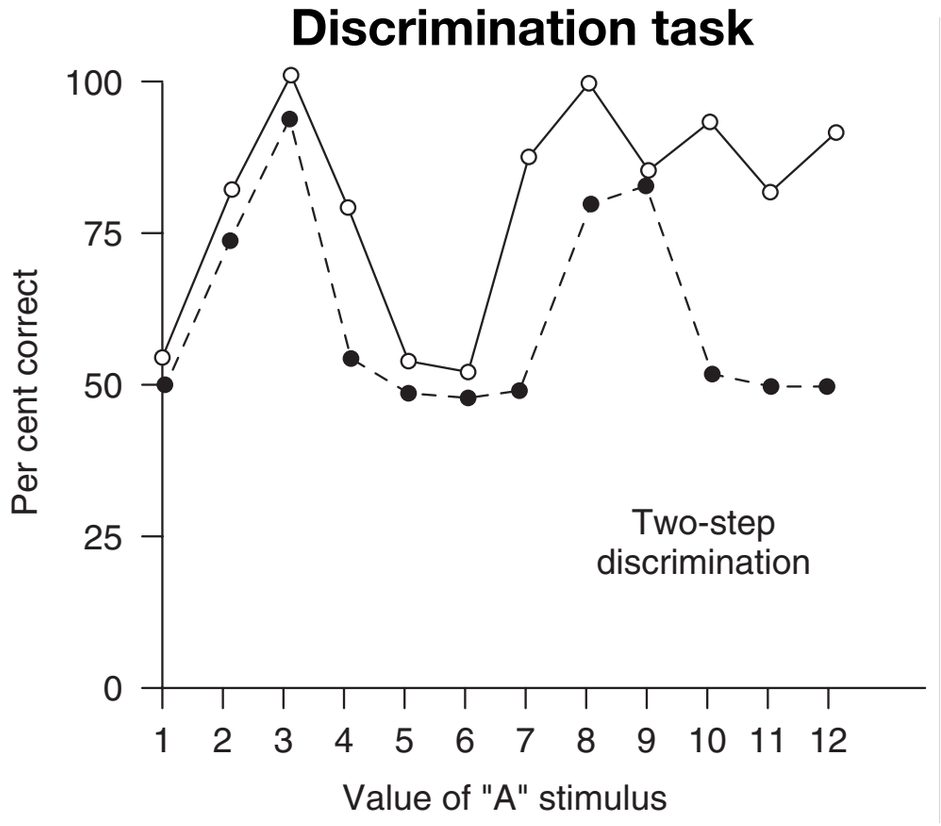
Categorical perception for artificial visual categories

A link between categorization and discrimination

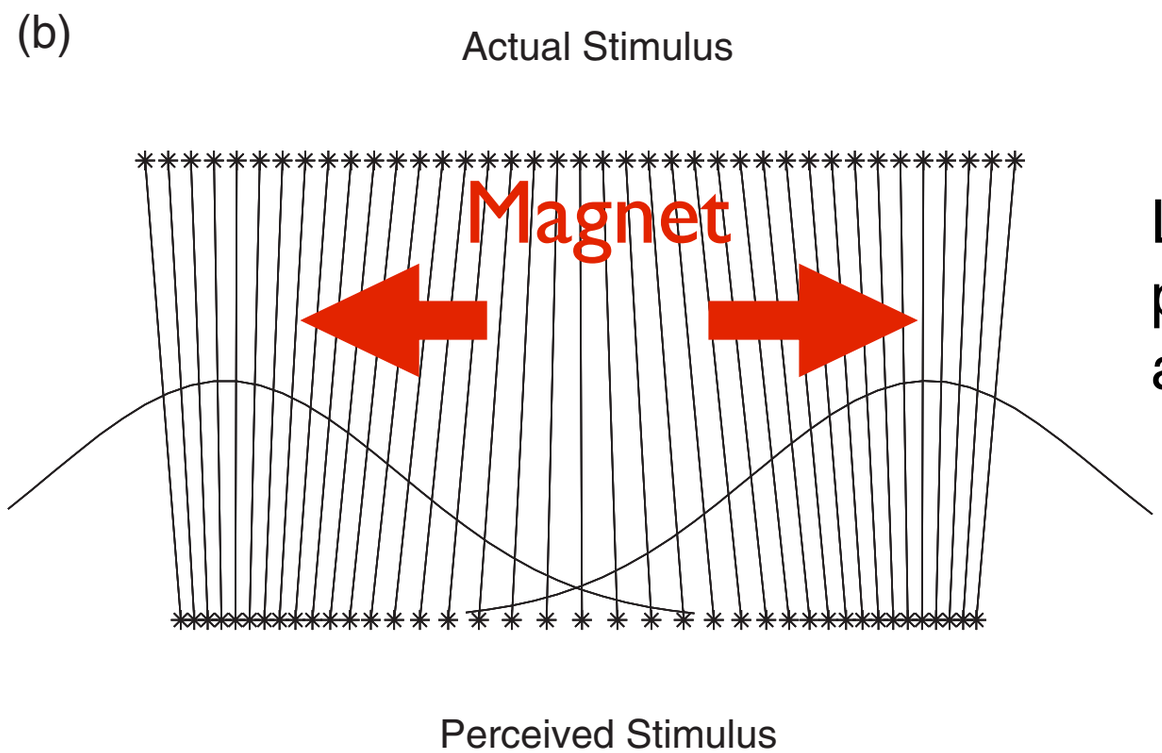


Goldstone, R. L. (1994). Influences of categorization on perceptual discrimination. Journal of Experimental Psychology: General, 123(2), 178.

Categorical perception: A link between categorization and discrimination



Categorical perception is very closely related to the “perceptual magnet effect” in speech studied by Pat Kuhl et al., which describes categories as *pulling your perception* toward the categories centers



Let’s try to understand the perceptual magnet effect through a Bayesian model

Psychological Review
2009, Vol. 116, No. 4, 752–782

0033-295X/09/$12.00 DOI: 10.1037/a0017196

The Influence of Categories on Perception: Explaining the Perceptual Magnet Effect as Optimal Statistical Inference

Naomi H. Feldman
Brown University

Thomas L. Griffiths
University of California, Berkeley

James L. Morgan
Brown University

A variety of studies have demonstrated that organizing stimuli into categories can affect the way the stimuli are perceived. We explore the influence of categories on perception through one such phenomenon, the perceptual magnet effect, in which discriminability between vowels is reduced near prototypical vowel sounds. We present a Bayesian model to explain why this reduced discriminability might occur: It arises as a consequence of optimally solving the statistical problem of perception in noise. In the optimal solution to this problem, listeners' perception is biased toward phonetic category means because they use knowledge of these categories to guide their inferences about speakers' target productions. Simulations show that model predictions closely correspond to previously published human data, and novel experimental results provide evidence for the predicted link between perceptual warping and noise. The model unifies several previous accounts of the perceptual magnet effect and provides a framework for exploring categorical effects in other domains.

Keywords: perceptual magnet effect, categorical perception, speech perception, Bayesian inference, rational analysis

The influence of categories on perception is well known in domains ranging from speech sounds to artificial categories of objects. Liberman, Harris, Hoffman, and Griffith (1957) first described categorical perception of speech sounds, noting that listeners' perception conforms to relatively sharp identification boundaries between categories of stop consonants and that whereas between-category discrimination of these sounds is nearly perfect, within-category discrimination is little better than chance. Similar patterns have been observed in the perception of colors (Davidoff, Davies, & Roberson, 1999), facial expressions (Etcoff & Magee, 1992), and familiar faces (Beale & Keil, 1995), as well as the representation of objects belonging to artificial categories that are learned over the course of an experiment (Goldstone, 1994; Goldstone, Lippa, & Shiffrin, 2001). All of these categorical effects are characterized by better discrimination of between-category contrasts than within-category contrasts, although the magnitude of the effect varies between domains.

In this article, we develop a computational model of the influence of categories on perception through a detailed investigation of one such phenomenon, the *perceptual magnet effect* (Kuhl, 1991), which has been described primarily in vowels. The perceptual magnet effect involves reduced discriminability of speech sounds near phonetic category prototypes. For several reasons, speech sounds, particularly vowels, provide an excellent starting point for assessing a model of the influence of categories on perception. Vowels are naturally occurring, highly familiar stimuli that all listeners have categorized. As discussed later, a precise two-

Naomi H. Feldman and James L. Morgan, Department of Cognitive and Linguistic Sciences, Brown University; Thomas L. Griffiths, Department of Psychology, University of California, Berkeley.

Bayesian model of speech perception

speaker

listener

noise

T

S

P(SIT)

P(TIS)

goal of perception

noise model produces actual stimulus

intended utterance from speaker

$$P(T|S) = \frac{P(S|T)P(T)}{P(S)}$$

- Speaker produces a speech sound T.
- Noise perturbs T into percept S (internal and external noise possible).
- The listener calculates the posterior P(T|S) with goal of reconstructing the original sound T.

Bayesian model of speech perception

The speaker makes a sound production T.
Noise in the air perturbs T into S.

Prior on utterance (Gaussian)

$$P(T) = N(\mu_c, \sigma_c^2)\,.$$

Likelihood (Gaussian)

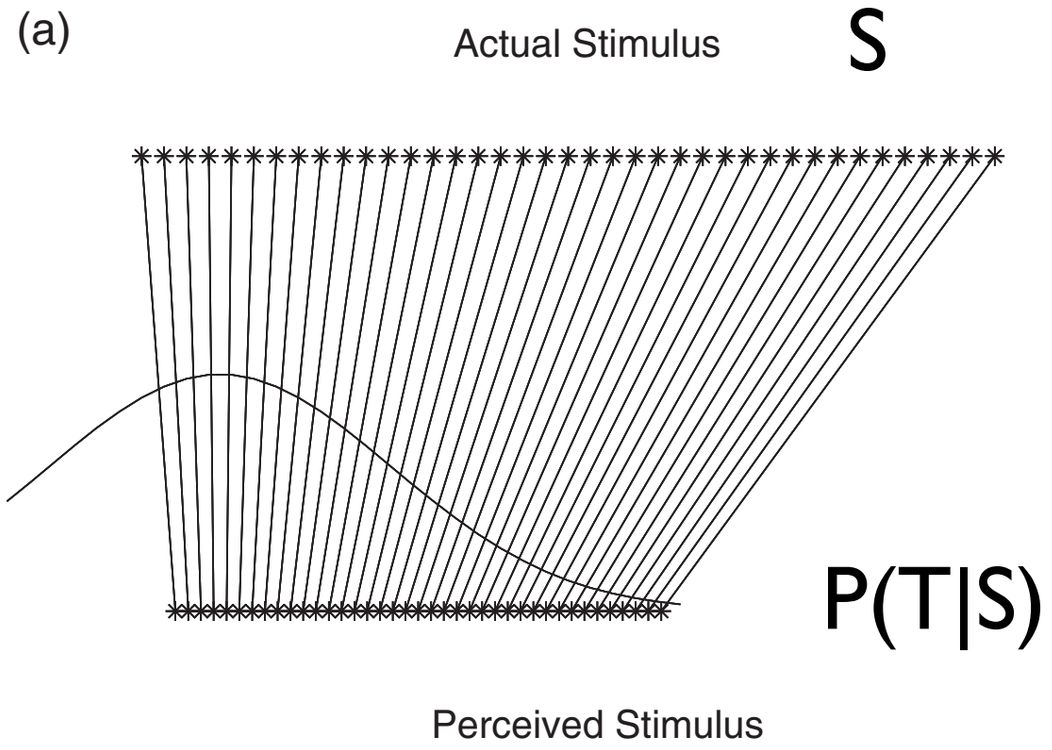
$$P(S \mid T) = N(T, \sigma_S^2)$$

Posterior

$$P(T \mid S) = \frac{P(S \mid T)P(T)}{P(S)}$$

$$= N\left(\frac{\sigma_c^2 S + \sigma_S^2 \mu_c}{\sigma_c^2 + \sigma_S^2}, \frac{\sigma_c^2 \sigma_S^2}{\sigma_c^2 + \sigma_S^2}\right)$$

If the stimulus is noisy, pull your perception towards the category you think it comes from.



Posterior is Gaussian, where the mean is a **weighted average** between the **actual stimulus *S* and the prior mean μ_c.**

- If the perceptual noise is high (high σ_S), rely more on the prior category mean
- If the category is highly variable (high σ_c), rely more on the actual stimulus *S*

Key technical concept: Conjugate priors

When prior and posterior are in the same family, then we have a **conjugate prior** for the likelihood function.

prior	likelihood	posterior	common use case
Normal	Normal (unknown mean, known variance)	Normal	estimating mean of a continuous sample
Beta	Binomial	Beta	estimating fairness of a coin based on counts
Dirichelt	Multinomial	Dirichelt	estimating weights on k-sided dice based on counts
...			

This makes it **very easy to compute posterior distributions**, as it can be done in closed form with standard formulas.

https://en.wikipedia.org/wiki/Conjugate_prior

Bayesian model of speech perception: Multiple categories

The speaker makes a sound production T.
Noise in the air perturbs T into S.

Step 1) Bayesian classification of the speech sound in category *c*

$$p(c|S) = \frac{p(S|c)p(c)}{\sum_c p(S|c)p(c)}$$

$$p(S|c) = \int p(S|T)p(T|c)\ dT$$

(this term is another Gaussian distribution)

Step 2) Compute reconstruction of *T* as weighted mixture of posteriors

$$p(T|S) = \Sigma_c p(T|S,c)p(c|S)$$

(mixture of Gaussians)

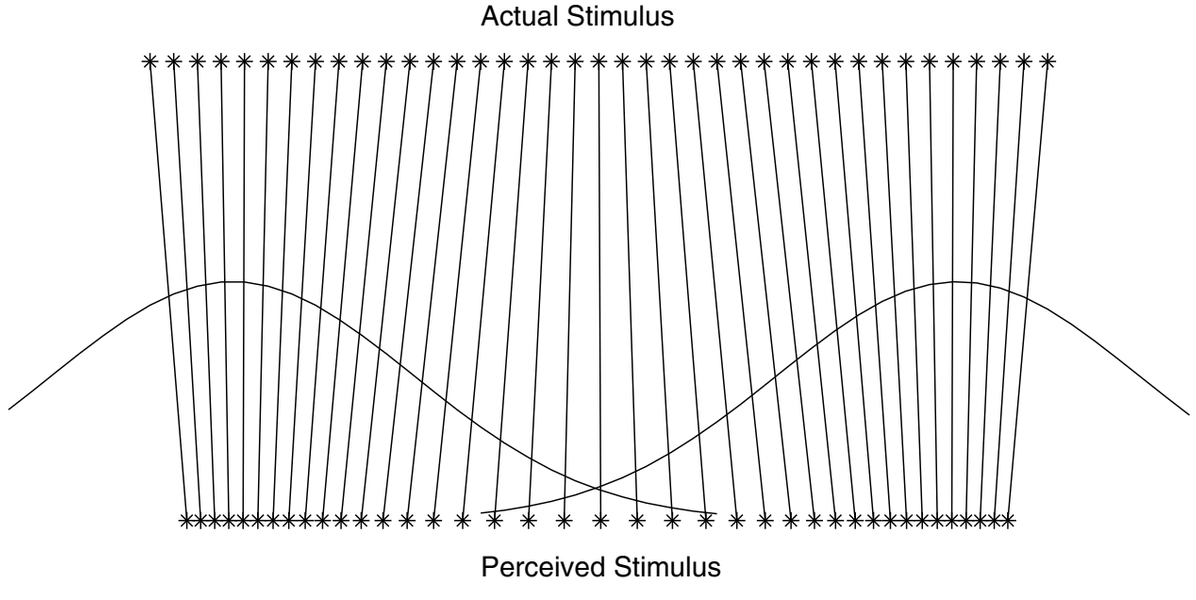
$$P(T|S,c) = N(\frac{\sigma_c^2 S + \sigma_S^2 \mu_c}{\sigma_c^2 + \sigma_S^2}, \frac{\sigma_c^2 \sigma_S^2}{\sigma_c^2 + \sigma_S^2})$$

(term is posterior from previous slide with known category)

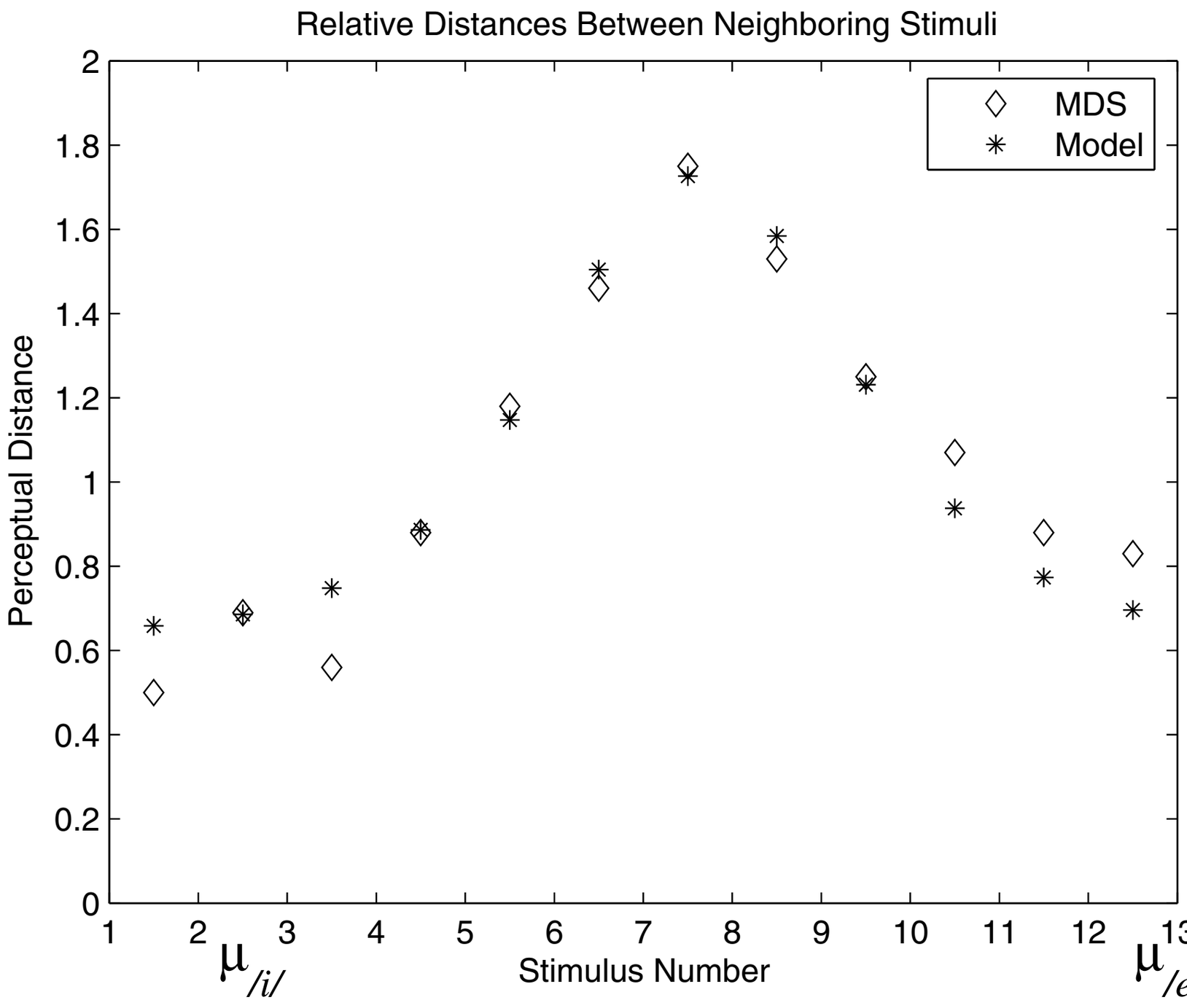
Posterior mean (expected value $E[x] = \sum_x xp(x)$)

$$E[T|S] = \frac{\sigma_c^2}{\sigma_c^2 + \sigma_S^2} S + \frac{\sigma_S^2}{\sigma_c^2 + \sigma_S^2} \Sigma_c p(c|S)\mu_c$$

- If the perceptual noise is high (high σ_S), rely more on the category means
- If the category is highly variable (high σ_c), rely more on the actual stimulus *S*



Comparing the Bayesian model to perceptual data



participant data
Bayesian model

Computing perceptual distance

- Participant discrimination judgments converted to perceptual distance using multi-dimensional scaling (MDS)
- Bayesian model predictions computed as $E[T|S]$

Conclusions from the Bayesian models of the perceptual magnet effect

• Categories influence perception in a range of domains: speech, color, faces, etc...

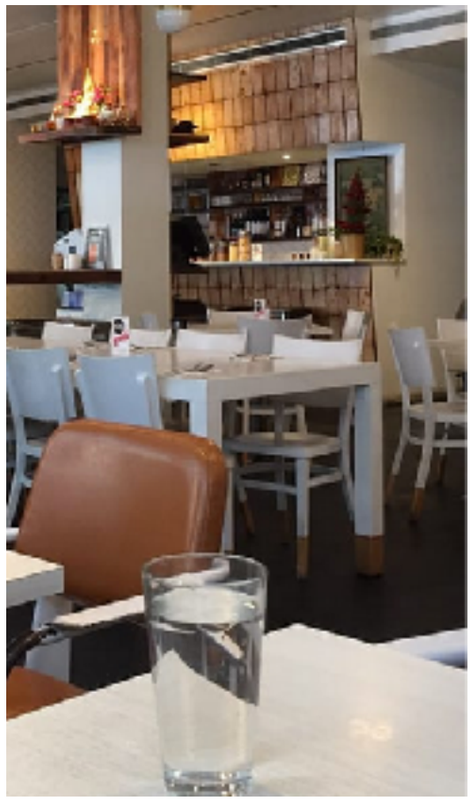
Although it's clear that categories influence perception, it's not clear WHY they should

- There are many other models of categorical perception and perceptual magnet effect, but they don't really answer the "why" question.

The Bayesian model suggests why perception should have this characteristic: It's a rational adaption for perceiving/reconstructing stimuli under noise.

Implications for understanding behavioral data: User ratings

Actual object



T

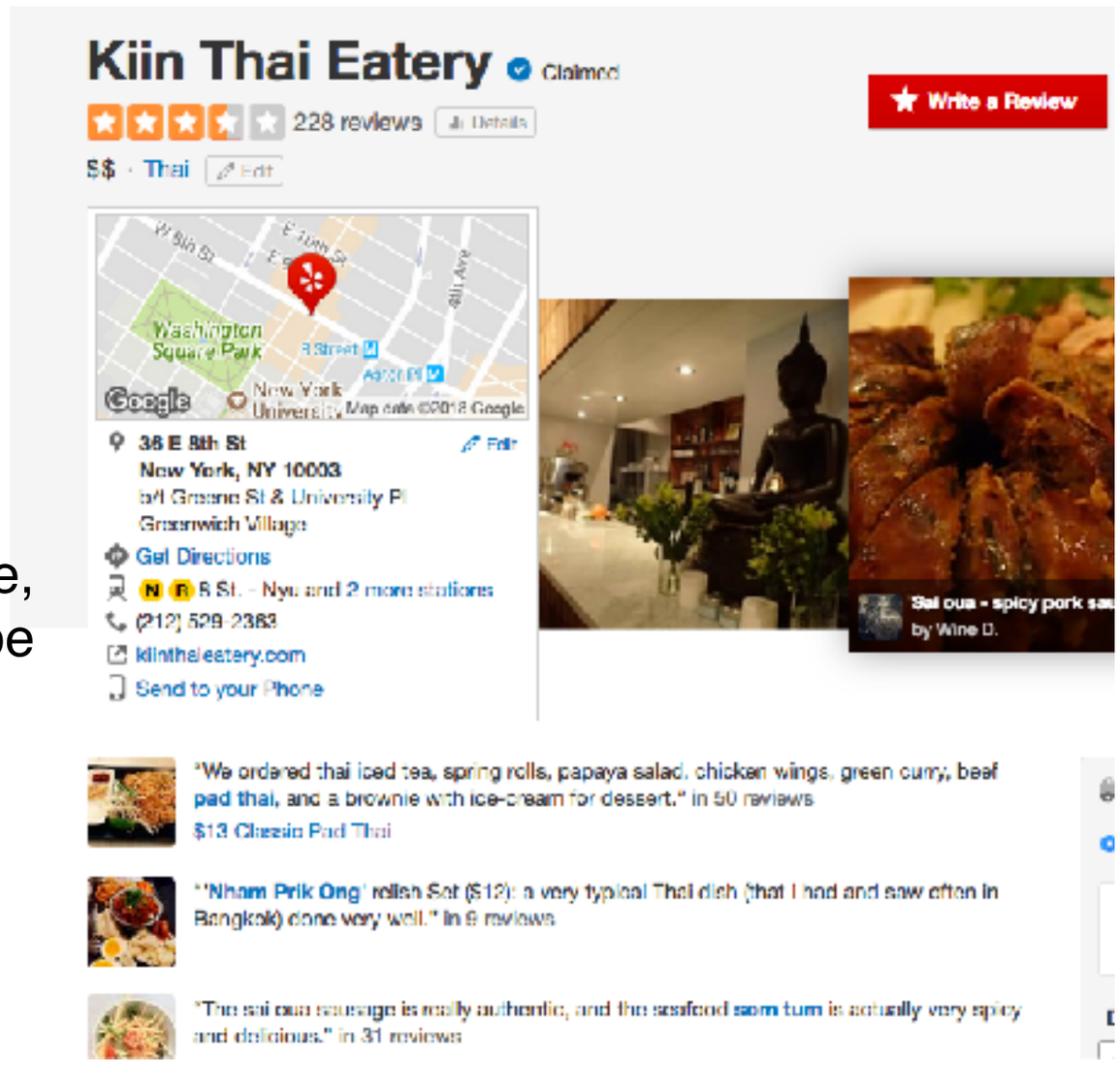
noise
(in perception,
memory, etc.)

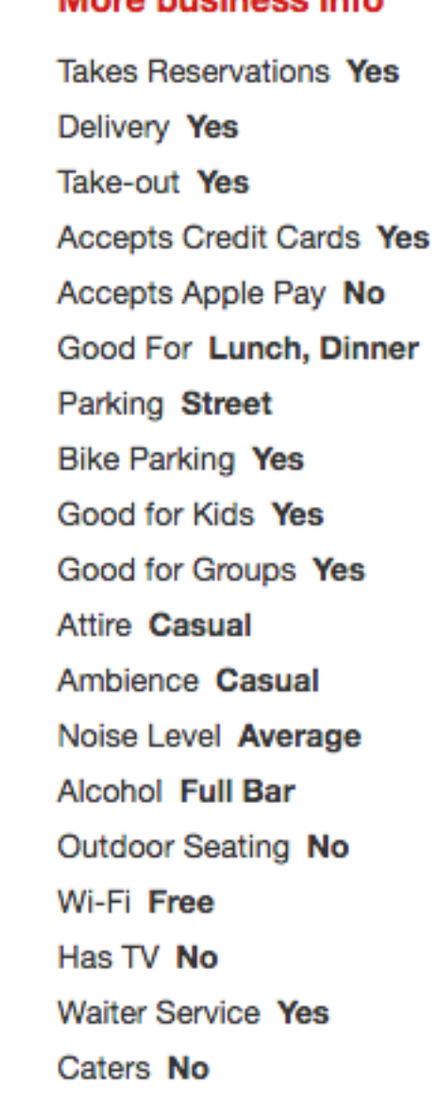
S

P(TIS)

Perceived experience may
warped by the category c

Price, quality, and many other attribute ratings (attire, good for groups, etc.) will be warped by category knowledge (Thai restaurant), especially if ratings are entered with a delay.





Predicting the future with Bayesian inference

PSYCHOLOGICAL SCIENCE

Research Article

Optimal Predictions in Everyday Cognition

Thomas L. Griffiths[1] and Joshua B. Tenenbaum[2]

[1]Department of Cognitive and Linguistic Sciences, Brown University, and [2]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology

ABSTRACT—*Human perception and memory are often explained as optimal statistical inferences that are informed by accurate prior probabilities. In contrast, cognitive judgments are usually viewed as following error-prone heuristics that are insensitive to priors. We examined the optimality of human cognition in a more realistic context than typical laboratory studies, asking people to make predictions about the duration or extent of everyday phenomena such as human life spans and the box-office take of movies. Our results suggest that everyday cognitive judgments follow the same optimal statistical principles as perception and memory, and reveal a close correspondence between people's implicit probabilistic models and the statistics of the world.*

If you were assessing the prospects of a 60-year-old man, how

Perry, Super, & Gallogly, 2001; Huber, Shiffrin, Lyle, & Ruys, 2001; Knill & Richards, 1996; Körding & Wolpert, 2004; Shiffrin & Steyvers, 1997; Simoncelli & Olshausen, 2001; Weiss, Simoncelli, & Adelson, 2002). In contrast—perhaps as a result of the great attention garnered by the work of Kahneman, Tversky, and their colleagues (e.g., Kahneman, Slovic, & Tversky, 1982; Tversky & Kahneman, 1974)—cognitive judgments under uncertainty are often characterized as the result of error-prone heuristics that are insensitive to prior probabilities. This view of cognition, based on laboratory studies, appears starkly at odds with the near-optimality of other human capacities, and with people's ability to make smart predictions from sparse data in the real world.

To evaluate how cognitive judgments compare with optimal statistical inferences in real-world settings, we asked people to predict the duration or extent of everyday phenomena such as human life spans and the gross of movies. We varied the phe-

Let’s make some predictions

You stopped by a friend’s apartment, and she has been watching a movie for 15 minutes. What would you predict for the length of the movie in total?

You stopped by a friend’s apartment, and she has been watching a movie for 75 minutes. What would you predict for the length of the movie in total?

Let’s make some predictions

You stopped by a friend’s apartment, and she has been watching a movie for 15 minutes. What would you predict for the length of the movie in total?

You stopped by a friend’s apartment, and she has been watching a movie for 75 minutes. What would you predict for the length of the movie in total?

A movie has grossed 15 million dollars at the box office, but you don’t know how long it’s been running. How much will it gross in total?

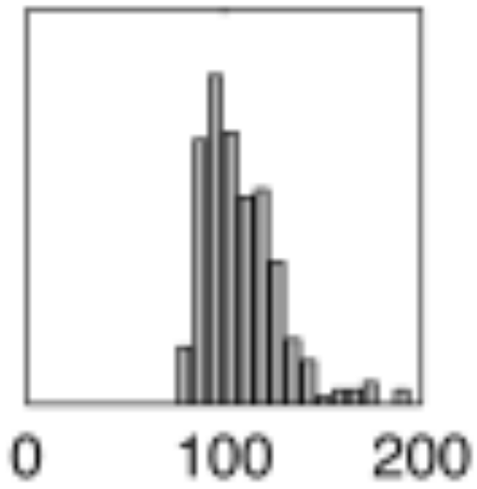
A movie has grossed 75 million dollars at the box office, but you don’t know how long it’s been running. How much will it gross in total?

Let's make some predictions

You stopped by a friend's apartment, and she has been watching a movie for 15 minutes. What would you predict for the length of the movie in total?

You stopped by a friend's apartment, and she has been watching a movie for 75 minutes. What would you predict for the length of the movie in total?

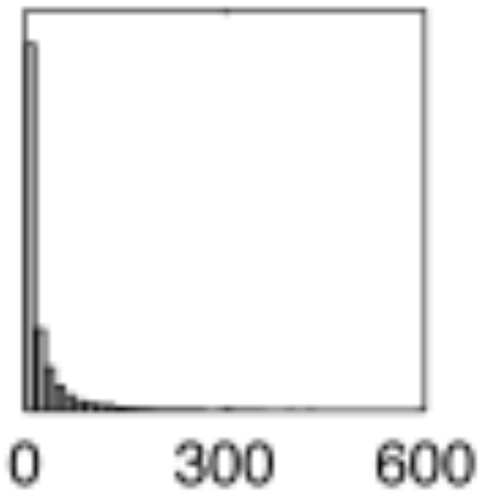
Movie runtimes



A movie has grossed 15 million dollars at the box office, but you don't know how long it's been running. How much will it gross in total?

A movie has grossed 75 million dollars at the box office, but you don't know how long it's been running. How much will it gross in total?

Movie grosses



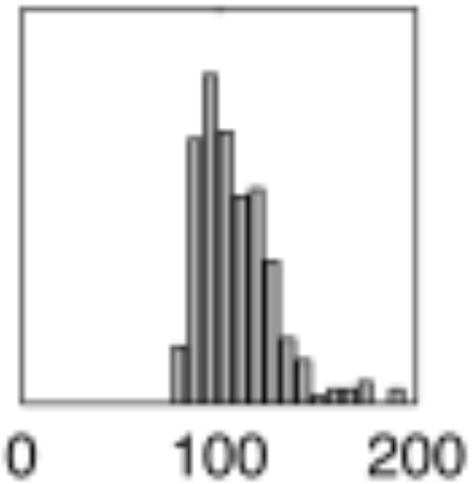
Simple Bayesian model of predicting the future

A movie has grossed 15 million dollars at the box office, but you don't know how long it's been running. How much will it gross total?

t_{total} : the total quantity you are estimating

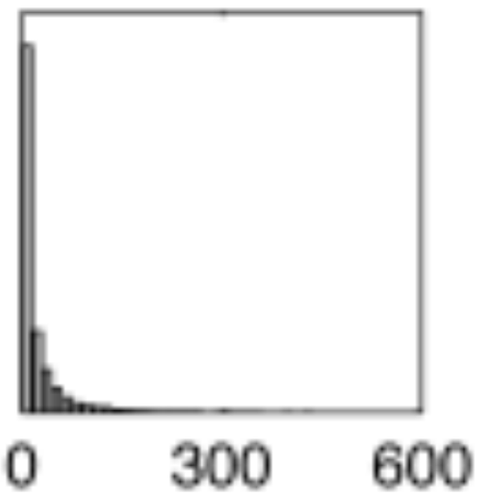
t : the current quantity you are given (current runtime of movie, current gross, etc.)

Movie runtimes



t_{total}

Movie grosses



t_{total}

Bayesian estimation problem

posterior ← $P(t_{total}|t)$ = likelihood ← $P(t|t_{total})$ prior ← $P(t_{total})$ / $P(t)$

$$P(t_{total}|t) = \frac{P(t|t_{total})P(t_{total})}{P(t)}$$

Likelihood

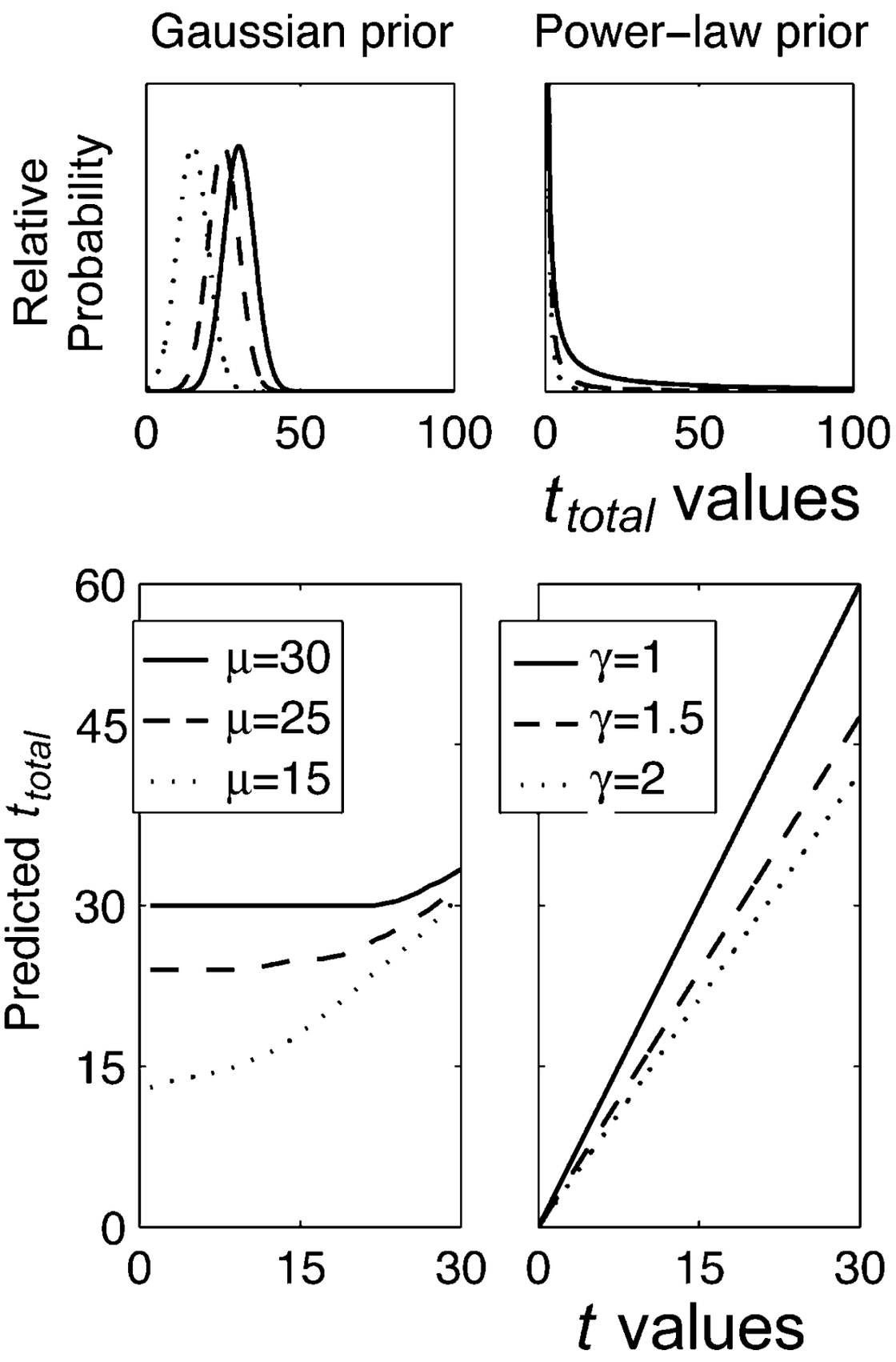
$$P(t|t_{total}) = 1/t_{total}$$

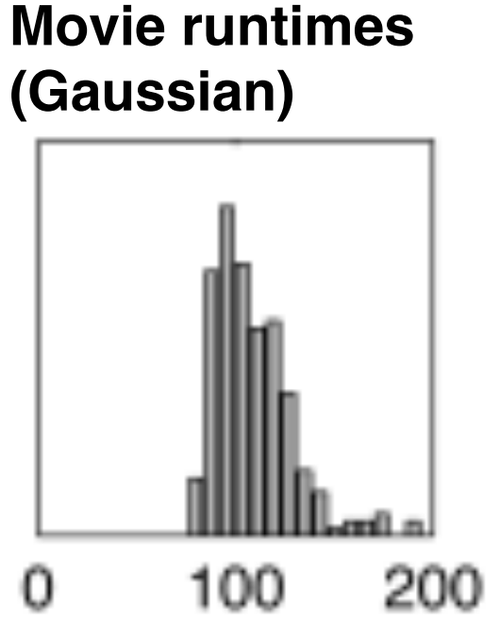
Assumption: you are equally likely to encounter a quantity at any point across its lifespan (movie / person / etc.)

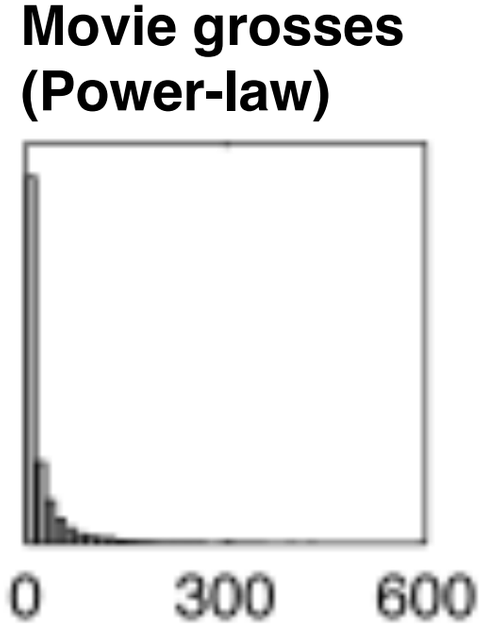
Prior

$P(t_{total})$ is estimated from real world statistics

Different priors have qualitatively different predictions







Gaussian prior

$$P(t_{total}) \propto \exp(-\frac{1}{2\sigma^2}(t_{total}-\mu)^2)$$

Power-law prior

$$P(t_{total}) \propto t_{total}^{-\gamma}$$

Posterior

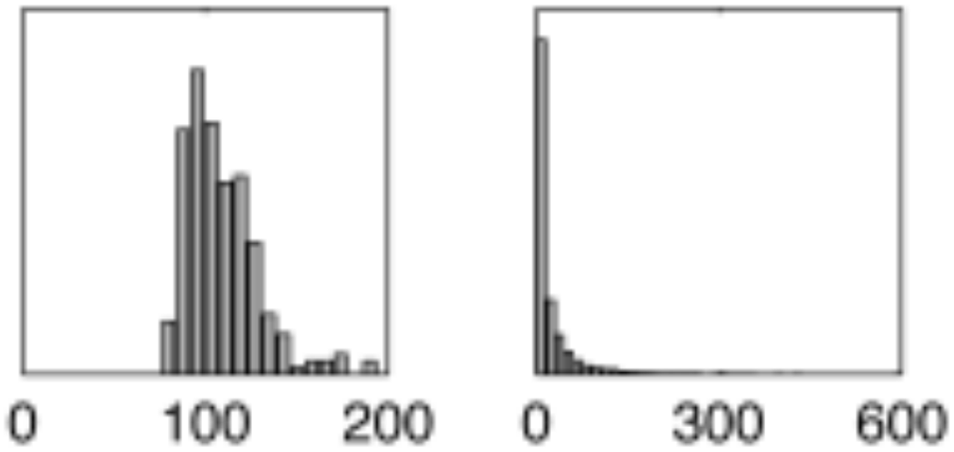
$$P(t_{total}|t) = \frac{P(t|t_{total})P(t_{total})}{P(t)}$$

Different priors have qualitatively different predictions: Comparison with human data

Movie runtimes (Gaussian)

Movie grosses (Power-law)

Prior



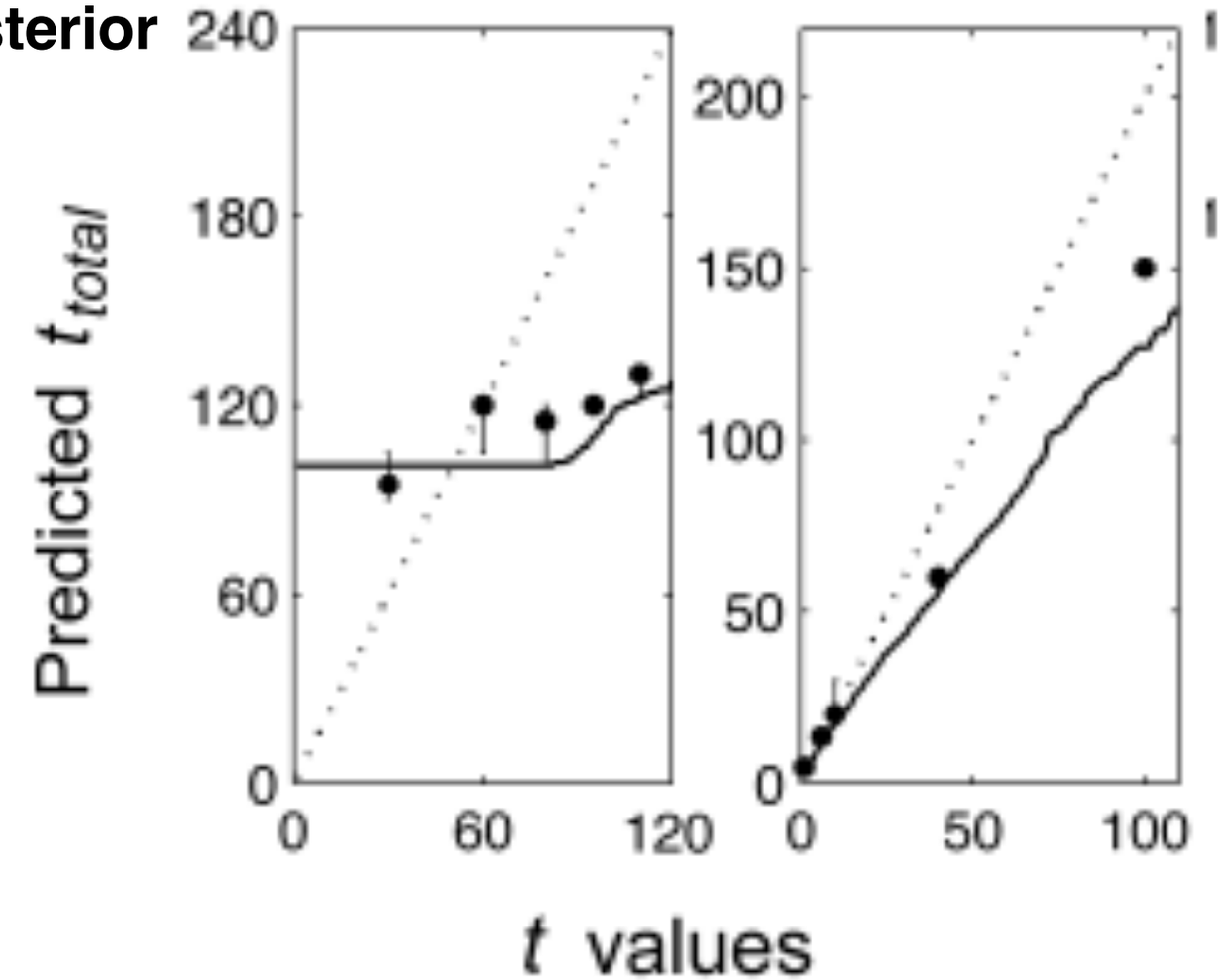
Posterior

Black dots are median prediction of human participants

Solid lines are optimal Bayesian posterior predictions

$$P(t_{total}|t) = \frac{P(t|t_{total})P(t_{total})}{P(t)}$$

Posterior



Optimal Bayesian predictions

For movie runtimes, predict the mean unless the runtime has already exceeded it.

For movie grosses, multiply the current gross by roughly 1.5

Patterns of prediction across a range of domains

Poem lengths: If your friend read you her favorite line of poetry, and told you it was line 5 of a poem, what would you predict for the total length of the poem?

Life spans: Insurance agencies employ actuaries to make predictions about people's life spans—the age at which they will die—based upon demographic information. If you were assessing an insurance case for an 18-year-old man, what would you predict for his life span?

Baking times for cakes: Imagine you are in somebody's kitchen and notice that a cake is in the oven. The timer shows that it has been baking for 35 minutes. What would you predict for the total amount of time the cake needs to bake?

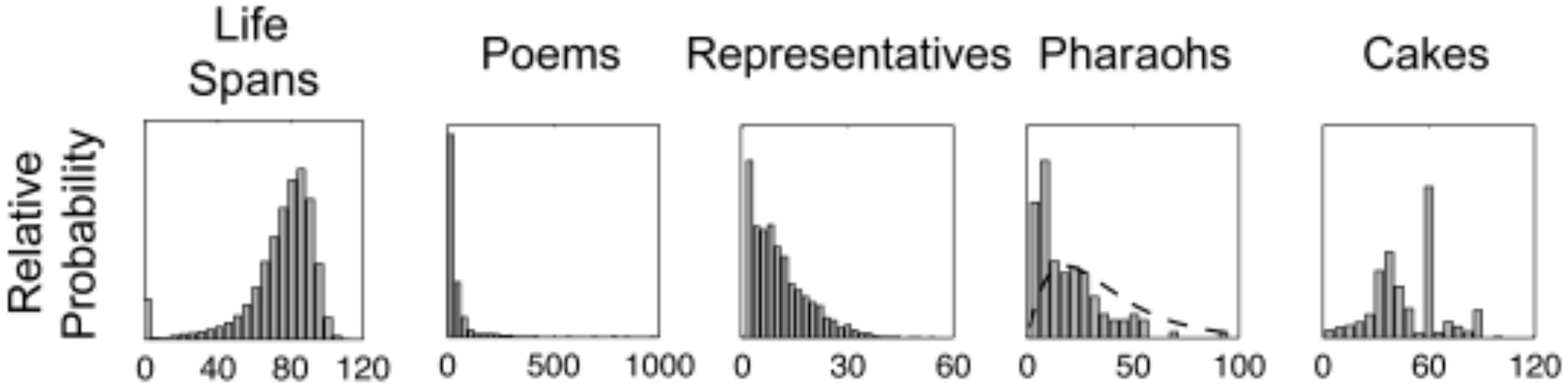
Waiting times: If you were calling a telephone box office to book tickets and had been on hold for 3 minutes, what would you predict for the total time you would be on hold?

Reigns of pharaohs: If you opened a book about the history of ancient Egypt to a page listing the reigns of the pharaohs, and noticed that at 4000 BC a particular pharaoh had been ruling for 11 years, what would you predict for the total duration of his reign?

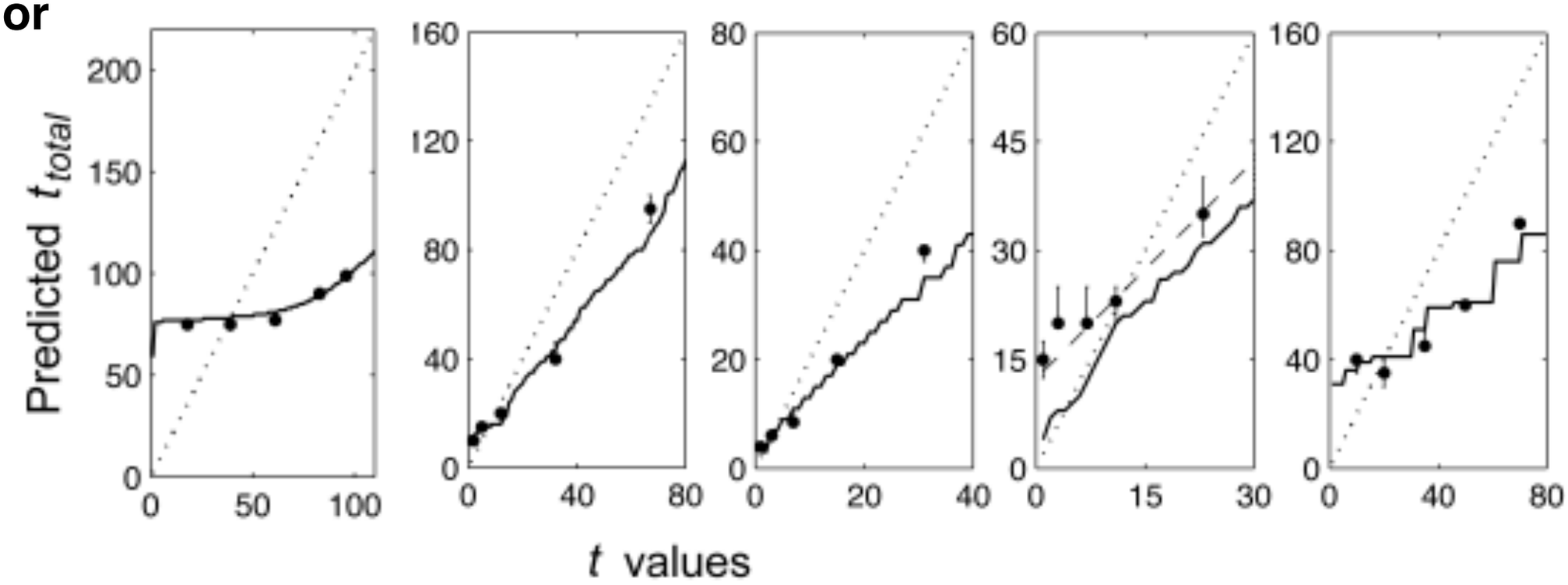
Terms of U.S. representatives: If you heard a member of the House of Representatives had served for 15 years, what would you predict his total term in the House would be?

Patterns of prediction across a range of domains

Prior



Posterior



Black dots are median prediction of participants

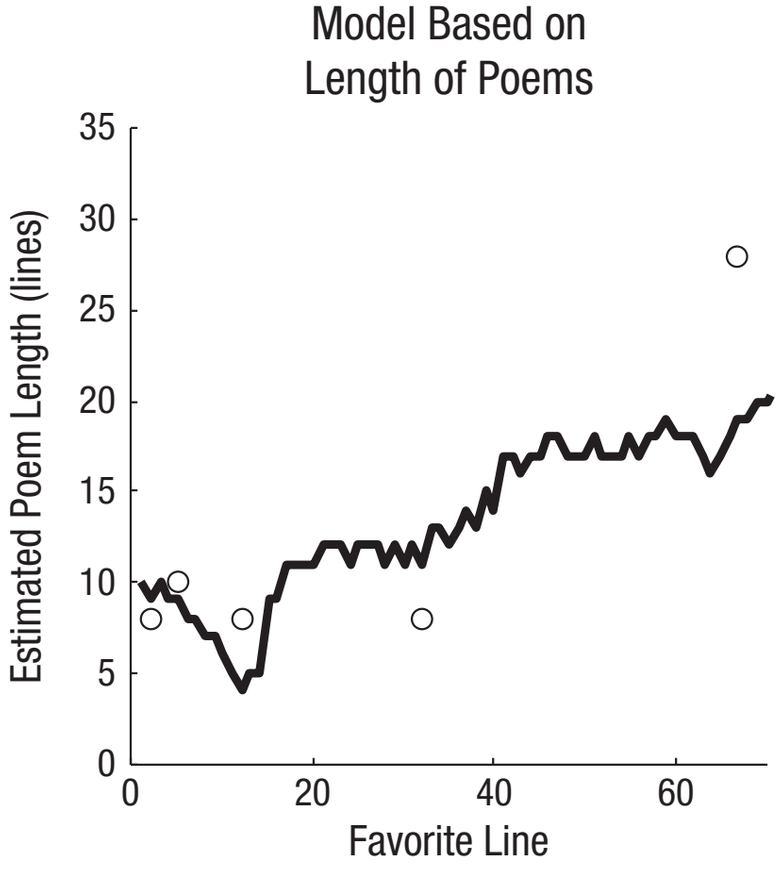
Solid lines are optimal Bayesian predictions

Critique of “Optimal predictions in everyday cognition”

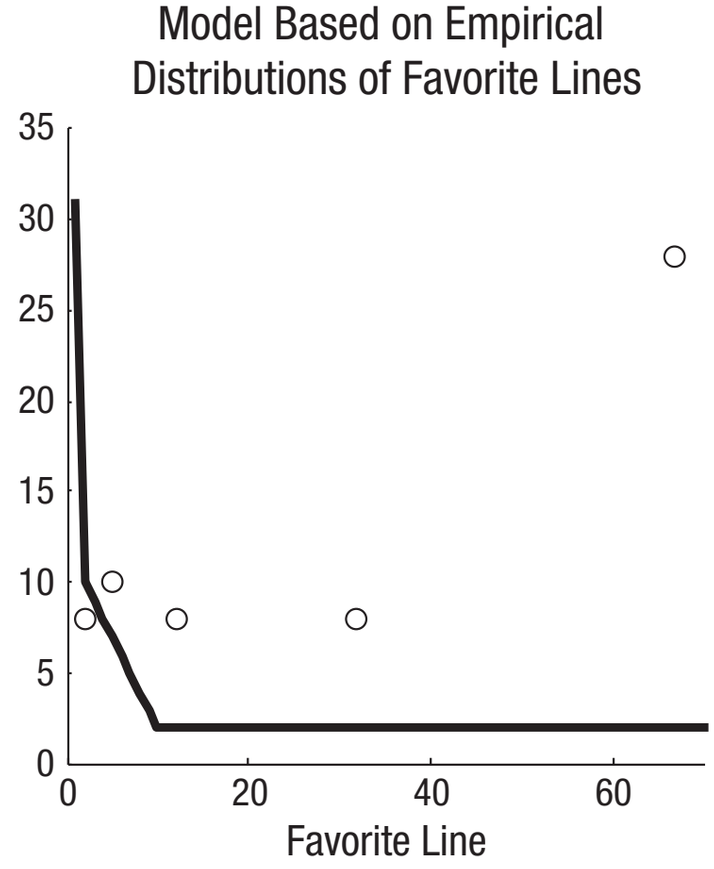
Marcus, G. and Davis, E. (2013) “How robust are probabilistic models of higher-level cognition?”

White dots are mean prediction of participants

Solid lines are Bayesian predictions

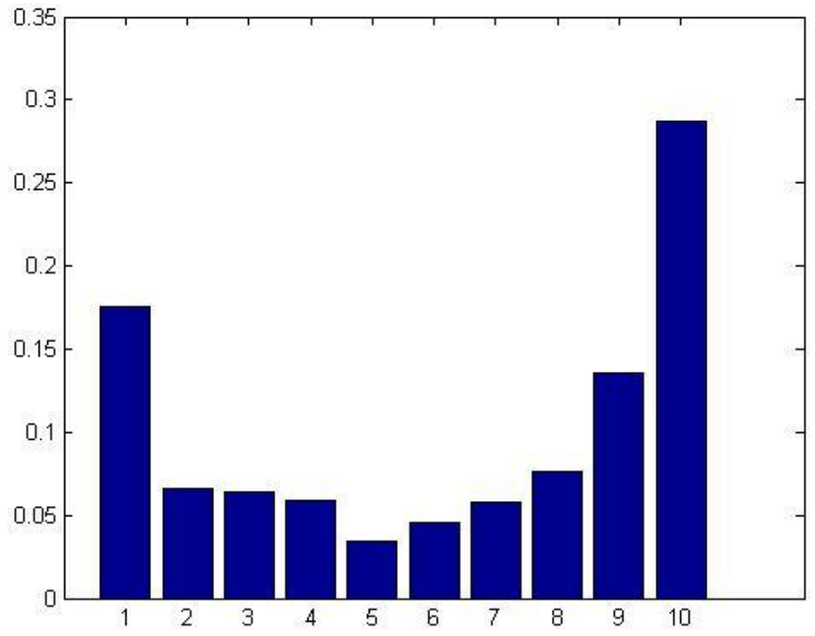


If Griffiths and Tenenbaum data is replotted to only show “additional length” on y-axis, the predictions can be impressive.



If distribution of favorite lines is taken into account, the predictions are way off.

Favorite lines are not uniformly distributed across poem.



Distribution of favorite lines per decile of poem.

Conclusions from optimal predictions in everyday cognition

- Critique for Marcus and Davis notwithstanding, there are surprisingly close fit between people's predictions and optimal Bayesian predictions.

- Implications

1. In many cases, people seem to accurately absorb the statistics of their environment for everyday quantities.

2. In addition, people use these learned statics in accordance with Bayesian inference.

3. The simplifying assumptions of "equal likelihood of encounter across timespan" could also be important, given Marcus and Davis critique.

Typical use cases of Bayesian inference

Here is our posterior distribution:

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

Usually, we want to compute the posterior expectation of some function $\phi(\cdot)$
(Bayesian hypothesis averaging)

$$E[\phi(h)|D] = \sum_h \phi(h)P(h|D)$$

(for discrete hypotheses)

$$E[\phi(h)|D] = \int \phi(h)P(h|D)dh$$

(for continuous hypotheses)

Examples of $\phi(\cdot)$ we have seen so far

$\phi(h) = 1\{y \in C\}$ in number game (is new number y in the hypothesis?)

$\phi(h) = h$ for perceptual magnet model, we want the posterior mean

$\phi(h) = h$ for optimal predictions, we want the posterior mean (or posterior median, which is what is used in paper)

The computational challenges of Bayesian inference

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

$$E[\phi(h)|D] = \sum_h \phi(h)P(h|D) \qquad E[\phi(h)|D] = \int \phi(h)P(h|D)dh$$

The case of discrete hypotheses *h* (e.g., the number game)*:*

- In most cases, there are so many hypotheses *h* that it is intractable to enumerate them all

The case of continuous hypotheses *h* (e.g., perceptual magnet, optimal predictions in everyday cognition)*:*

- In some cases, we can use a conjugate prior or analytically compute the posterior
- Unfortunately, in most cases, the posterior does not have a simple form that we can work with.

In practice, we usually need to resort to approximate Bayesian inference. And we also want general purpose computational tools that don't require special-purpose derivations for each model.

Monte Carlo methods for approximate Bayesian inference

$$E[\phi(h) \mid D] = \sum_h \phi(h) P(h \mid D) \approx \frac{1}{M} \sum_m \phi(h^{(m)})$$

where samples $h^{(1)}, \ldots, h^{(M)}$ are generated from $P(h|D)$

As *M* approaches infinity, the sample mean converges to its expected value (law of large numbers)

[Note: there are other popular approaches for approximate Bayesian inference, but we will focus on Monte Carlo methods since they are the most general]

We're going to discuss three Monte Carlo algorithms for Bayesian inference:

- Rejection sampling (for discrete data *D* only)
- Importance sampling
- Metropolis-Hastings algorithm (example of Markov Chain Monte Carlo)

Rejection sampling

- Sample hypotheses $h^{(m)}$ from the prior $P(h)$ and data $D^{(m)}$ from the likelihood $P(D|h)$
- If your sample data $D^{(m)}$ exactly matches your target data D, store $h^{(m)}$ as an independent sample from posterior $P(h|D)$

Rejection sampling

(note, this is different than the “rejection sampler” covered in MacKay reading)

Goal of approximate inference:

$$E[\phi(h)|D] \approx \frac{1}{M} \sum_{m} \phi(h^{(m)})$$

where samples $h^{(1)}, \ldots, h^{(M)}$ are generated from $P(h|D)$

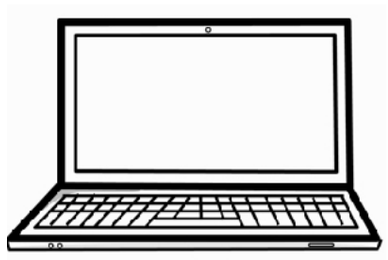
Simple algorithm for a rejection sampler:

$m \leftarrow 1$
while $m < M$ **do**
  sample $h^{(m)} \sim P(h)$
  sample $D^{(m)} \sim P(D|h^{(m)})$
  if $D^{(m)}$ and D match exactly **then** *(if sampled and real data match)*
    accept $h^{(m)}$ as a sample
    $m \leftarrow m + 1$
  end if
end while

Pros and cons:

pros: very simple to implement
cons: extremely inefficient; only works for discrete data D

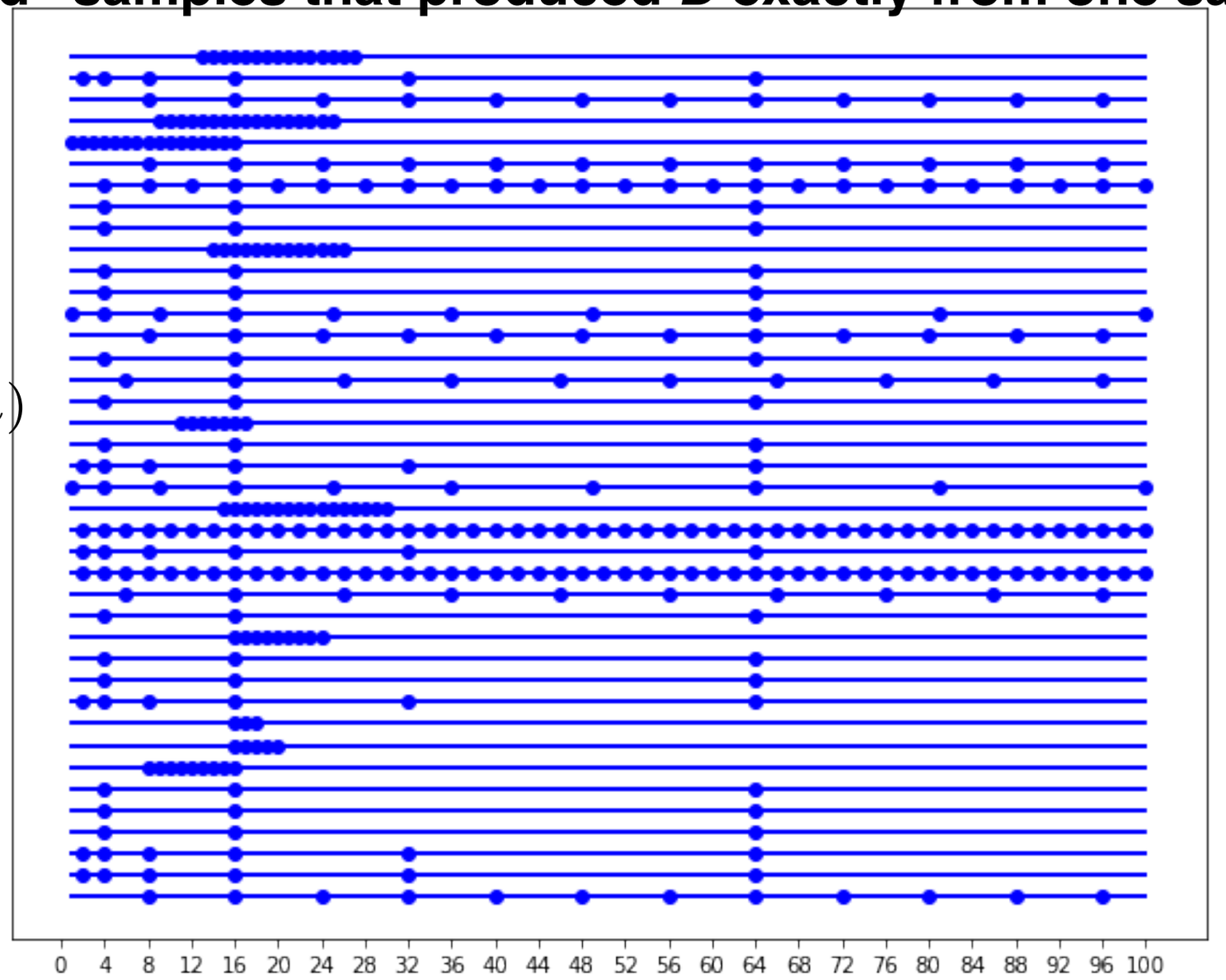
Example: rejection sampling for the number game



16

rows are $h^{(m)}$

"filtered" samples that produced *D* exactly from one sample

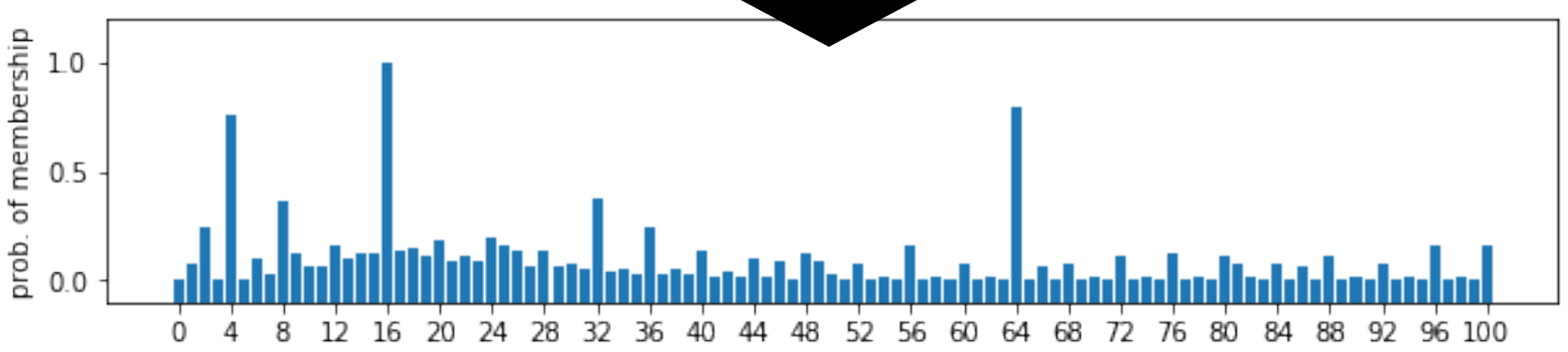


$$E[\phi(h)|D] \approx \frac{1}{M}\sum_m \phi(h^{(m)})$$

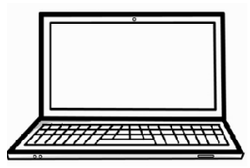
$$\phi(h) = 1\{y \in C\}$$

is new number y in the hypothesis?

compute prob. of membership (average)

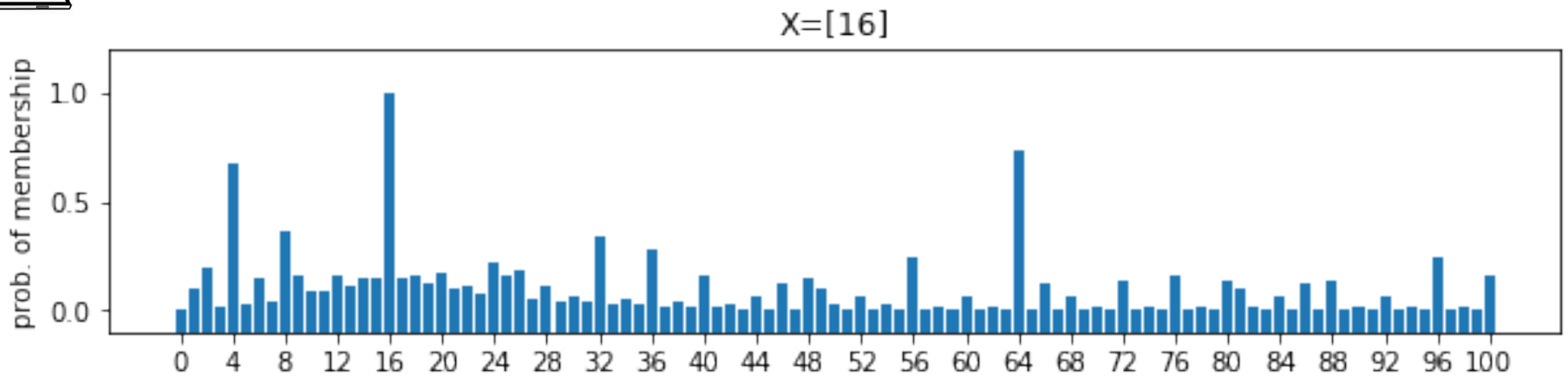


Example: rejection sampling for the number game

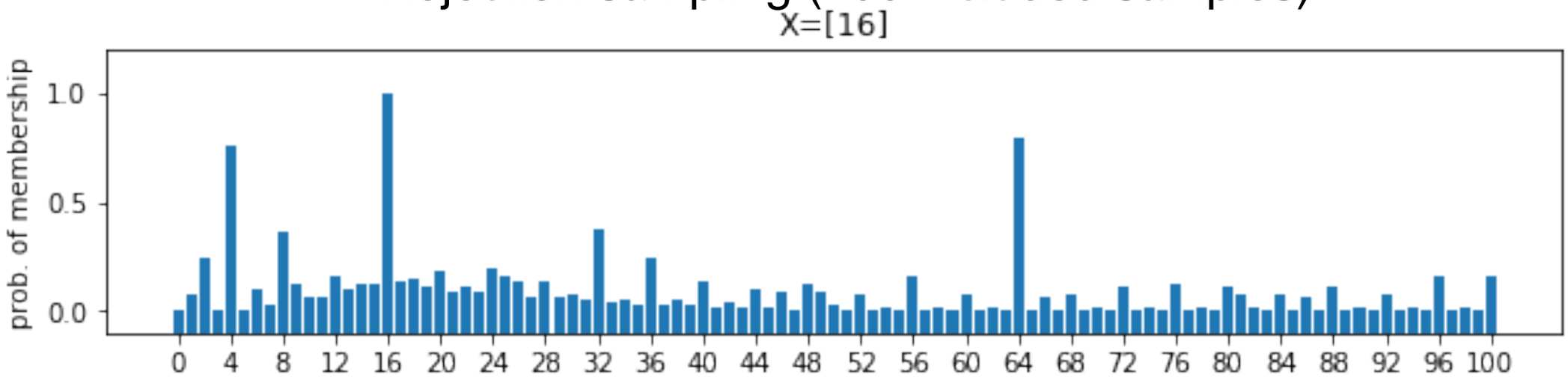


→ 16

Exact Bayesian inference



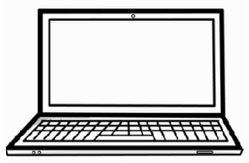
Rejection sampling (100 included samples)



Efficiency is only about 2% for the set [16]
(meaning we throw away 98% of samples, or we need about 4900 samples to get the desired 100)

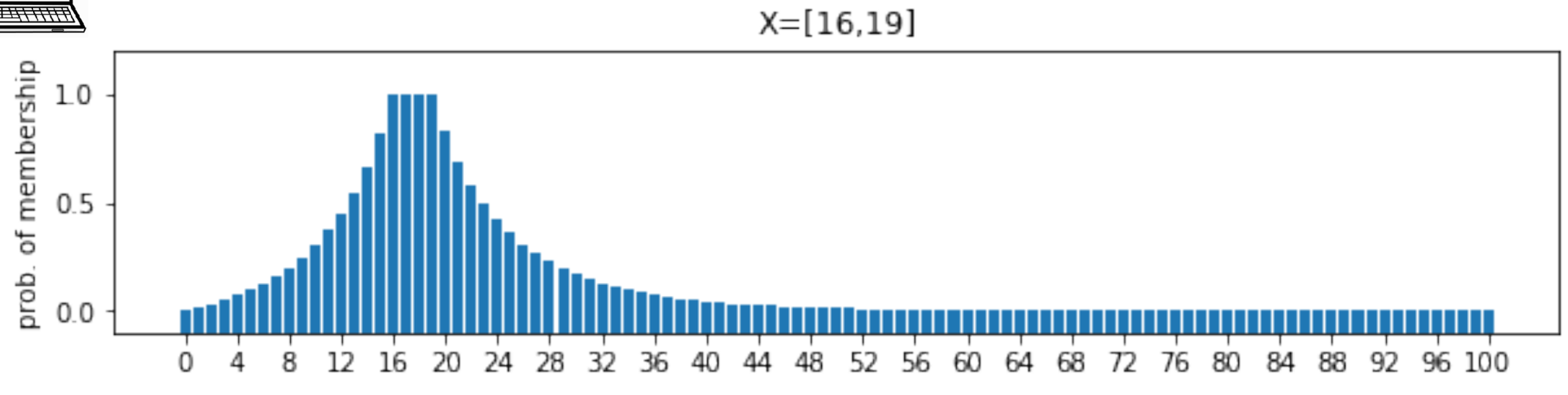
Example: rejection sampling for the number game

This algorithm scales very badly as we get more data.

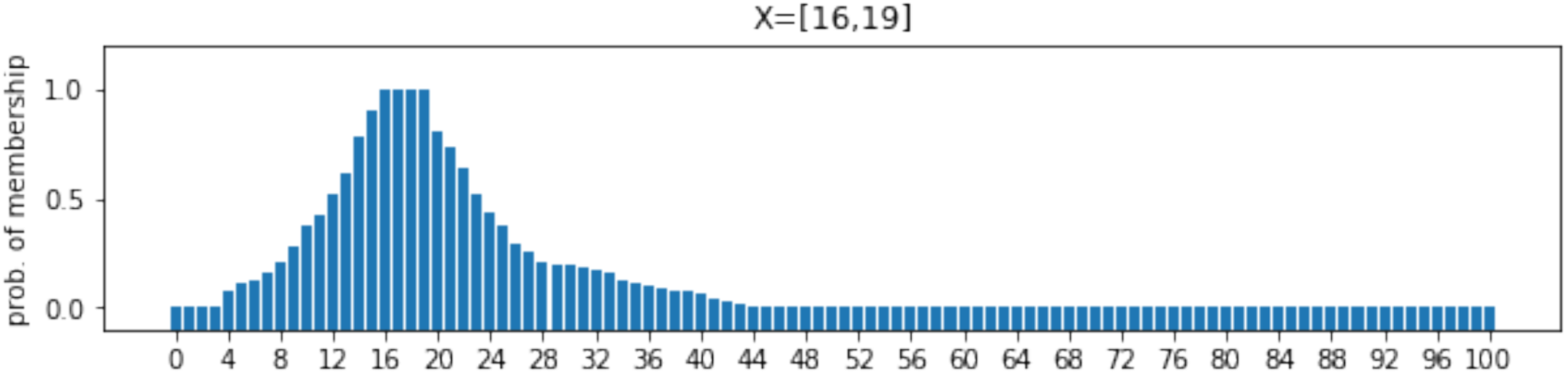


→ 16, 19

Exact Bayesian inference



Rejection sampling (100 included samples)

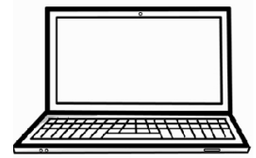


Efficiency is REALLY BAD, accepting only 0.04% for the set [16,19]
(we need about 265,000 samples to get 100 we can use)

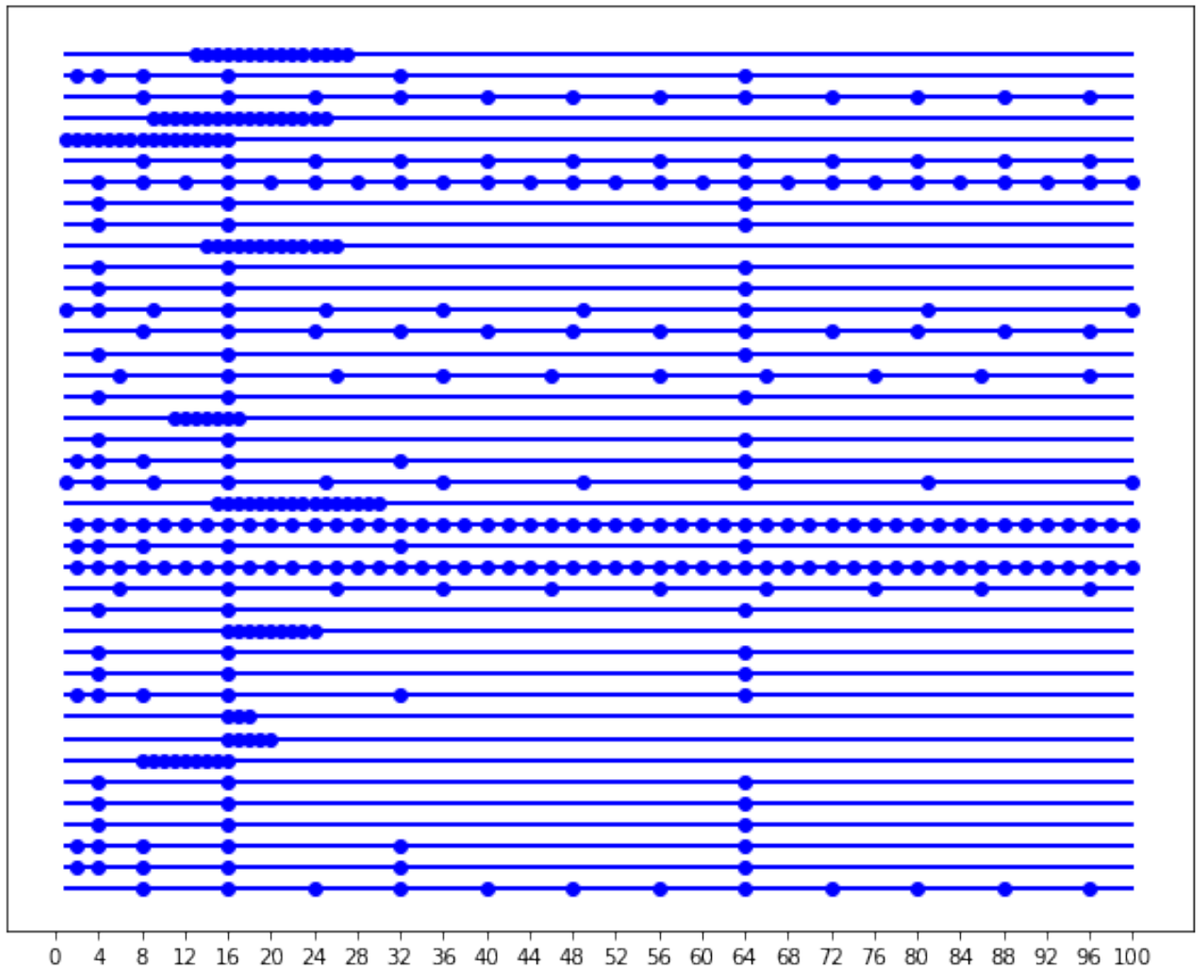
Important aside: Probabilistic inference is very flexible!

$$E[\phi(h)|D] \approx \frac{1}{M}\sum_{m}\phi(h^{(m)})$$

rows are samples $h^{(m)}$



➡ 16



If we can compute the posterior, or draw samples from the posterior, we can automatically reason about a huge range of questions $\phi(\cdot)$ given a single set of generated samples

Examples of reusing the sample for new queries

- Is 64 a member of the set? **(probability is 0.73)**
- Are both 36 and 64 members of the set? **(0.36)**
- Is there a member of the set greater than or equal to 80? **(0.27)**
- If we sample a new number from the hypothesis, what is the chance it will be 64? (**0.16)**
- If we sample a new number from the hypothesis, what is the chance it will be 80? **(0.02)**

Important aside: Probabilistic inference is very flexible!

Reusing samples is an example of the flexibility of probabilistic inference.

Flexible reasoning is natural in Bayesian models, but it is difficult to capture in neural networks trained with supervised learning, or model-free reinforcement learning.

Inference flexibility is not specific to rejection sampling, but to Bayesian models in general.

Importance sampling

- Sample hypotheses $h^{(m)}$ from a surrogate distribution $Q(h)$
- Re-weight the samples to approximate your target posterior $P(h|D)$.

Importance sampling

We want to approximate posterior expectation:

$$E[\phi(h)|D] = \sum_h \phi(h) P(h|D)$$

$$= \sum_h \phi(h) \frac{P(h|D)}{Q(h)} Q(h)$$

introduce a distribution Q we can easily sample from (and which is nonzero where the posterior is non-zero)

$$\approx \frac{1}{M} \sum_m \phi(h^{(m)}) \frac{P(h^{(m)}|D)}{Q(h^{(m)})}$$

draw samples from Q

importance sampling

$$= \frac{1}{M} \sum_m w^{(m)} \phi(h^{(m)}) \quad \text{for } w^{(m)} = \frac{P(h^{(m)}|D)}{Q(h^{(m)})}$$

where samples $h^{(1)}, \ldots, h^{(M)}$ are generated from $Q(h^{(m)})$

More commonly, we don't know normalizing constant for either P or Q, so we use:

$$E[\phi(h)|D] \approx \frac{1}{\sum_m w^{(m)}} \sum_m w^{(m)} \phi(h^{(m)})$$

Importance sampling

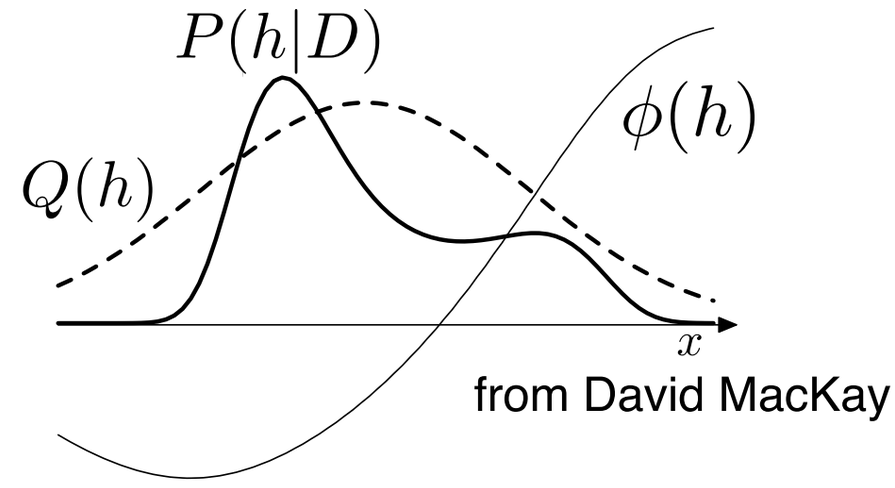
$$E[\phi(h)|D] \approx \frac{1}{\sum_m w^{(m)}} \sum_m w^{(m)} \phi(h^{(m)})$$

where samples $h^{(1)}, \ldots, h^{(M)}$ are generated from $Q(h^{(m)})$

Strategy:

We replace average over all hypotheses with a set of weighted samples, which correct for discrepancy between posterior and Q

$$w^{(m)} = \frac{P(h^{(m)}|D)}{Q(h^{(m)})}$$



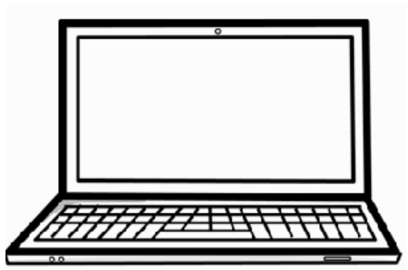
from David MacKay

We can set Q to be the prior, in order to get **"Likelihood weighted sampling"**

Pros and cons:

pros: far more efficient than rejection sampling, and works for continuous data
cons: its effectiveness strongly depends on how close Q is to the posterior

Example: likelihood weighted sampling



16

rows are samples from prior

$$Q(h) \leftarrow P(h)$$

$$E[\phi(h)|D] \approx \frac{1}{\sum_m w^{(m)}} \sum_m w^{(m)} \phi(h^{(m)})$$

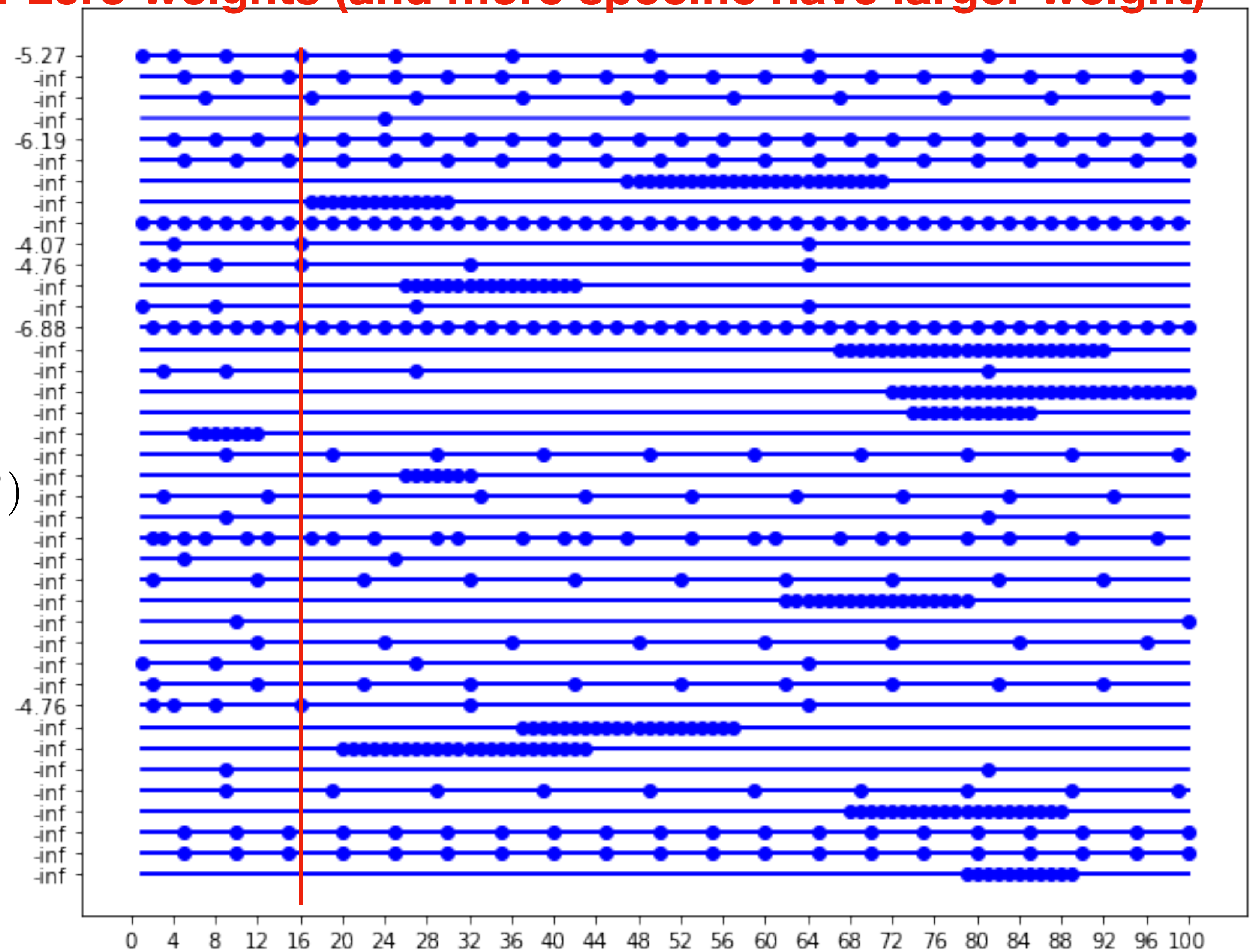
$$\phi(h) = 1\{y \in C\}$$

is new number y in the hypothesis?

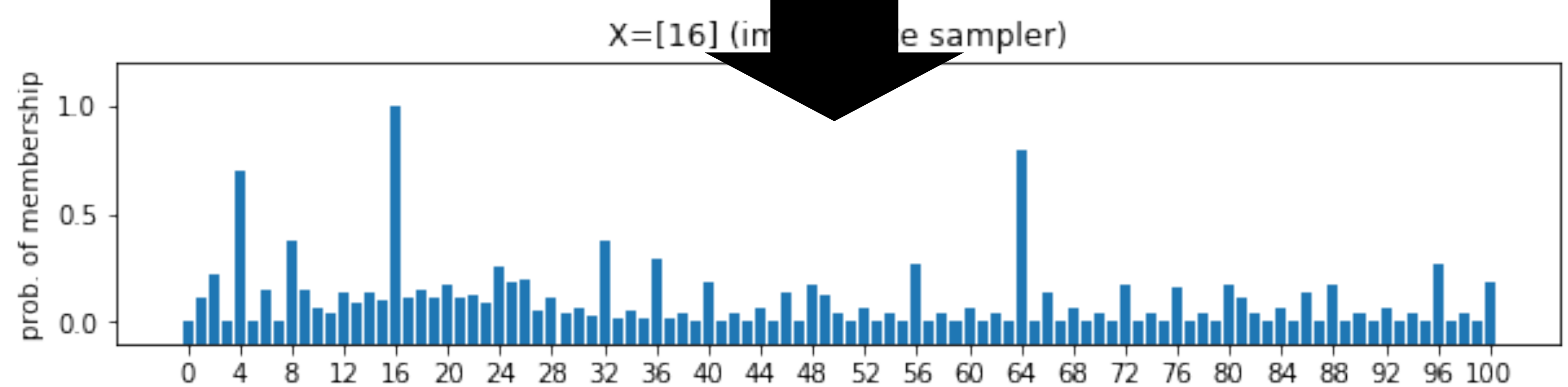
non-zero weights (and more specific have larger weight)

$h^{(m)}$

$\log(w^{(m)})$

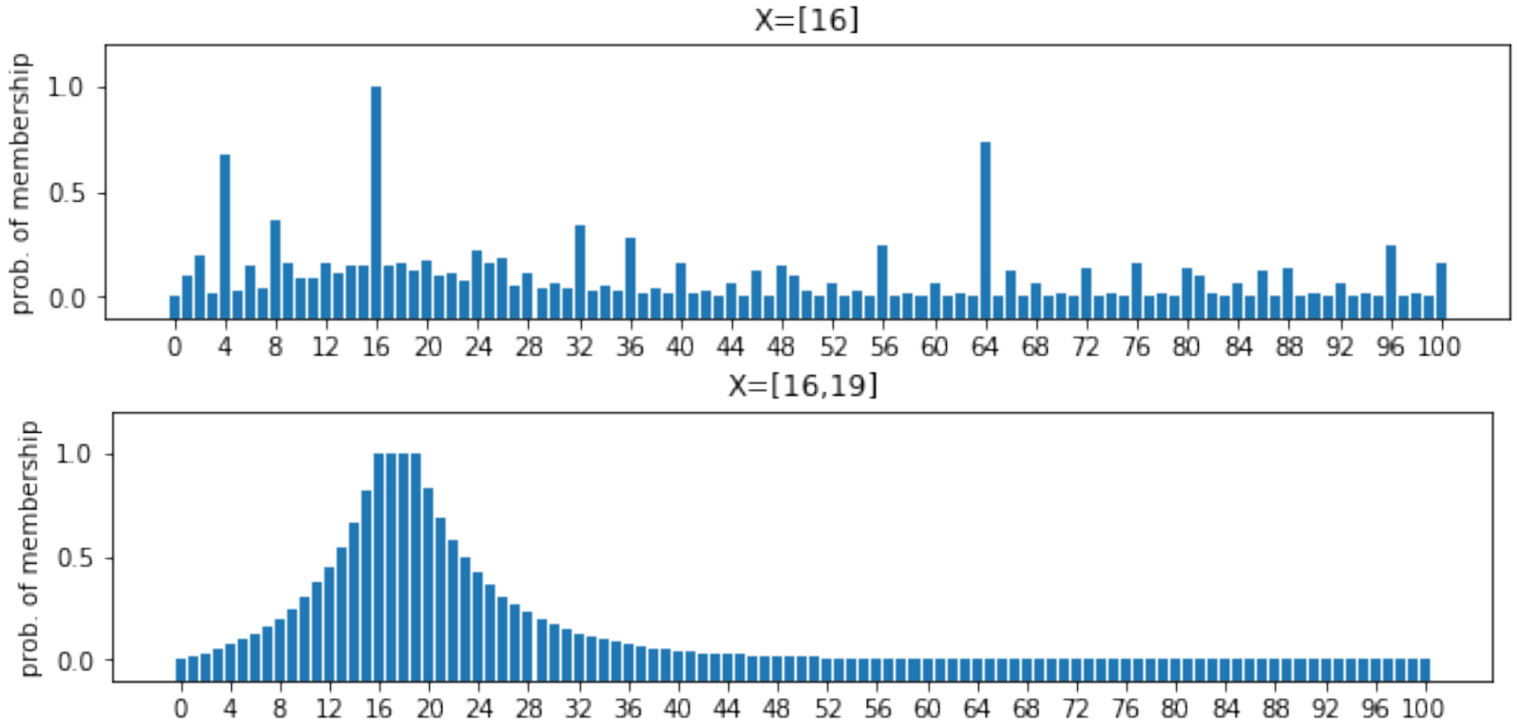


compute prob. of membership (weighted average)

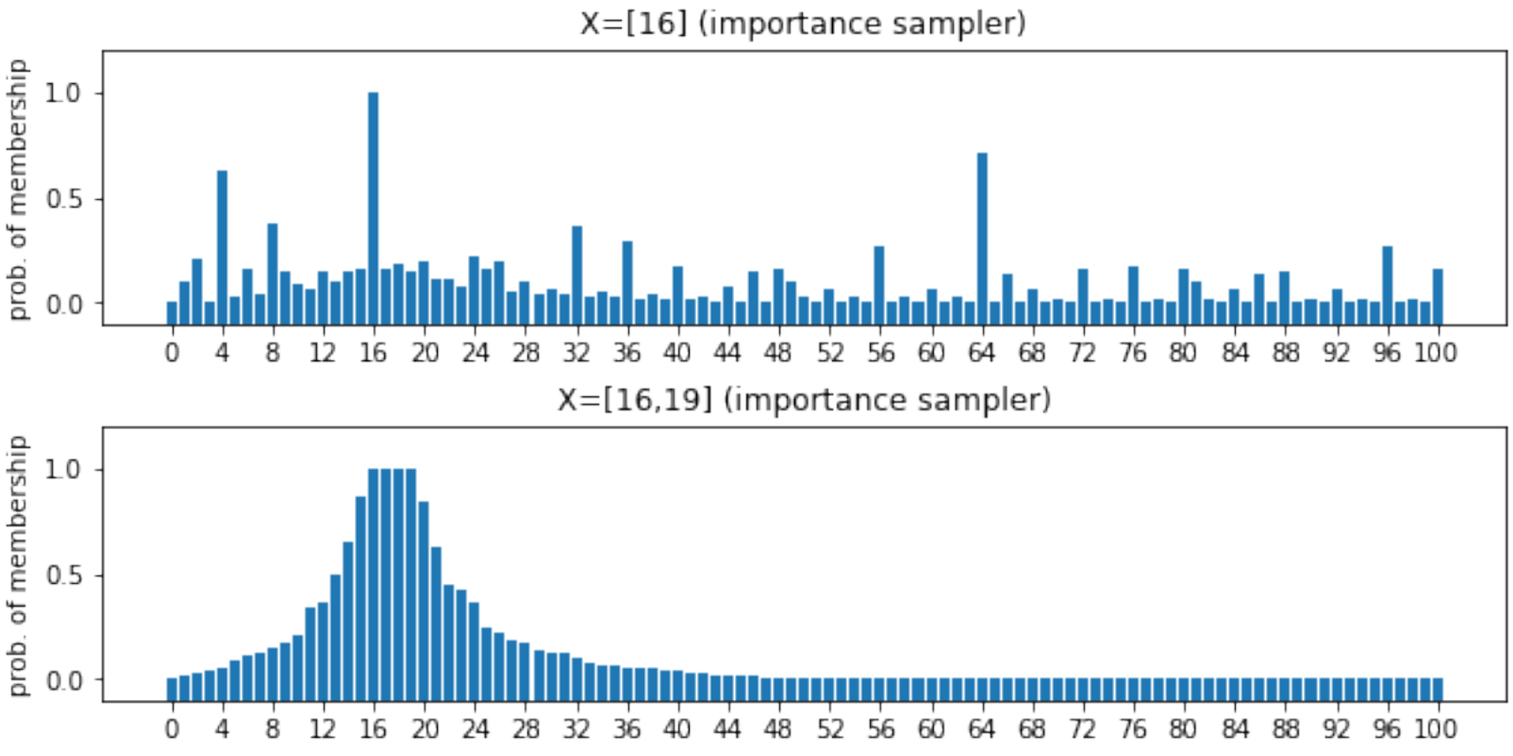


Example: likelihood weighted sampling

Exact Bayesian inference



Importance sampling (with only 2000 samples)

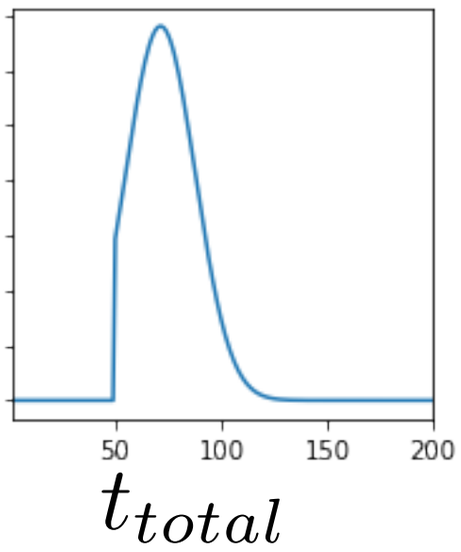


Example: Importance sampling for “Optimal Predictions"

Examples of non-standard posterior distributions

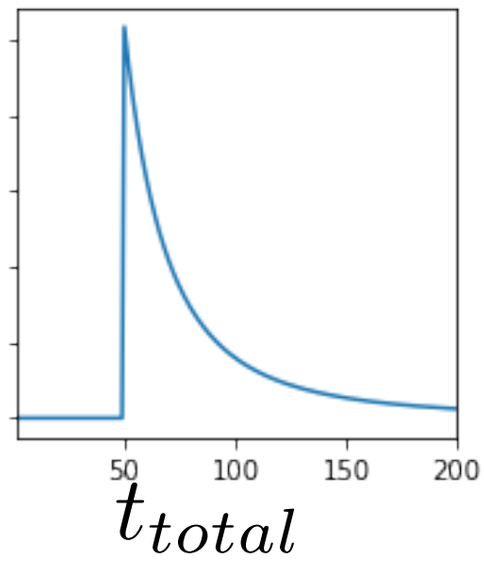
Lifespan posterior

$P(t_{total}|t = 50)$



Movie gross posterior

$P(t_{total}|t = 50)$



Posterior mean $E[t_{total}|t = 50]$

Exact inference:	74.3	82.6
Importance sampler (with 400 samples)	74.0	82.8

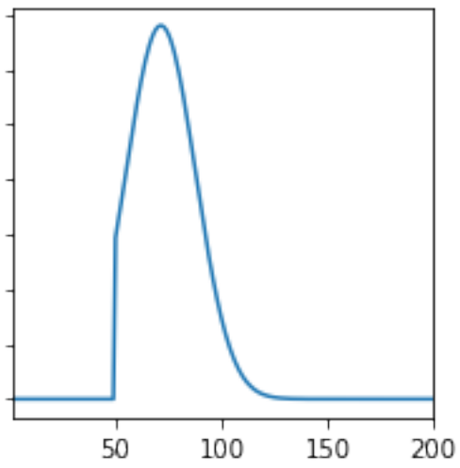
$$Q(t_{total}) = \text{Uniform}(1, 200)$$

Example: Importance sampling for “Optimal Predictions" (in more detail…)

Lifespan posterior

$P(t_{total}|t=50)$

relative probability



t_{total}

Posterior: $P(t_{total}|t) = \frac{P(t|t_{total})P(t_{total})}{P(t)}$

Likelihood: $P(t|t_{total}) = 1/t_{total}$ for $t < t_{total}$, 0 otherwise

Prior: $P(t_{total}) = N(t_{total}|\mu, \sigma_2^2)$

Importance sampling

$$E[\phi(h)|D] \approx \frac{1}{\sum_m w^{(m)}} \sum_m w^{(m)} \phi(h^{(m)})$$

Algorithm…

$t_{total}^{(1)}, \ldots, t_{total}^{(M)}$ sample from uniform Q

$$w^{(m)} = \frac{\frac{1}{t_{total}^{(m)}} N(t_{total}^{(m)}|\mu, \sigma_2^2)}{\frac{1}{200}}$$

compute weights for $t_{total} > 50$, 0 otherwise

$$E[t_{total}|t=50] \approx \frac{1}{\sum_m w^{(m)}} \sum_m w^{(m)} t_{total}^{(m)}$$

$$Q(t_{total}) = \textbf{Uniform}(1,200) = \frac{1}{200}$$

$$P(t_{total}|t) \propto \frac{1}{t_{total}} N(t_{total}|\mu, \sigma_2^2)$$

for $t < t_{total}$, 0 otherwise

Markov Chain Monte Carlo (MCMC)

- You have a single hypothesis $h^{(t)}$ in mind at any one time, and you make a small stochastic adjustment to $h^{(t)}$ to produce $h^{(t+1)}$
- The series of hypotheses $h^{(1)},\ldots,h^{(T)}$ (the Markov chain) converges in distribution to your target posterior $P(h|D)$

Metropolis-Hastings algorithm

(example of Markov Chain Monte Carlo (MCMC))

Goal of approximate inference:

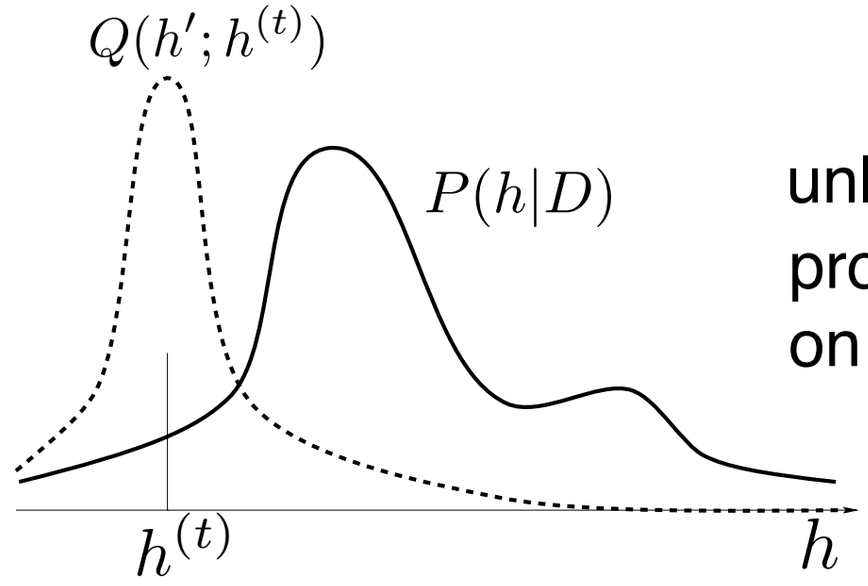
$$E[\phi(h)|D] \approx \frac{1}{T}\sum_t \phi(h^{(t)})$$

where the sequence of samples $h^{(1)}, \ldots, h^{(T)}$ converges to the posterior $P(h|D)$

Proposal distribution:

$$Q(h'; h^{(t)})$$

proposed sample h'



unlike importance sampling proposal h' depends on the current sample $h^{(t)}$

Acceptance ratio:

$$a = \frac{P(h'|D)Q(h^{(t)}; h')}{P(h^{(t)}|D)Q(h'; h^{(t)})}$$

If $a \geq 1$ then the new state is accepted.

Otherwise, the new state is accepted with probability a

If the state is accepted, we set $h^{(t+1)} \leftarrow h'$

If the state is rejected, we set $h^{(t+1)} \leftarrow h^{(t)}$ **(warning! common mistake)**

Metropolis-Hastings algorithm

(example of Markov Chain Monte Carlo (MCMC))

Goal of approximate inference:

$$E[\phi(h)|D] \approx \frac{1}{T}\sum_t \phi(h^{(t)})$$

where the sequence of samples $h^{(1)}, \ldots, h^{(T)}$ converges to the posterior $P(h|D)$

Full Metropolis-Hastings algorithm:

pick initial $h^{(1)}$

for $t \leftarrow 1 \ldots (T-1)$ **do**

  sample $h' \sim Q(h'|h^{(t)})$

  $a = \frac{P(h'|D)Q(h^{(t)}|h')}{P(h^{(t)}|D)Q(h'|h^{(t)})}$

  if $a \geq 1$ **then**

    $h^{(t+1)} \leftarrow h'$

  else

    $h^{(t+1)} \leftarrow h'$ with probability a

    otherwise, $h^{(t+1)} \leftarrow h^{(t)}$

  end if

end for

(important note: in computing acceptance probability 'a', we can safely ignore the normalizing constant P(D) in the posterior — it cancels out— which we often don't know for complex models!)

$$P(h|D) = \frac{P(D|H)P(H)}{P(D)}$$

Metropolis-Hastings algorithm and MCMC

Tricks of the trade:

- If we use a symmetric distribution for *Q*, like a Gaussian, we can simplify the acceptance ratio to:

$$a = \frac{P(h' \mid D)}{P(h^{(t)} \mid D)}$$

- Samples are correlated with one another, so you typically throw out the samples at the beginning of your chain (called **burn in**)

- It is good practice to run multiple chains with different starting points, to examine convergence.

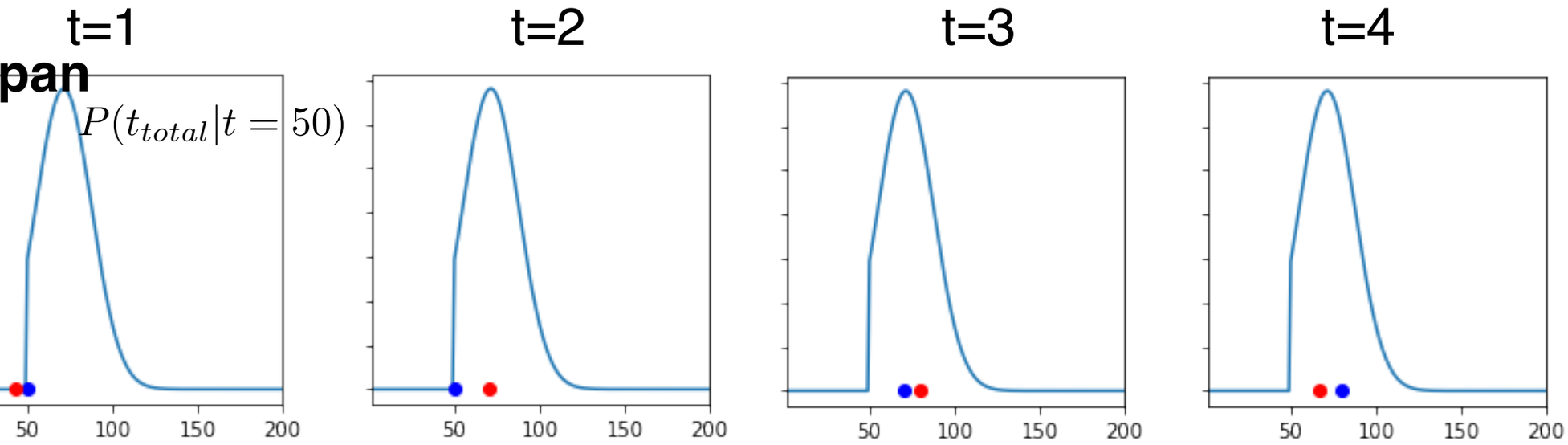
- MCMC can be used as a stochastic search algorithm as well, when searching for the "best hypothesis" by choosing highest-scoring sample

$$h^* = \textbf{argmax}_h \; P(h \mid D)$$

Pros and cons:

- pros: very general; choosing *Q* is important, but it does not need to be as carefully constructed as an importance sampler does
- cons: samples are correlated with each other; it can take a very long time to converge

Example of Metropolis-Hastings for "Optimal Predictions"



$h^{(1)} = 50$ → $h^{(2)} = 50$ $h^{(3)} = 70.7$ $h^{(4)} = 79$

$h' = 42.7$ $h' = 70.7$ $h' = 79$ $h' = 67$

$a = 0$ $a = 2.3$ $a = 0.89$ $a = 1.09$

REJECT **ACCEPT** **ACCEPT** **ACCEPT**

Acceptance ratio:

$$a = \frac{P(h')}{P(h^{(t)})}$$

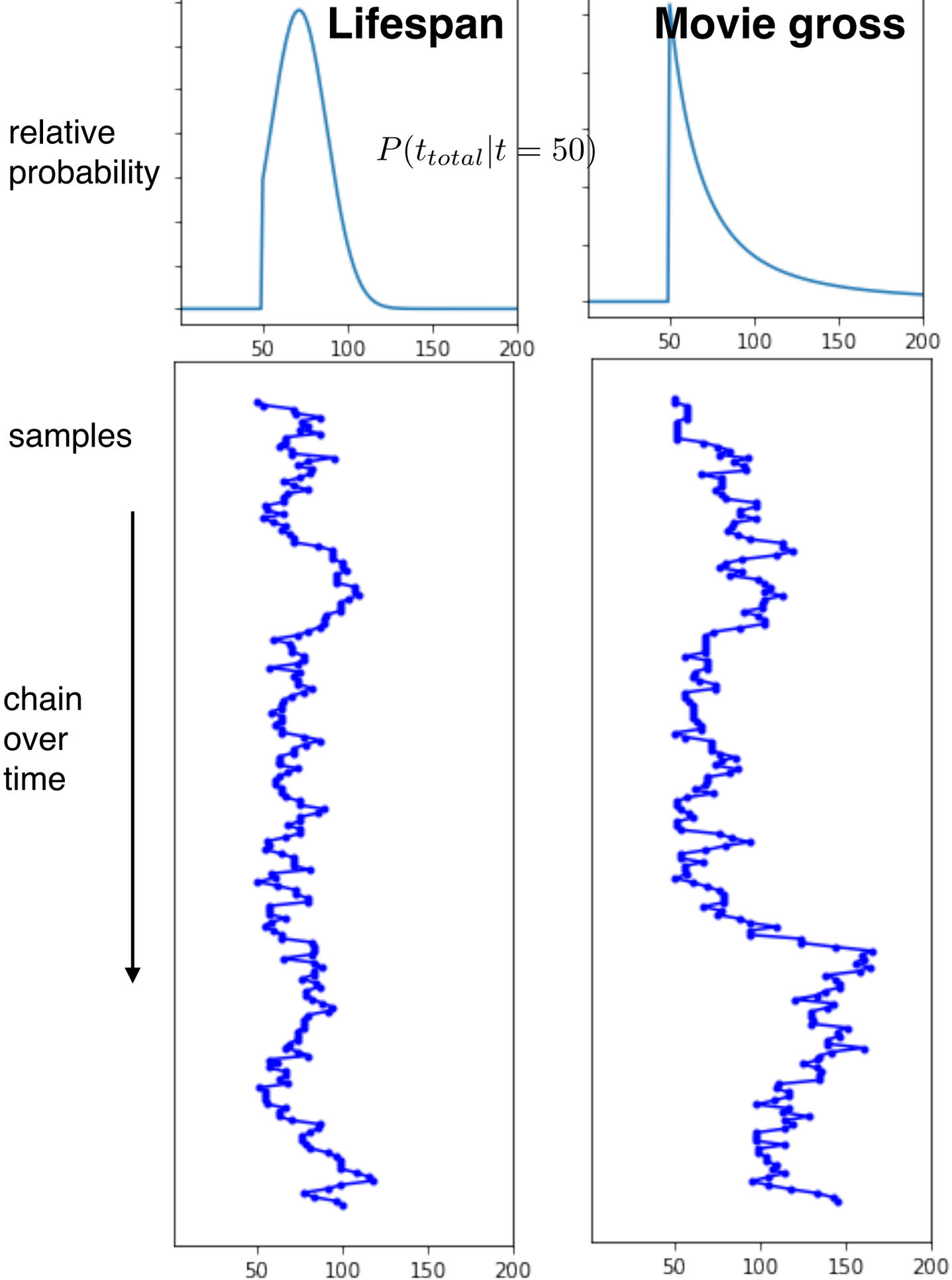
If $a \geq 1$ then the new state is accepted.

Otherwise, the new state is accepted with probability a

Proposal function:

$$Q(h'; h^{(t)}) = N(h^{(t)}, 15)$$

Example of Metropolis-Hastings for "Optimal Predictions"



Posterior mean $E[t_{total}|t = 50]$

	Lifespan	Movie
Exact Inference:	74.3	82.6
Metropolos-Hastings (MCMC; 1000 samples)	74.4	83.0

[Samples track region of high probability!]

Probabilistic programming

- Probabilistic programming is a powerful approach for writing Bayesian models
- The probabilistic model is defined in a structured description language (much like a programming language) using random elements
- Due to random elements, every time the program executes it returns a different output
- Convenient when the prior is too complex to write down as a set of hypotheses, or the model is awkward to write as a probabilistic graphical model (see lecture on graphical models)
- This is a very general way to think about Bayesian modeling — most Bayesian models can be written as simple probabilistic programs

Probabilistic programming: A simple example

Preliminary definitions

```
def flip(theta=0.5):
    return random.random() < theta
```

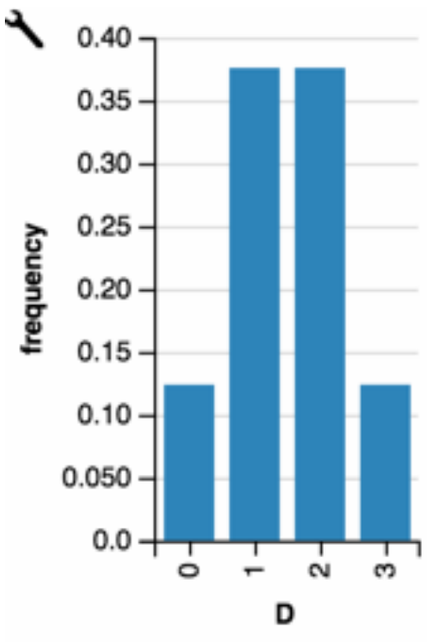
Simple probabilistic program

```
A = flip()
B = flip()
C = flip()
D = A + B + C
```

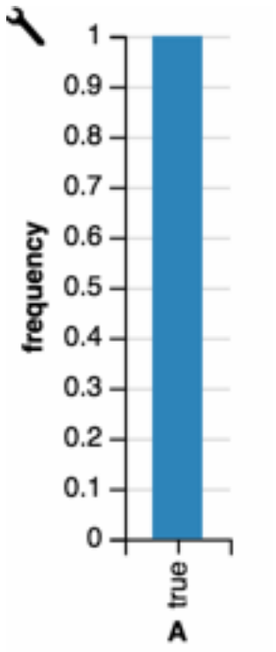
Bayesian inference

(again, notice productivity of reasoning abilities!)

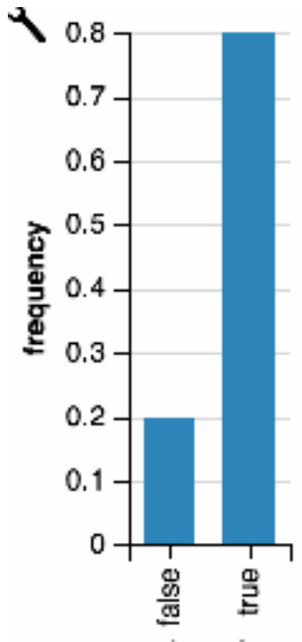
$P(D)$



$P(A|D = 3)$



$P(A|D \geq 2)$



Example from Noah Goodman and Josh Tenenbaum https://probmods.org/

Probabilistic programming: Another example

Simple probabilistic program (yet more complex than before)

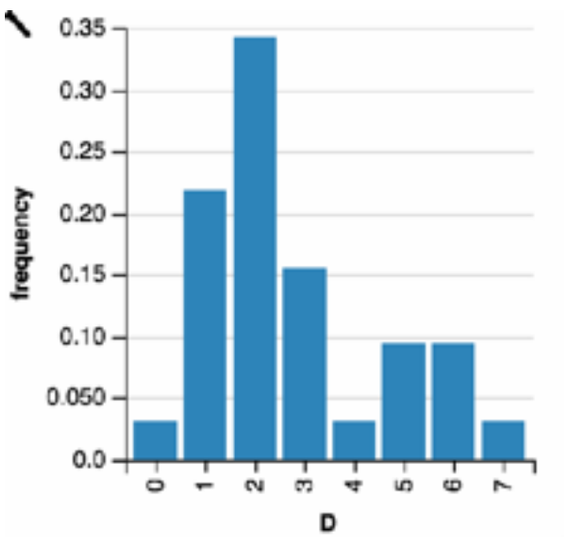
```
A = flip()
B = flip()
C = flip()
if C:
    D = A + B + C
else:
    E = flip()
    F = (2*flip())**2
    D = A + B + C +E + F
```

Key idea: A probabilistic program is a generative process for producing data

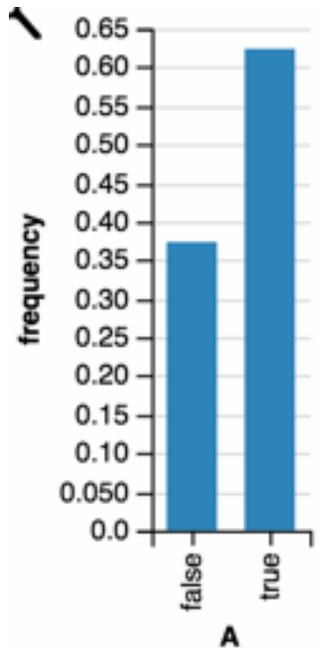
Hypotheses are then all possible ways the data could have been generated

Bayesian inference

$P(D)$



$P(A|D \geq 2)$



Key resource on probabilistic programming perspective to cognitive modeling (probmods.org)

Probabilistic Models of Cognition

2nd Edition

by Noah D. Goodman & Joshua B. Tenenbaum

This book explores the probabilistic approach to cognitive science, which models learning and reasoning as inference in complex probabilistic models. We examine how a broad range of empirical phenomena, including intuitive physics, concept learning, causal reasoning, social cognition, and language understanding, can be modeled using a functional probabilistic programming language called WebPPL.

Citation

N. D. Goodman and J. B. Tenenbaum (2016). *Probabilistic Models of Cognition* (2nd ed.). Retrieved 2018-4-2 from `https://probmods.org/` [bibtex]

Open source

- Book content
  Markdown code for the book chapters
- WebPPL
  A probabilistic programming language for the web

Previous edition

The first edition of this book used the probabilistic programming language Church and can be found here.

Chapters

1. Introduction
   A brief introduction to the philosophy.
2. Generative models
   Representing working models with probabilistic programs.
3. Conditioning
   Asking questions of models by conditional inference.
4. Patterns of inference
   Causal and statistical dependence. Conditional dependence.
5. Models for sequences of observations
   Generative models of the relations between data points
6. Inference about inference
   Models on models on models

Key principles of Bayesian models of cognition

- Start with analyzing the computational problem that has to be solved, and describe it as a problem of Bayesian inference

- A successful computational level model provides strong constraints when developing an algorithmic and implementational level model

- Bayesian inference provides a flexible framework for testing different hypotheses about representation, without having to worry about how to define special algorithms for inference and learning