

Monte Carlo and the mind

Tom Griffiths

Department of Psychology

Program in Cognitive Science

University of California, Berkeley

Two uses of Monte Carlo methods

1. For solving problems of probabilistic inference involved in developing computational models
2. As a source of hypotheses about how the mind might solve problems of probabilistic inference

Answers and expectations

- For a function $f(x)$ and distribution $P(x)$, the expectation of f with respect to P is

$$E_{P(x)}[f(x)] = \sum_x f(x)P(x)$$

- The expectation is the average of f , when x is drawn from the probability distribution P

Answers and expectations

$$E_{P(x)}[f(x)] = \sum_x f(x)P(x)$$

- Example 1: The average # of spots on a die roll
 - $x = 1, \dots, 6$, $f(x) = x$, $P(x)$ is uniform
- Example 2: The probability two observations belong to the same mixture component
 - x is an assignment of observations to components, $f(x) = 1$ if observations belong to same component and 0 otherwise, $P(x)$ is posterior over assignments

The Monte Carlo principle

- The expectation of f with respect to P can be approximated by

$$E_{P(x)}[f(x)] \approx \frac{1}{n} \sum_{i=1}^n f(x_i)$$

where the x_i are sampled from $P(x)$

- Example 1: the average # of spots on a die roll

The Monte Carlo principle

The law of large numbers

Average number of spots

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Number of rolls

When simple Monte Carlo fails

- Efficient algorithms for sampling only exist for a relatively small number of distributions

When simple Monte Carlo fails

- Efficient algorithms for sampling only exist for a relatively small number of distributions
- Sampling from distributions over large discrete state spaces is computationally expensive
 - mixture model with n observations and k components, k^n possible component assignment for observations

When simple Monte Carlo fails

- Efficient algorithms for sampling only exist for a relatively small number of distributions
- Sampling from distributions over large discrete state spaces is computationally expensive
 - mixture model with n observations and k components, k^n possible component assignment for observations
- Sometimes we want to sample from distributions for which we only know the probability of each state up to a multiplicative constant

Why Bayesian inference is hard

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

Evaluating the posterior probability of a hypothesis requires summing over all hypotheses

(statistical physics: computing partition function)

Modern Monte Carlo methods

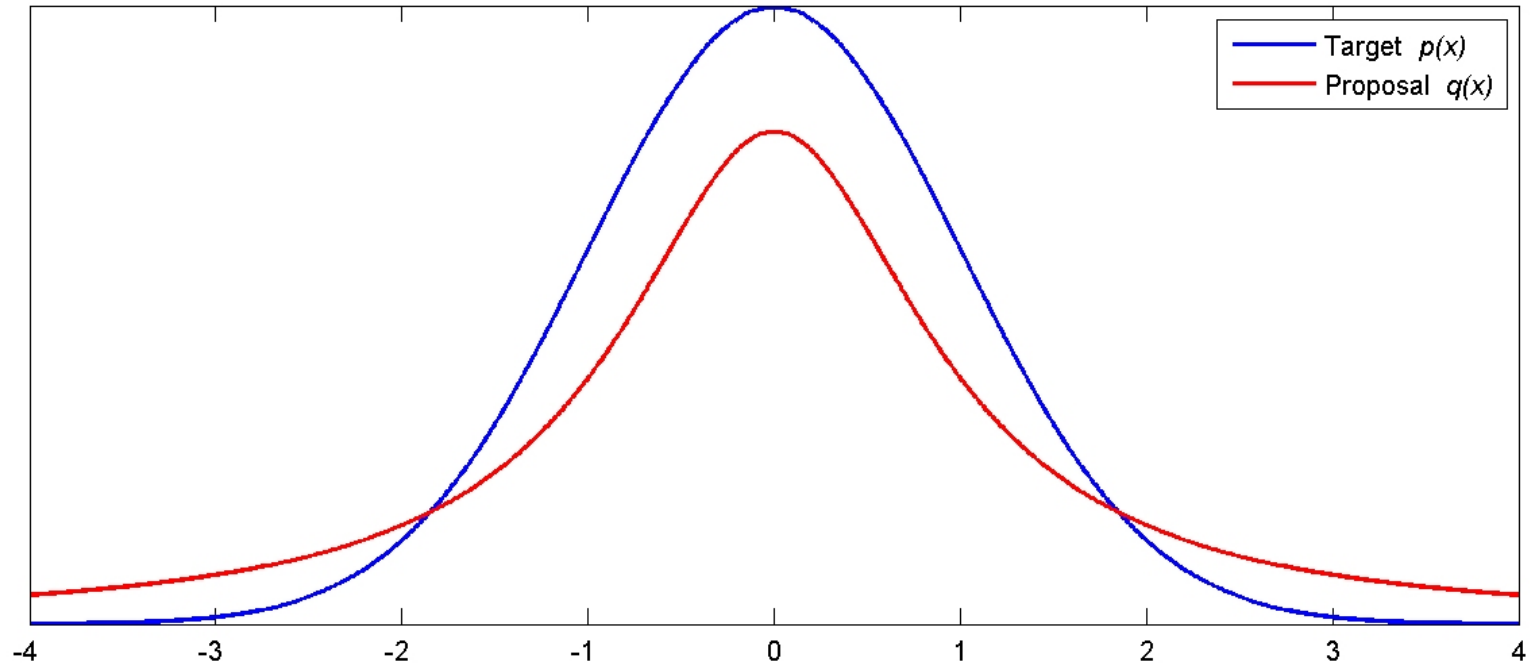
- Sampling schemes for distributions with large state spaces known up to a multiplicative constant
- Two approaches:
 - importance sampling
 - Markov chain Monte Carlo
- (Major competitors... variational inference, sophisticated numerical quadrature methods)

Importance sampling

Basic idea: generate from the wrong distribution, assign weights to samples to correct for this

$$\begin{aligned} E_{p(x)} [f(x)] &= \int f(x) p(x) dx \\ &= \int f(x) \frac{p(x)}{q(x)} q(x) dx \\ &\approx \frac{1}{n} \sum_{i=1}^n f(x_i) \frac{p(x_i)}{q(x_i)} \quad \text{for } x_i \sim q(x) \end{aligned}$$

Importance sampling



works when sampling from proposal is easy, target is hard

An alternative scheme...

$$E_{p(x)}[f(x)] \approx \frac{1}{n} \sum_{i=1}^n f(x_i) \frac{p(x_i)}{q(x_i)} \quad \text{for } x_i \sim q(x)$$

$$E_{p(x)}[f(x)] \approx \frac{\sum_{i=1}^n f(x_i) \frac{p(x_i)}{q(x_i)}}{\sum_{i=1}^n \frac{p(x_i)}{q(x_i)}} \quad \text{for } x_i \sim q(x)$$

works when $p(x)$ is known up to a multiplicative constant

Optimal importance sampling

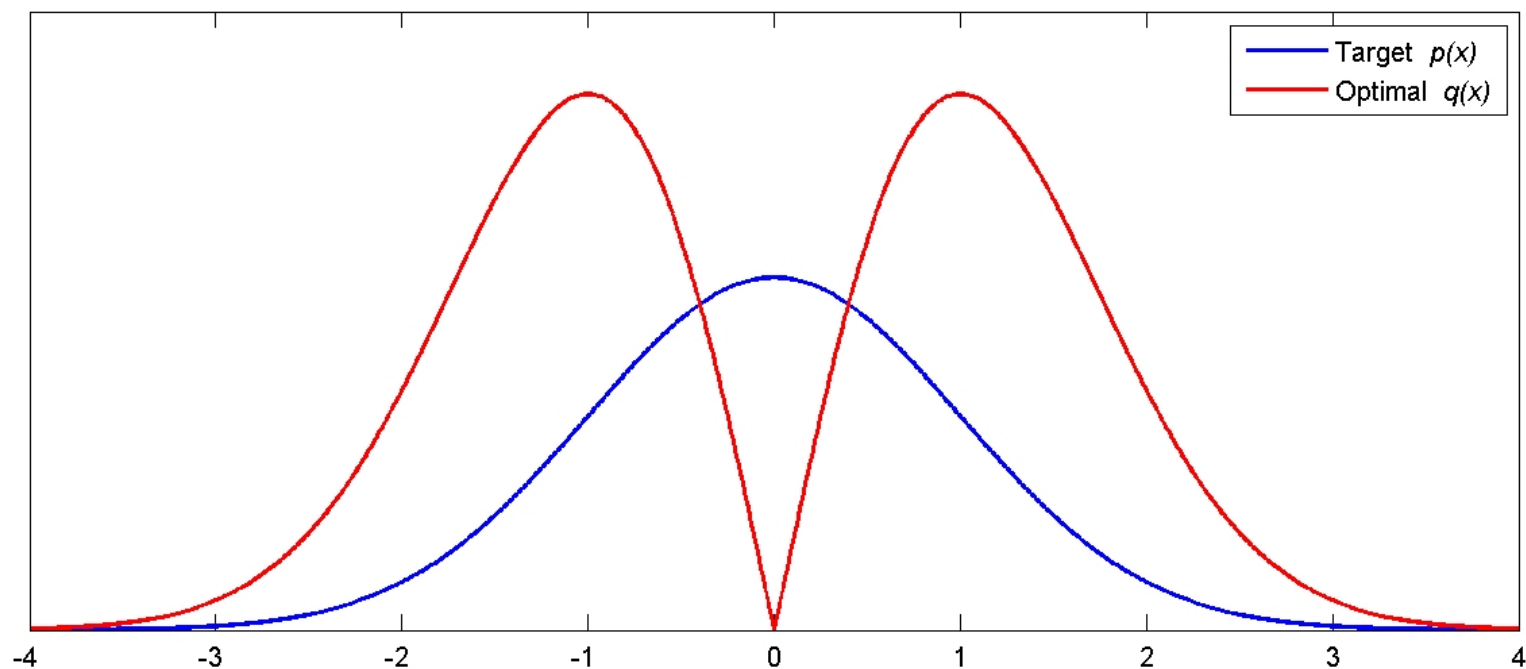
- Asymptotic variance is

$$\sigma_{IS}^2 = E_{p(x)} \left[(f(x) - E_{p(x)}[f(x)])^2 \frac{p(x)}{q(x)} \right]$$

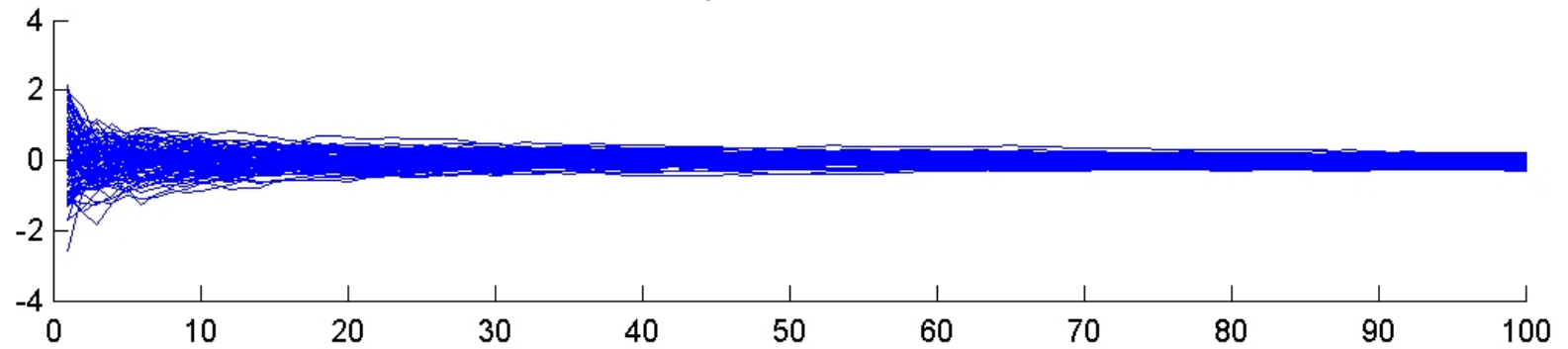
- This is minimized by

$$q(x) \propto |f(x) - E_{p(x)}[f(x)]| p(x)$$

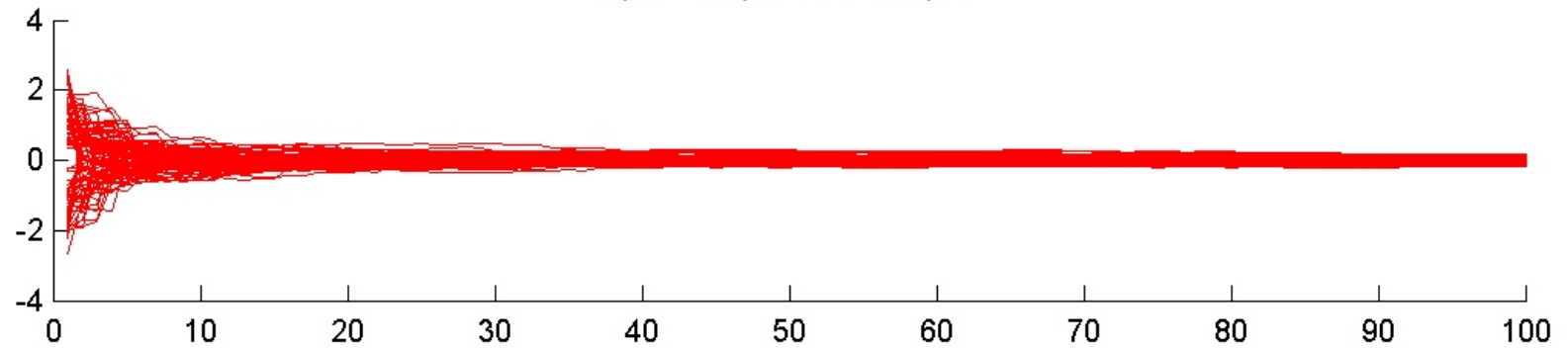
Optimal importance sampling



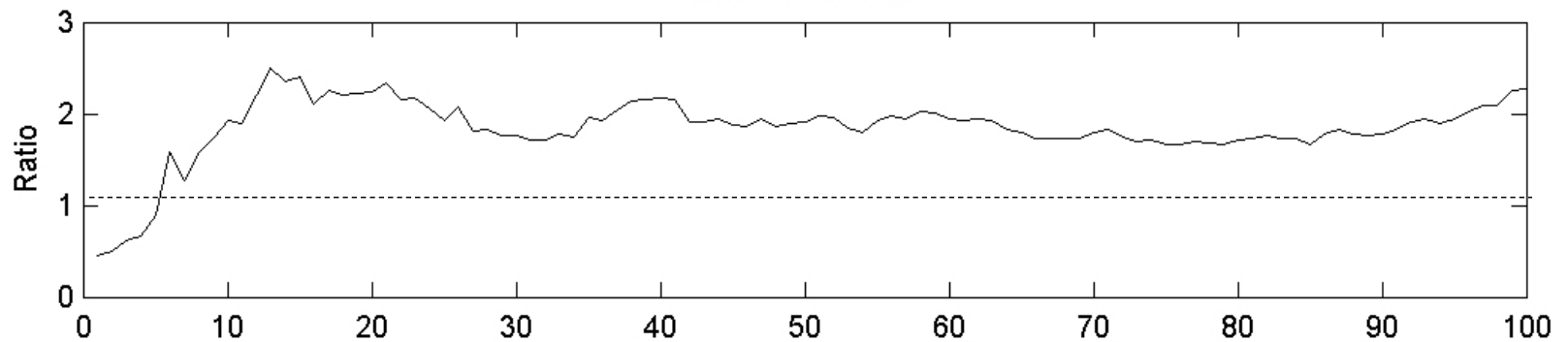
Simple Monte Carlo



Optimal importance sampler



Ratio of variances



Likelihood weighting

- A particularly simple form of importance sampling for posterior distributions
- Use the prior as the proposal distribution
- Weights:

$$\frac{p(h | d)}{p(h)} = \frac{p(d | h)p(h)}{p(d)p(h)} = \frac{p(d | h)}{p(d)} \propto p(d | h)$$

Approximating Bayesian inference

$$E_{p(h|d)}[f(h)] \approx \frac{\sum_i p(d | h^{(i)}) f(h^{(i)})}{\sum_i p(d | h^{(i)})}$$

Sample from the prior, weight by the likelihood

Exemplar models

- Assume decisions are made by storing previous events in memory, then activating by similarity
- For example, categorization:

$$P_{choice}(c | x) = \frac{\sum_i s(x, x^{(i)}) I(x^{(i)} \in c)}{\sum_i s(x, x^{(i)})}$$

where $x^{(i)}$ are exemplars, $s(x, x^{(i)})$ is similarity, $I(x^{(i)} \in c)$ is 1 if $x^{(i)}$ is from category c

(e.g., Nosofsky, 1986)

Exemplar models

- Assume decisions are made by storing previous events in memory, then activating by similarity
- General version:

$$response(x) = \frac{\sum_i s(x, x^{(i)}) f(x^{(i)})}{\sum_i s(x, x^{(i)})}$$

where $x^{(i)}$ are exemplars, $s(x, x^{(i)})$ is similarity, $f(x^{(i)})$ is quantity of interest

Equivalence

$$response(x) = \frac{\sum_i s(x, x^{(i)}) f(x^{(i)})}{\sum_i s(x, x^{(i)})}$$

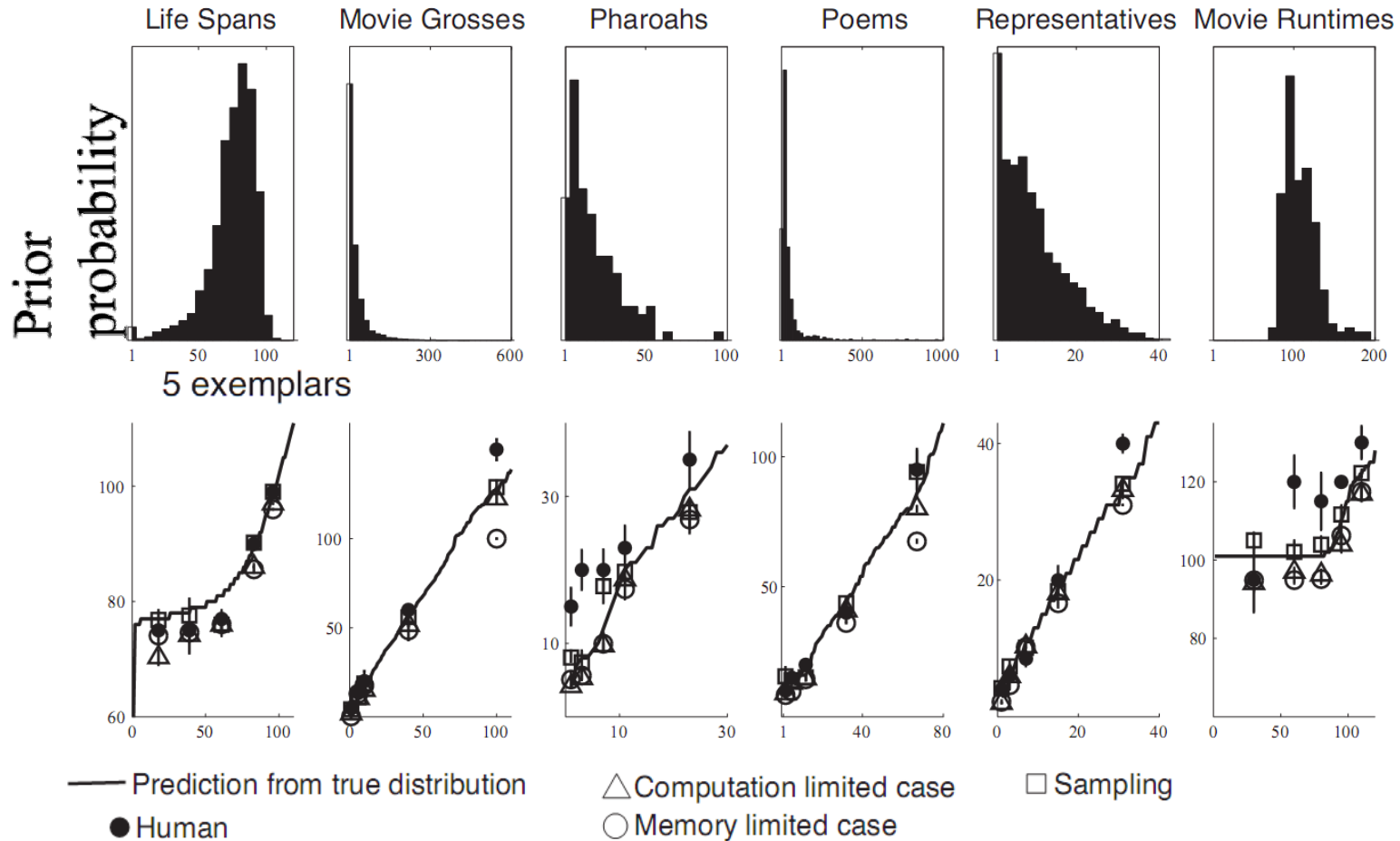
$$E_{p(h|d)}[f(h)] \approx \frac{\sum_i p(d | h^{(i)}) f(h^{(i)})}{\sum_i p(d | h^{(i)})}$$

Bayes can be approximated using exemplar models, storing hypotheses sampled from prior

Predicting the future

- Assume people store examples of t_{total} in memory, and activate after observing t
 - likelihood is 0 for $t_{total} < t$, else $1/t_{total}$
- Explore different kinds of constraints:
 - “memory limited”: limit total number recalled
 - “computation limited”: limit total number $> t_{total}$

Predicting the future



Importance sampling

- A general scheme for sampling from complex distributions that have simpler relatives
- Simple methods for sampling from posterior distributions in some cases (easy to sample from prior, prior and posterior are close)
- Can be more efficient than simple Monte Carlo
- Links to exemplar models in psychology
- Also provides a solution to the question of how people can update beliefs as data come in...

Updating distributions over time...

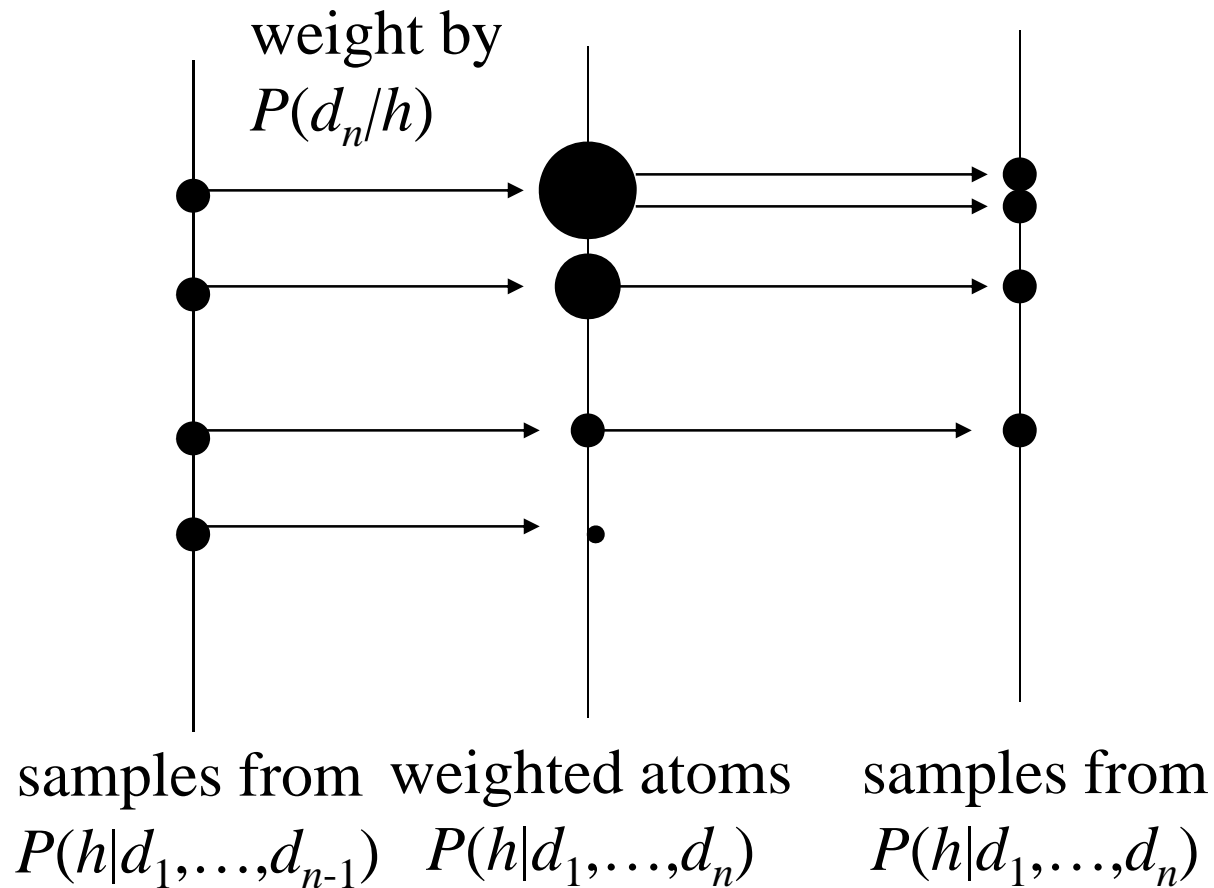
- Computational costs are compounded when data are observed incrementally...
 - recompute $P(h|d_1, \dots, d_n)$ after observing d_n
- Exploit “yesterday’s posterior is today’s prior”

$$P(h | d_1, \dots, d_n) \propto P(d_n | h)P(h | d_1, \dots, d_{n-1})$$

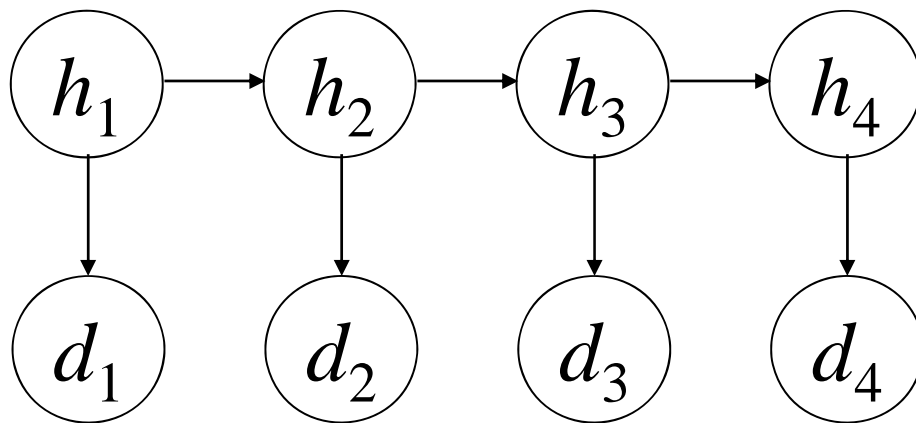
- Repeatedly using importance sampling results in an algorithm known as a “particle filter”

Particle filter

$$P(h \mid d_1, \dots, d_n) \propto P(d_n \mid h)P(h \mid d_1, \dots, d_{n-1})$$



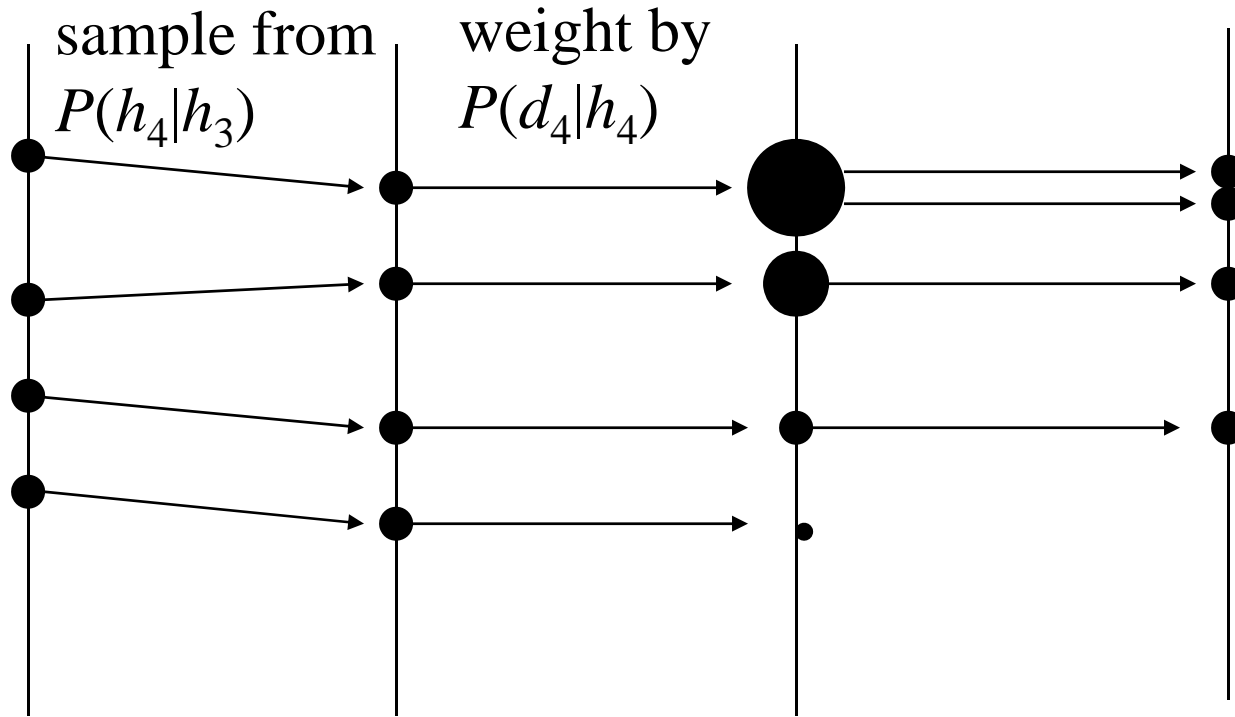
Dynamic hypotheses



$$\begin{aligned} P(h_4 \mid d_1, \dots, d_4) &\propto P(d_4 \mid h_4) P(h_4 \mid d_1, \dots, d_3) \\ &= P(d_4 \mid h_4) \sum_{h_3} P(h_4 \mid h_3) P(h_3 \mid d_1, \dots, d_3) \end{aligned}$$

Particle filters

$$P(h_4 | d_1, \dots, d_4) \propto P(d_4 | h_4) \sum_{h_3} P(h_4 | h_3) P(h_3 | d_1, \dots, d_3)$$



samples from $P(h_3 | d_1, \dots, d_3)$
 samples from $P(h_4 | d_1, \dots, d_3)$
 weighted atoms $P(h_4 | d_1, \dots, d_4)$
 samples from $P(h_4 | d_1, \dots, d_4)$

The promise of particle filters

- A general scheme for defining rational process models of updating over time (Sanborn et al., 2006)
- Model limited memory, and produce order effects (cf. Kruschke, 2006)
- Used to define rational process models of...
 - categorization (Sanborn et al., 2006)
 - associative learning (Daw & Courville, 2008)
 - changepoint detection (Brown & Steyvers, 2009)
 - sentence processing (Levy et al., 2009)

Two uses of Monte Carlo methods

1. For solving problems of probabilistic inference involved in developing computational models
2. As a source of hypotheses about how the mind might solve problems of probabilistic inference

Three

~~Two~~ uses of Monte Carlo methods

1. For solving problems of probabilistic inference involved in developing computational models
2. As a source of hypotheses about how the mind might solve problems of probabilistic inference
3. As a way to explore people's subjective probability distributions

Human learning

Categorization

Causal learning

Function learning

Representations

Language

Experiment design

...

Machine learning

Density estimation

Graphical models

Regression

Nonparametric Bayes

Probabilistic grammars

Inference algorithms

...

Two deep questions

- What are the biases that guide human learning?
- What do mental representations look like?



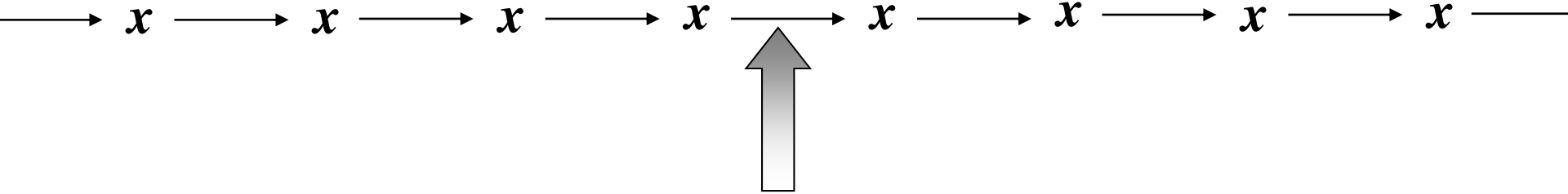
$$\lim_{t \rightarrow \infty} P(x^{(t)} = i \mid x^{(0)}) = \pi_i$$

Two deep questions

- What are the biases that guide human learning?
 - prior probability distribution on hypotheses, $P(h)$
- What do mental representations look like?
 - distribution over objects x in category c , $P(x|c)$

Develop ways to sample from these distributions

Markov chains



Transition matrix

$$\mathbf{T} = P(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})$$

- Variables $\mathbf{x}^{(t+1)}$ independent of history given $\mathbf{x}^{(t)}$
- Converges to a *stationary distribution* under easily checked conditions (i.e., if it is ergodic)

Markov chain Monte Carlo

- Sample from a target distribution $P(\mathbf{x})$ by constructing Markov chain for which $P(\mathbf{x})$ is the stationary distribution
- Two main schemes:
 - Gibbs sampling
 - Metropolis-Hastings algorithm

Gibbs sampling

Particular choice of proposal distribution

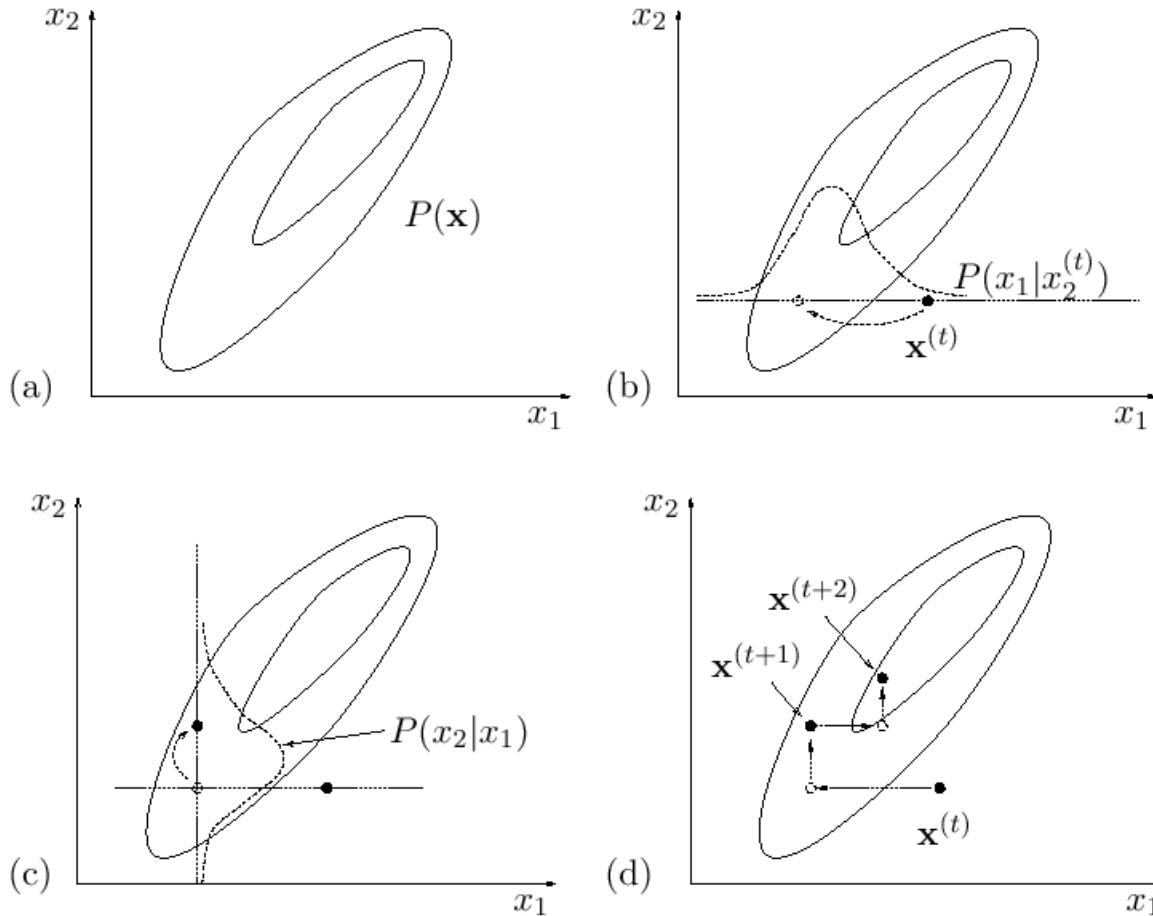
For variables $\mathbf{x} = x_1, x_2, \dots, x_n$

Draw $x_i^{(t+1)}$ from $P(x_i/\mathbf{x}_{-i})$

$$\mathbf{x}_{-i} = x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \dots, x_n^{(t)}$$

(this is called the *full conditional* distribution)

Gibbs sampling



Iterated learning

(Kirby, 2001)

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

What are the consequences of learners
learning from other learners?

Objects of iterated learning

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

How do constraints on learning (inductive biases)
influence cultural universals?

Analyzing iterated learning

$$P_L(h|d)$$

$$P_L(h|d)$$

$P_P(d|h)$
QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

$$P_P(d|h)$$

$P_L(h|d)$: probability of inferring hypothesis h from data d

$P_P(d|h)$: probability of generating data d from hypothesis h

Analyzing iterated learning

$$d_0 \xrightarrow{P_L(h|d)} h_1 \xrightarrow{P_P(d|h)} d_1 \xrightarrow{P_L(h|d)} h_2 \xrightarrow{P_P(d|h)} d_2 \xrightarrow{P_L(h|d)} h_3 \xrightarrow{\quad} \dots$$

A Markov chain on hypotheses

$$h_1 \xrightarrow{\sum_d P_P(d|h) P_L(h|d)} h_2 \xrightarrow{\sum_d P_P(d|h) P_L(h|d)} h_3 \xrightarrow{\quad} \dots$$

A Markov chain on data

$$d_0 \xrightarrow{\sum_h P_L(h|d) P_P(d|h)} d_1 \xrightarrow{\sum_h P_L(h|d) P_P(d|h)} d_2 \xrightarrow{\quad} \dots$$

Iterated Bayesian learning

$$P_L(h|d)$$

$$P_L(h|d)$$

$P_P(d|h)$
QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this pictu

$$P_P(d|h)$$

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this pictu

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this pictu

Assume learners *sample* from their posterior distribution:

$$P_L(h | d) = \frac{P_P(d | h)P(h)}{\sum_{h' \in H} P_P(d | h')P(h')}$$

Stationary distributions

- Markov chain on h converges to the prior, $P(h)$
- Markov chain on d converges to the “prior predictive distribution”

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

(Griffiths & Kalish, 2005)

Explaining convergence to the prior

$$P_L(h|d)$$

$$P_L(h|d)$$

$P_P(d|h)$
QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture

$P_P(d|h)$

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

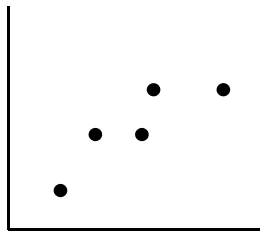
QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

- Intuitively: data acts once, prior many times
- Formally: iterated learning with Bayesian agents is a *Gibbs sampler* on $P(d,h)$

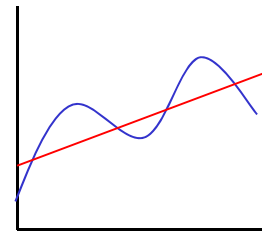
(Griffiths & Kalish, 2007)

Iterated function learning

data



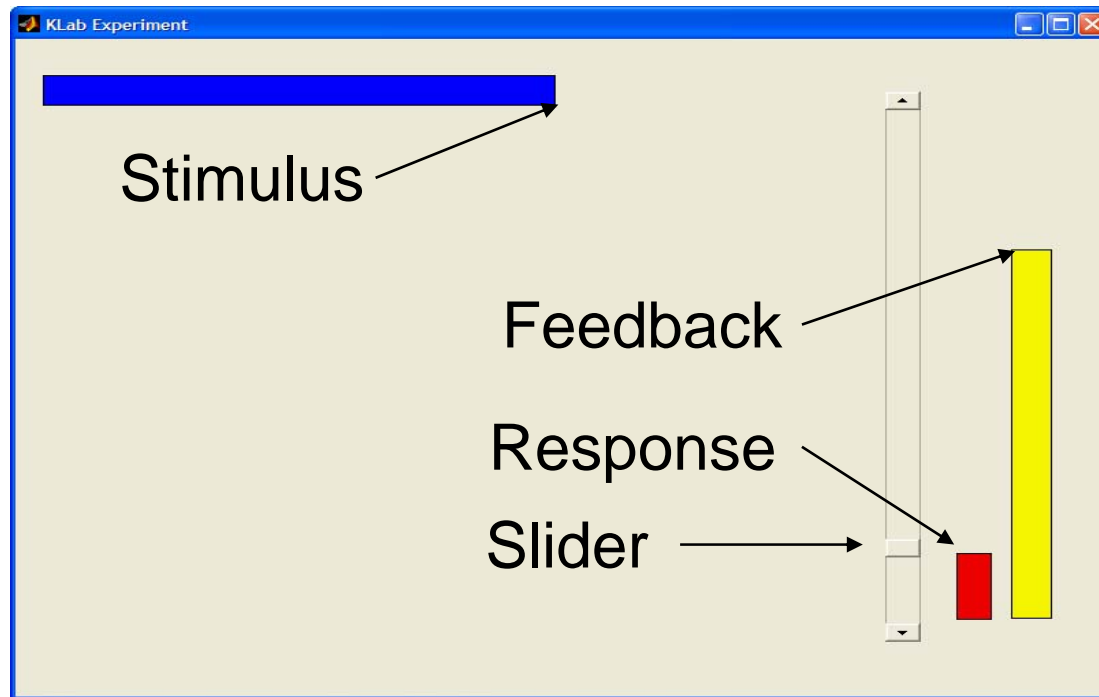
hypotheses



- Each learner sees a set of (x,y) pairs
- Makes predictions of y for new x values
- Predictions are data for the next learner

(Kalish, Griffiths, & Lewandowsky, 2007)

Function learning experiments



Examine iterated learning with different initial data

Initial
data



Iteration

1

2

3

4

5

6

7

8

9

Iterated predicting the future

data

A movie has made
\$30 million so far

hypotheses

\$60 million total

- Each learner sees values of t
- Makes predictions of t_{total}
- The next value of t is chosen from $(0, t_{total})$

(Lewandowsky, Griffiths & Kalish, 2009)

Movie grosses

Poems

Chains of predictions

t_{total}

QuickTime¹
decompr
are needed to see

t_{total}
ire.

Iteration

Iteration

(Lewandowsky, Griffiths & Kalish, 2009)

Stationary distributions

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

(Lewandowsky, Griffiths & Kalish, 2009)

Identifying inductive biases

- Concept learning



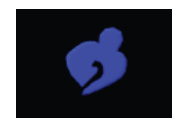
(Griffiths, Christian, & Kalish, 2008)

- Reproduction from memory



(Xu & Griffiths, 2008)

- Learning linguistic frequencies



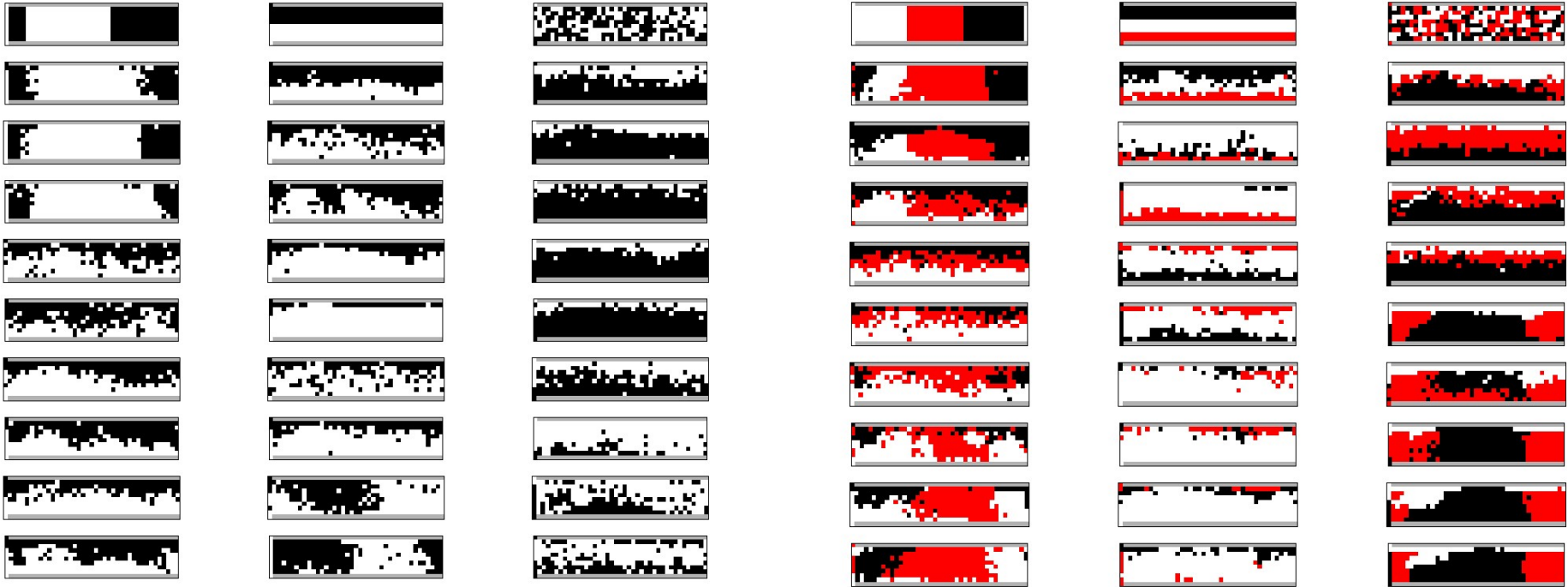
“DUP”

(Reali & Griffiths, 2009)

- All show convergence to priors...

Comparing to universals...

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.



(Xu, Dowman, & Griffiths, in press)

Metropolis-Hastings algorithm

(Metropolis et al., 1953; Hastings, 1970)

Step 1: propose a state (we assume symmetrically)

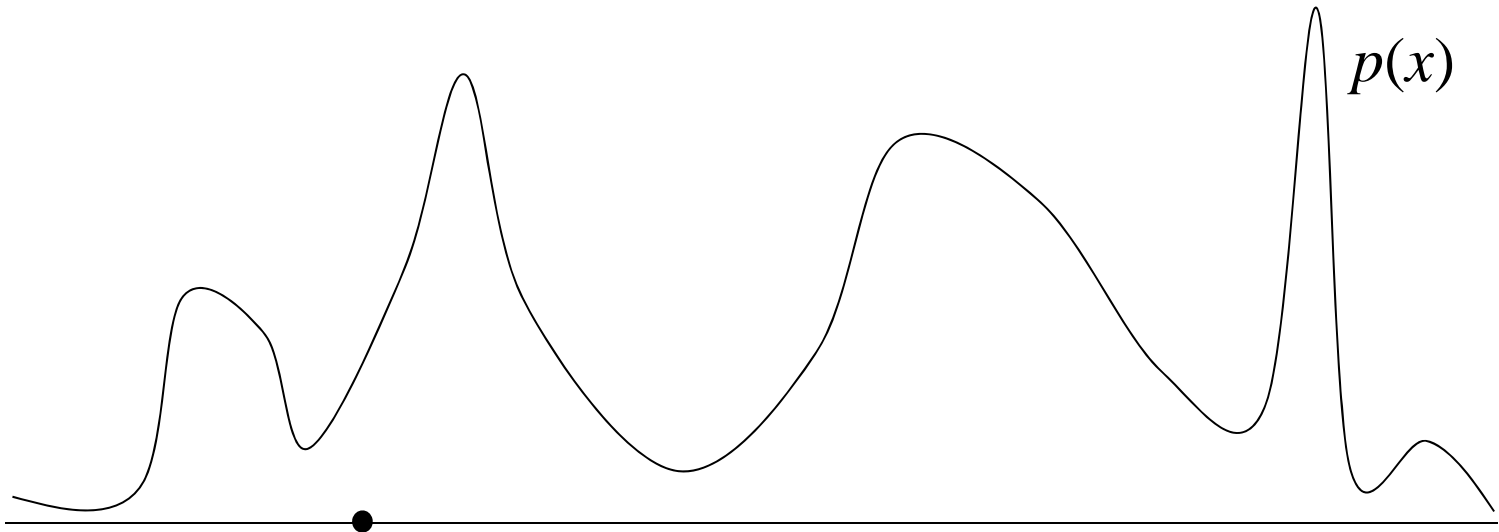
$$Q(x^{(t+1)}|x^{(t)}) = Q(x^{(t)}|x^{(t+1)})$$

Step 2: decide whether to accept, with probability

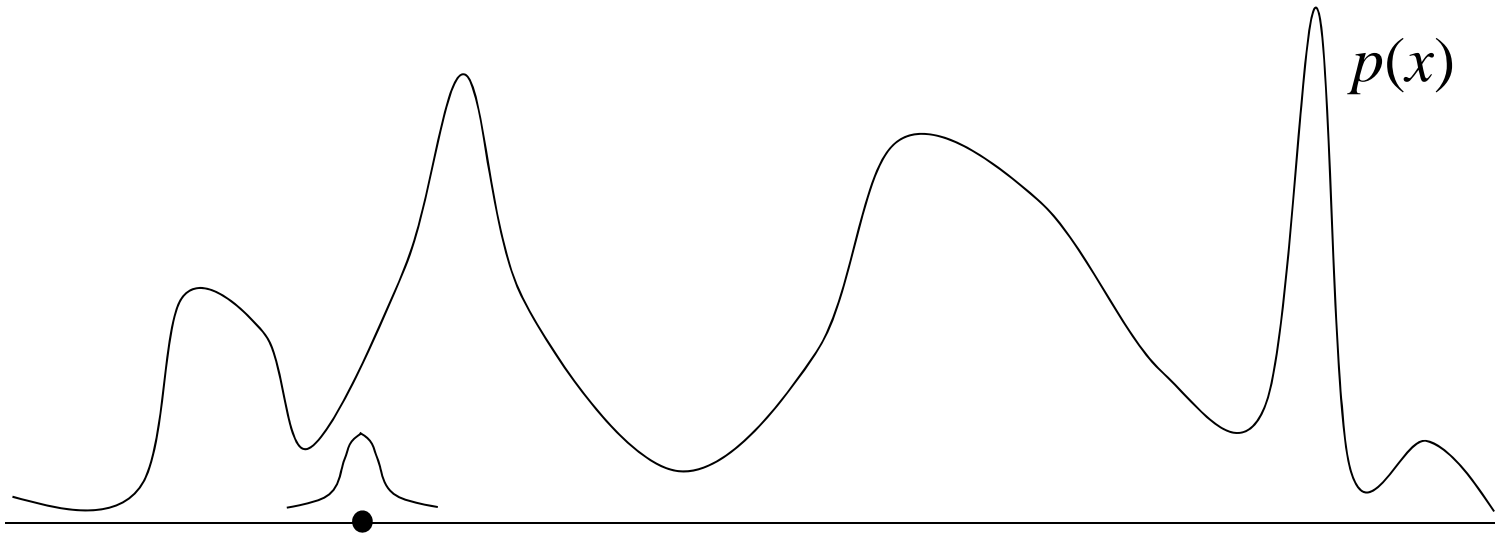
$$A(x^{(t+1)}, x^{(t)}) = \min \left(1, \frac{p(x^{(t+1)})}{p(x^{(t)})} \right) \quad \text{Metropolis acceptance function}$$

$$A(x^{(t+1)}, x^{(t)}) = \frac{p(x^{(t+1)})}{p(x^{(t+1)}) + p(x^{(t)})} \quad \text{Barker acceptance function}$$

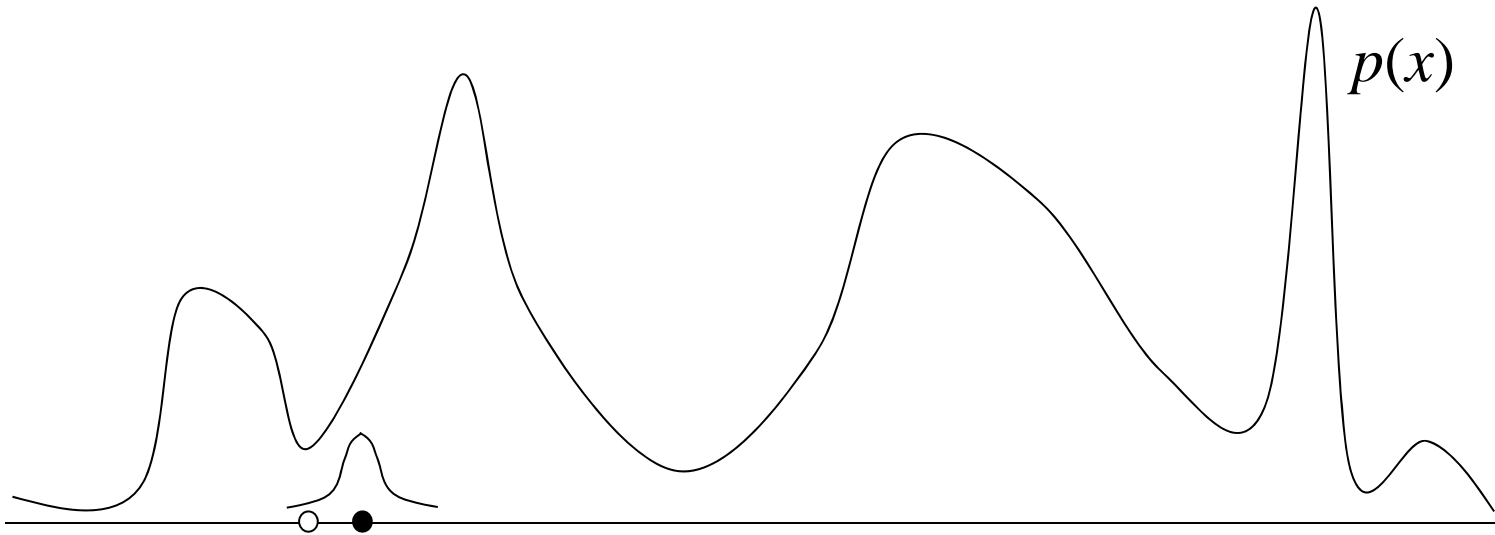
Metropolis-Hastings algorithm



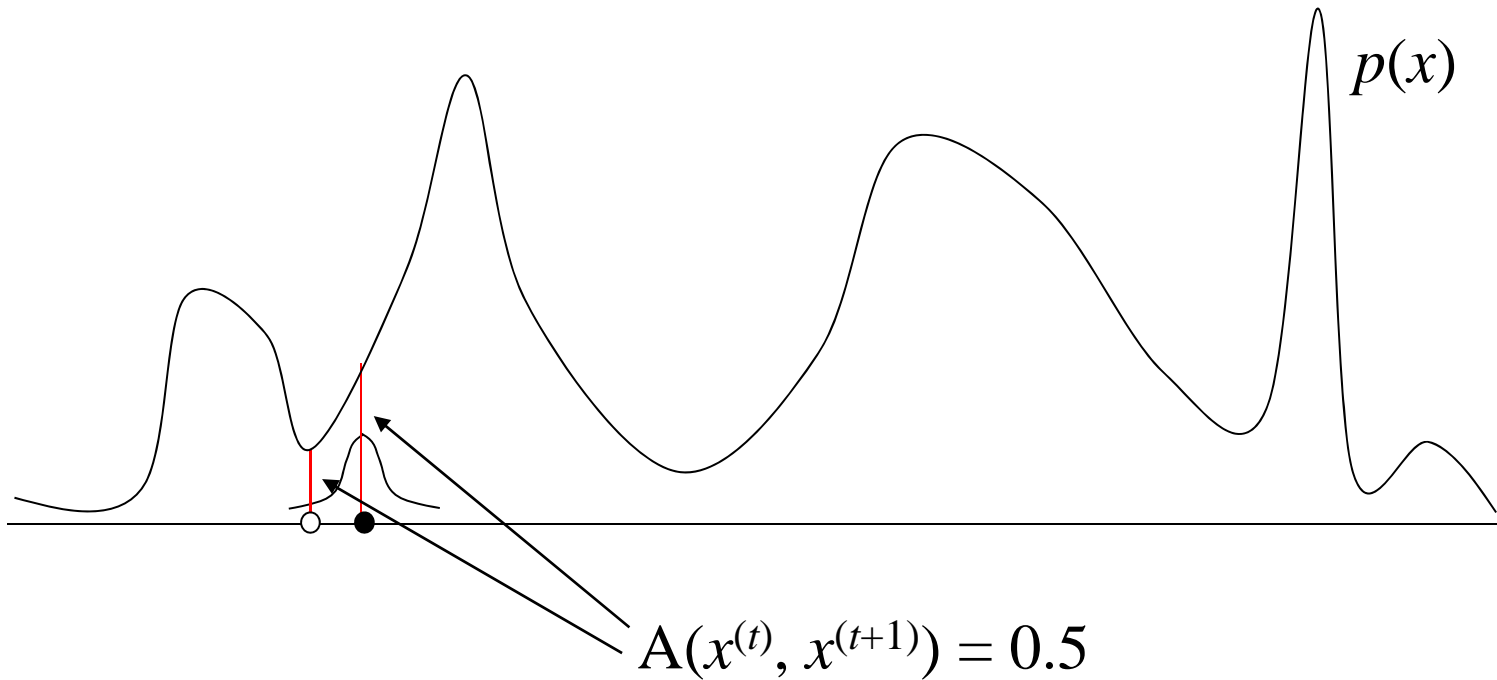
Metropolis-Hastings algorithm



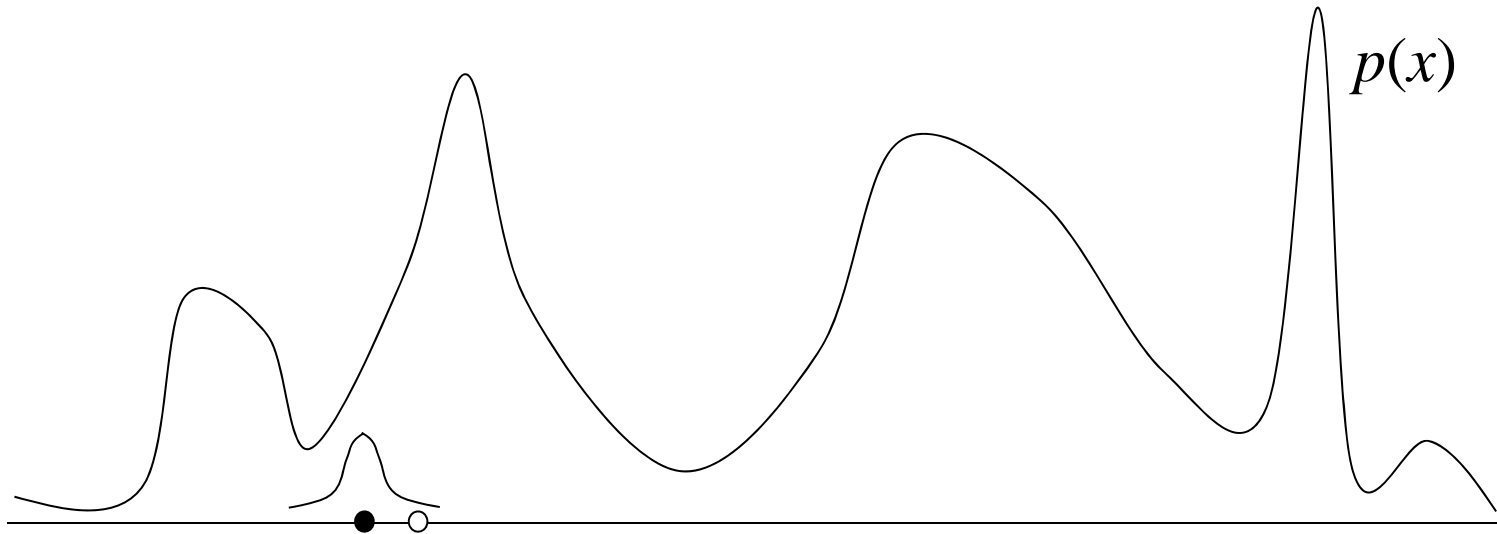
Metropolis-Hastings algorithm



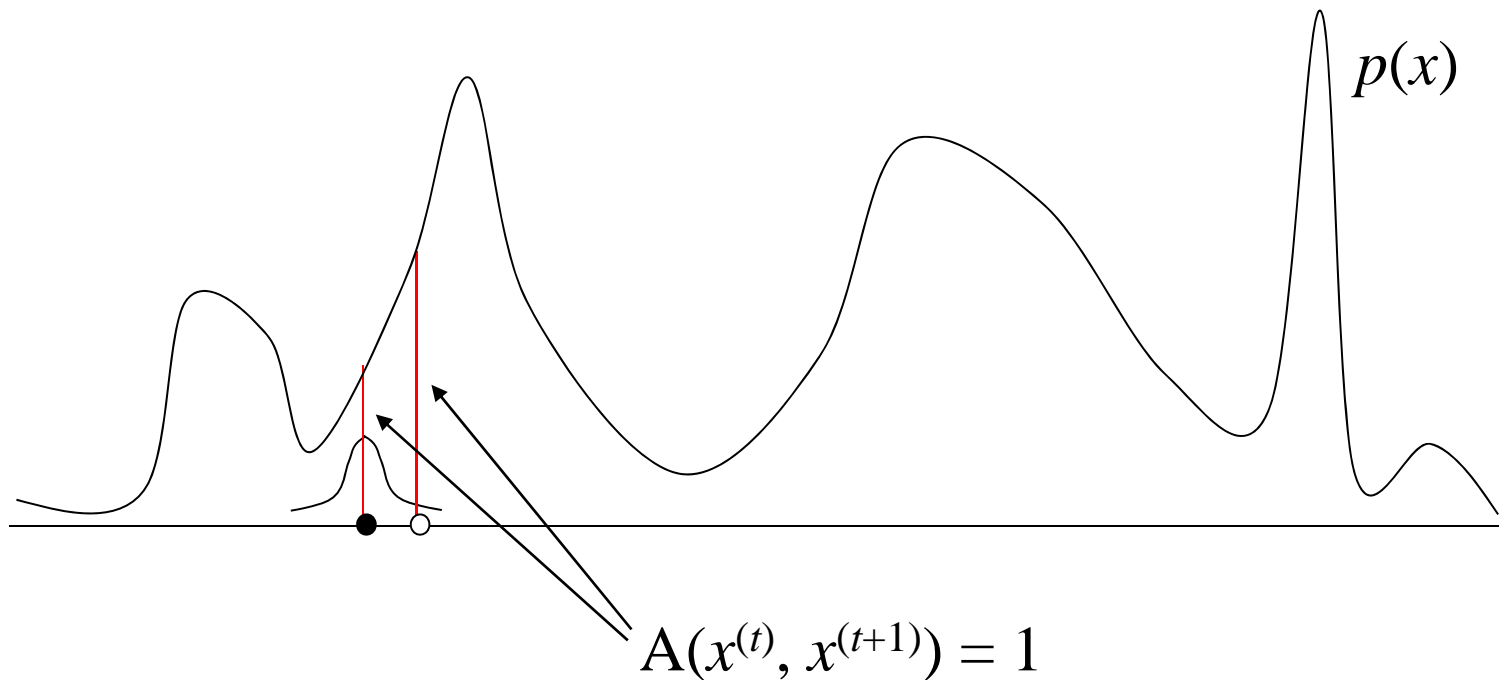
Metropolis-Hastings algorithm



Metropolis-Hastings algorithm



Metropolis-Hastings algorithm



Sampling subjective quantities

- Assume we want to gather information about a subjectively represented non-negative quantity $f(x)$
- If people's responses follow the Luce choice rule...

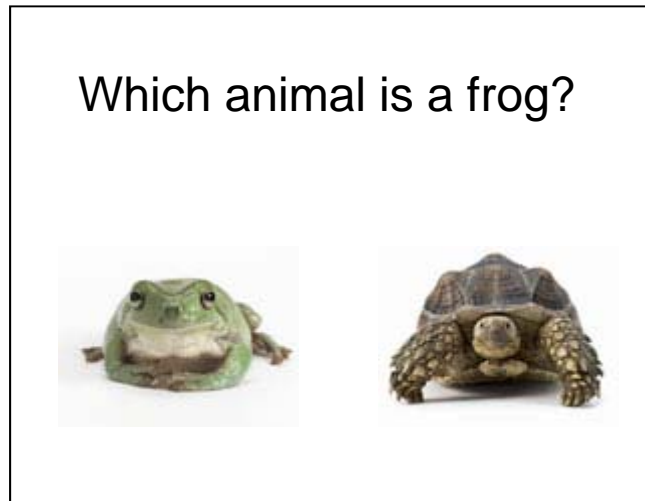
$$P_{\text{choice}}(x_1) = \frac{f(x_1)}{f(x_1) + f(x_2)}$$

- ...we can sample from $p(x) \propto f(x)$ using only pairwise comparison of alternatives (a 2AFC task)

(Sanborn & Griffiths, 2008)

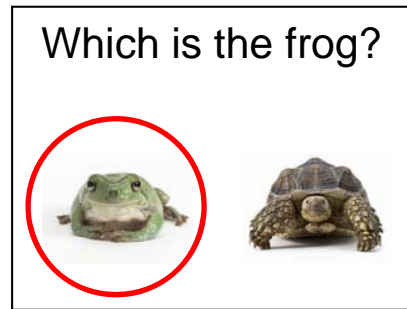
An example: categorization

Assume that a category (e.g., frogs) is represented by a subjective probability distribution, $p(x|c)$

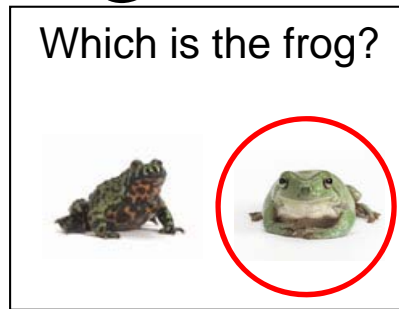


$$P_{\text{choice}}(x_1) = \frac{p(x_1 | c)}{p(x_1 | c) + p(x_2 | c)}$$

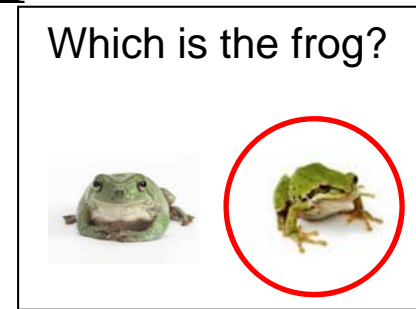
Collecting the samples



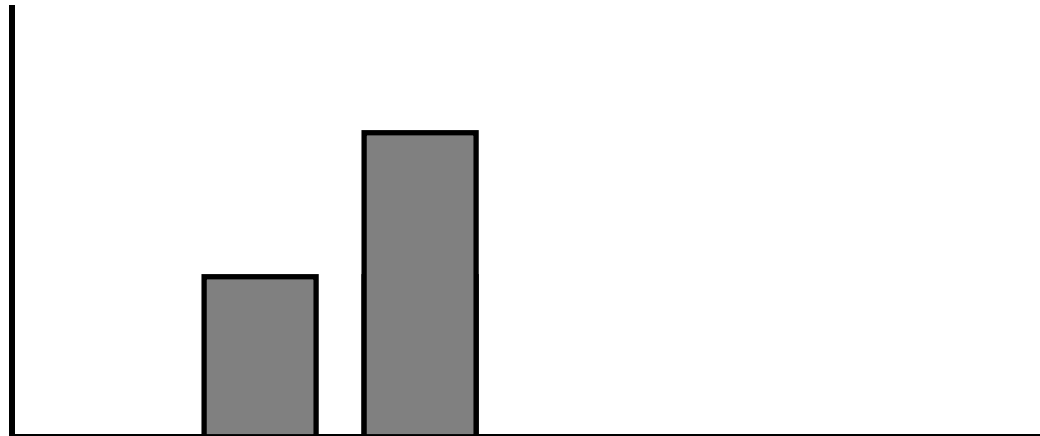
Trial 1



Trial 2

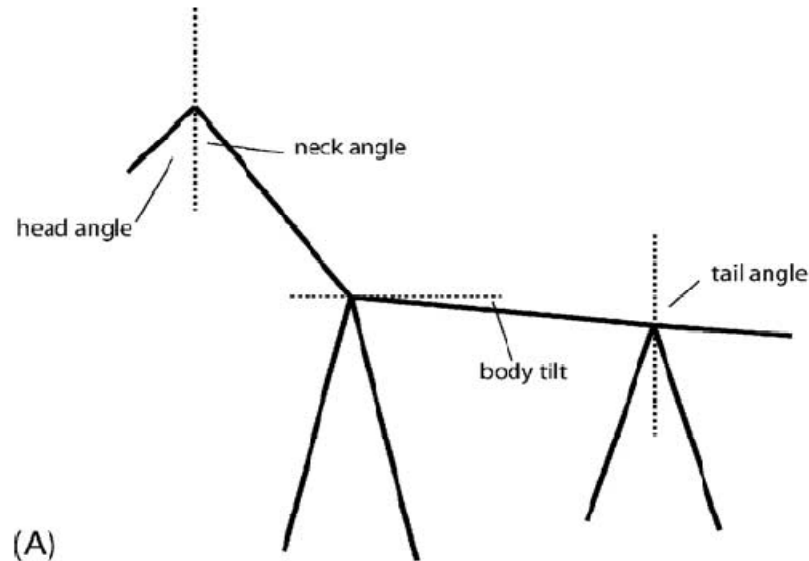


Trial 3



Sampling from natural categories

Examined distributions for four natural categories:
giraffes, horses, cats, and dogs

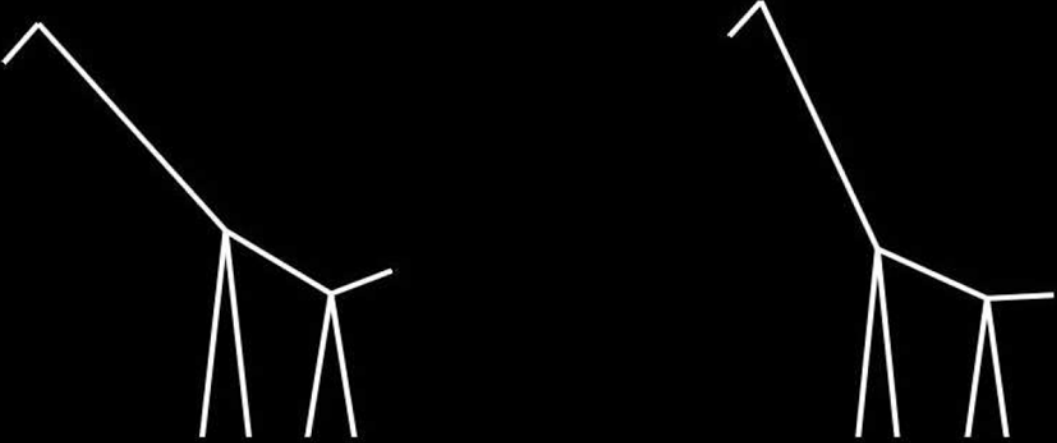


Presented stimuli with nine-parameter stick figures

(Olman & Kersten, 2004)

Choice task

Which animal is a giraffe?



Button 1

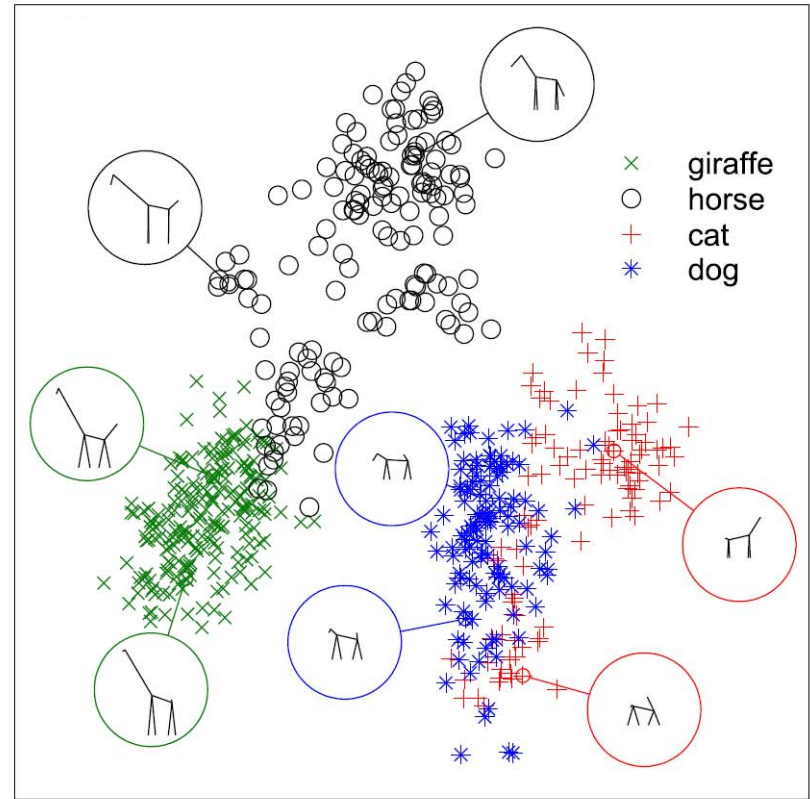
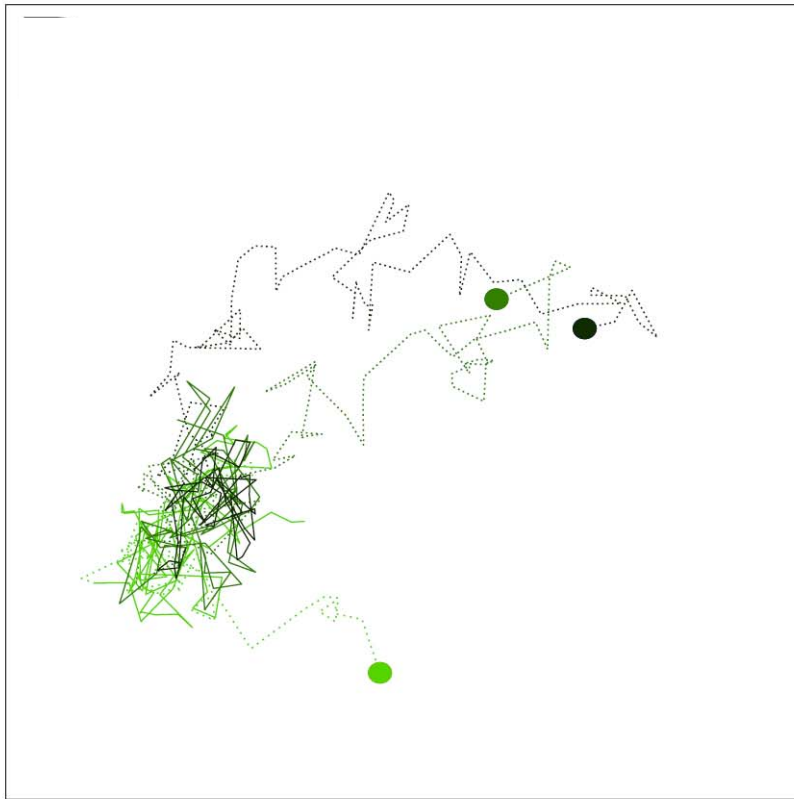
Button 2

1 trials remaining.

The image shows a choice task on a black background. At the top, the text 'Which animal is a giraffe?' is displayed. Below this, two identical line drawings of a giraffe are shown. Each drawing consists of a long neck, a head with a small horn, and four legs. Below the first drawing is the text 'Button 1', and below the second drawing is the text 'Button 2'. At the bottom of the screen, the text '1 trials remaining.' is displayed.

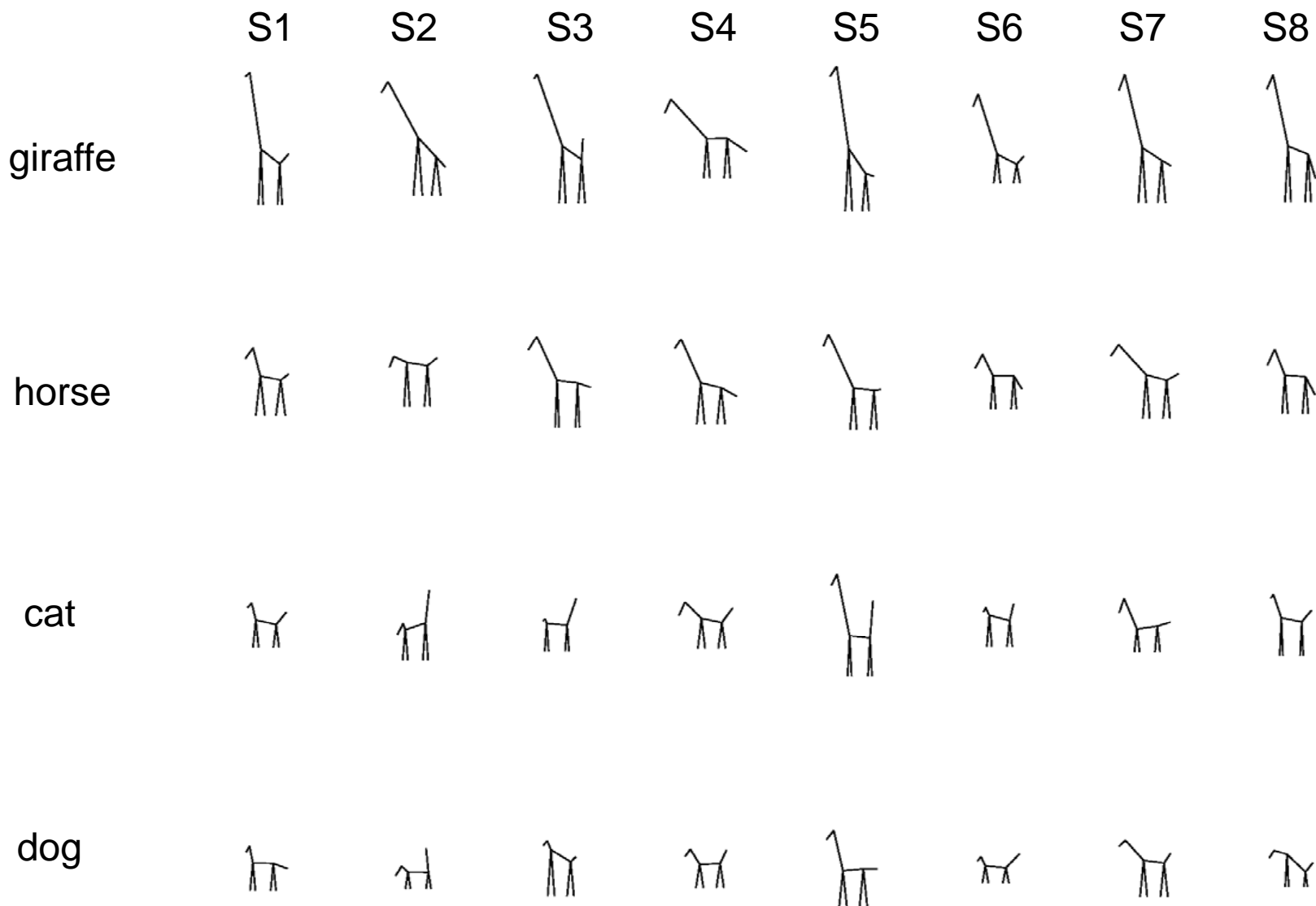
Samples from Subject 3

(projected onto plane from LDA)

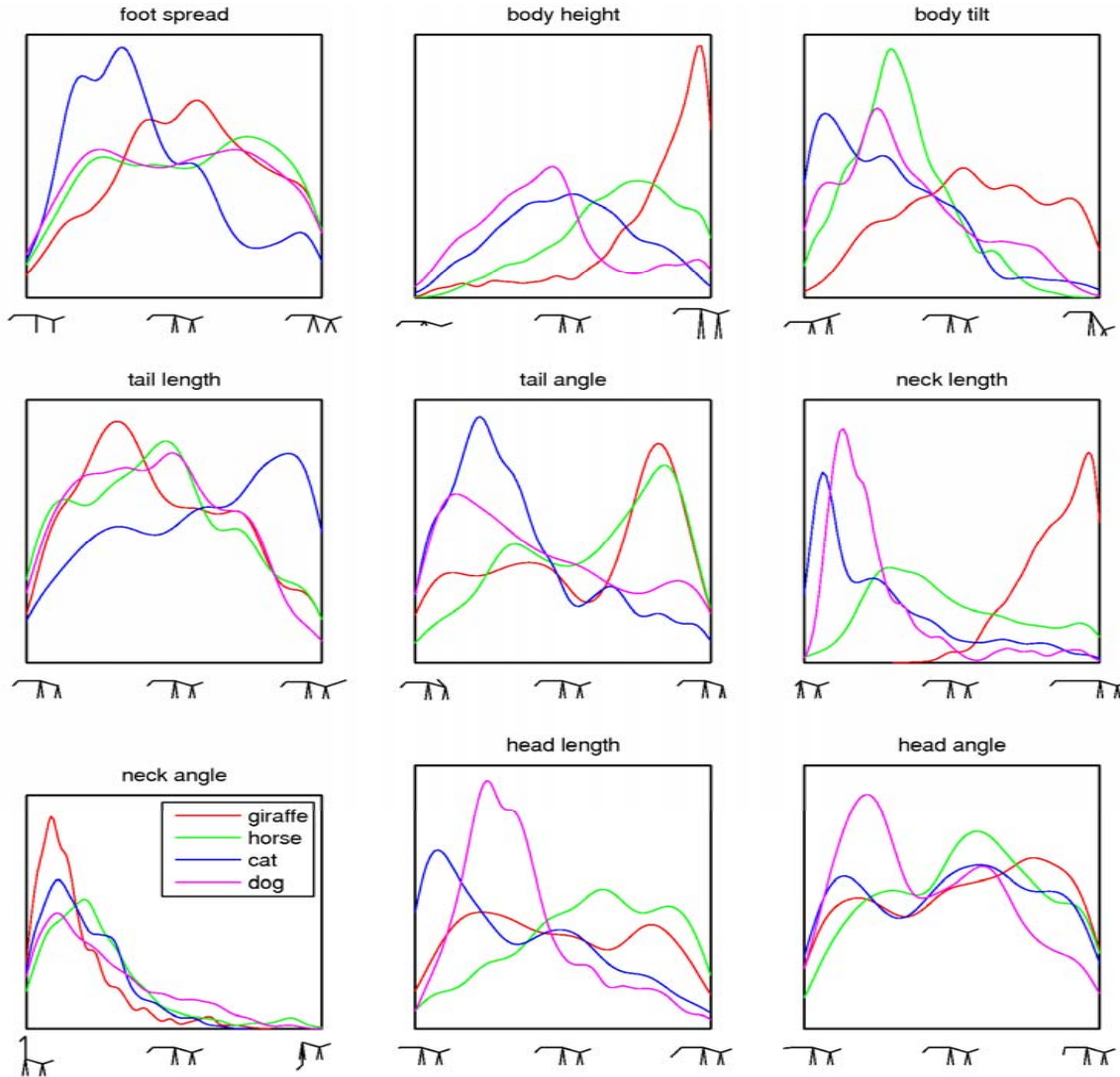


(Sanborn & Griffiths, 2008)

Mean animals by subject



Marginal densities (aggregated across subjects)



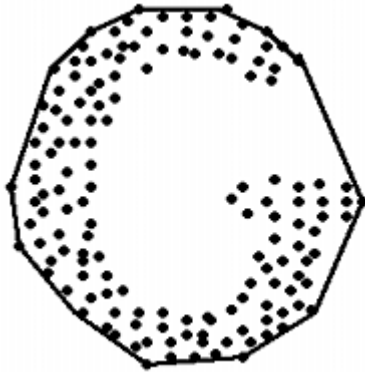
Giraffes are distinguished by neck length, body height and body tilt

Horses are like giraffes, but with shorter bodies and nearly uniform necks

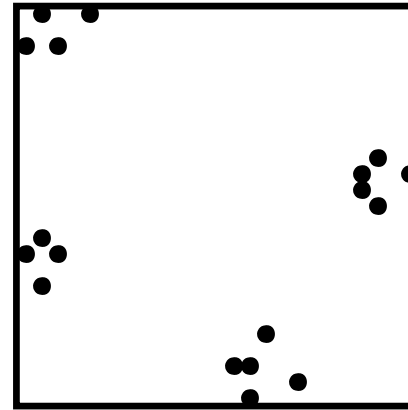
Cats have longer tails than dogs

Relative volume of categories

Convex Hull



Minimum Enclosing Hypercube

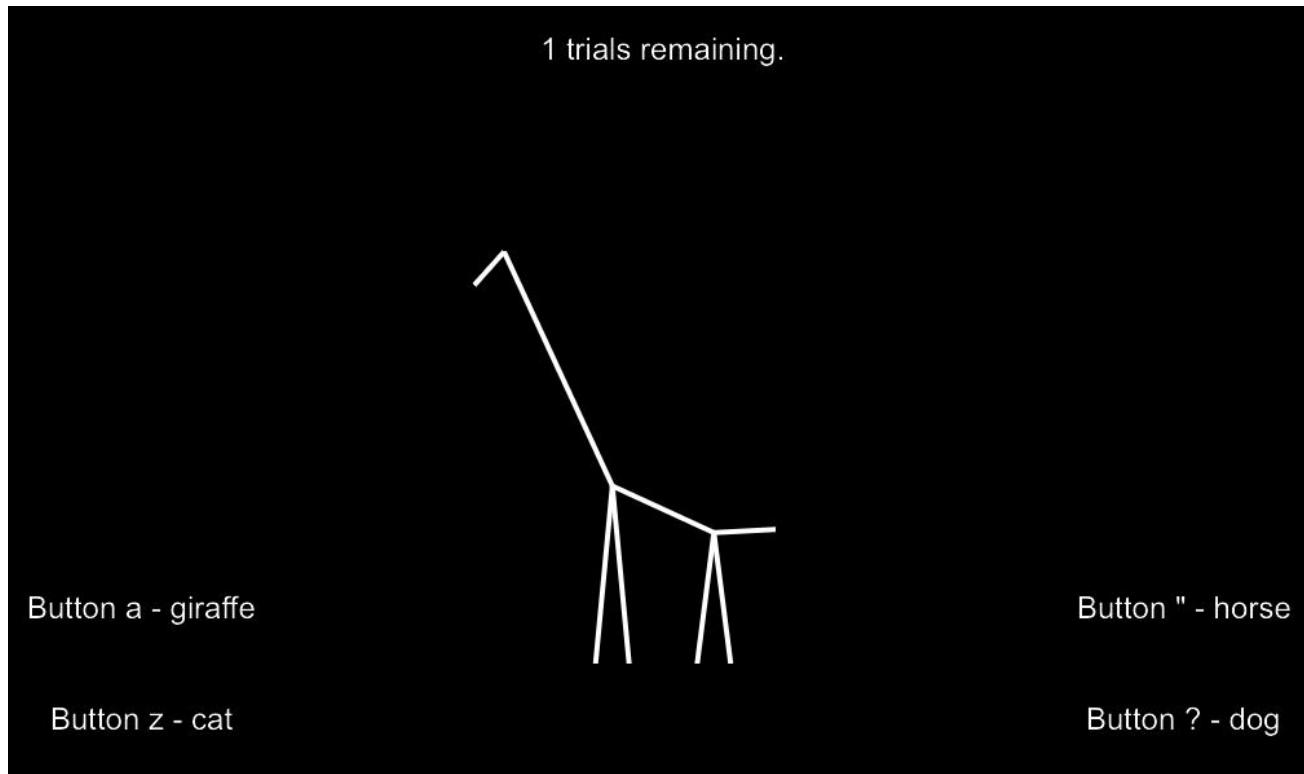


Convex hull content divided by enclosing
hypercube content

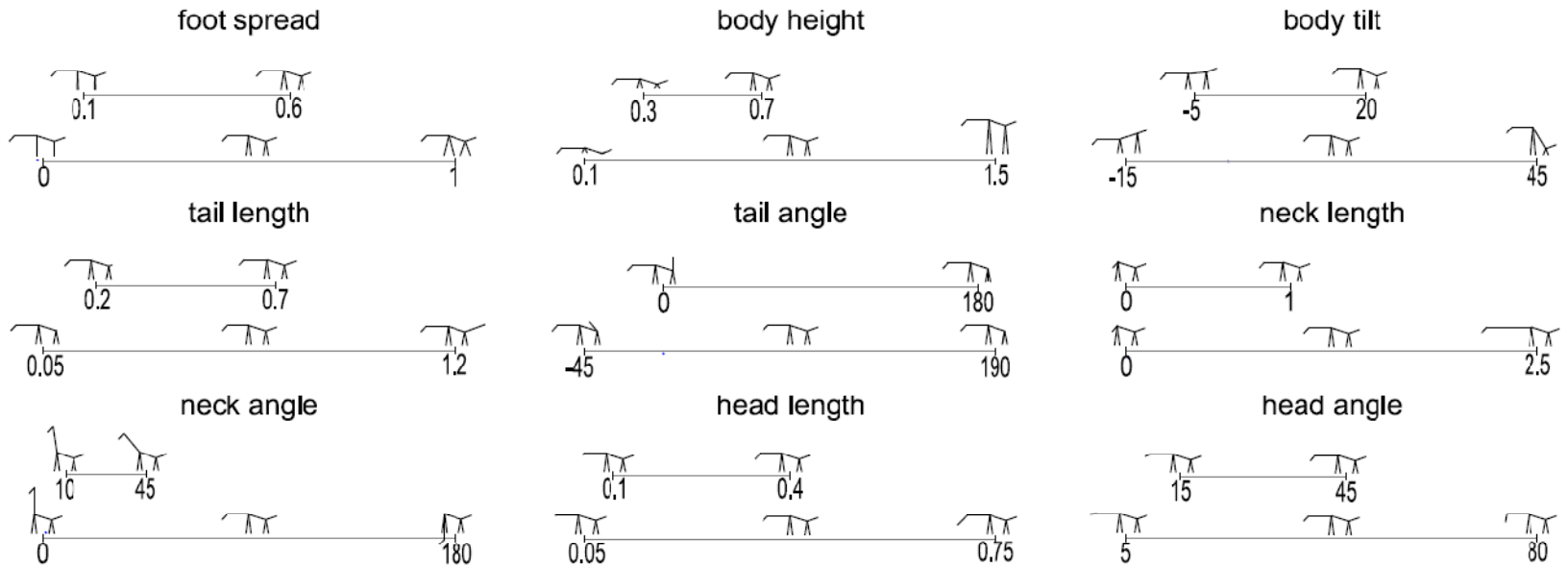
| Giraffe | Horse | Cat | Dog |
|---------|---------|---------|---------|
| 0.00004 | 0.00006 | 0.00003 | 0.00002 |

Discrimination method

(Olman & Kersten, 2004)











Parameter space for discrimination



Restricted so that most random draws were animal-like

MCMC and discrimination means

| | giraffe | horse | cat | dog |
|----------------|---|---|---|---|
| MCMC |  |  |  |  |
| Discrimination |  |  |  |  |

MCMC and the mind

- Markov chain Monte Carlo provides a way to sample from subjective probability distributions
- Many interesting questions can be framed in terms of subjective probability distributions
 - inductive biases (priors)
 - mental representations (category distributions)
- Other MCMC methods may provide further empirical methods...
 - Gibbs for categories, adaptive MCMC, ...

Papers:

<http://cocosci.berkeley.edu>

Questions:

tom_griffiths@berkeley.edu

