Part IV: Monte Carlo and nonparametric Bayes

Outline

Monte Carlo methods

Nonparametric Bayesian models

Outline

Monte Carlo methods

Nonparametric Bayesian models

The Monte Carlo principle

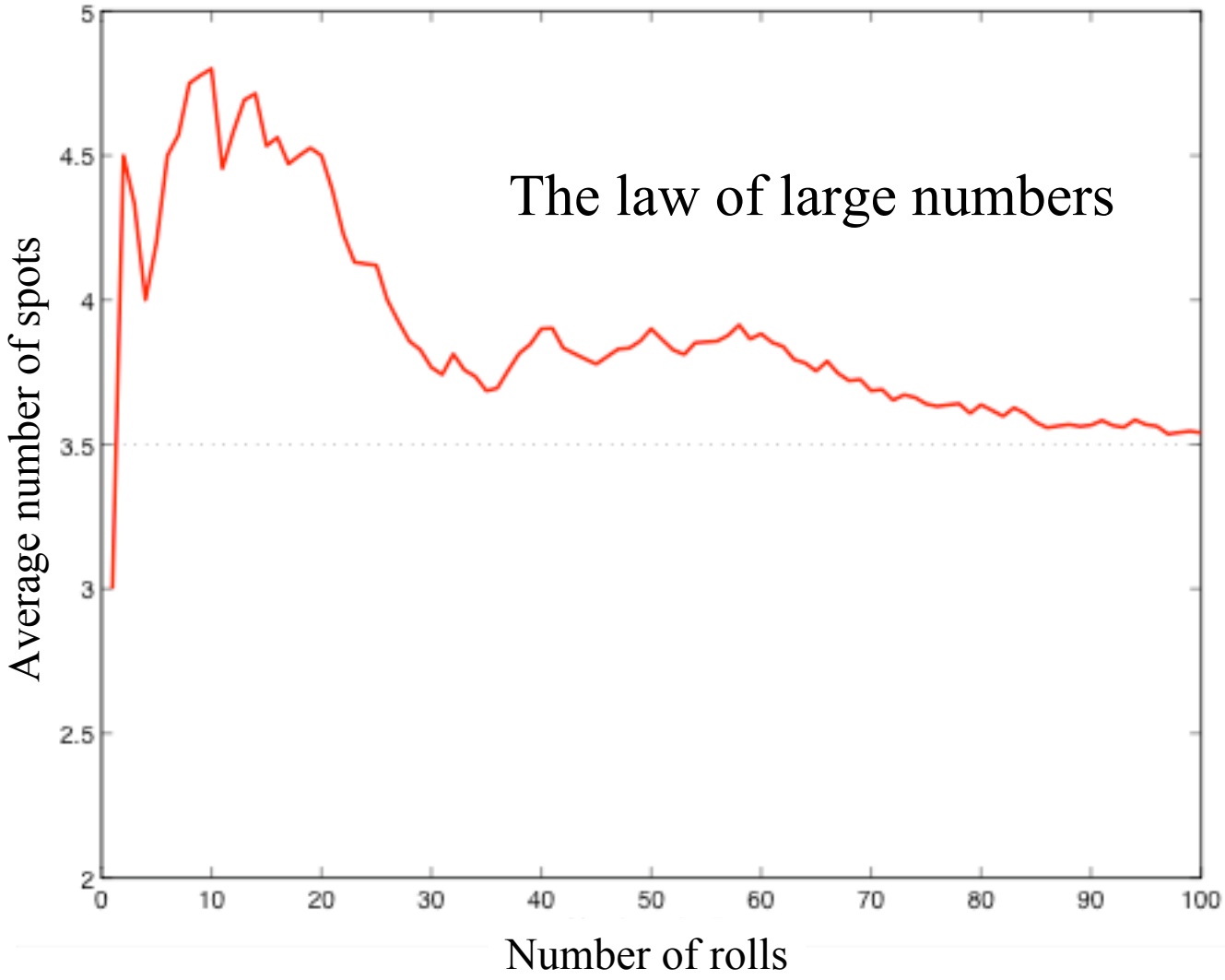
- The expectation of f with respect to P can be approximated by

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

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

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

The Monte Carlo principle



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

Making Bayesian inference easier

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

Evaluating the posterior probability of a hypothesis requires considering all hypotheses

Modern Monte Carlo methods let us avoid this

Modern Monte Carlo methods

- Sampling schemes for distributions with large state spaces known up to a multiplicative constant

- Two approaches:
  - importance sampling (and particle filters)
  - Markov chain Monte Carlo

Importance sampling

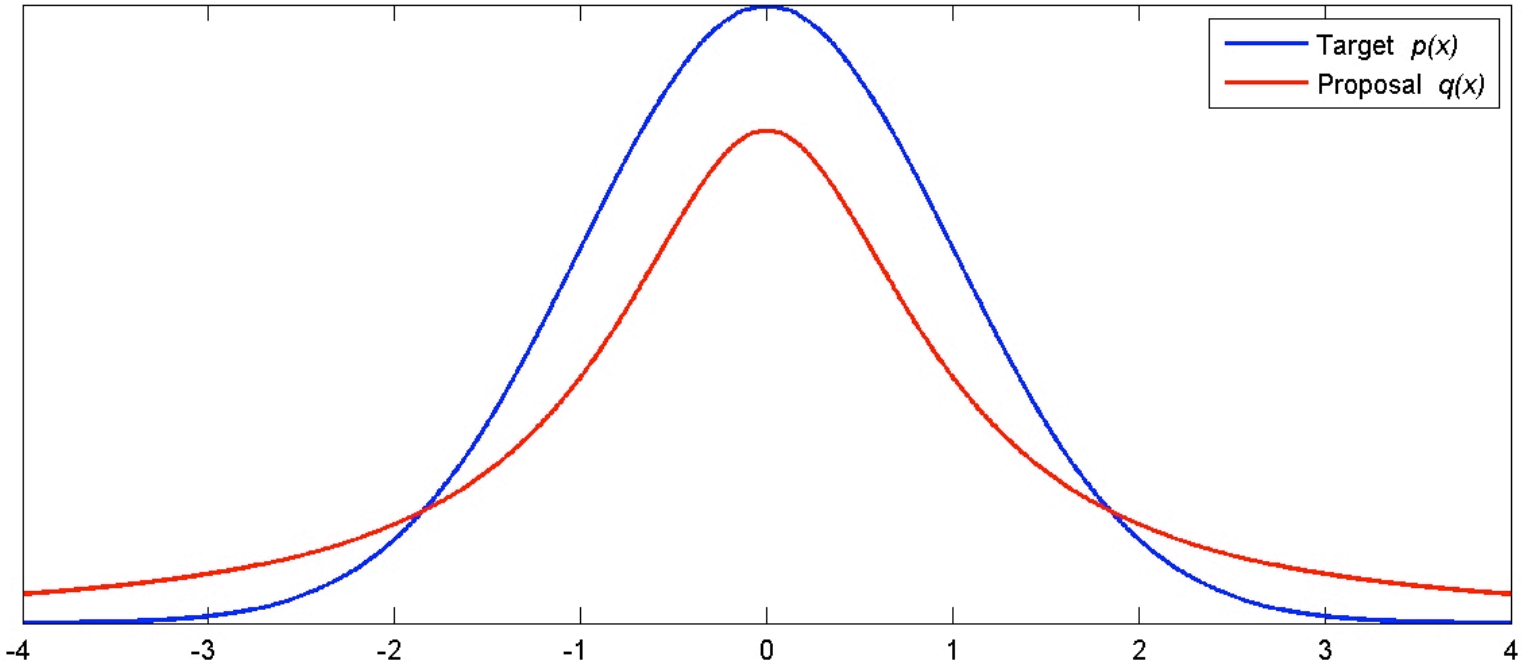
Basic idea: generate from the wrong distribution, assign weights to samples to correct for this

$$E_{p(x)}\left[f(x)\right] = \int f(x)\,p(x)\,dx$$

$$= \int f(x)\,\frac{p(x)}{q(x)}\,q(x)\,dx$$

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

Importance sampling



works when sampling from proposal is easy, target is hard

An alternative scheme…

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

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

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

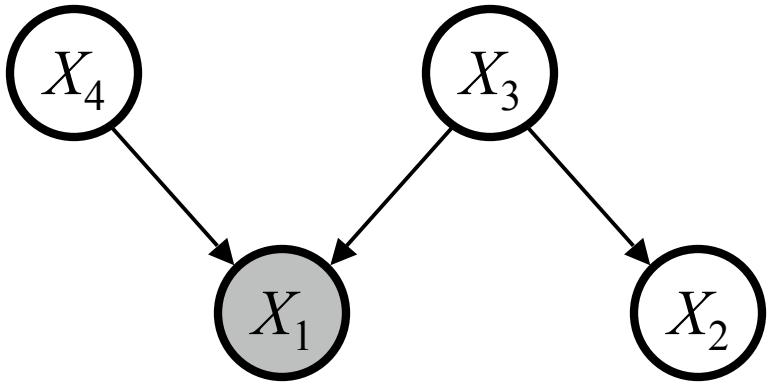
Likelihood weighting

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

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

Likelihood weighting

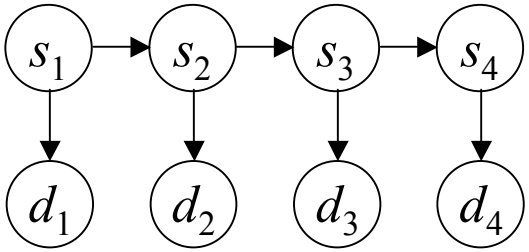
- Generate samples of all variables except observed variables
- Assign weights proportional to probability of observed data given values in sample



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
  - particularly for, e.g., tail probabilities
- Also provides a solution to the question of how people can update beliefs as data come in…

Particle filtering



We want to generate samples from $P(s_4|d_1, \ldots, d_4)$

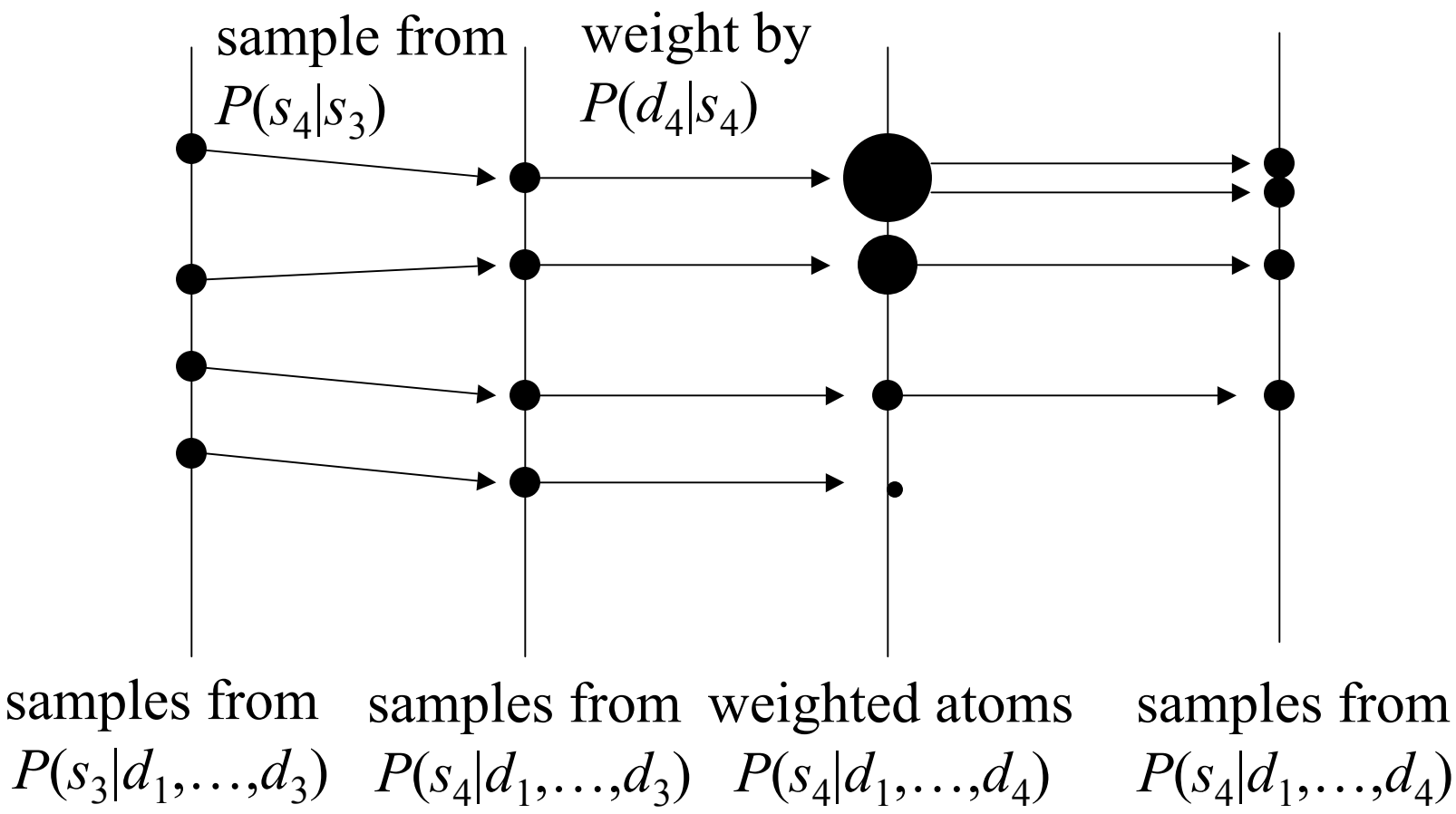
$$P(s_4 \mid d_1,...,d_4) \propto P(d_4 \mid s_4)P(s_4 \mid d_1,...,d_3)$$

$$= P(d_4 \mid s_4)\sum_{s_3} P(s_4 \mid s_3)P(s_3 \mid d_1,...,d_3)$$

We can use likelihood weighting if we can sample from $P(s_4|s_3)$ and $P(s_3|d_1, \ldots, d_3)$

Particle filtering

$$P(s_4 \mid d_1,...,d_4) \propto P(d_4 \mid s_4) \sum_{s_3} P(s_4 \mid s_3) P(s_3 \mid d_1,...,d_3)$$



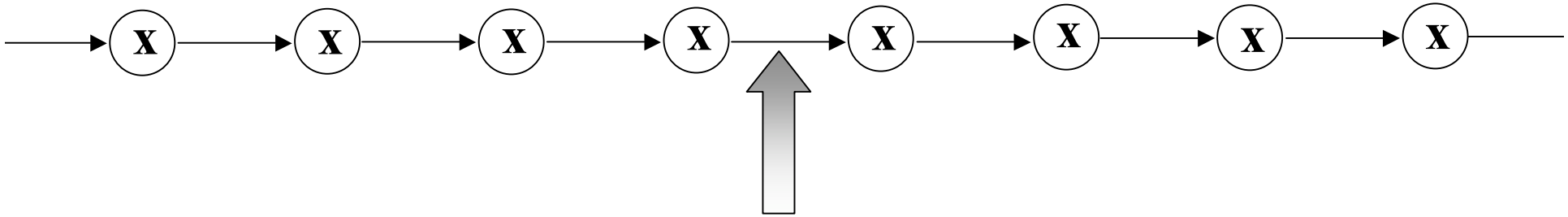
The promise of particle filters

- People need to be able to update probability distributions over large hypothesis spaces as more data become available
- Particle filters provide a way to do this with limited computing resources…
  - maintain a fixed finite number of samples
- Not just for dynamic models
  - can work with a fixed set of hypotheses, although this requires some further tricks for maintaining diversity

Markov chain Monte Carlo

- Basic idea: construct a *Markov chain* that will converge to the target distribution, and draw samples from that chain
- Just uses something proportional to the target distribution (good for Bayesian inference!)
- Can work in state spaces of arbitrary (including unbounded) size (good for nonparametric Bayes)

Markov chains



Variables $\mathbf{x}^{(t+1)}$ independent of all previous variables given immediate predecessor $\mathbf{x}^{(t)}$

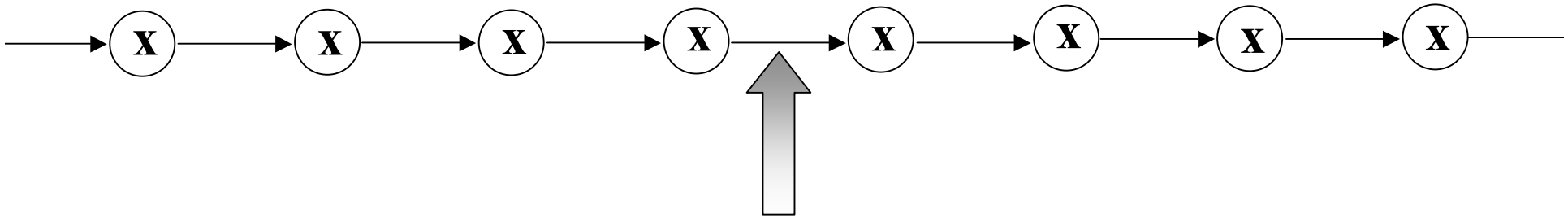
An example: card shuffling

- Each state $\mathbf{x}^{(t)}$ is a permutation of a deck of cards (there are 52! permutations)
- Transition matrix **T** indicates how likely one permutation will become another
- The transition probabilities are determined by the shuffling procedure
  - riffle shuffle
  - overhand
  - one card

Convergence of Markov chains

- Why do we shuffle cards?
- Convergence to a uniform distribution takes only 7 riffle shuffles…
- Other Markov chains will also converge to a *stationary distribution*, if certain simple conditions are satisfied (called "ergodicity")
  - e.g. every state can be reached in some number of steps from every other state

Markov chain Monte Carlo



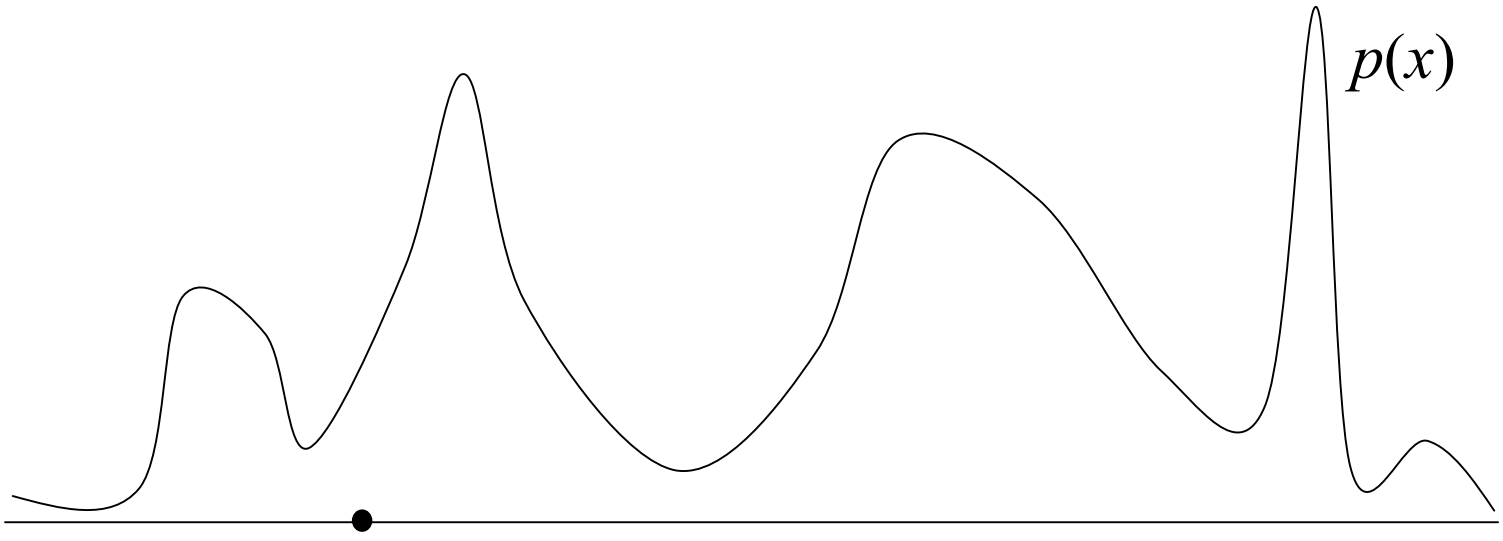
- States of chain are variables of interest
- Transition matrix chosen to give target distribution as stationary distribution

Metropolis-Hastings algorithm

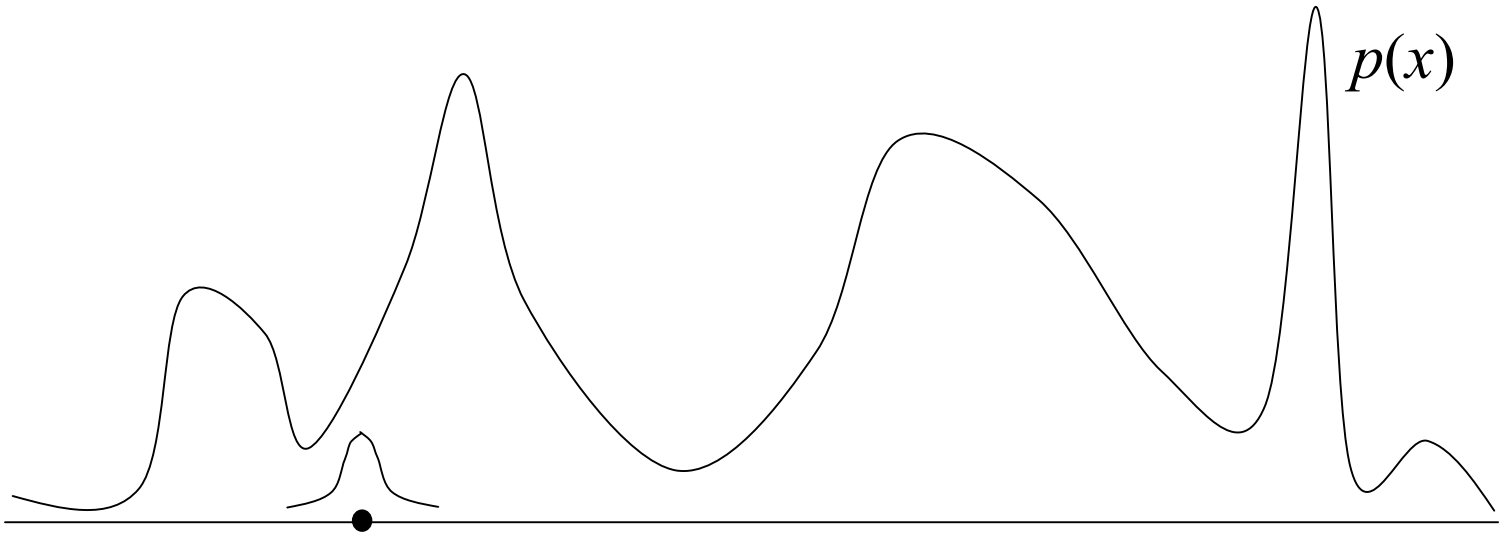
- Transitions have two parts:
  - proposal distribution: $Q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})$
  - acceptance: take proposals with probability

$$A(\mathbf{x}^{(t)},\mathbf{x}^{(t+1)}) = \min\left(1, \frac{P(\mathbf{x}^{(t+1)})\, Q(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)})}{P(\mathbf{x}^{(t)})\, Q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})}\right)$$

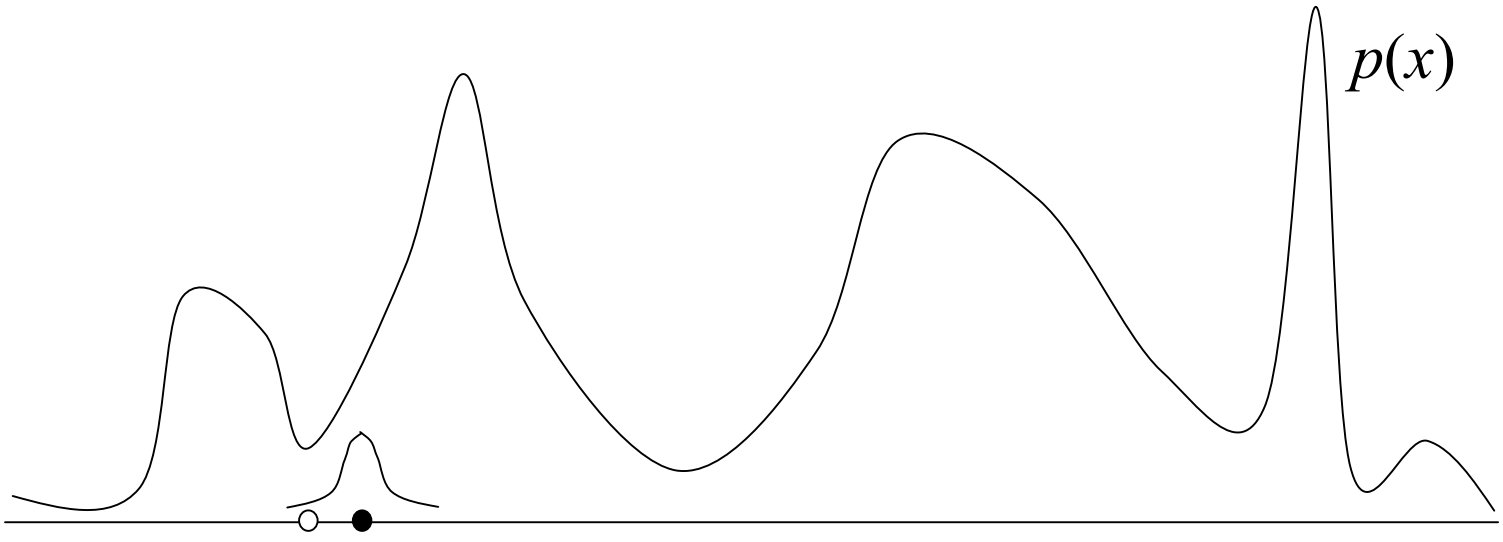
Metropolis-Hastings algorithm



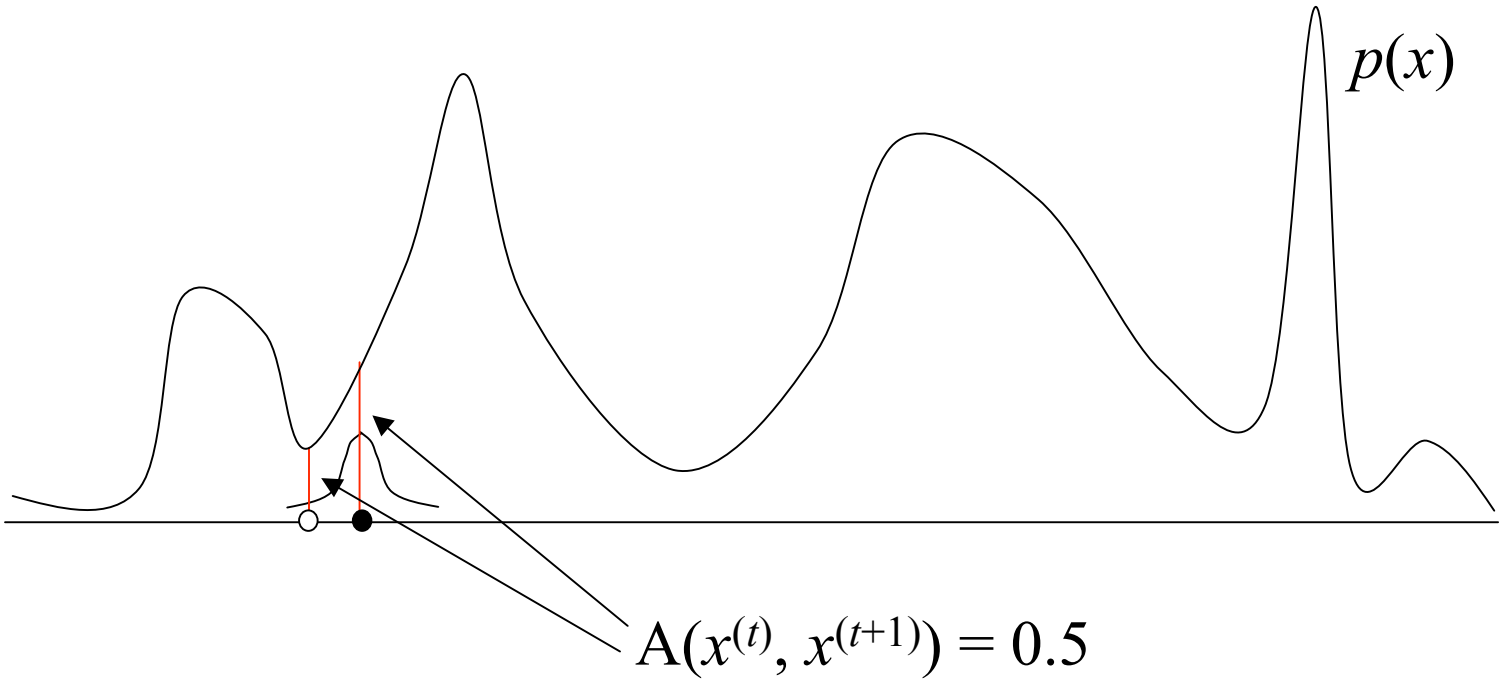
Metropolis-Hastings algorithm



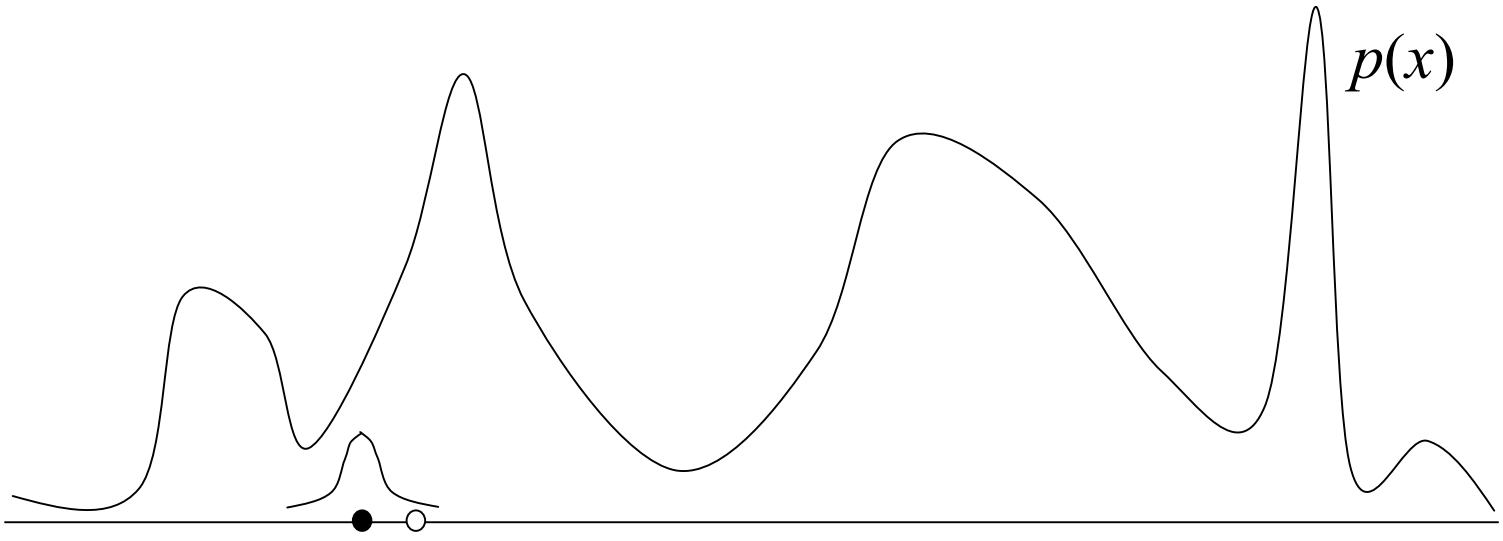
Metropolis-Hastings algorithm



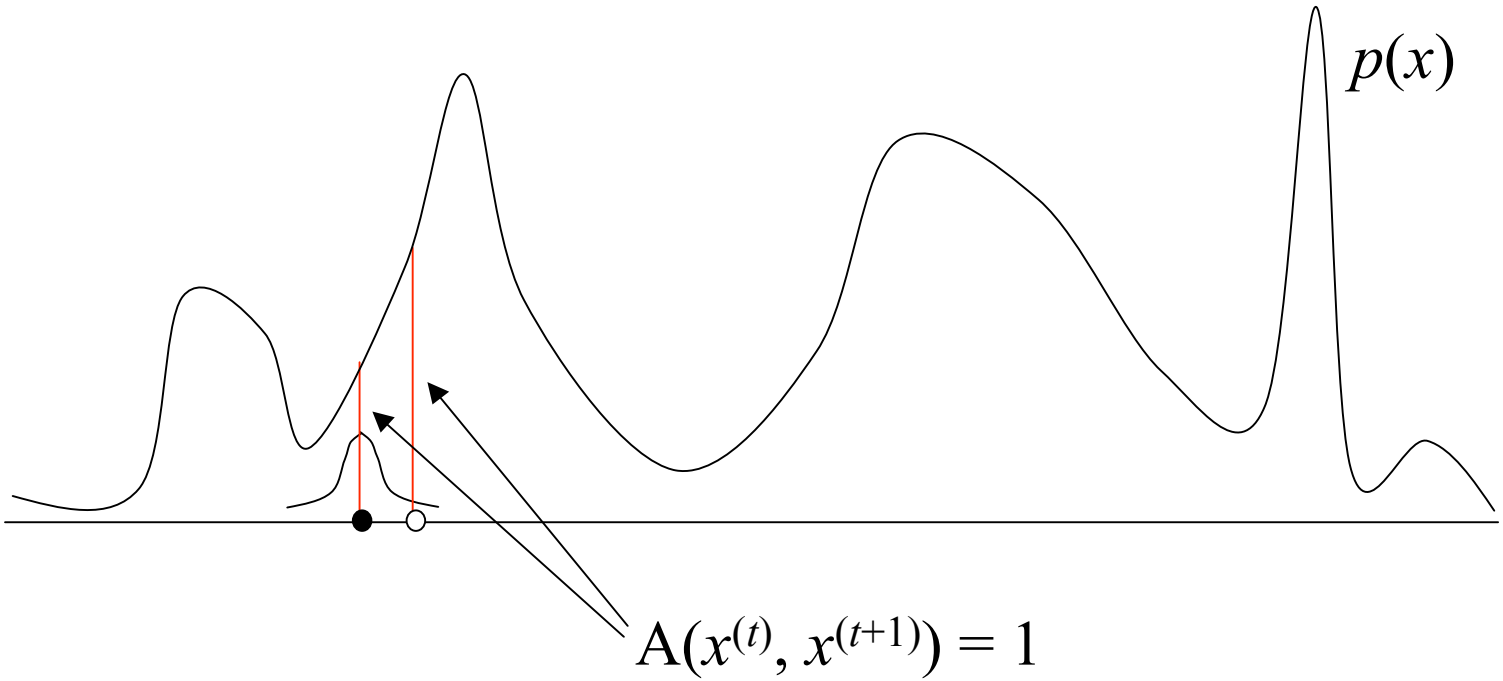
Metropolis-Hastings algorithm



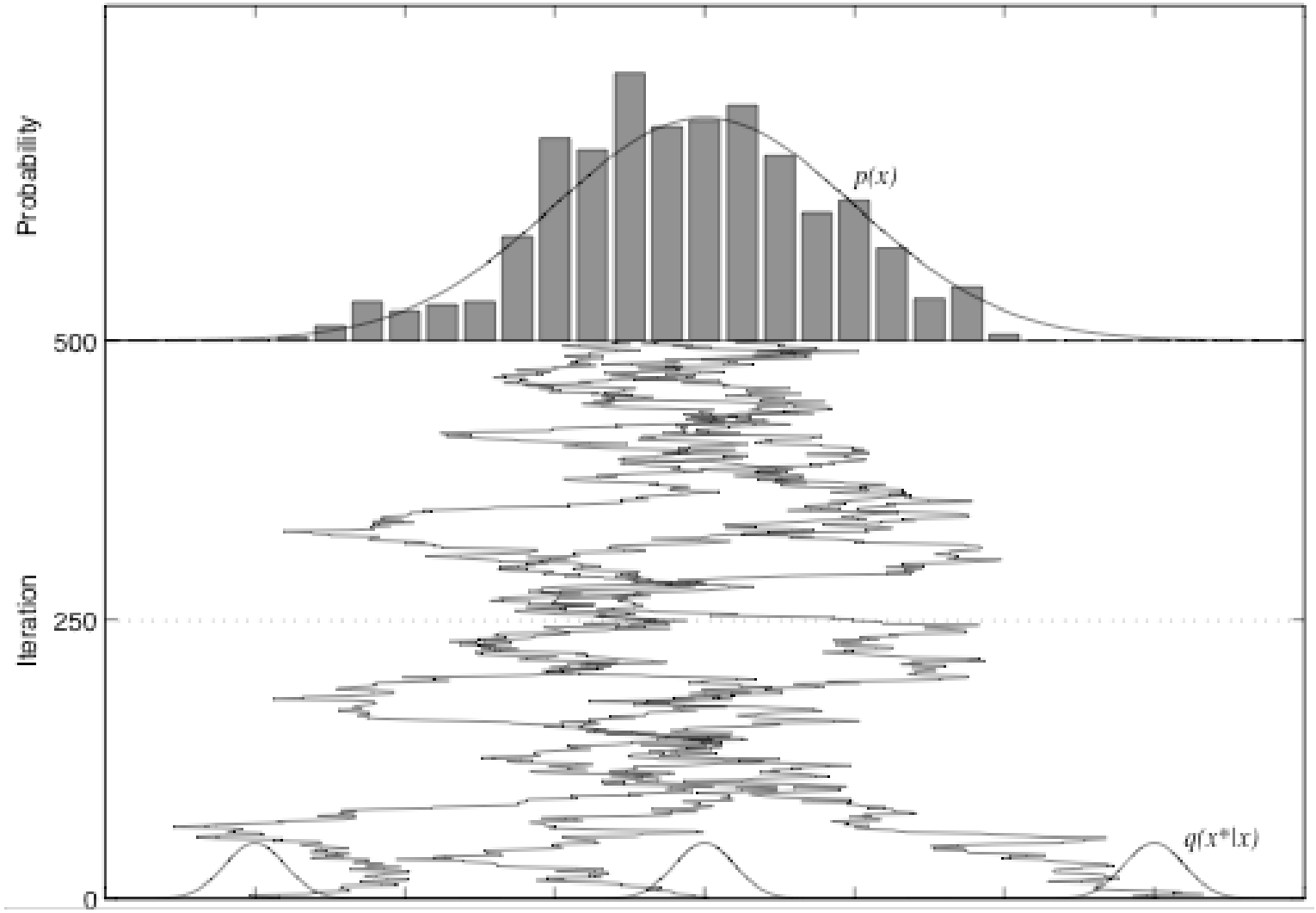
Metropolis-Hastings algorithm



Metropolis-Hastings algorithm



Metropolis-Hastings in a slide



Gibbs sampling

Particular choice of proposal distribution

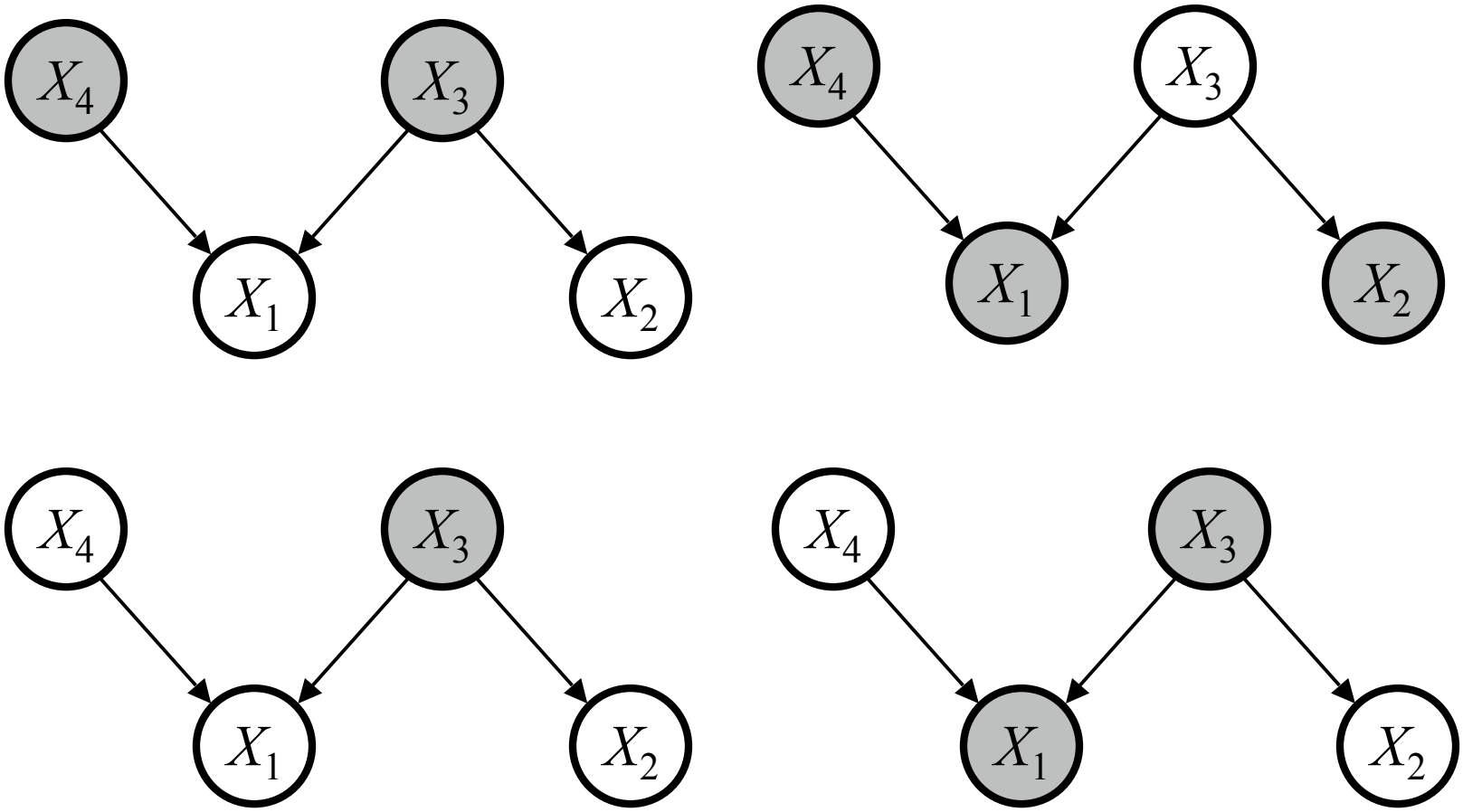
For variables $\mathbf{x} = x_1, x_2, \ldots, x_n$

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

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

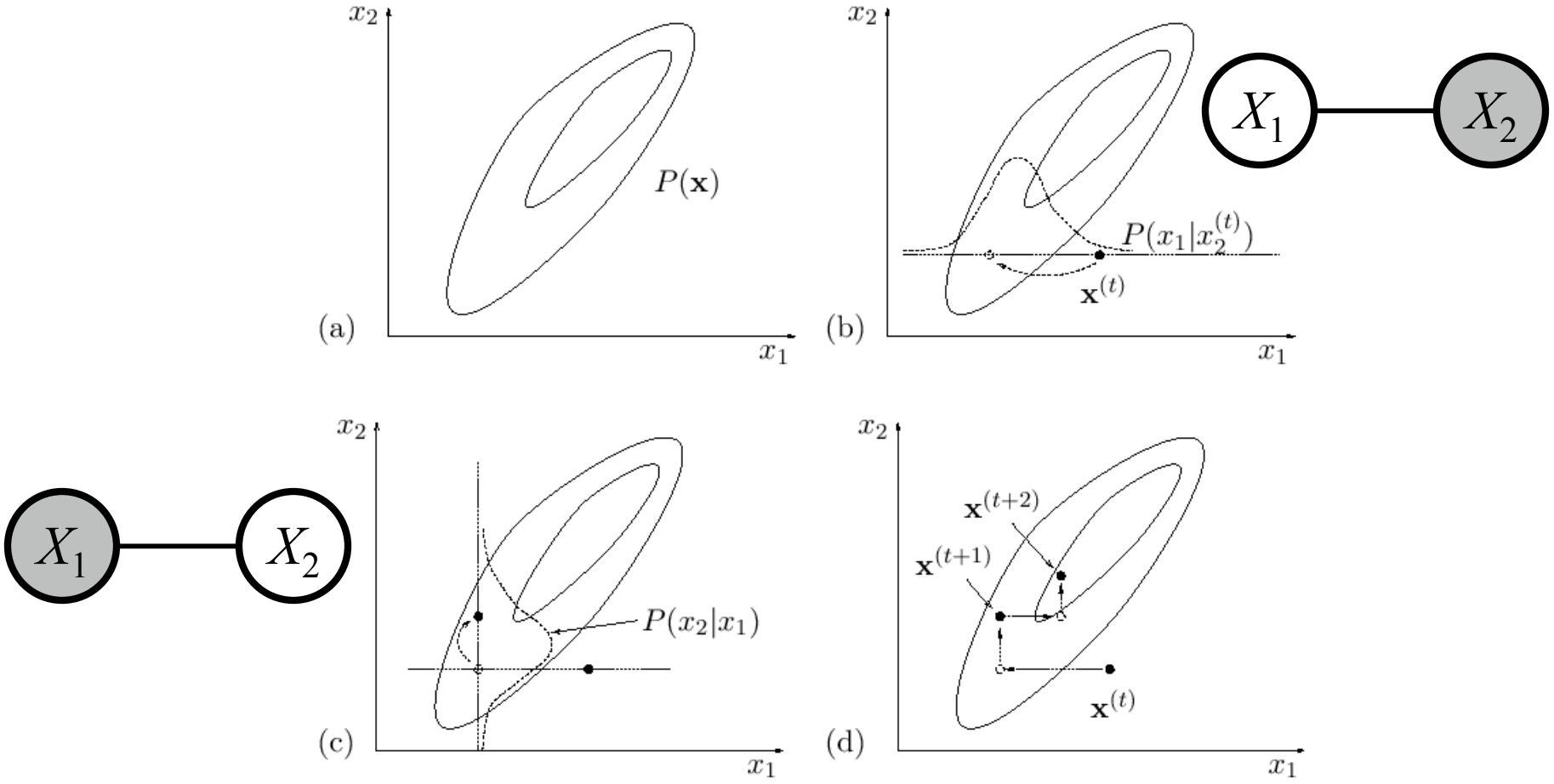
(this is called the *full conditional* distribution)

In a graphical model…



Sample each variable conditioned on its *Markov blanket*

Gibbs sampling



(MacKay, 2002)

The magic of MCMC

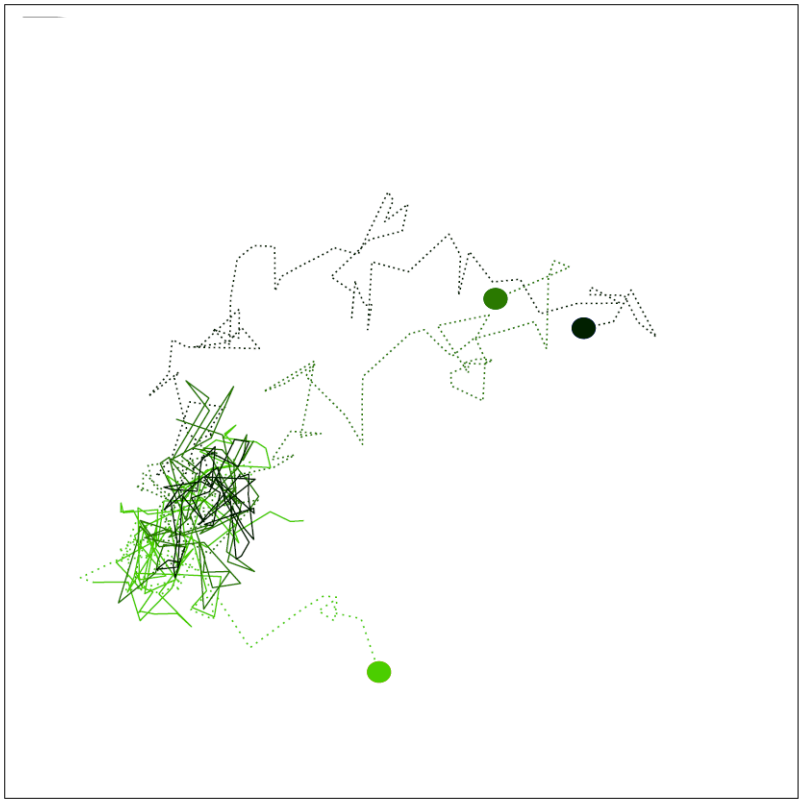
- Since we only ever need to evaluate the relative probabilities of two states, we can have huge state spaces (much of which we rarely reach)
- In fact, our state spaces can be *infinite*
  - common with nonparametric Bayesian models
- But… the guarantees it provides are asymptotic
  - making algorithms that converge in practical amounts of time is a significant challenge

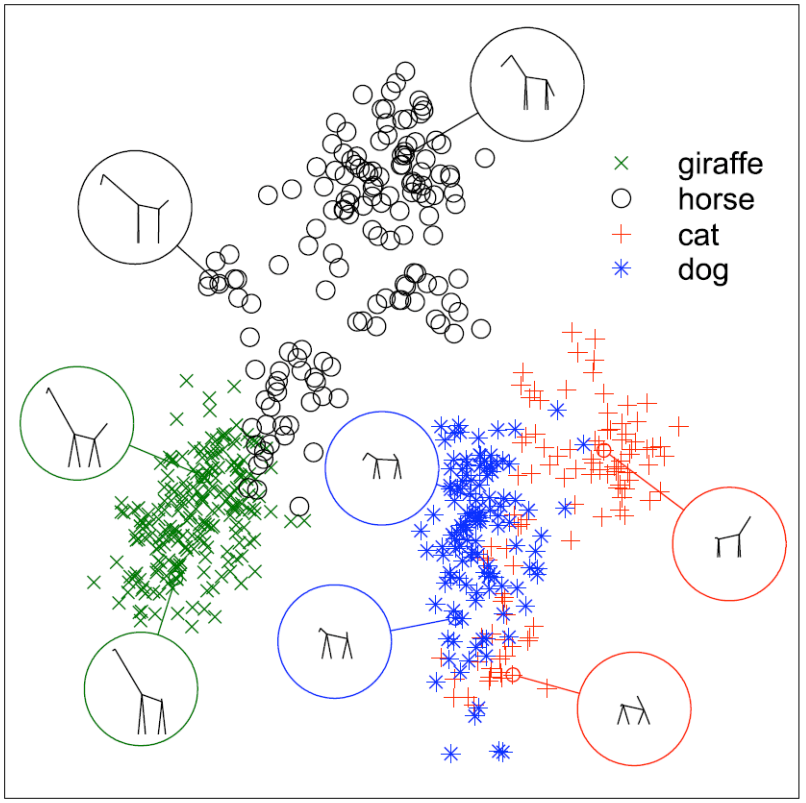
MCMC and cognitive science

- The main use of MCMC is for probabilistic inference in complex models
- The Metropolis-Hastings algorithm seems like a good metaphor for aspects of development…
- A form of cultural evolution can be shown to be equivalent to Gibbs sampling (Griffiths & Kalish, 2007)
- We can also use MCMC algorithms as the basis for experiments with people…

Samples from Subject 3

(projected onto plane from LDA)





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

Outline

Monte Carlo methods

Nonparametric Bayesian models

Nonparametric Bayes

- Nonparametric models…
  - can capture distributions outside parametric families
  - have infinitely many parameters
  - grow in complexity with the data
- Provide a way to automatically determine how much structure can be inferred from data
  - how many clusters?
  - how many dimensions?

How many clusters?

Nonparametric approach:
Dirichlet process mixture models

Mixture models

- Each observation is assumed to come from a single (possibly previously unseen) cluster
- The probability that the *i*th sample belongs to the *k*th cluster is

$$p(z_i = k \mid x_i) \propto p(x_i \mid z_i = k)\, p(z_i = k)$$

- Where $p(x_i|z_i=k)$ reflects the structure of cluster k (e.g. Gaussian) and $p(z_i=k)$ is its prior probability

Dirichlet process mixture models

- Use a prior that allows infinitely many clusters (but finitely many for finite observations)
- The *i*th sample is drawn from the *k*th cluster with probability

$$P(k) = \begin{cases} \frac{n_k}{i+\alpha}, & n_k > 0 \text{ (i.e., } k \text{ is old)} \\ \frac{\alpha}{i+\alpha}, & n_k = 0 \text{ (i.e., } k \text{ is new)} \end{cases}$$

where α is a parameter of the model

(known as the "Chinese restaurant process")

Nonparametric Bayes and cognition

- Nonparametic Bayesian models are useful for answering questions about how much structure people should infer from data
- Many cognitive science questions take this form
  - how should we represent categories?
  - what features should we identify for objects?

Nonparametric Bayes and cognition

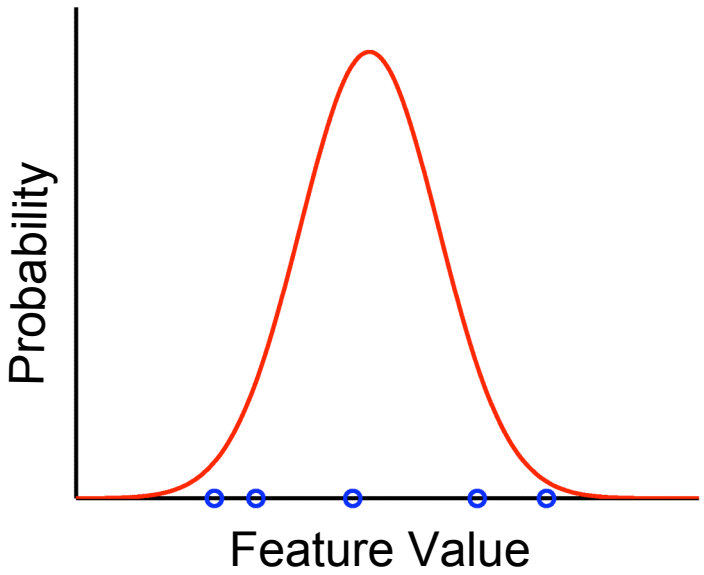
- Nonparametic Bayesian models are useful for answering questions about how much structure people should infer from data
- Many cognitive science questions take this form
  - how should we represent categories?
  - what features should we identify for objects?

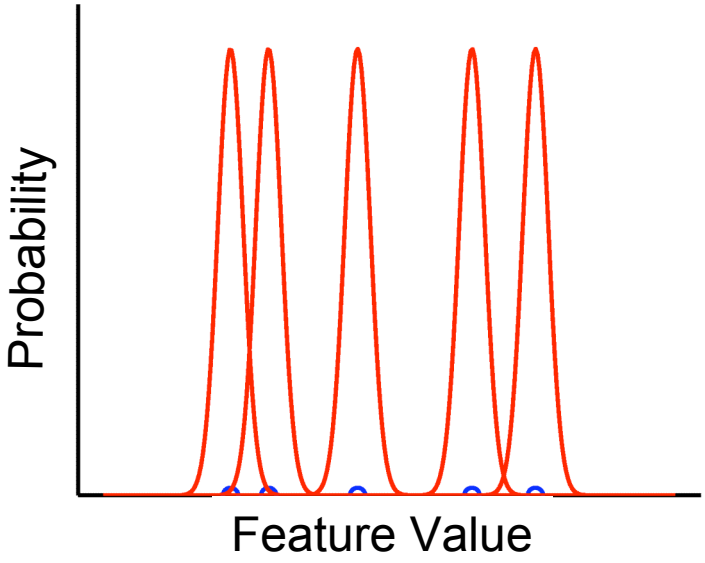
The Rational Model of Categorization
(RMC; Anderson 1990; 1991)

- Computational problem: predicting a feature based on observed data
  - assume that category labels are just features
- Predictions are made on the assumption that objects form clusters with similar properties
  - each object belongs to a single cluster
  - feature values likely to be the same within clusters
  - the number of clusters is unbounded

Representation in the RMC

Flexible representation can interpolate between prototype and exemplar models





The "optimal solution"

The probability of the missing feature (i.e., the category label) taking a certain value is

$$P(j|F_n) = \sum_{x_n} P(j|x_n, F_n) P(x_n|F_n)$$

posterior over partitions

where j is a feature value, F_n are the observed features of a set of n objects, and x_n is a partition of objects into clusters

The prior over partitions

- An object is assumed to have a constant probability of joining same cluster as another object, known as the *coupling probability*
- This allows some probability that a stimulus forms a new cluster, so the probability that the *i*th object is assigned to the *k*th cluster is

$$P(k) = \begin{cases} \frac{cn_k}{(1-c)+ci} & n_k > 0 \text{ (i.e., } k \text{ is old)} \\ \frac{(1-c)}{(1-c)+ci} & n_k = 0 \text{ (i.e., } k \text{ is new)} \end{cases}$$

Equivalence

Neal (1998) showed that the prior for the RMC and the DPMM are the same, with

$$\alpha = (1 - c)/c$$

RMC prior:
$$P(k) = \begin{cases} \frac{cn_k}{(1-c)+ci} & n_k > 0 \text{ (i.e., } k \text{ is old)} \\ \frac{(1-c)}{(1-c)+ci} & n_k = 0 \text{ (i.e., } k \text{ is new)} \end{cases}$$

DPMM prior:
$$P(k) = \begin{cases} \frac{n_k}{i+\alpha}, & n_k > 0 \text{ (i.e., } k \text{ is old)} \\ \frac{\alpha}{i+\alpha}, & n_k = 0 \text{ (i.e., } k \text{ is new)} \end{cases}$$

The computational challenge

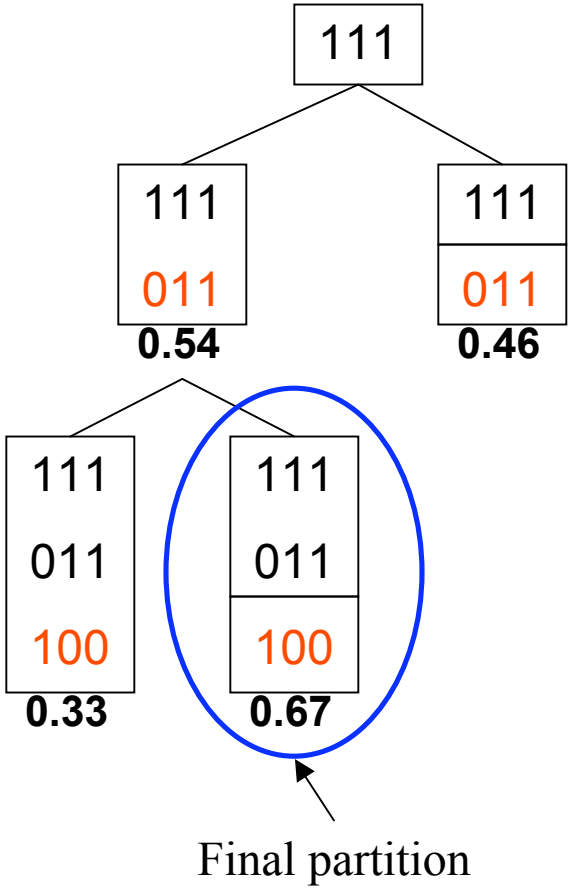
The probability of the missing feature (i.e., the category label) taking a certain value is

$$P(j|F_n) = \sum_{x_n} P(j|x_n, F_n)P(x_n|F_n)$$

where j is a feature value, F_n are the observed features of a set of n objects, and x_n is a partition of objects into groups

n	1	2	3	4	5	6	7	8	9	10
$\|x_n\|$	1	2	5	15	52	203	877	4140	21147	115975

Anderson's approximation



- Data observed sequentially
- Each object is deterministically assigned to the cluster with the highest posterior probability
- Call this the "Local MAP"
  - choosing the cluster with the maximum *a posteriori* probability

Two uses of Monte Carlo methods

1. For solving problems of probabilistic inference involved in developing computational models

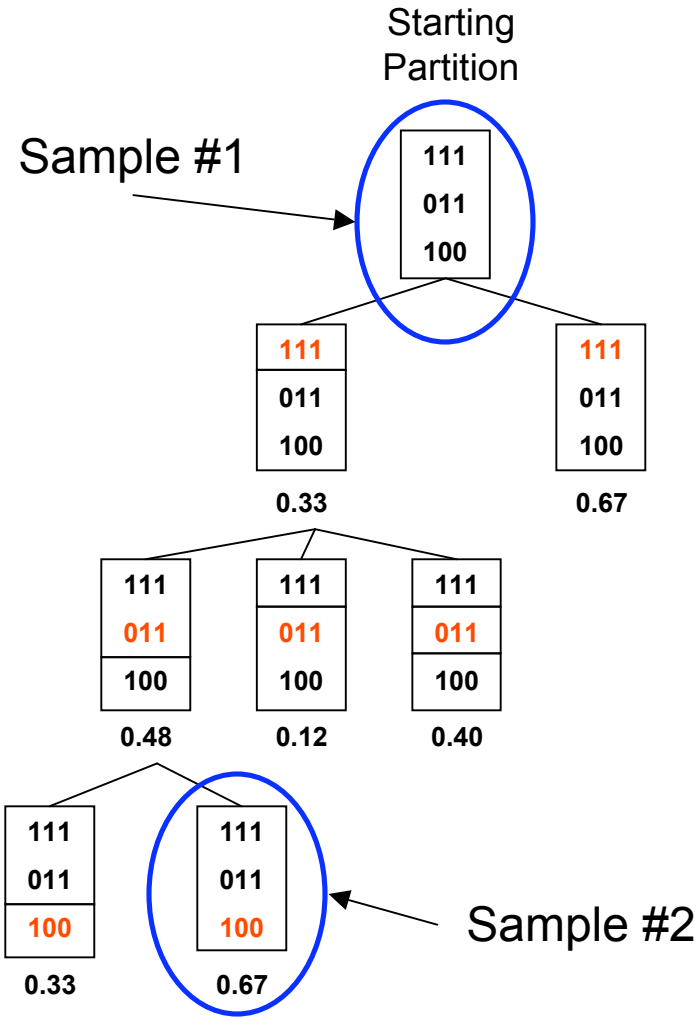
2. As a source of hypotheses about how the mind might solve problems of probabilistic inference

Alternative approximation schemes

- There are several methods for making approximations to the posterior in DPMMs
  - Gibbs sampling
  - Particle filtering
- These methods provide asymptotic performance guarantees (in contrast to Anderson's procedure)

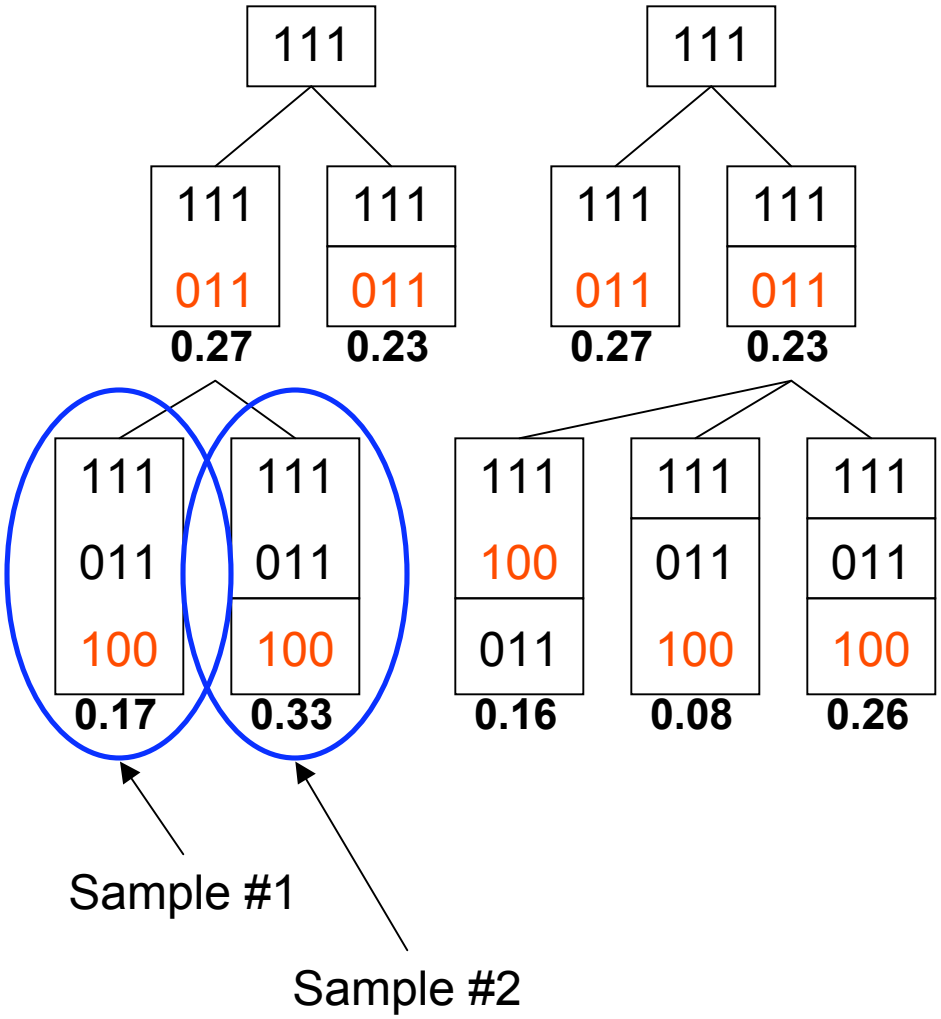
(Sanborn, Griffiths, & Navarro, 2006)

Gibbs sampling for the DPMM



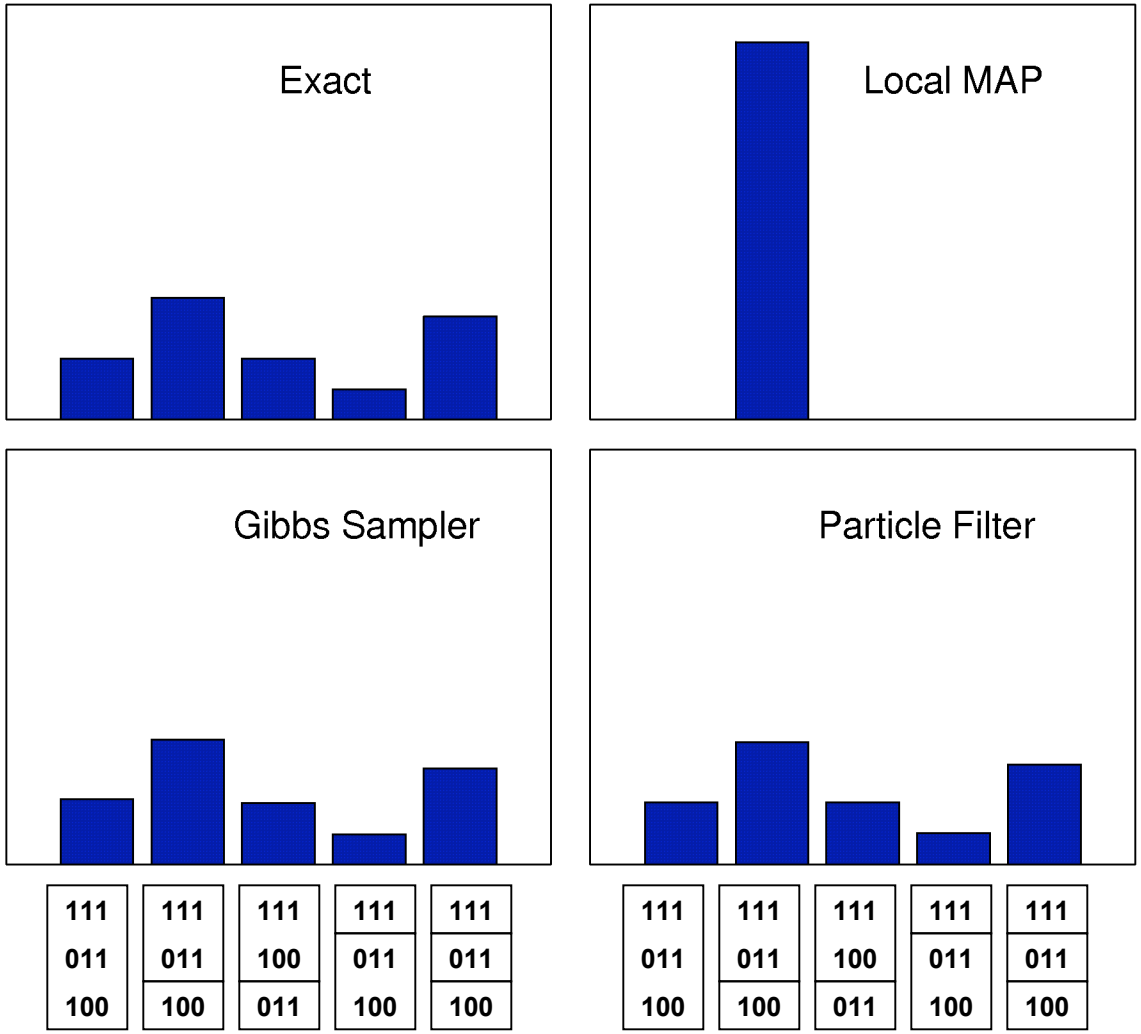
- All the data are required at once (a batch procedure)
- Each stimulus is sequentially assigned to a cluster based on the assignments of all of the remaining stimuli
- Assignments are made probabilistically, using the full conditional distribution

Particle filter for the DPMM



- Data are observed sequentially
- The posterior distribution at each point is approximated by a set of "particles"
- Particles are updated, and a fixed number of are carried over from trial to trial

Approximating the posterior



- For a single order, the Local MAP will produce a single partition
- The Gibbs sampler and particle filter will approximate the exact DPMM distribution

Order effects in human data

- The probabilistic model underlying the DPMM does not produce any order effects
  - follows from exchangeability
- But… human data shows order effects

  (e.g., Medin & Bettger, 1994)
- Anderson and Matessa tested local MAP predictions about order effects in an unsupervised clustering experiment

  (Anderson, 1990)

Anderson and Matessa's Experiment

Front-Anchored Order	End-Anchored Order
scadsporm	snadstirb
scadstirm	snekstirb
sneksporb	scadsporm
snekstirb	sceksporm
sneksporm	sneksporm
snekstirm	snadsporm
scadsporb	scedstirb
scadstirb	scadstirb

- Subjects were shown all sixteen stimuli that had four binary features
- Front-anchored ordered stimuli emphasized the first two features in the first eight trials; end-anchored ordered emphasized the last two

Anderson and Matessa's Experiment

	Proportion that are Divided Along a Front-Anchored Feature				
	Experimental Data	Local MAP	Particle Filter (1)	Particle Filter (100)	Gibbs Sampler
Front-Anchored Order	0.55	1.00	0.59	0.50	0.48
End-Anchored Order	0.30	0.00	0.38	0.50	0.49

A “rational process model”

- A rational model clarifies a problem and serves as a benchmark for performance
- Using a psychologically plausible approximation can change a rational model into a “rational process model”
- Research in machine learning and statistics has produced useful approximations to statistical models which can be tested as general-purpose psychological heuristics

Nonparametric Bayes and cognition

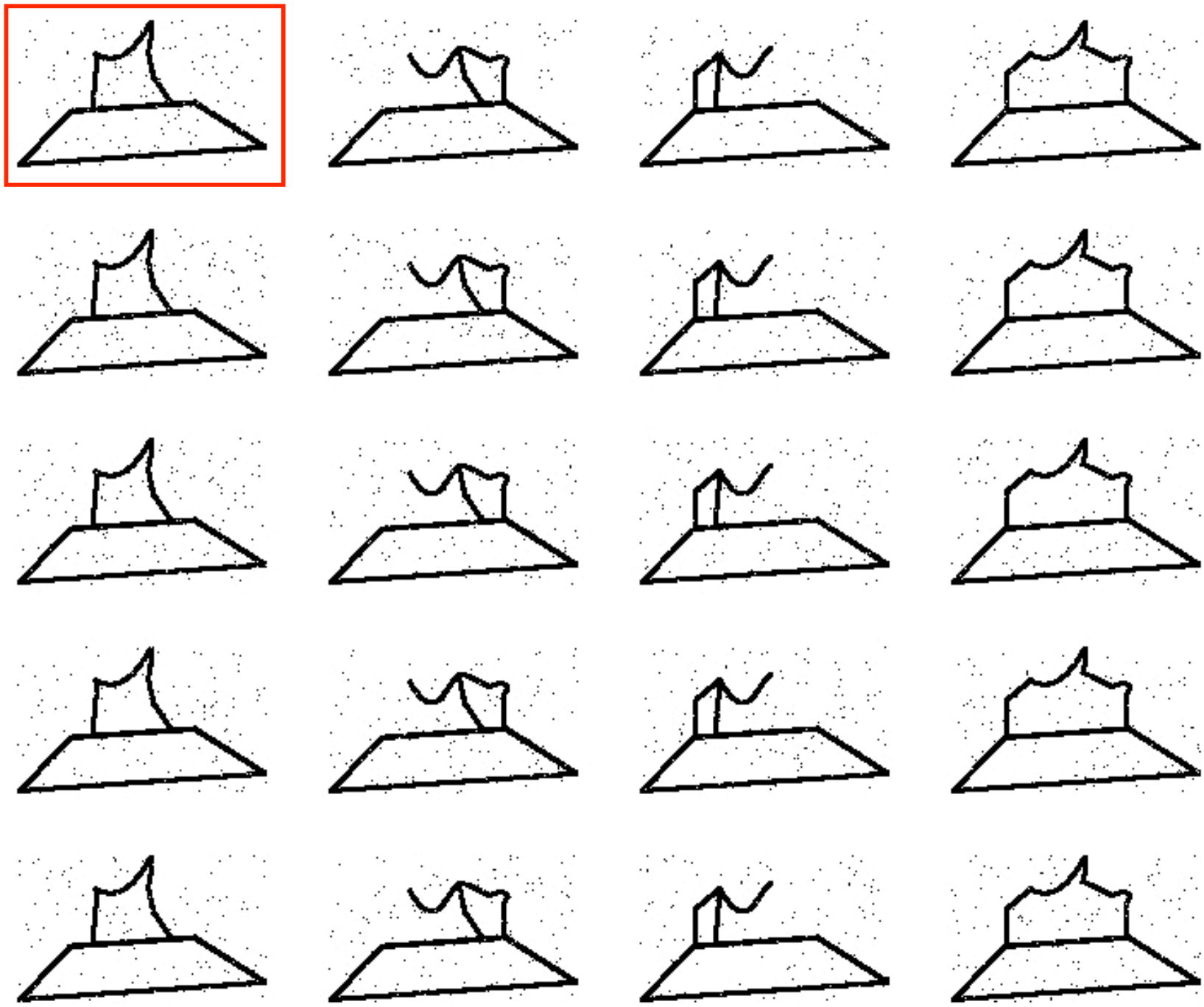
- Nonparametric Bayesian models are useful for answering questions about how much structure people should infer from data
- Many cognitive science questions take this form
  - how should we represent categories?
  - what features should we identify for objects?

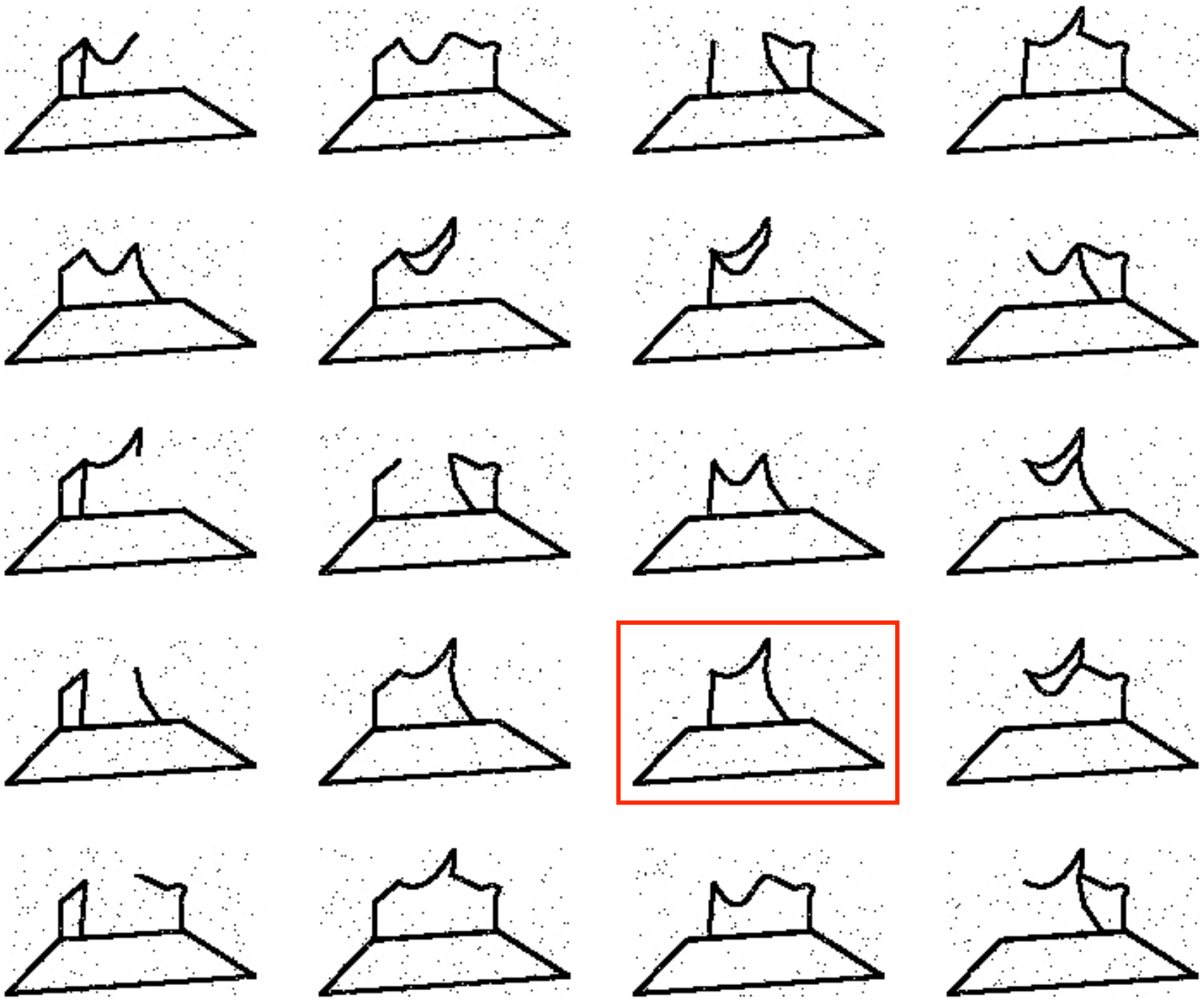
Learning the features of objects

- Most models of human cognition assume objects are represented in terms of abstract features
- What are the features of this object?

- What determines what features we identify?

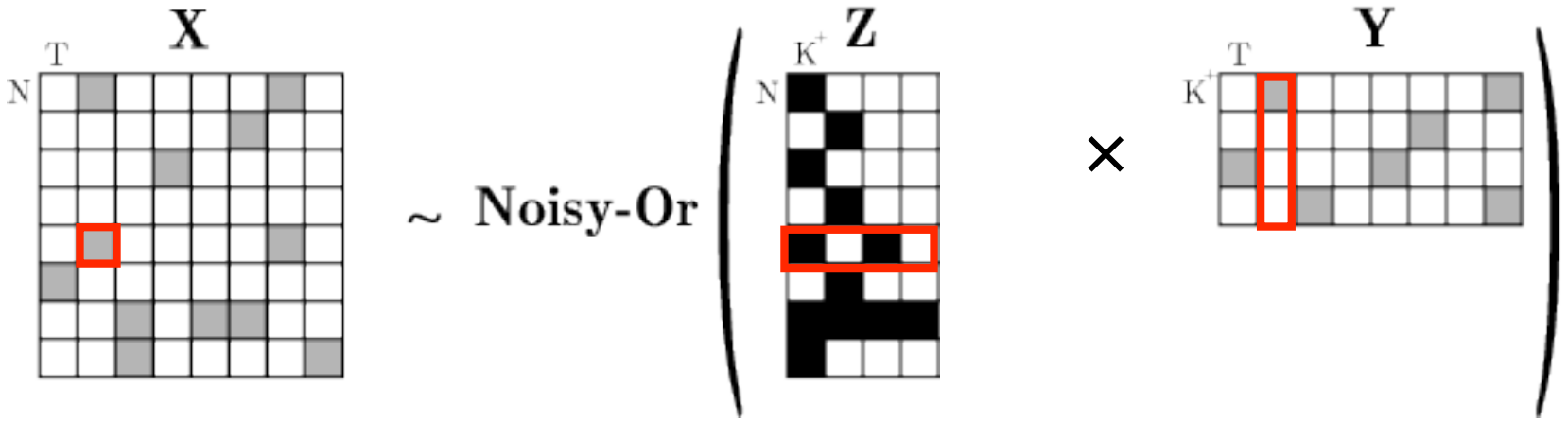
(Austerweil & Griffiths, 2009)





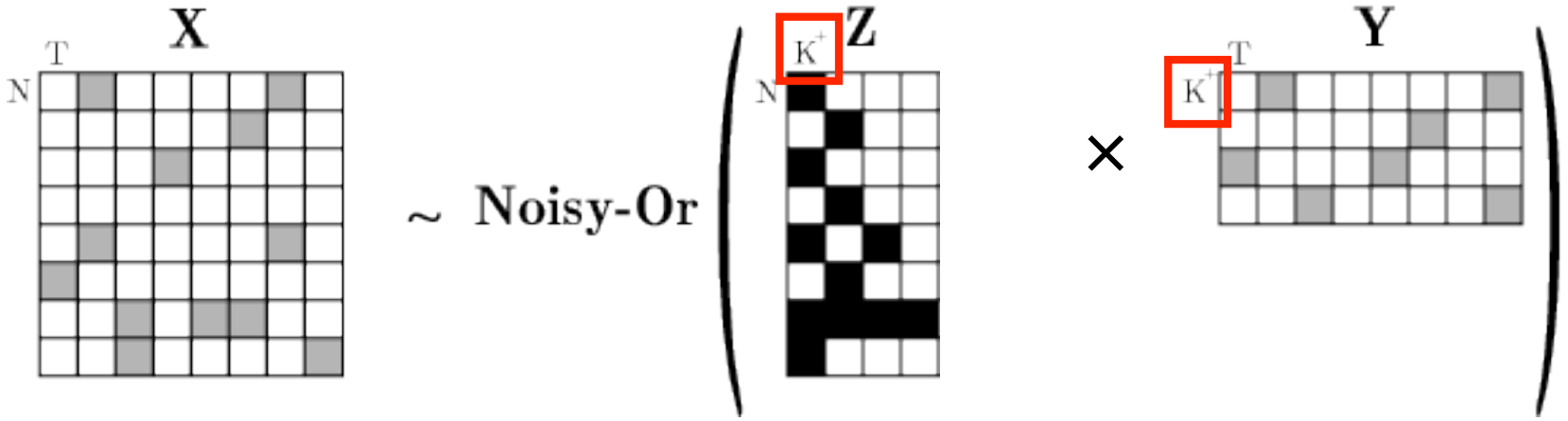
Binary matrix factorization

$$P(x_{i,t} = 1|\mathbf{Z}, \mathbf{Y}) = 1 - (1 - \lambda)^{<\mathbf{z}_{i,:}, \mathbf{y}_{:,t}>}(1 - \epsilon)$$



Binary matrix factorization

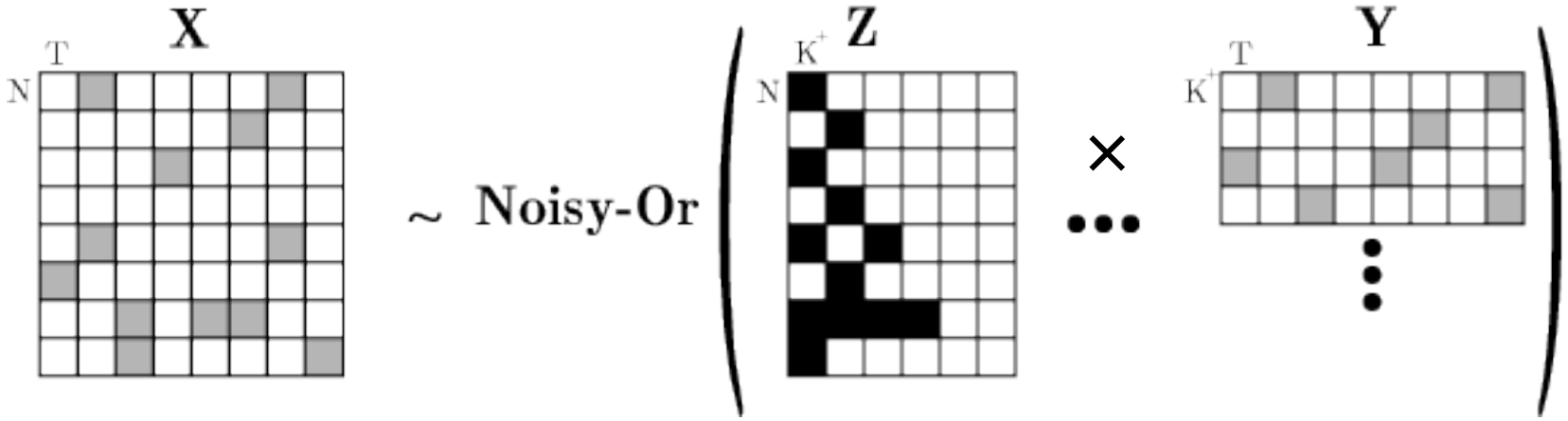
$$P(x_{i,t} = 1 | \mathbf{Z}, \mathbf{Y}) = 1 - (1 - \lambda)^{<\mathbf{z}_{i,:}, \mathbf{y}_{:,t}>}(1 - \epsilon)$$



How should we infer the number of features?

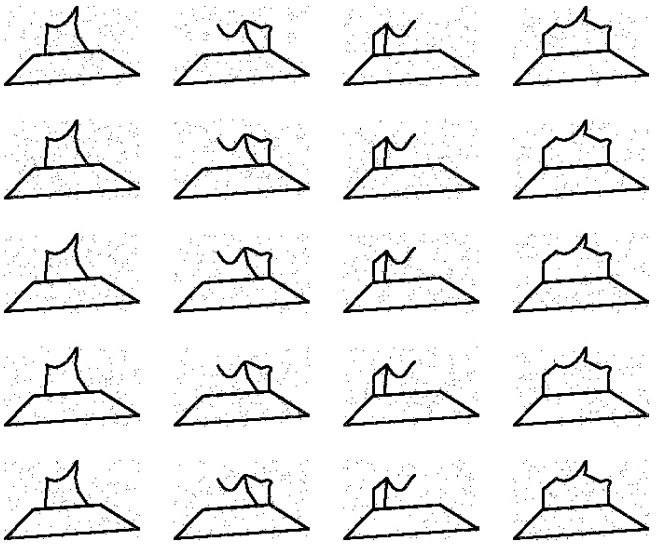
The nonparametric approach

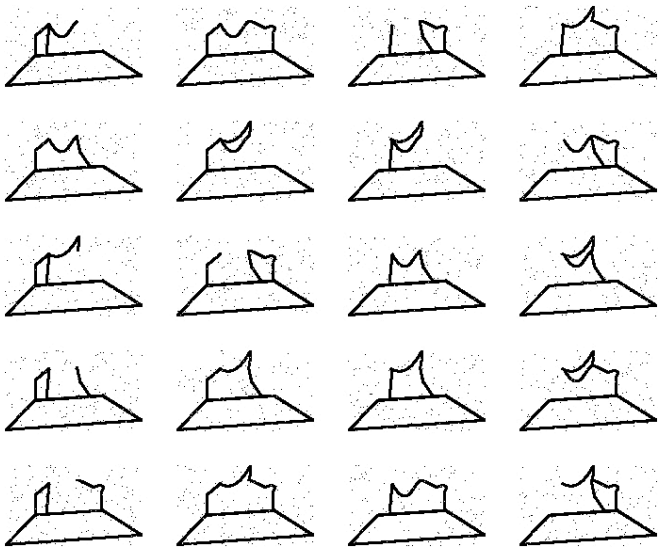
Assume that the total number of features is unbounded, but only a finite number will be expressed in any finite dataset

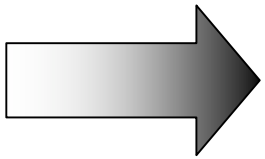


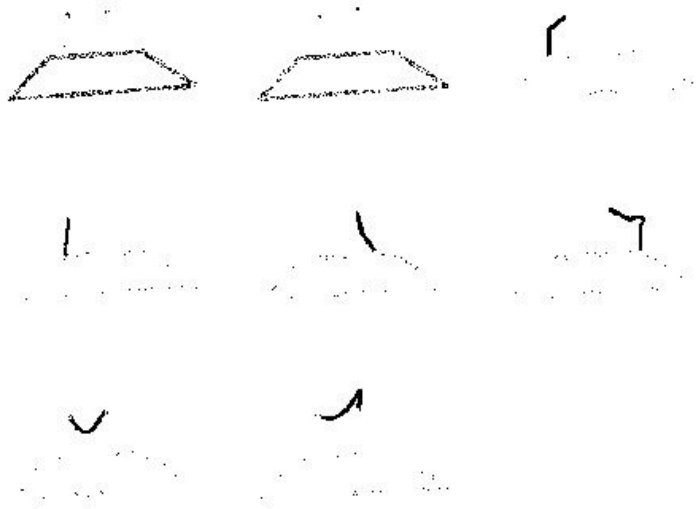
Use the Indian buffet process as a prior on **Z**

(Griffiths & Ghahramani, 2006)









(Austerweil & Griffiths, 2009)

Summary

- Sophisticated tools from Bayesian statistics can be valuable in developing probabilistic models of cognition…
- Monte Carlo methods provide a way to perform inference in probabilistic models, and a source of ideas for process models and experiments
- Nonparametric models help us tackle questions about how much structure to infer, with unbounded hypothesis spaces
- We look forward to seeing what you do with them!