Bayesian inference and generative models in neuroscience

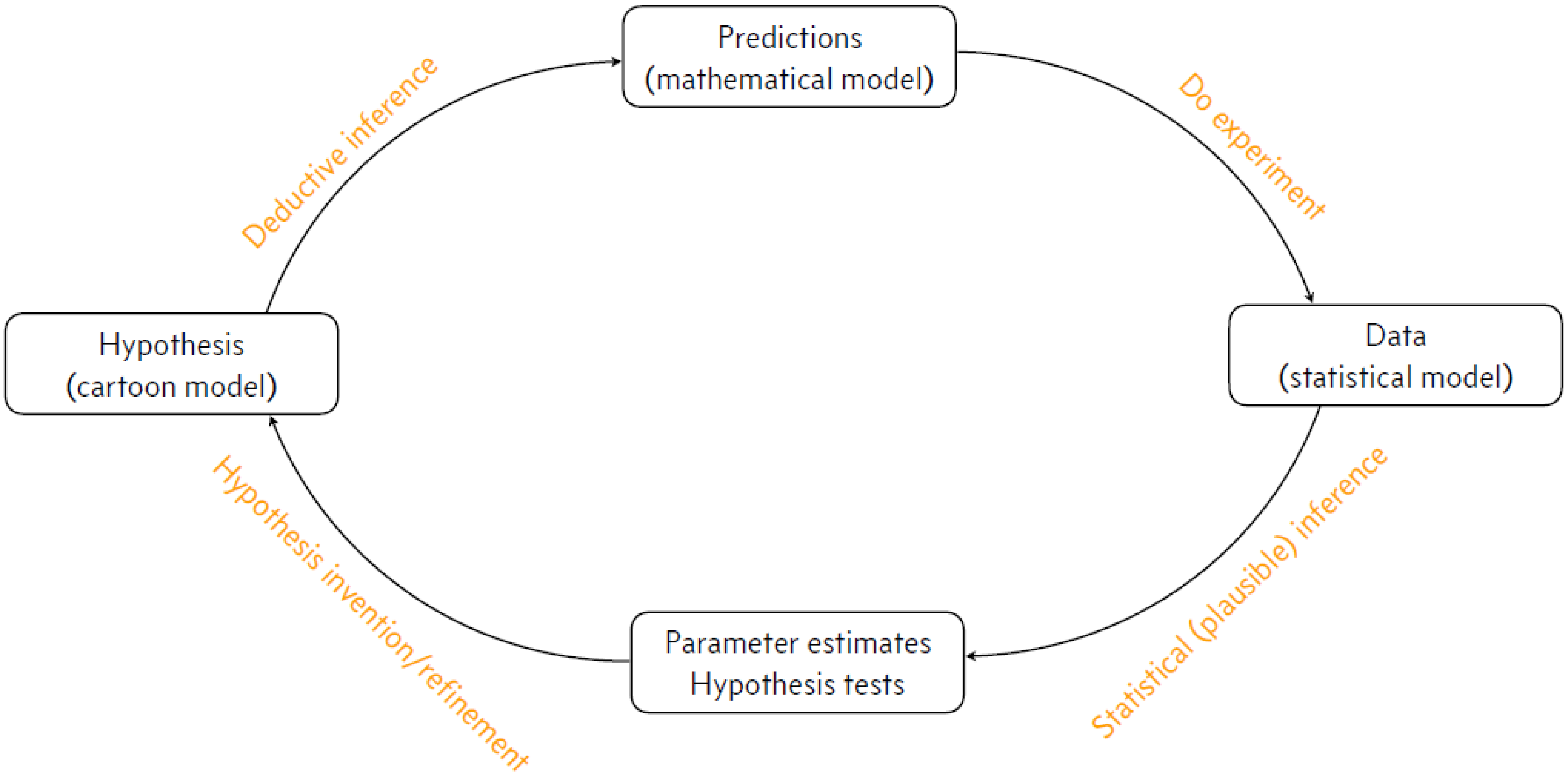
Nhat Le, Peer lecture 3/31/2019

What is the goal of doing (neuroscience) experiments?

Is it…

- To further knowledge?
- To test a hypothesis?
- To explore and observe?
- To demonstrate a method?
- To graduate?

The scientific process



Frequentist vs Bayes

- ***Frequentist probability***: P(A) represents long-run frequency over a large run of repetitions of the experiment.

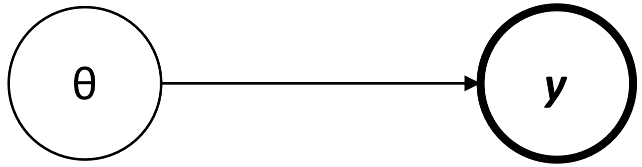
- ***Bayesian probability***: P(A) represents the degree of belief / plausibility about A

Generative (statistical) model

A generative model describes our beliefs about:

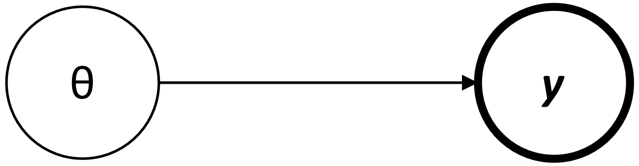
- The sources of error or uncertainty in the data

- Uncertainty in the underlying parameters of the model



Data is generated from hidden (latent) parameters

- ‘Data’: any observable measurements
  - Firing rate
  - Behavior
  - Protein levels
  - Sentences in a language
- ‘Hidden’ parameters govern the generation of data
  - Resting potential of neurons
  - Emotional state
  - Gene sequence
  - Grammar rules



Bayes rule

Parameter (not observed)

Data (observed)

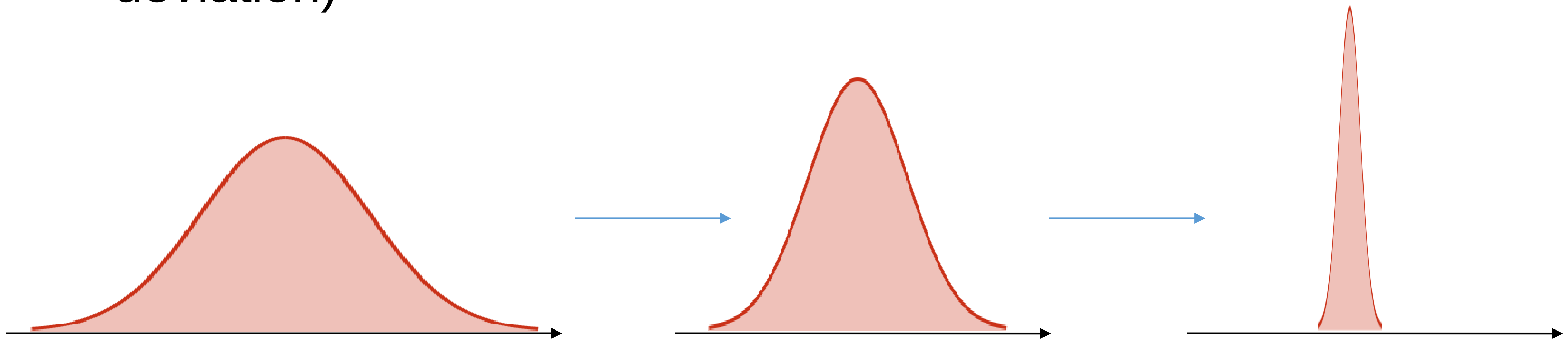
$$P(\theta \mid y) = \frac{P(y \mid \theta)\, P(\theta)}{P(y)}.$$

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}.$$

Our job is to infer hidden parameters, given observed data

- Bayesian analysis is composed of three steps:

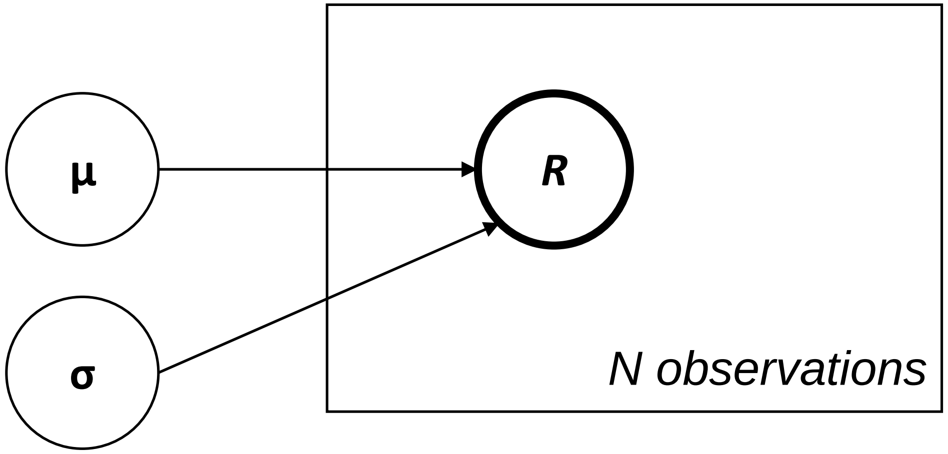
1. Build a model, i.e. define a likelihood $p(D|\theta)$ and a prior $p(\theta)$
2. Compute the posterior $p(\theta|D)$
3. Report some summary of the posterior (mean and standard deviation)



An example: neuron firing rate

- Average firing rate R modelled as normally distributed
- ***Parameters of the model***: mean μ and standard deviation σ
- ***Observed data***: average firing rates R for different trials

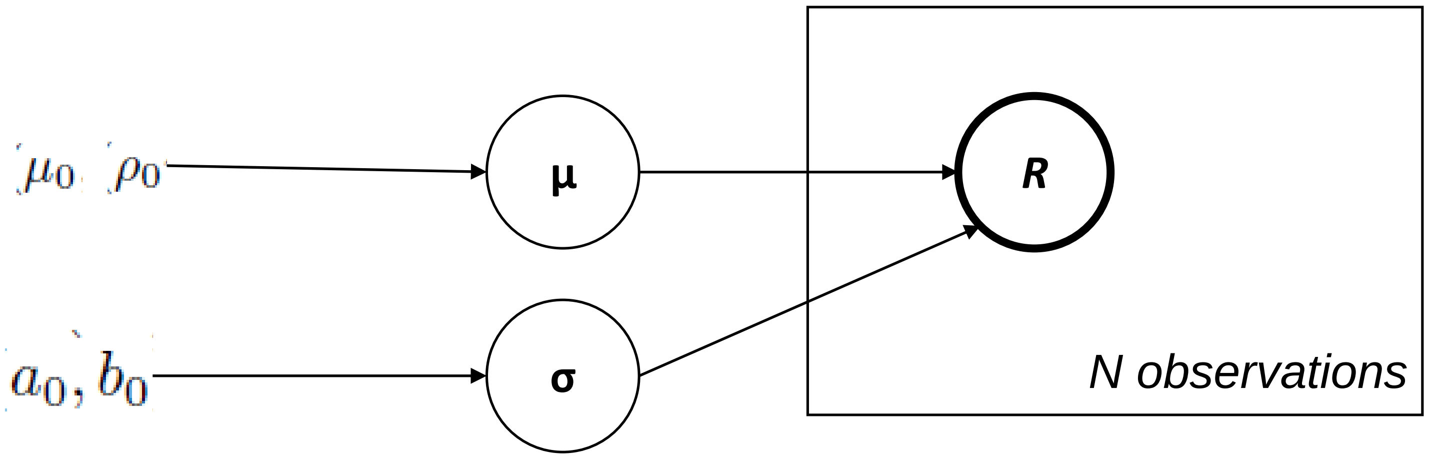
Inference objective: infer the likely values of μ and σ based on observed firing rates.



An example: neuron firing rate

- Average firing rate *R* modelled as normally distributed
- ***Parameters of the model***: mean μ and standard deviation σ
- ***Observed data***: average firing rates for different trials

Inference objective: infer the likely values of μ and σ based on observed firing rates.



$$y_n \overset{iid}{\sim} \mathcal{N}(\mu, \tau^{-1}), \text{ where } \tau = (\sigma^2)^{-1}$$

$$\tau \sim \text{Gamma}(a_0, b_0)$$

$$\mu \mid \tau \sim \mathcal{N}(\mu_0, (\rho_0 \tau)^{-1})$$

Exact Inference

- Evaluate the posterior directly using Bayes' rule
- Main challenge: compute the evidence (denominator)

$$P(\theta \mid y) = \frac{P(y \mid \theta)\, P(\theta)}{\boxed{P(y)}}.$$

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}.$$

Conjugate priors

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}.$$

Posterior can be easily evaluated if likelihood and priors come from ‘conjugate families’

Likelihood	Conjugate prior	Posterior update
$x \sim \text{Normal}(\mu=z, 1)$	$z \sim \text{Normal}(\mu_0, \sigma_0^2)$	$z \mid x \sim \text{Normal}\left(\frac{x + \mu_0/\sigma_0^2}{1 + \sigma_0^2}, \frac{1}{1 + \sigma_0^2}\right)$
$x \sim \text{Normal}(\mu=0, \sigma^2=z)$	$z \sim \text{InvGamma}(\alpha, \beta)$	$z \mid x \sim \text{InvGamma}(\alpha + ½, \beta + ½\, x^2)$
$x \sim \text{Bernoulli}(p=z)$	$z \sim \text{Beta}(\alpha, \beta)$	$z \mid x \sim \text{Beta}(\alpha + x, \beta + 1 - x)$

For a list of conjugate pairs: https://en.wikipedia.org/wiki/Conjugate_prior

Approximate inference

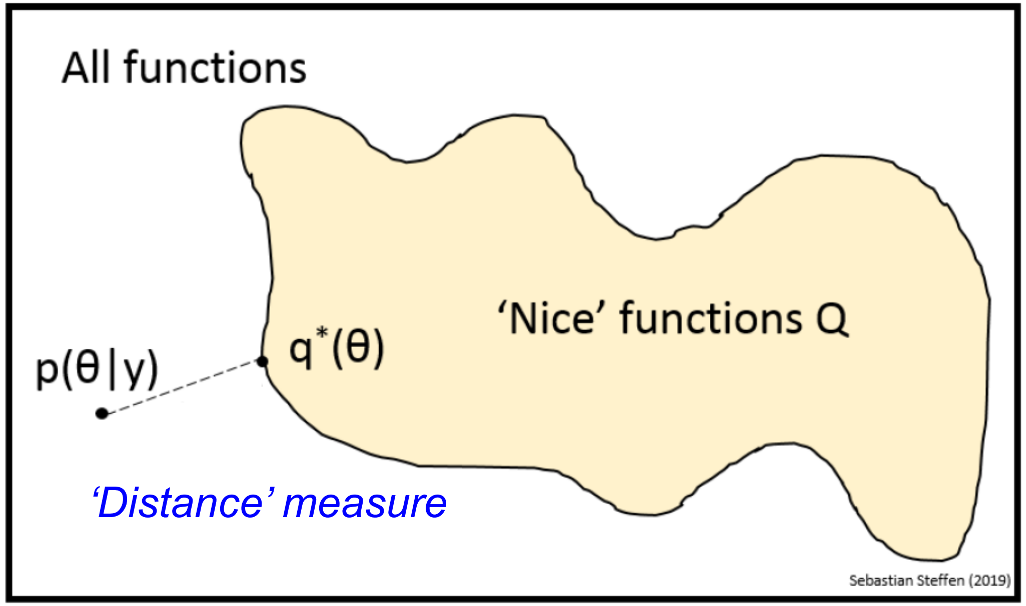
- Most of the time, exact inference is not possible

→ Have to resort to approximate inference

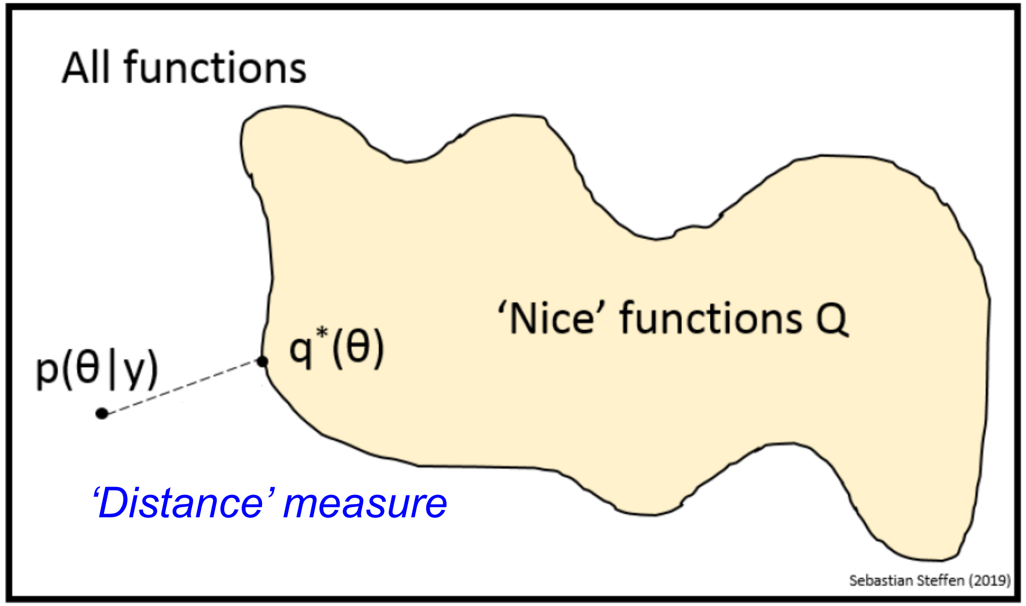
- Two ways of doing this
  - Approximate inference by optimization
  - Approximate inference by sampling

Approximate inference by optimization (aka Variational inference)

- Can't evaluate posterior P(θ | y) directly
- **Strategy**: approximate it with some distribution q(θ)

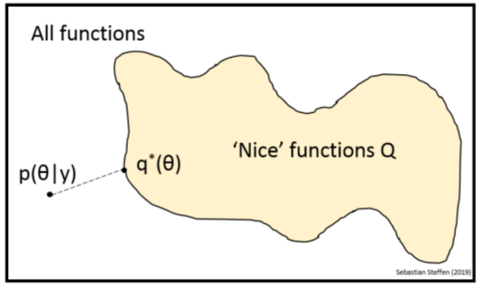


Choices



1. Choice of distance measure is KL-divergence
2. Choice of 'nice' functions is often the mean-field approximation
3. Choice of optimization procedure is coordinate ascent

Objective: minimize distance



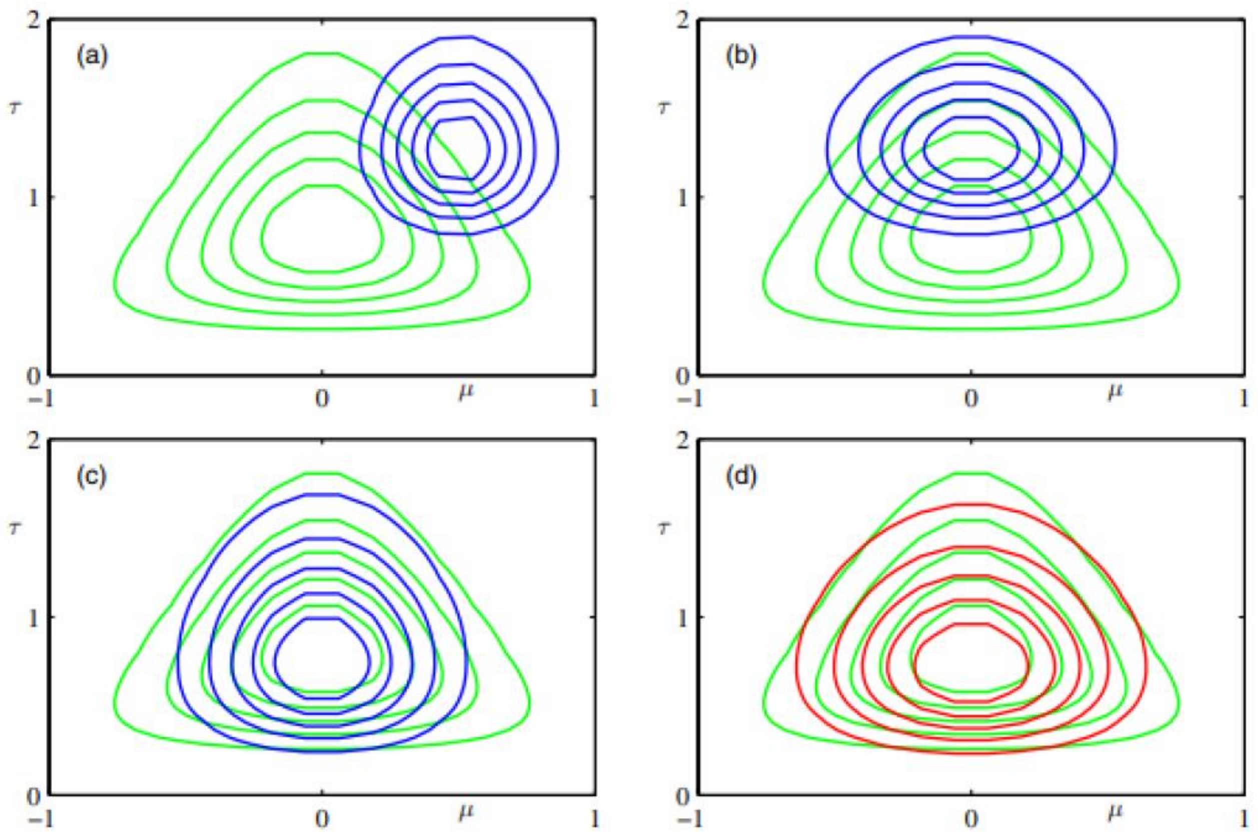
$$D_{KL}[\,Q(z)\,||\,p(z|x)\,] = -E_{z\sim Q}\left[\log \frac{p(z|x)}{Q(z)}\right]$$

$$= -E_{z\sim Q}\left[\log p(z,x) - \log Q(z)\right] + \text{const.}$$

ELBO (evidence lower bound)

Lower bound on log p(x)

To minimize distance, perform coordinate ascent



Free energy

- *Free energy is the ELBO!*

- Brief explanation:

$$P(\tilde{o}, \tilde{s}, \pi, a, b, d, \beta)$$

The world's generative model,
a joint distribution over observations, outcome probabilities, state-related variables, and the agent's policies

$$Q(\tilde{s}, \pi, a, b, d, \beta)$$

The agent's approximation of the generative model

Agent's objective: minimize KL(P || Q) → maximize ELBO = minimize free energy

Approximate inference by sampling

- Can't evaluate posterior $P(\theta \mid y)$ directly
- **Strategy**: draw samples from the posterior

Markov Chain Monte Carlo (MCMC)

Stanisław Ulam

Monte Carlo inference

- We want to sample from some distribution
  - In this case, a posterior p(z|x)
- We can't sample from p directly, but maybe we can evaluate it
  - Or maybe we can only evaluate an unnormalised version of it, e.g. p(z, x)

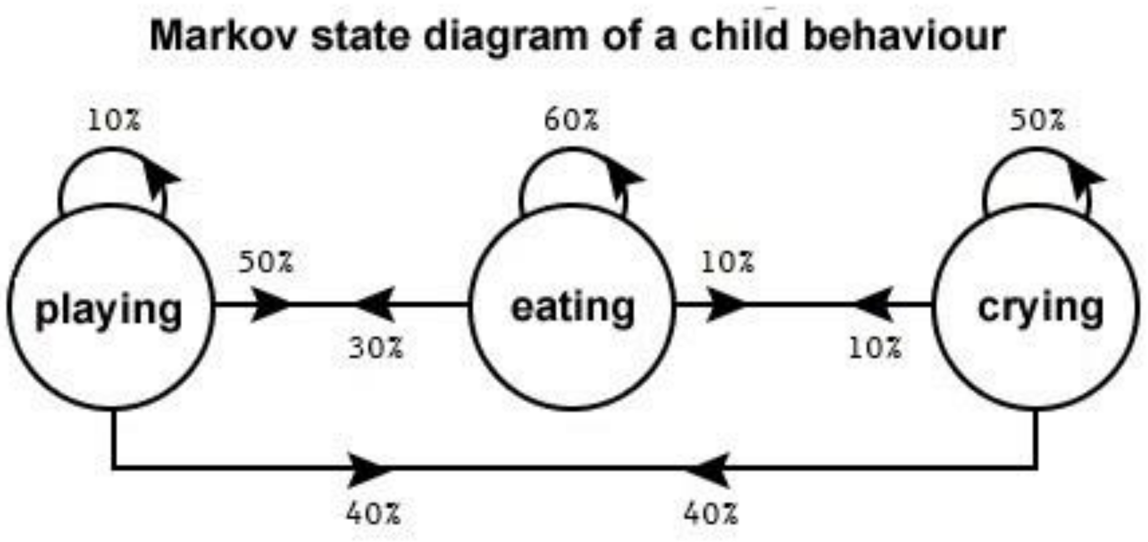
Take samples from some other distribution (e.g. prior) and transform/reweight/etc. them so that they become samples from the posterior

Sampling techniques

- Many different ways to sample
  - Importance sampling
  - Rejection sampling
  - MCMC (Markov Chain Monte Carlo)
    - Gibbs sampling
    - Slice sampling
    - HMC
    - NUTS

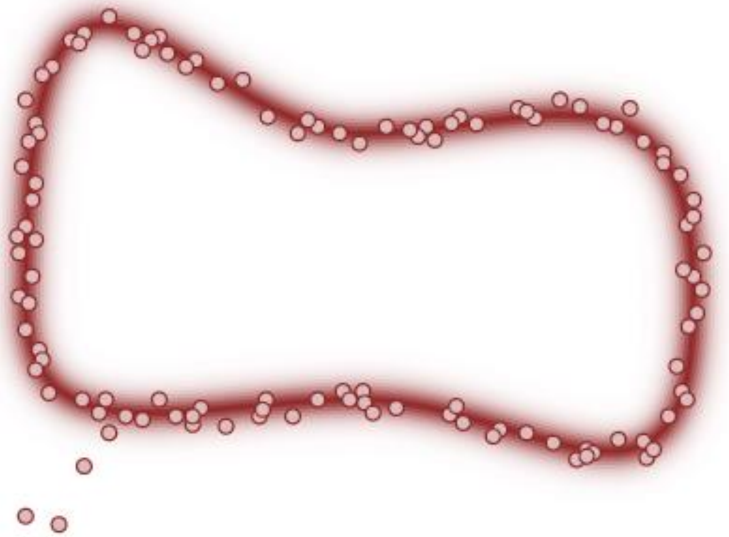
A Markov Chain

A random walk in which the next step depends only on where you are now



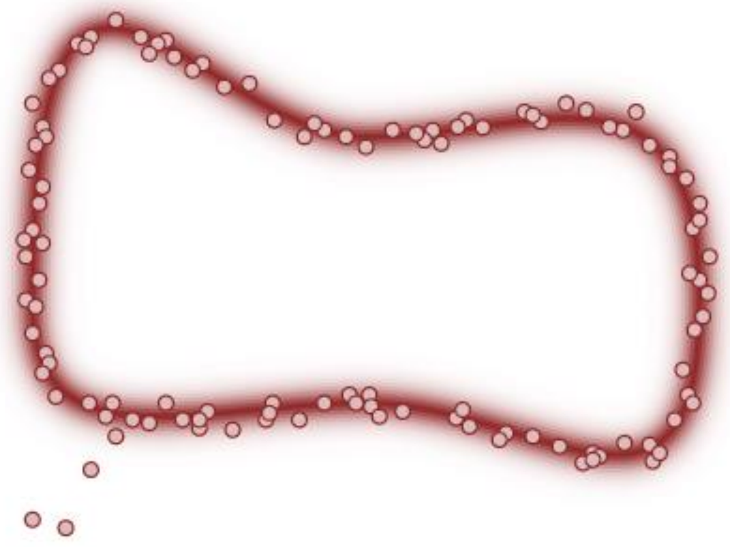
MCMC methods

- Idea: take random walks in the parameter space to sample from the target distribution



MCMC convergence

- To ensure that the final samples indeed come from the target distribution, we need to satisfy a few conditions
  - Detailed balance: eventually converges to the target distribution
  - Ergodicity: able to get to any point in parameter space in finite time



Some MCMC algorithms

1. Random-walk Metropolis-Hastings

   - Propose new candidate states, accept/reject the proposal with some probability

2. Gibbs sampling

   - For *n* parameters, fix (*n* – 1) of them and sample from the conditional distribution

3. Hamiltonian Monte Carlo

   - Make use of extra momentum variables to flow through parameter space

Demo of MCMC in action

How can I get started with these inferences for my data analysis?

Stan will help you!

State-of-the-art platform for statistical modeling

Supports:

- Full Bayesian statistical inference with MCMC sampling (NUTS, HMC)
- Approximate Bayesian inference with variational inference (ADVI)

Interfaces with most popular data analysis languages (Python, MATLAB, R)

Simple syntax

```
data {
  int<lower=0> N;           // number of neurons measured
  real y[N];                // firing rates

  // prior parameters
  real mu_0;
  real<lower=0> rho_0;
  real<lower=0> alpha_0;
  real<lower=0> beta_0;
}
```

Simple syntax

```
parameters {
  real mu;
  real<lower=0> tau;  //precision
}

model {
  tau ~ gamma(alpha_0, beta_0);
  mu ~ normal(mu_0, 1 / (rho_0 * tau));
  for( n in 1:N )
    y[n] ~ normal(mu, 1 / tau);
}
```