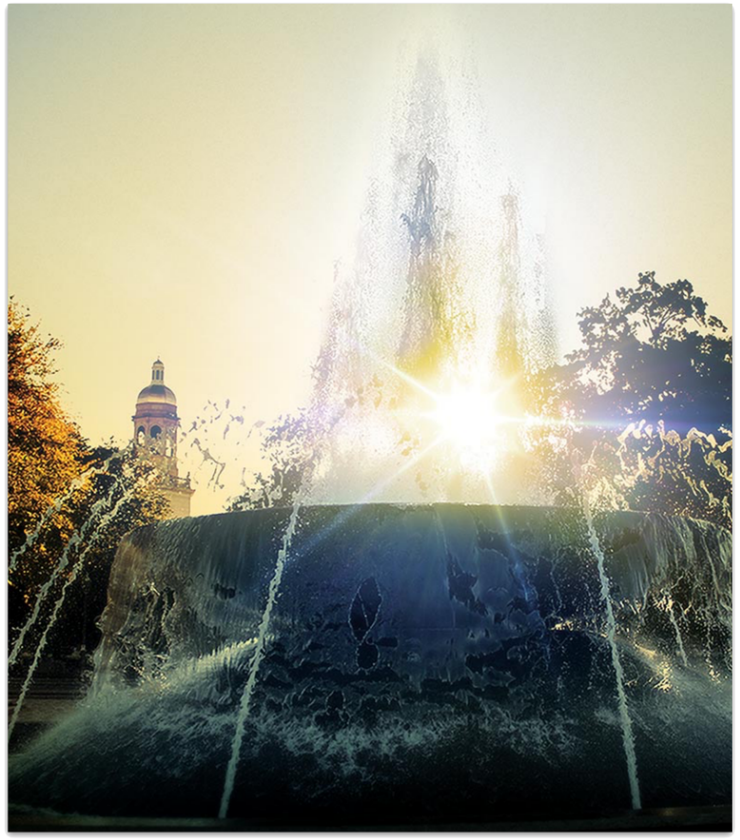
Practical Strategies for Bayesian Computation



Introduction

The unique computational challenges of Bayesian methods

Approximately computing the exact posterior

Exactly computing an approximate posterior

Software tools for Bayesian computation

Baylor University
COLLEGE OF ARTS & SCIENCES
Statistical Science

David Kahle

Introduction

Bayesian statistics

In Bayesian statistics, probability distributions = belief distributions

Used to determine degrees of belief about propositions

Basic setup: five distributions of interest

Data model = family of probability distributions concerning provenance of outcomes

Usually a parametric model, $\theta \in \Theta$. "True" θ is fixed but unknown.

Viewed as a function of parameters for fixed data over Θ, the density of the data model is the *likelihood*

Prior distribution = belief distribution over Θ *before* seeing any data

Goal of analysis: Inference involving parameters after observing data

Posterior distribution = belief distribution over Θ *after* seeing data

Result of coherently updating prior belief in the light of data to arrive at an updated belief distribution

Bayes' theorem = mechanism used to coherently synthesize prior and data

Bayesian statistics

Bayes' theorem = mechanism used to coherently synthesize prior and data

Posterior — Likelihood — Prior — Un-normalized posterior

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y},\boldsymbol{\theta})}{p(\mathbf{y})} = \frac{p(\mathbf{y},\boldsymbol{\theta})}{\int_{\boldsymbol{\Theta}} p(\mathbf{y},\boldsymbol{\theta})\,d\boldsymbol{\theta}} = \frac{l(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\Theta}} l(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})\,d\boldsymbol{\theta}} \propto l(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta}) =: \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})$$

Prior predictive distribution = distribution over outcomes integrating over prior

$$p(\mathbf{y}) = \int_{\boldsymbol{\Theta}} p(\mathbf{y},\boldsymbol{\theta})\,d\boldsymbol{\theta} = \int_{\boldsymbol{\Theta}} p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})\,d\boldsymbol{\theta}$$

Prior predictive

Mixture of distributions in data model weighted by prior

Also denote this quantity Z(y), the "partition function"

Also the expectation of the likelihood w.r.t. the prior

Posterior predictive distribution = distribution over outcomes integrating over posterior

$$p(\mathbf{y}'|\mathbf{y}) = \int_{\boldsymbol{\Theta}} p(\mathbf{y}',\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta} = \int_{\boldsymbol{\Theta}} p(\mathbf{y}'|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}$$

Posterior predictive

Mixture of distributions in data model weighted by posterior

Bayes' theorem

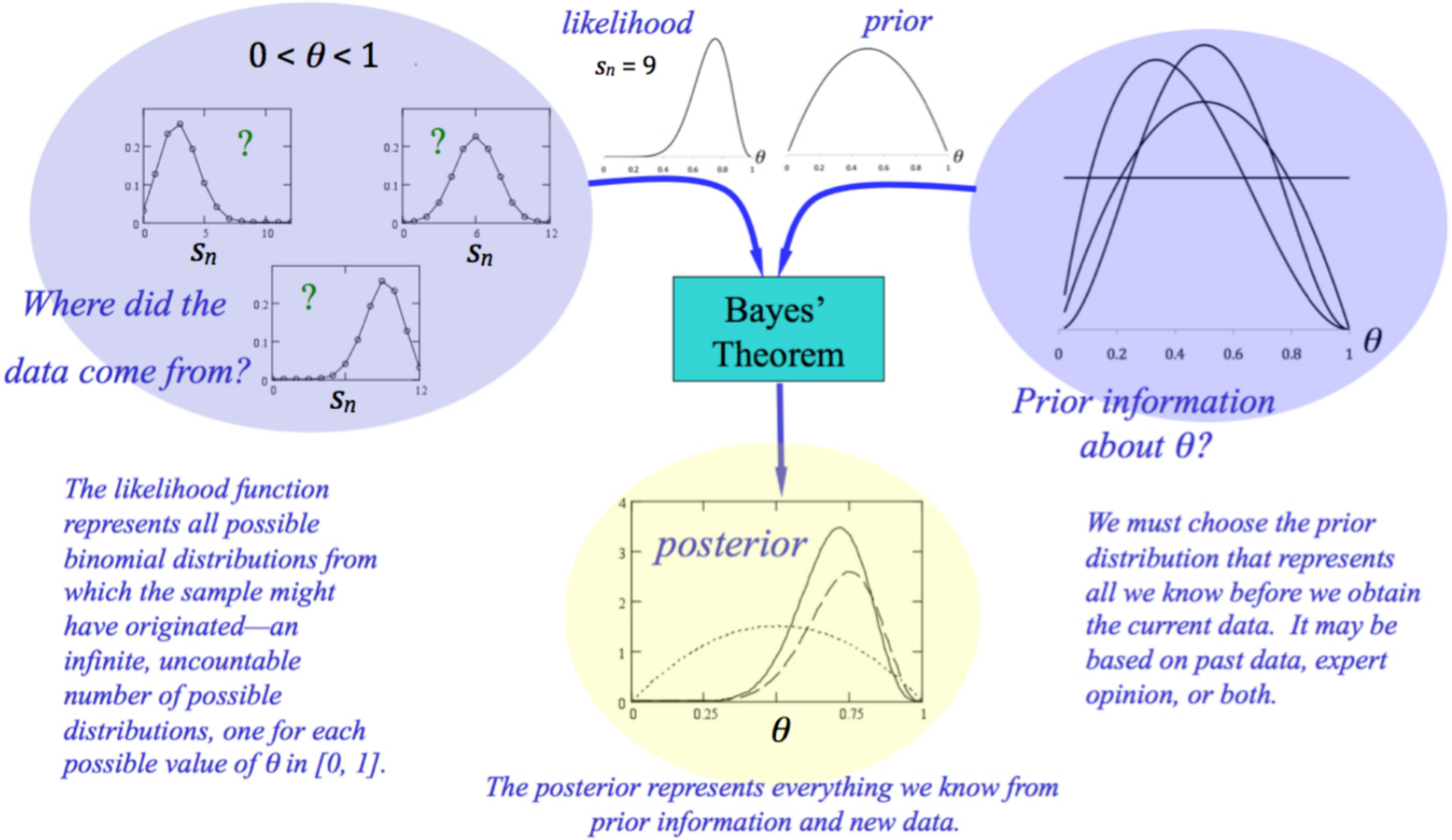


Figure 1.1: A schematic representation of combining prior and likelihood using Bayes' theorem.

Bayes' theorem

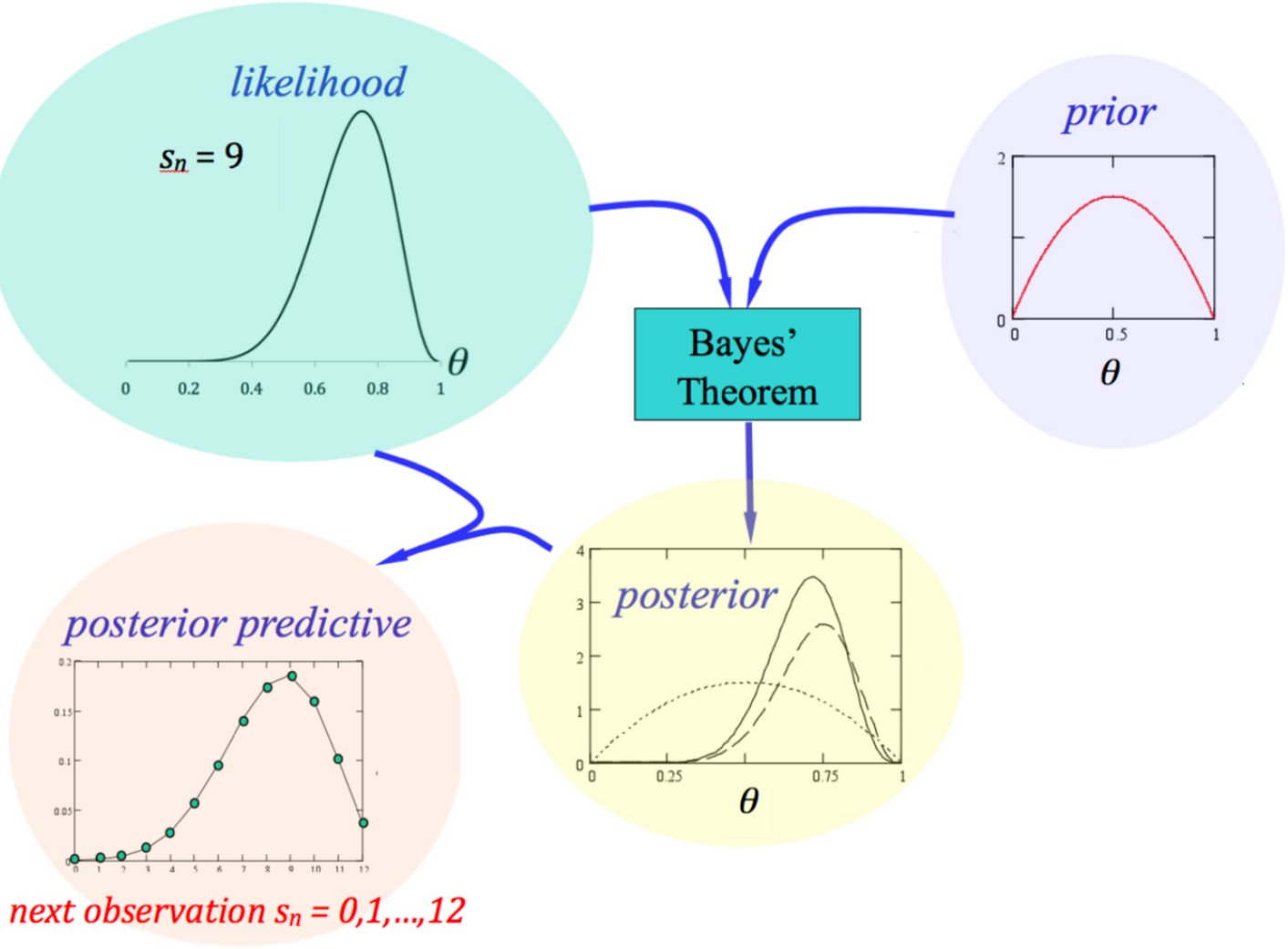


Figure 1.2: The general procedure for Bayesian prediction.

Bayesian statistics

Posterior distribution = Central object of interest in Bayesian statistics

However: posterior distribution is never the end of an analysis

In practice, summaries are desired

Bayes estimator = mean of posterior distribution

MAP estimator = mode of posterior distribution

Credible intervals = probability intervals concerning θ based off p(θ|y)

Decision rules can be created by inverting credible intervals

Quantities of interest can be written as expectations with respect to posterior

Mean: g(x) = x

Variance: g(x) = $(x - \mu_\theta)^2$

Probabilities: g(x) = 1[x ∈ A]

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{y})}[g(\boldsymbol{\theta})] = \int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{y})\, d\boldsymbol{\theta}$$

"*Computing the posterior*" *(CTP)* = computing p(θ|y) or expectations $E_{p(\theta|y)}[g(\theta)]$

running example

Suppose $Y_1, \ldots, Y_n \sim N(\mu, \sigma^2)$ iid are observed, n fixed and known

Inferential target: $\theta = (\mu, \sigma^2)$

Observe values $y_1, \ldots, y_n$

Using previous notation

$$l(\boldsymbol{\theta}|\mathbf{y}) \;=\; l(\mu, \sigma^2|\mathbf{y}) \;=\; (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2\right\} \qquad \boldsymbol{\Theta} = \mathbb{R}\times\mathbb{R}_+$$

Our choice of prior will vary depending on the method being introduced

But generally we will assume it flat over $-2 \le \mu \le 2$ and $0 \le \sigma^2 \le 4$

$$p(\mu, \sigma^2|\mathbf{y}) = \frac{l(\mu, \sigma^2|\mathbf{y})p(\mu, \sigma^2)}{\int_0^4\int_{-2}^2 l(\mu, \sigma^2|\mathbf{y})p(\mu, \sigma^2)\,d\mu\,d\sigma^2} = \frac{l(\mu, \sigma^2|\mathbf{y})\frac{1}{16}}{\int_0^4\int_{-2}^2 l(\mu, \sigma^2|\mathbf{y})\frac{1}{16}\,d\mu\,d\sigma^2}$$

Approximately Computing the Exact Posterior

approximately computing the post

Most basic approach to CTP is to approximate solutions to the problem posed

Ex: Compute the Bayes estimator for (μ,σ^2)

$$\widehat{\theta} = (\widehat{\mu}, \widehat{\sigma}^2) = \left(\mathbb{E}_{p(\mu,\sigma^2|\mathbf{y})}[\mu], \mathbb{E}_{p(\mu,\sigma^2|\mathbf{y})}[\sigma^2]\right)$$

$$\widehat{\mu} = \mathbb{E}_{p(\mu,\sigma^2|\mathbf{y})}[\mu] = \int_0^4 \int_{-2}^2 \mu \; p(\mu, \sigma^2|\mathbf{y}) \, d\mu d\sigma^2$$

$$p(\mu, \sigma^2|\mathbf{y}) = \frac{l(\mu, \sigma^2|\mathbf{y})\frac{1}{16}}{\int_0^4 \int_{-2}^2 l(\mu, \sigma^2|\mathbf{y})\frac{1}{16} \, d\mu \, d\sigma^2}$$

$$l(\mu, \sigma^2|\mathbf{y}) = (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2\right\}$$

Two approaches:

1. Deterministic approximations

Fixed strategies; when two statisticians use the same method, they get the same answer

2. Stochastic (Monte Carlo) approximations

Algorithms based on random numbers that rely on laws of large numbers (LLN) and central limit theorem (CLT) arguments

deterministic strategies

symbolic integration

In very rare circumstances, the integrals can be performed symbolically

Find antiderivatives and apply the fundamental theorem of calculus

In real problems, this essentially always fails

Worth looking at how it fails

$$\widehat{\mu} = \mathbb{E}_{p(\mu,\sigma^2|\mathbf{y})}[\mu] = \int_0^4 \int_{-2}^2 \mu \; p(\mu, \sigma^2|\mathbf{y}) \, d\mu d\sigma^2$$

This integral is structurally very similar

If we were computing probabilities the integrand would be identical

$$\int_{\Theta} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y}) \, d\boldsymbol{\theta} = \int_{\Theta} l(\boldsymbol{\theta}|\mathbf{y}) p(\boldsymbol{\theta}) \, d\boldsymbol{\theta}$$

$$p(\mu, \sigma^2|\mathbf{y}) = \frac{l(\mu, \sigma^2|\mathbf{y}) \frac{1}{16}}{\int_0^4 \int_{-2}^2 l(\mu, \sigma^2|\mathbf{y}) \frac{1}{16} \, d\mu \, d\sigma^2}$$

Constant not involving μ or σ^2

$$l(\mu, \sigma^2|\mathbf{y}) = (2\pi\sigma^2)^{-n/2} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu)^2 \right\}$$

Gaussian integral: provably can't represent in terms of elementary functions

deterministic strategies

conjugate priors

Sometimes $p(\theta)$ can be selected from a family so that it combines with the likelihood to make the kernel of $l(\theta|y)p(\theta)$ recognizable as a common distribution

Families that do this are called conjugate families: $p(\theta) = p(\theta|\eta)$ implies $l(\theta|y)p(\theta) = p(\theta|\eta')$

Priors from conjugate families are *conjugate priors*

Other parameters are simple functions of prior parameters and data

There are tables of conjugate families

binomial-beta poisson-gamma multinomial-dirichlet (much bigger list)

Present scenario: if $p(\mu,\sigma^2) = p(\mu|\sigma^2)\, p(\sigma^2)$ = (Normal)(Inv-Gamma)

$p(\mu,\sigma^2|y)$ is also (Normal)(Inv-Gamma) but with different parameters

Expectations with respect to well known distributions tend to be solved

Either via high-quality numerical implementations or via Monte Carlo

Limitation: Most practical modeling likelihoods don't admit conjugate families

deterministic strategies

numerical integration

Numerical integration = use numerical analysis to approximate integrals

Typically a linear combination of the values of l(θ|y)p(θ) at points in domain of integration

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{l(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\Theta}} l(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})\,d\boldsymbol{\theta}} = \frac{\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})}{\int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}$$

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{y})}[g(\boldsymbol{\theta})] = \int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta} = \frac{\int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}{\int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}$$

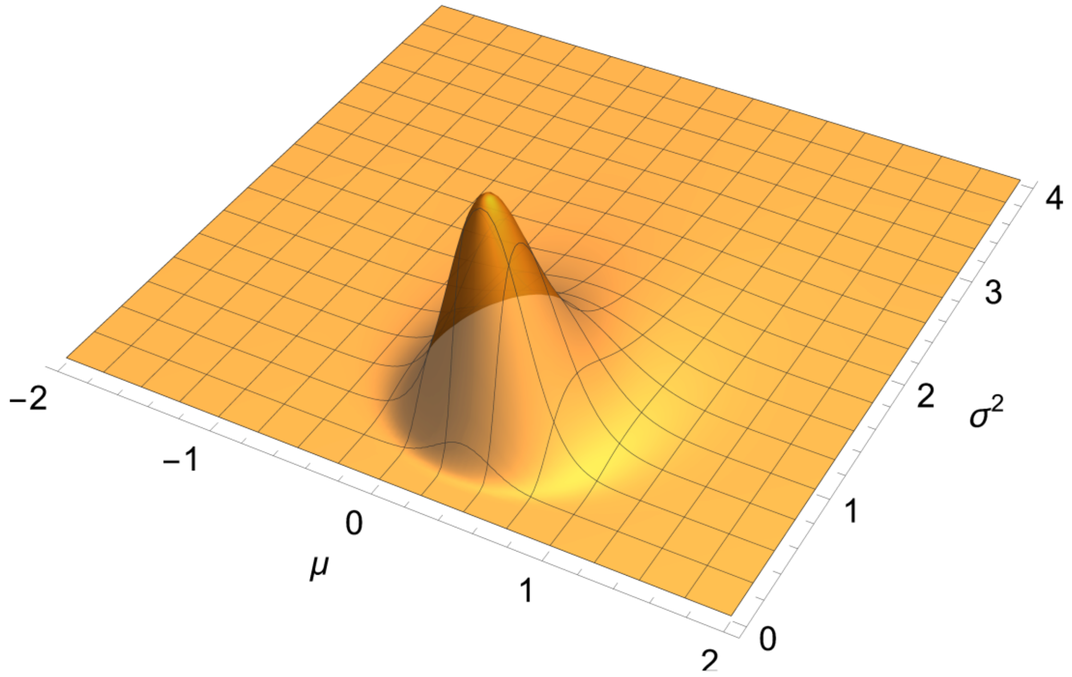
p̃(θ|y) is usually a well-behaved function, we can often approximate

$$Z(\mathbf{y}) = \int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta} \approx \sum_{i=1}^{N} \widetilde{p}(\boldsymbol{\theta}_i|\mathbf{y})\,w_i \quad \text{and similarly} \quad \int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta} \approx \sum_{i=1}^{N} g(\boldsymbol{\theta}_i)\widetilde{p}(\boldsymbol{\theta}_i|\mathbf{y})\,w_i$$

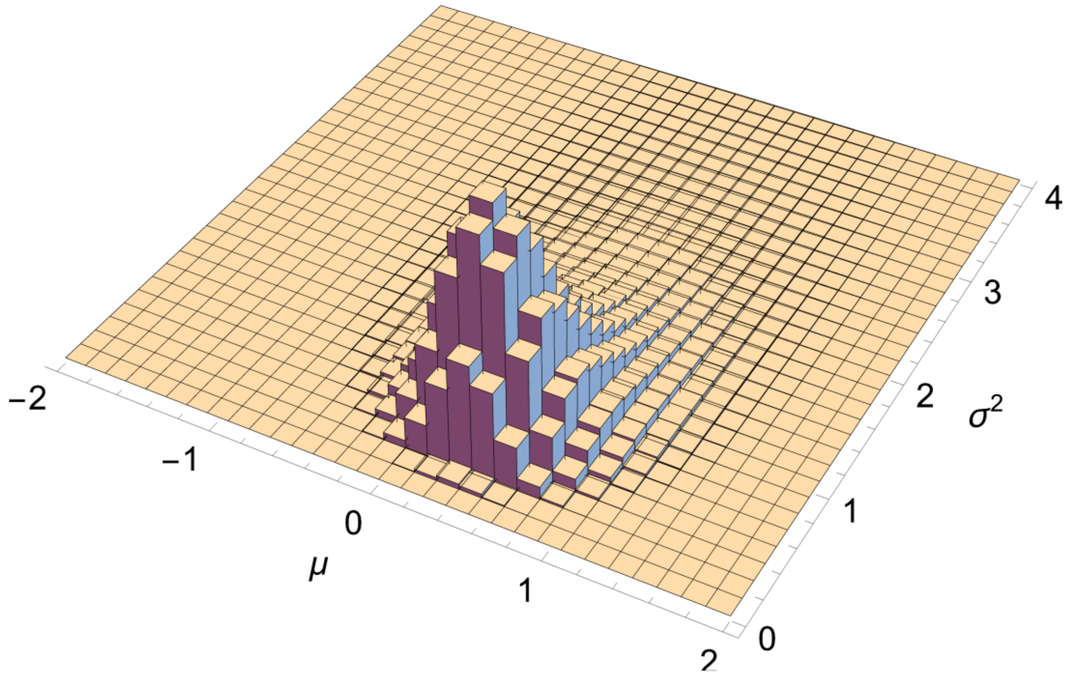
$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{y})}[g(\boldsymbol{\theta})] = \frac{\int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}{\int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}} \approx \frac{\sum_{i=1}^{N} g(\boldsymbol{\theta}_i)\widetilde{p}(\boldsymbol{\theta}_i|\mathbf{y})\,w_i}{\sum_{i=1}^{N} \widetilde{p}(\boldsymbol{\theta}_i|\mathbf{y})\,w_i}$$

deterministic strategies

numerical integration



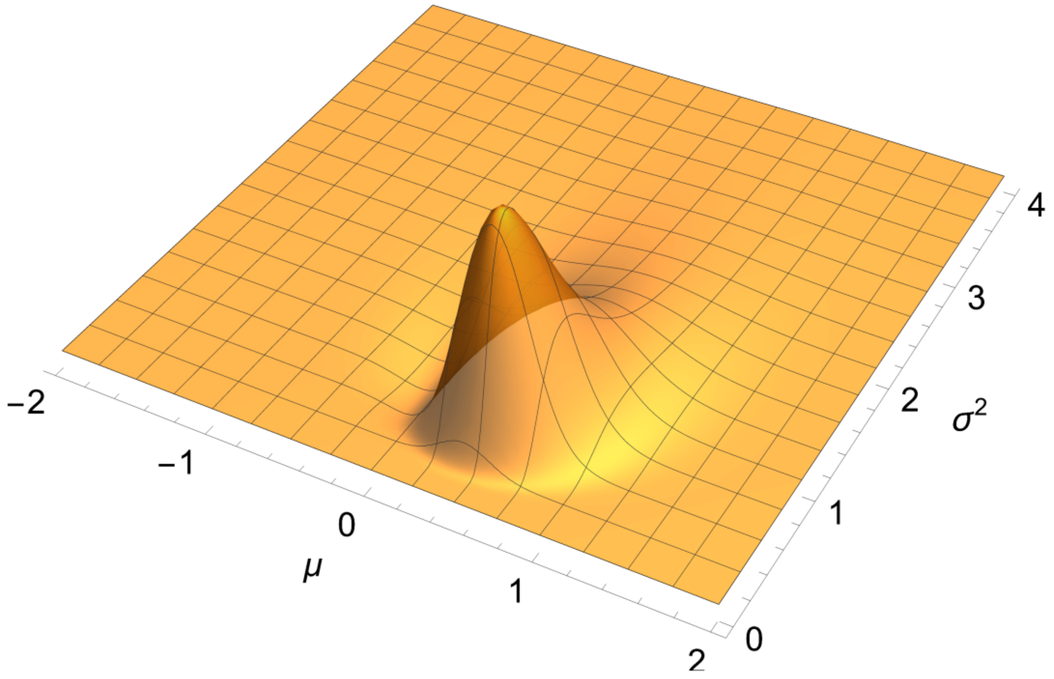
$\tilde{p}(\mu,\sigma^2|y)$



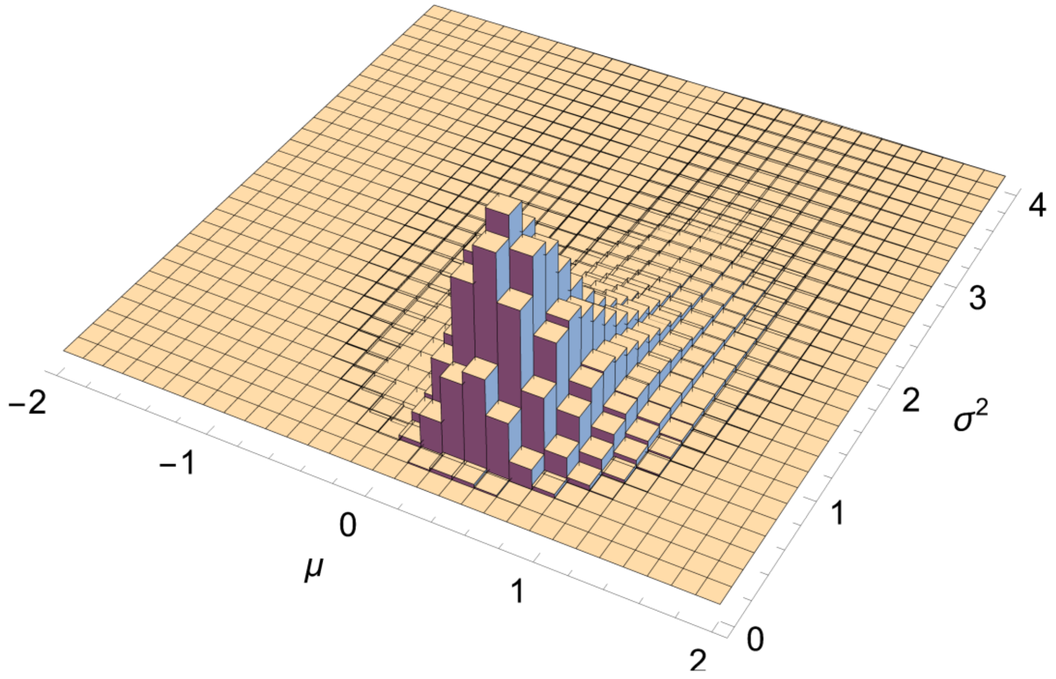
$\tilde{p}(\mu_i,\sigma_i^2|y)$

deterministic strategies

numerical integration



$\mu\ \tilde{p}(\mu,\sigma^2|y)$



$\mu_i\ \tilde{p}(\mu_i,\sigma_i^2|y)$

deterministic strategies

numerical integration

Numerical integration = use numerical analysis to approximate integrals

Typically a linear combination of the values of l(θ|y)p(θ) at points in domain of integration

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{l(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\Theta}} l(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})\,d\boldsymbol{\theta}} = \frac{\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})}{\int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}$$

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{y})}[g(\boldsymbol{\theta})] = \int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta} = \frac{\int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}{\int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}$$

p̃(θ|y) is usually a well-behaved function, we can often approximate

$$Z(\mathbf{y}) = \int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta} \approx \sum_{i=1}^{N} \widetilde{p}(\boldsymbol{\theta}_i|\mathbf{y})\,w_i \quad \text{and similarly} \quad \int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta} \approx \sum_{i=1}^{N} g(\boldsymbol{\theta}_i)\widetilde{p}(\boldsymbol{\theta}_i|\mathbf{y})\,w_i$$

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{y})}[g(\boldsymbol{\theta})] = \frac{\int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}{\int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}} \approx \frac{\sum_{i=1}^{N} g(\boldsymbol{\theta}_i)\widetilde{p}(\boldsymbol{\theta}_i|\mathbf{y})\,w_i}{\sum_{i=1}^{N} \widetilde{p}(\boldsymbol{\theta}_i|\mathbf{y})\,w_i}$$

deterministic strategies

numerical integration

Numerical integration = use numerical analysis to approximate integrals

Typically a linear combination of the values of l(θ|y)p(θ) at points in domain of integration

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{l(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\Theta}} l(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})\,d\boldsymbol{\theta}} = \frac{\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})}{\int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}$$

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{y})}[g(\boldsymbol{\theta})] = \int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta} = \frac{\int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}{\int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta}}$$

p̃(θ|y) is usually a well-behaved function, we can often approximate

$$Z(\mathbf{y}) = \int_{\boldsymbol{\Theta}} \widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta} \approx \sum_{i=1}^{N} \widetilde{p}(\boldsymbol{\theta}_i|\mathbf{y})\,w_i \quad \text{and similarly} \quad \int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})\widetilde{p}(\boldsymbol{\theta}|\mathbf{y})\,d\boldsymbol{\theta} \approx \sum_{i=1}^{N} g(\boldsymbol{\theta}_i)\widetilde{p}(\boldsymbol{\theta}_i|\mathbf{y})\,w_i$$

The quality of the approximations are gauged purely by our computational power (N)

Limitation: Doesn't scale well into high dimensions; naive gridding is very inefficient

stochastic strategies

Often preferable to have draws from the distribution than the distribution itself

Law of large numbers (LLN): If $X_1, \ldots, X_n \sim p(x)$ with mean $E[X]$, $\bar{X}_n \rightarrow E[X]$

Applies to transformations $g(X_1), \ldots, g(X_n)$ to approximate $\overline{g(X)}_n \approx E[g(X)]$

Central limit theorem (CLT) can provide probabilistic bounds on how close

Swapping one hard problem for another

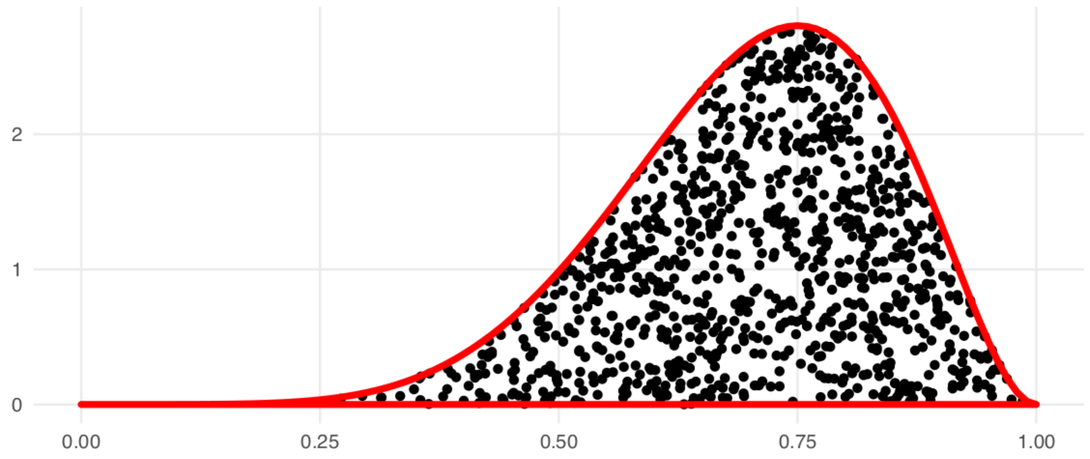
Integrating to compute expectations Sampling to compute expectations

Generating draws $\theta_1, \ldots, \theta_N \sim p(\theta|y)$ allows us to CTP $\overline{g(\theta)}_N \approx E_{p(\theta|y)}[g(\theta)]$

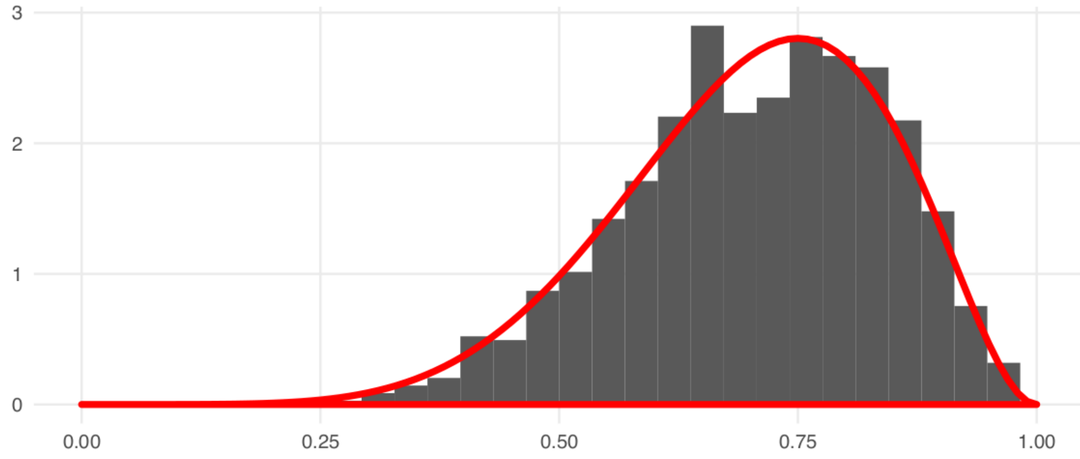
stochastic strategies

rejection sampling

Fundamental theorem of simulation: Sampling from the uniform distribution on the volume under the graph of a density p(x) is equivalent to sampling p(x)



Marginalize out height



More: the scale of the density doesn't matter; sampling under an un-normalized version of the density $\tilde{p}(x)$ is sufficient

stochastic strategies

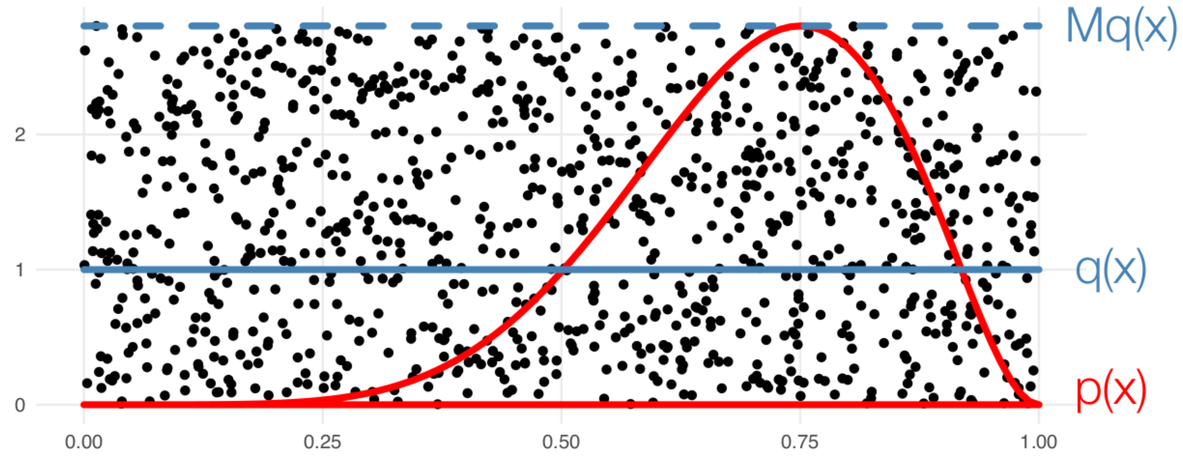
rejection sampling

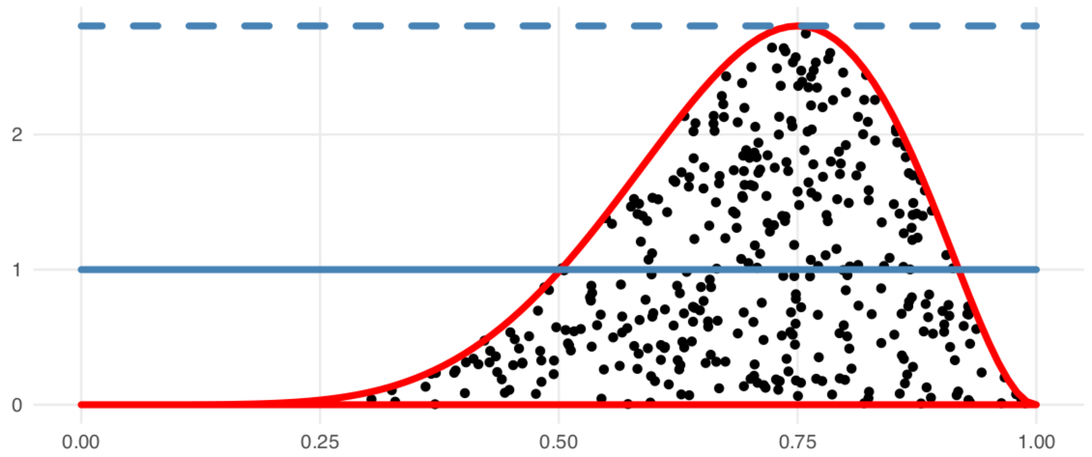
Rejection sampling: To sample from target p(x), sample from proposal q(x) that

(1) can be scaled to be everywhere larger than p(x); $p(x) \leq Mq(x)$ for all x, and

(2) can be evaluated, so that p(x) and q(x) can be determined for all x.

Then, sample $x' \sim q(x)$ and $u \sim \text{Unif}(0, Mq(x'))$. If $u \leq p(x')$, $x' \sim p(x)$.



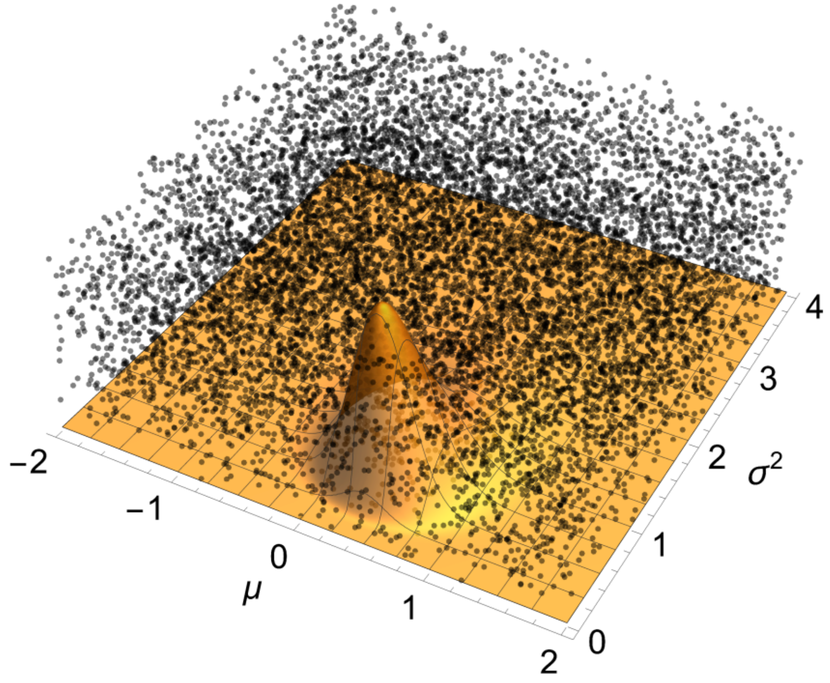


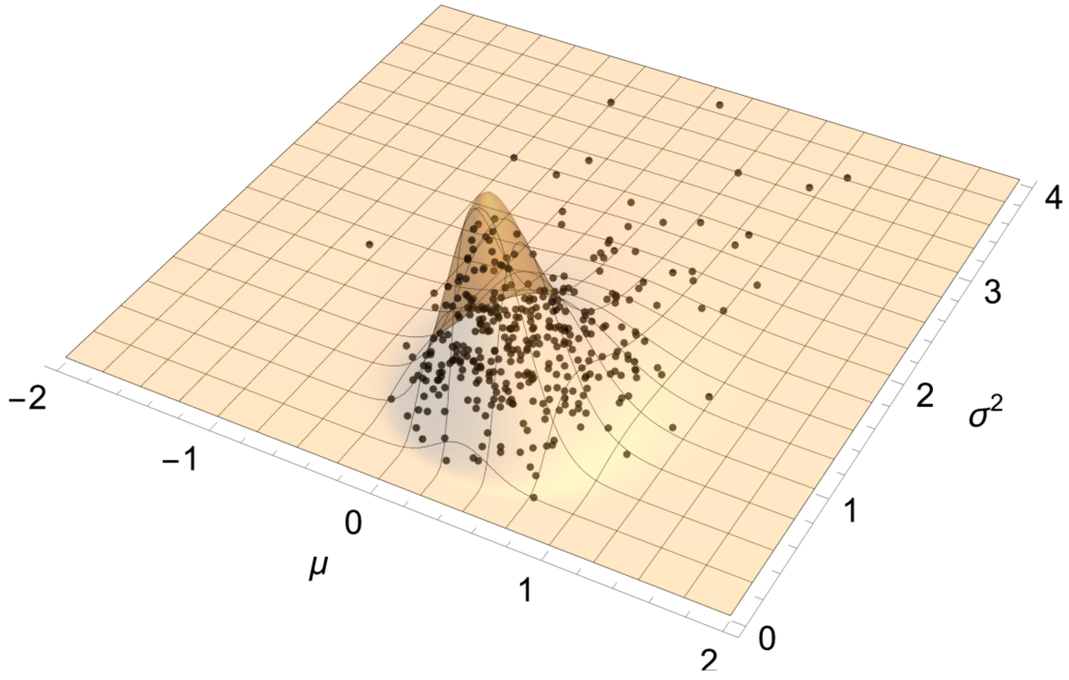
stochastic strategies

rejection sampling

Rejection sampling: To sample from target p(x), sample from proposal q(x) that

Then, sample x' ~ q(x) and u ~ Unif(0, Mq(x')). If u ≤ p(x'), x' ~ p(x).



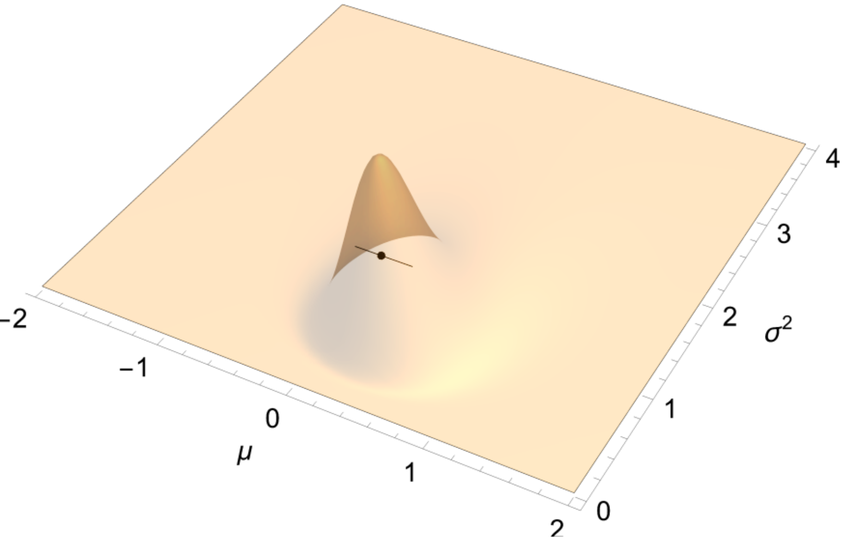


Limitation: Doesn't scale well into high dimensions; hard to get similar distribution

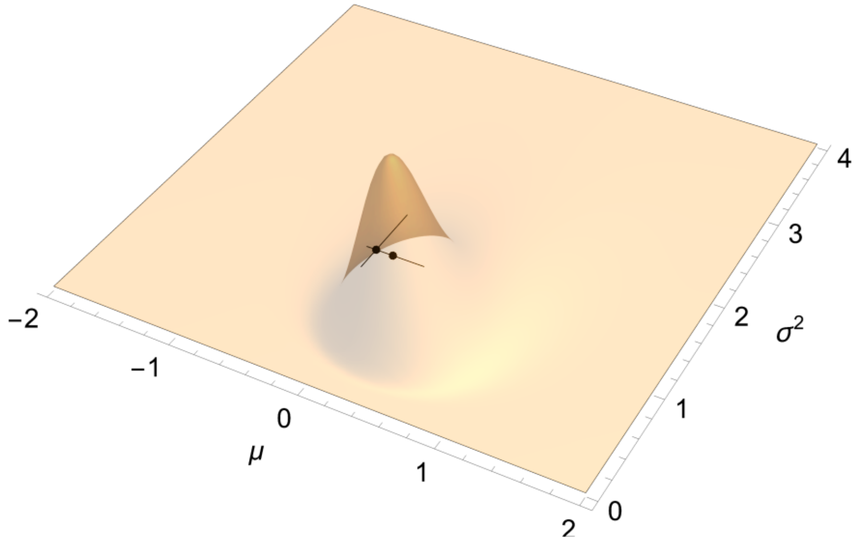
stochastic strategies

slice sampling

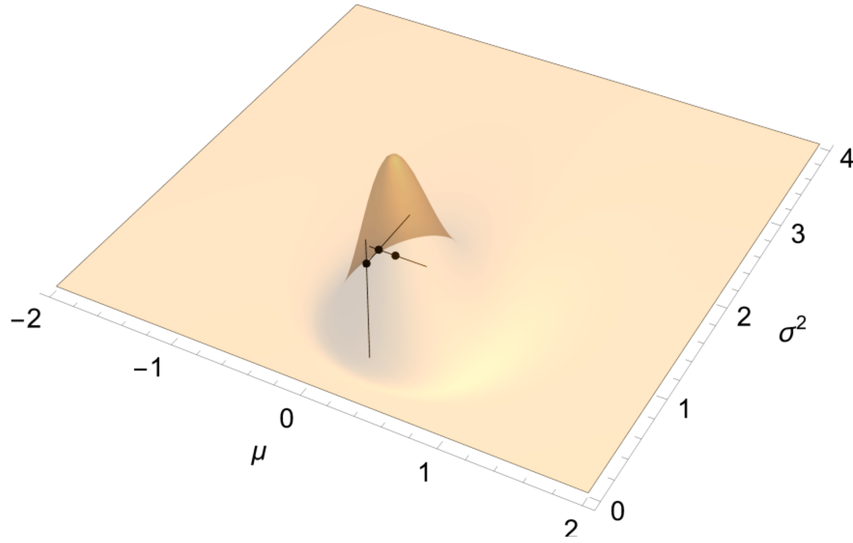
Slice sampling: starting from an arbitrary point under the surface, iteratively sample from the uniform distribution in each of the coordinates



Sample [μ|σ²,u]



Sample [σ²|μ,u]



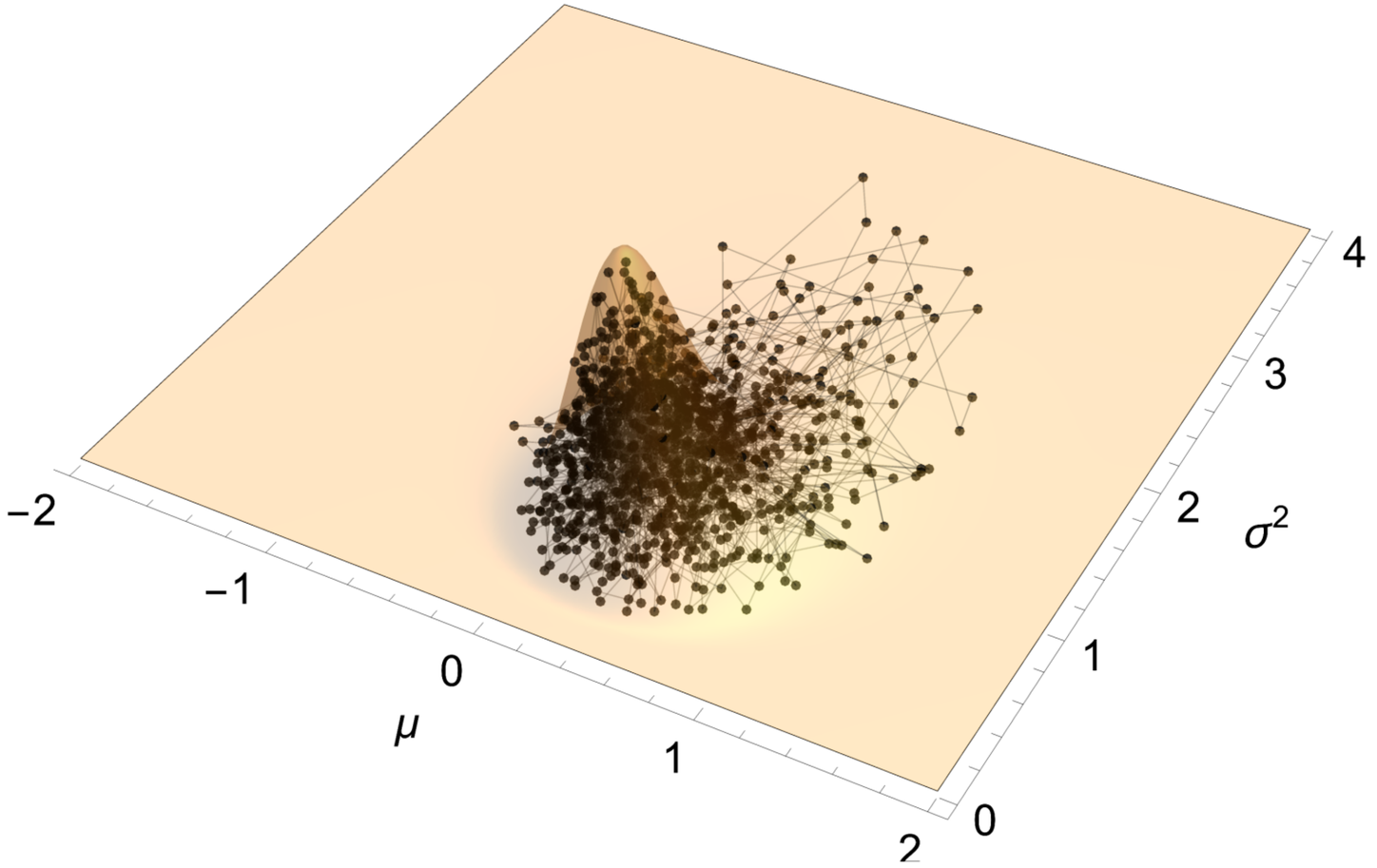
Sample [u|μ,σ²]

Sample (μ, σ², u), (μ, σ², u), ..., after a while, you're equally likely to be found anywhere under $\tilde{p}(\mu,\sigma^2|y)$

stochastic strategies

slice sampling

Slice sampling: starting from an arbitrary point under the surface, iteratively sample from the uniform distribution in each of the coordinates



stochastic strategies

Gibbs sampling

In the long run it doesn't matter whether you sample (μ, σ^2, u), (μ, σ^2, u), ...
or (μ, u), (μ, u), ..., (σ^2, u), (σ^2, u), ..., (μ, u), (μ, u), ..., (σ^2, u), (σ^2, u), ...

The resulting draws will still be equally likely to be found anywhere under the surface

But drawing (μ, u), (μ, u), ... simply keeps σ^2 fixed, and since we ultimately don't want u, this is equivalent to taking a draw $[\mu|\sigma^2]$

Similarly, (σ2, u), (σ2, u), ... is equivalent to drawing $[\sigma^2|\mu]$

Gibbs sampling = sample multivariate distributions by iteratively sampling the *full conditional* distributions, each variate given all others

Any method to sample those is valid, e.g. inverse transform and (adaptive) rejection

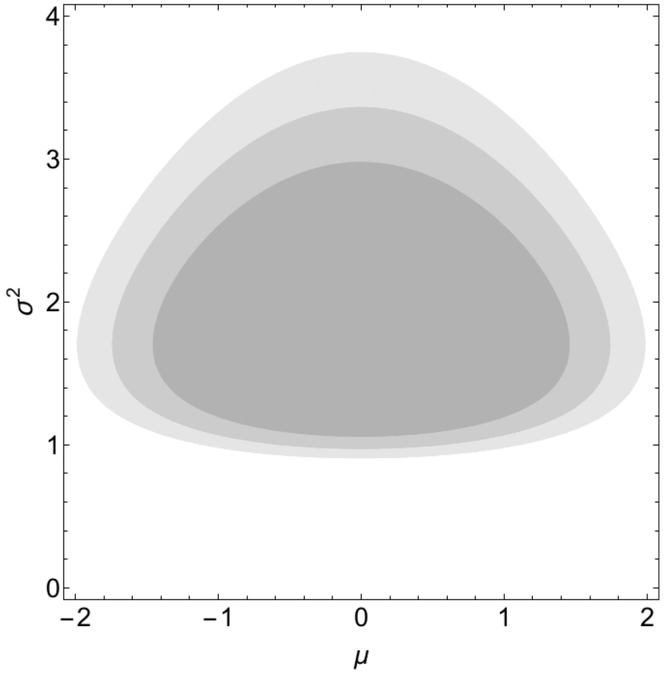
Sampling full conditionals is often very simple, even if they are not recognizable

Limitation: For distributions not aligned with axes, explores distribution slowly

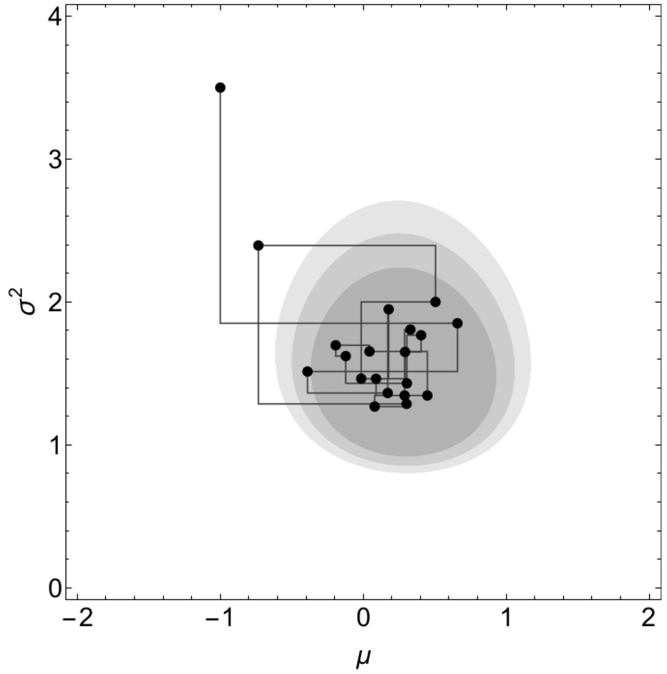
stochastic strategies

Gibbs sampling

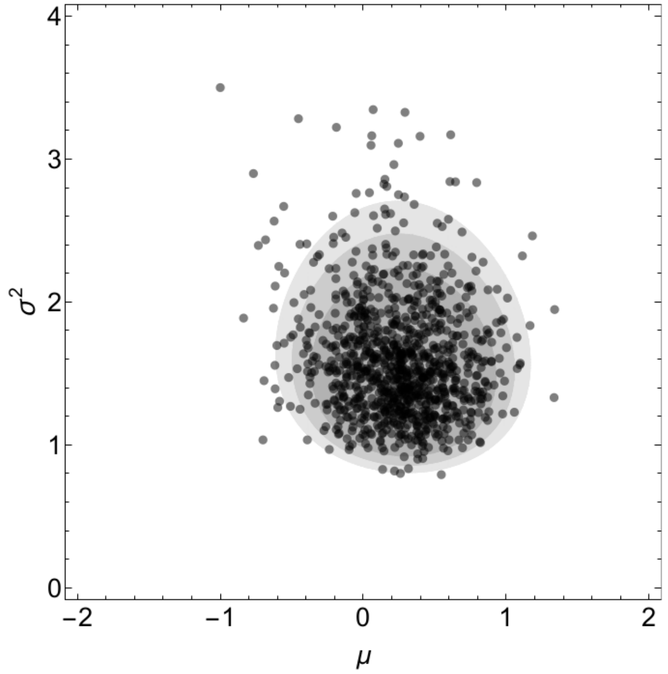
Normal example: if independent (Normal)(Inv-Gamma) priors are placed on (μ,σ^2), $[\mu|\sigma^2]$ is normal and $[\sigma^2|\mu]$ is Inv-Gamma, sampling is simple



$p(\mu,\sigma^2)$



$p(\mu,\sigma^2|y)$, 20 draws



$p(\mu,\sigma^2|y)$, 1k draws

stochastic strategies

Markov Chain Monte Carlo

The slice sampler and Gibbs sampler are members of a larger class of algorithms called *Markov chain Monte Carlo (MCMC) algorithms*

Like the rejection sampler, typically have a proposal step and an acceptance step, but the acceptance step is probabilistic, not just 0-1

The probability is called the *Metropolis-Hastings (acceptance) probability*

Intuition: if the random step is into a region of higher probability, it should be accepted.
If the random step is into a region of low probability, it should probably be rejected.

stochastic strategies

Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) is a MCMC algorithm that choses proposal steps via a physics simulation of a marble sliding around on a glass surface

The height of the surface is – log $\tilde{p}(\theta|y)$; the momentum of the marble is randomly generated (a "random flick")

The position of the marble is tracked for a time, stopped, and its state is proposed

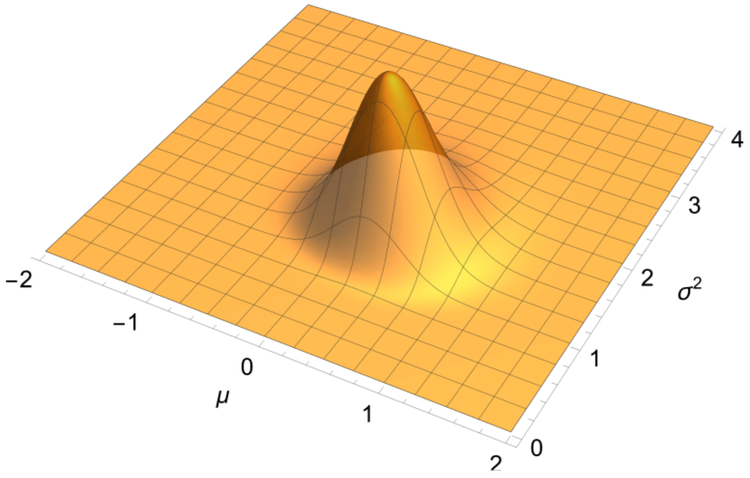
Marble is more likely to be in lower regions on the surface = higher probability regions

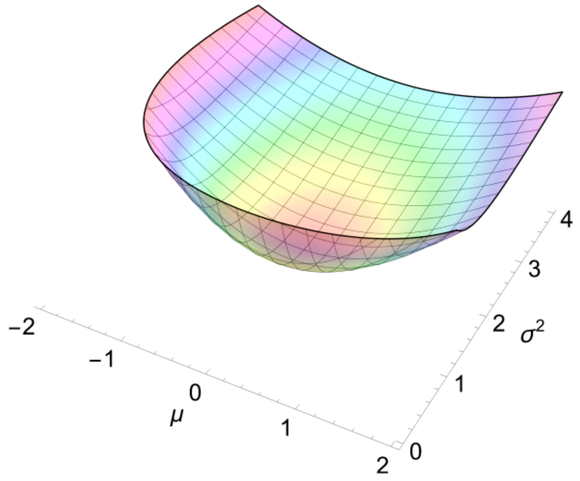
No-U-Turn sampler (NUTS) = HMC variant that helps automatically choose how long to run the physics simulation and how to discretize it

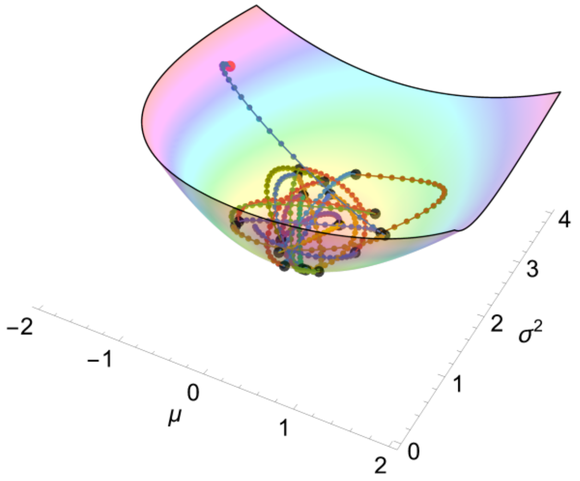
stochastic strategies

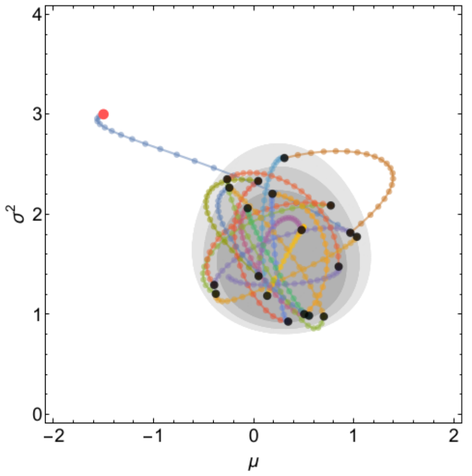
Hamiltonian Monte Carlo

Hamiltonian Monte Carlo is a MCMC algorithm that choses proposal steps via a physics simulation of a marble sliding around on a glass surface





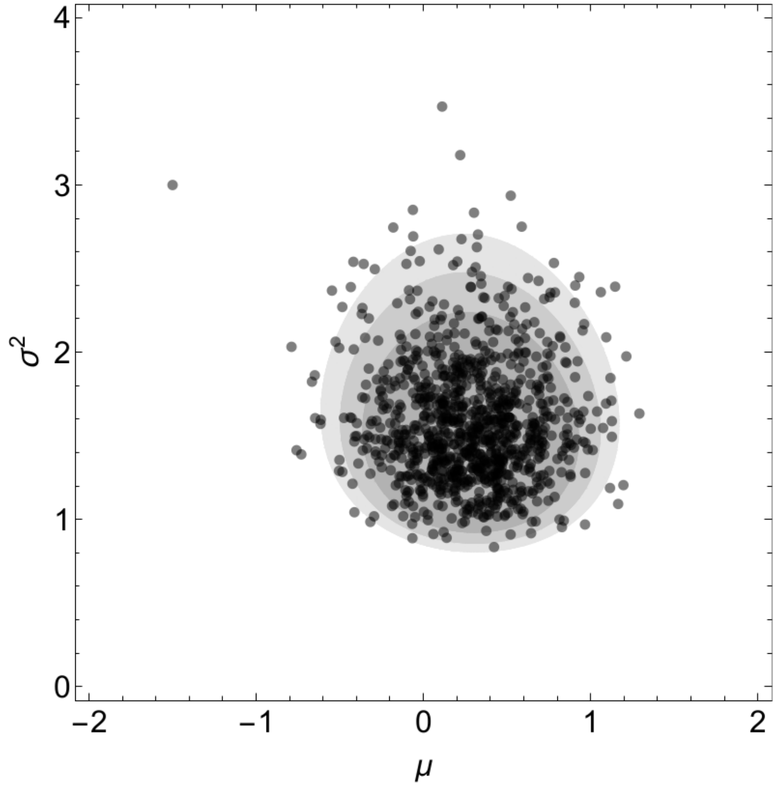




stochastic strategies

Hamiltonian Monte Carlo

Hamiltonian Monte Carlo is a MCMC algorithm that choses proposal steps via a physics simulation of a marble sliding around on a glass surface



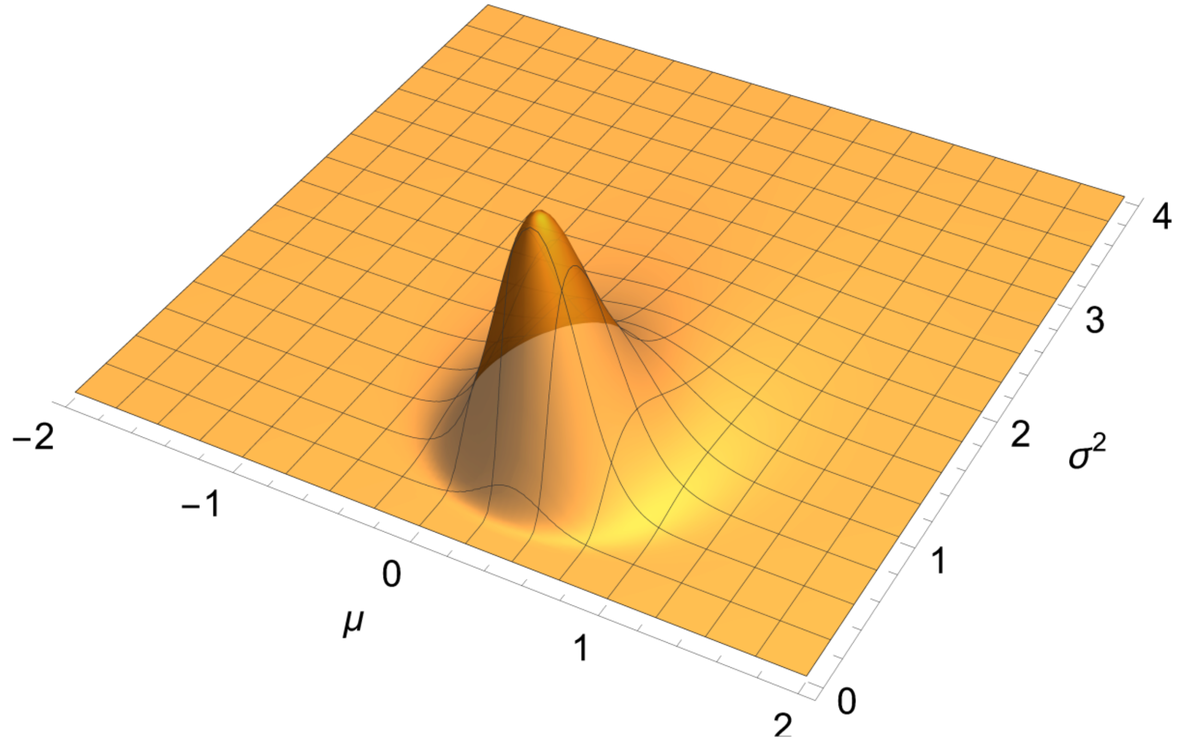
Limitation: HMC doesn't support discrete parameter spaces; requires computing gradients

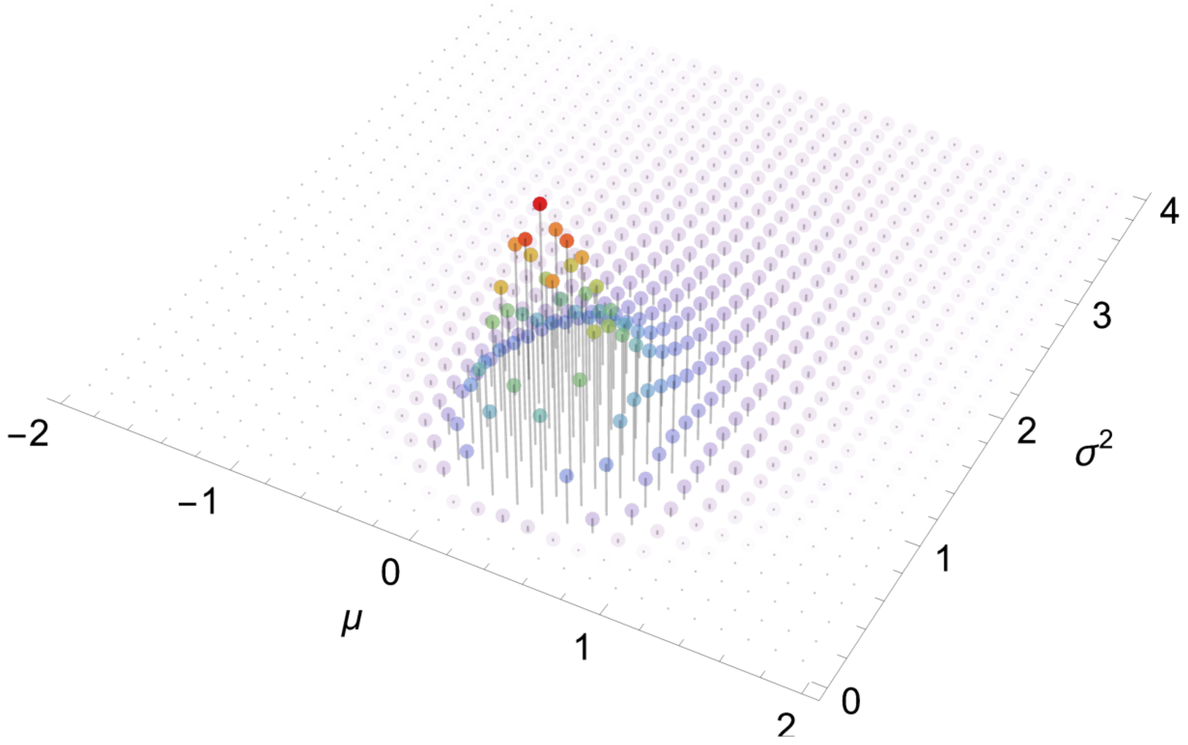
Exactly Computing an Approximate Posterior

discretization

Discretization is perhaps the simplest strategy to CTP

Very similar to naive numerical integration, but accommodates constraints better





Limitation: Doesn't scale well into high dimensions; naive gridding is very inefficient

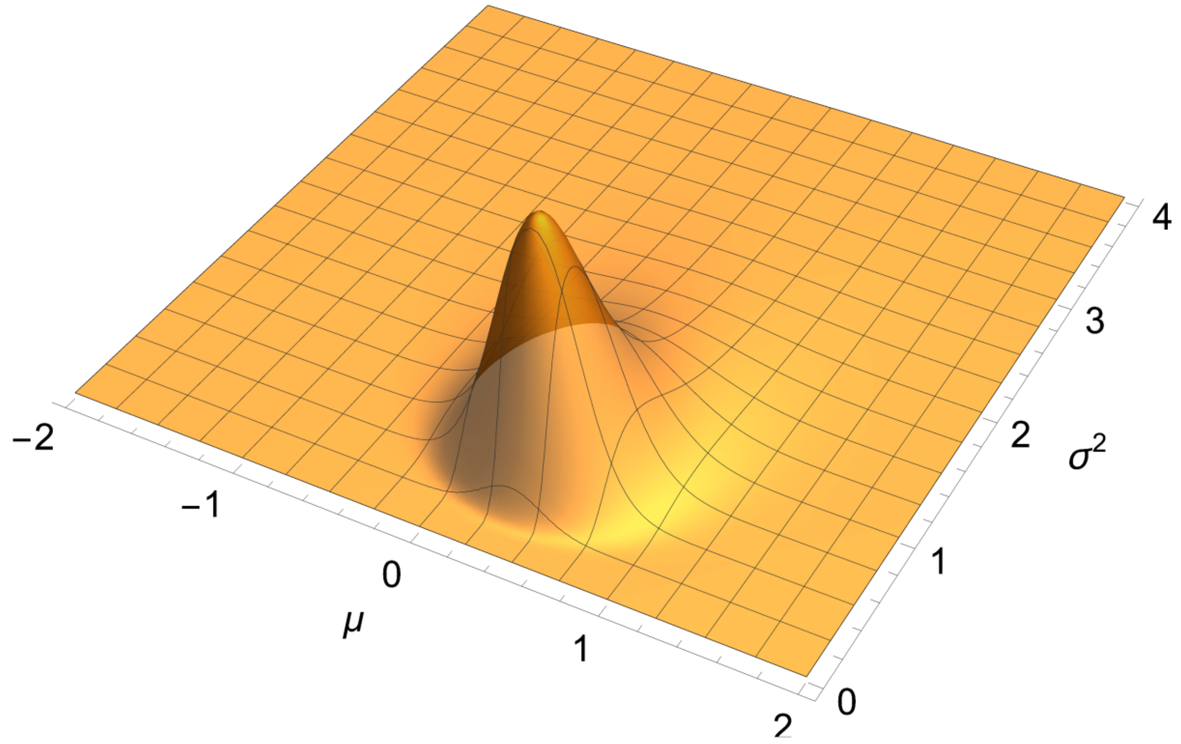
Laplace approximation

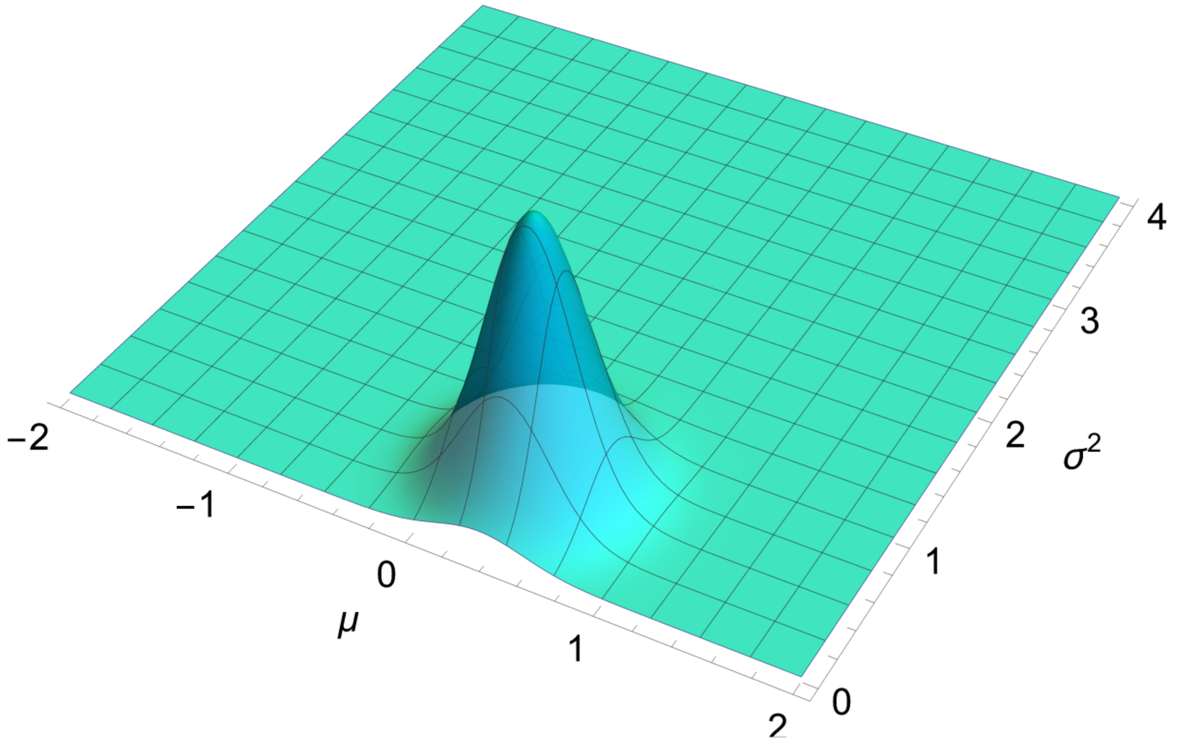
Laplace approximation: approximate the posterior with a multivariate normal

Mean = posterior mode

Covariance = inverse of observed Fisher information of the posterior

Hessian of log posterior evaluated at post. mode





Limitation: Approximation won't capture non-normal posterior features, e.g. asymmetry

Software Tools for Bayesian Computation

probabilistic programming languages

Probabilistic programming language (PPL) = A simple specification language for Bayesian models along with an engine that automates sampling

BUGS (Bayesian inference using Gibbs sampling) variants

Classic – 1989 – written in Modula-2 and ran on Windows and some Unix

Gibbs sampling using only inverse transform sampling, conjugate sampling, and adaptive rejection sampling

WinBUGS (1997), OpenBUGS (2005) – written in Component Pascal, increased modularity, open licensing, sampling methods (e.g. slice)

No longer maintained

JAGS (early 2000's) – self-described WinBUGS clone in C++, very modular, **rjags**, **runjags**

Appears to be in maintenance-only mode

Nimble (2017), MultiBUGS (2020) – Modern state-of-art BUGS engines, parallelism

Stan = NUTS-HMC sampling; models down compiled into C++ to run

Large user base, RStudio support, substantial documentation

other R abstractions

INLA = R package for integrated nested Laplace approximation for hierarchical models

Not available on CRAN, but easy to install

brms,**rstanarm** = R package for hierarchical modeling with Stan using **lme4**'s formula specification language that extends base R

Feels a lot like base R **stats** functions such as glm()

Many other packages, see the Bayesian statistics CRAN task view

Takeaways

rear-view of talk

Computation is a central bottleneck in Bayesian statistics due to the intractability of the posterior distribution

Approximately compute the exact posterior

Deterministic approaches

Symbolic methods

Conjugate priors

Numerical integration (quadrature)

Stochastic approaches

Rejection sampling

MCMC – Slice sampling, Gibbs sampling, MH algorithm

MCMC – Hamiltonian Monte Carlo

Exact computations on approximate posteriors

Discretization

Laplace approximation

Software Tools for Bayesian Computation

BUGS (Classic/Win/Open/Multi), JAGS, NIMBLE, Stan, brms/rstanarm