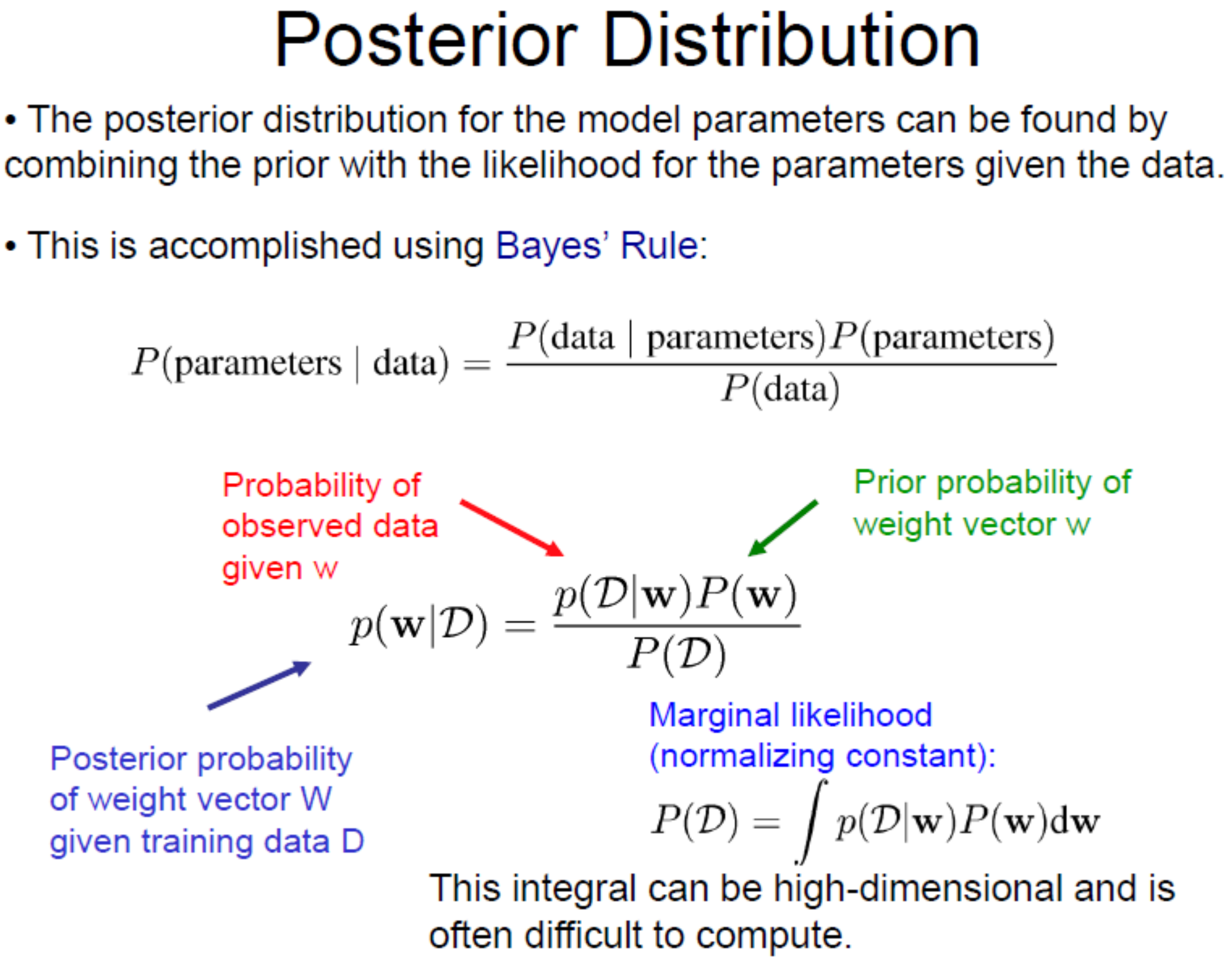
Lecture 12: Approximate inference

STA414, 2 April 2018

Based on slides by
Profs Michael Osborne, Stephen Roberts, & Richard Zemel

Recall this slide from Lecture 4, Bayesian methods:



Posterior Distribution

- The posterior distribution for the model parameters can be found by combining the prior with the likelihood for the parameters given the data.

- This is accomplished using Bayes' Rule:

$$P(\text{parameters} \mid \text{data}) = \frac{P(\text{data} \mid \text{parameters}) P(\text{parameters})}{P(\text{data})}$$

Probability of observed data given w

Prior probability of weight vector w

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})}$$

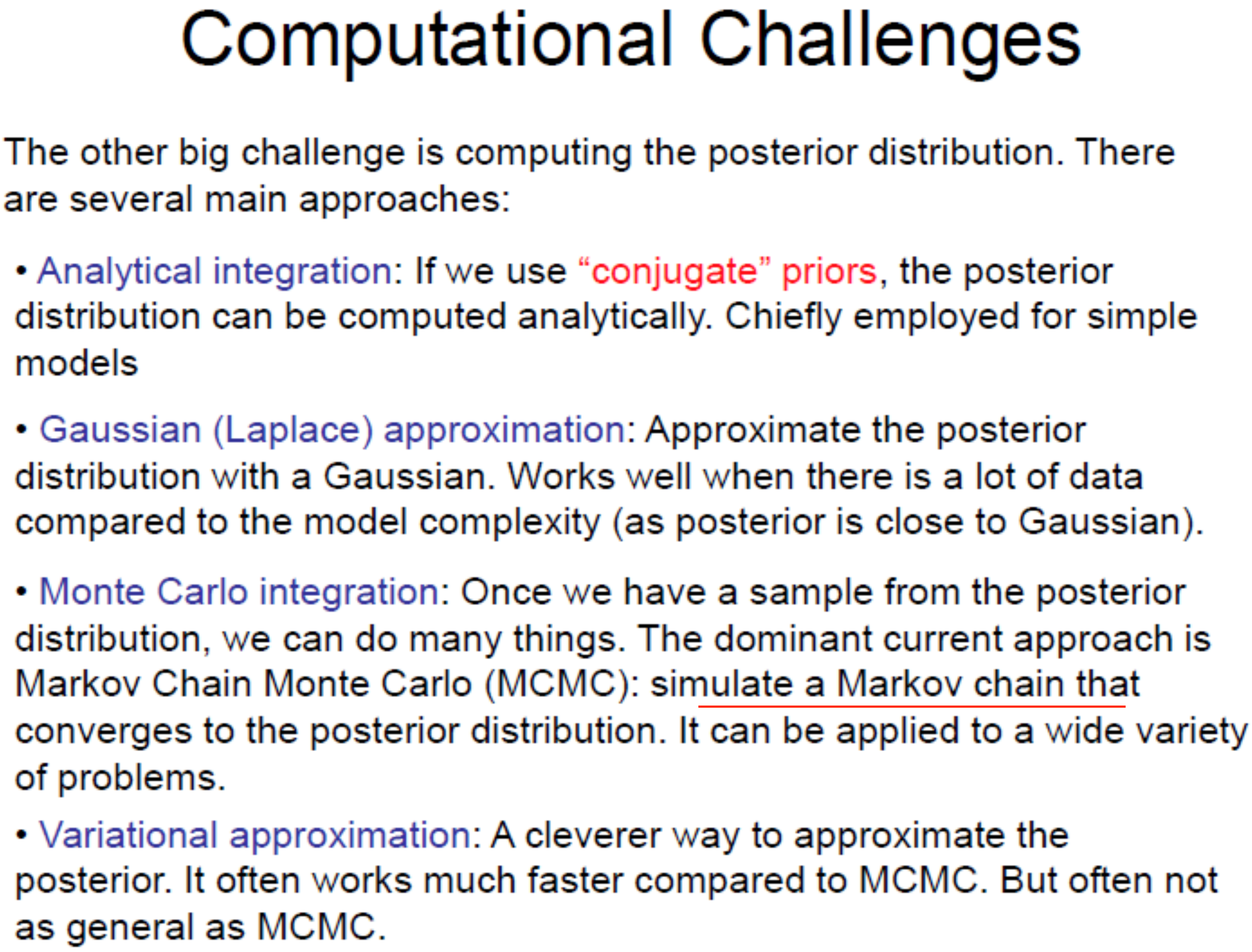
Posterior probability of weight vector W given training data D

Marginal likelihood (normalizing constant):

$$P(\mathcal{D}) = \int p(\mathcal{D}|\mathbf{w})P(\mathbf{w})\mathrm{d}\mathbf{w}$$

This integral can be high-dimensional and is often difficult to compute.

Recall this slide from Lecture 4, Bayesian methods:

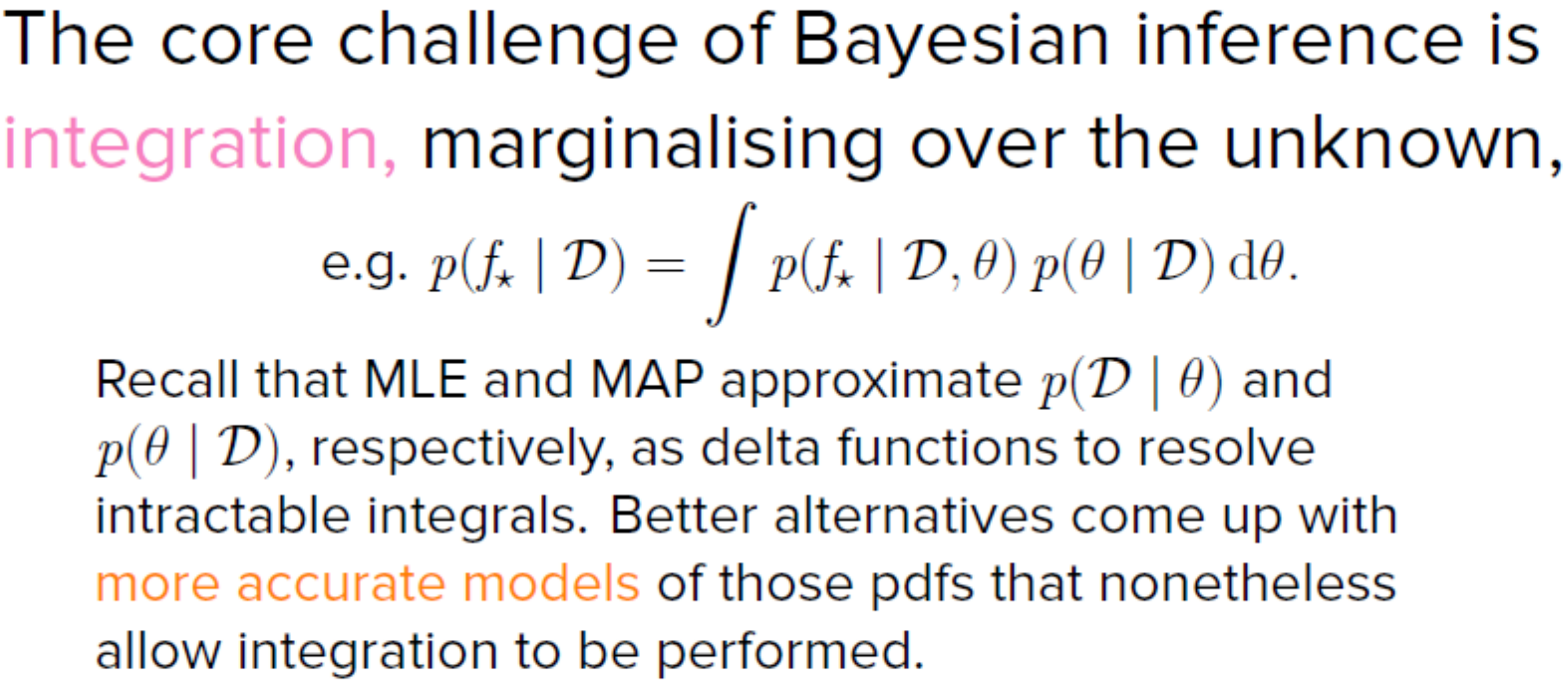


Computational Challenges

The other big challenge is computing the posterior distribution. There are several main approaches:

- Analytical integration: If we use "conjugate" priors, the posterior distribution can be computed analytically. Chiefly employed for simple models
- Gaussian (Laplace) approximation: Approximate the posterior distribution with a Gaussian. Works well when there is a lot of data compared to the model complexity (as posterior is close to Gaussian).
- Monte Carlo integration: Once we have a sample from the posterior distribution, we can do many things. The dominant current approach is Markov Chain Monte Carlo (MCMC): simulate a Markov chain that converges to the posterior distribution. It can be applied to a wide variety of problems.
- Variational approximation: A cleverer way to approximate the posterior. It often works much faster compared to MCMC. But often not as general as MCMC.

Computational Challenges:



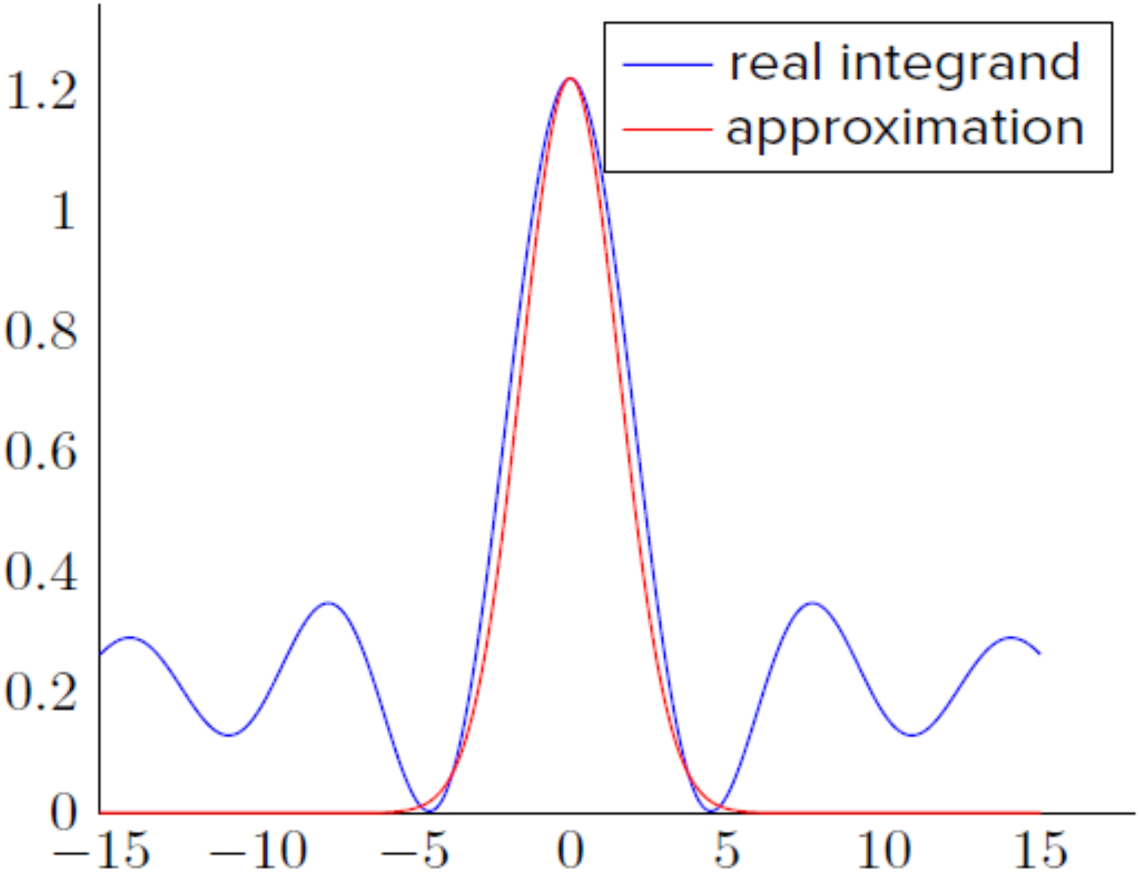
The core challenge of Bayesian inference is integration, marginalising over the unknown,

$$\text{e.g. } p(f_\star \mid \mathcal{D}) = \int p(f_\star \mid \mathcal{D}, \theta)\, p(\theta \mid \mathcal{D})\, \mathrm{d}\theta.$$

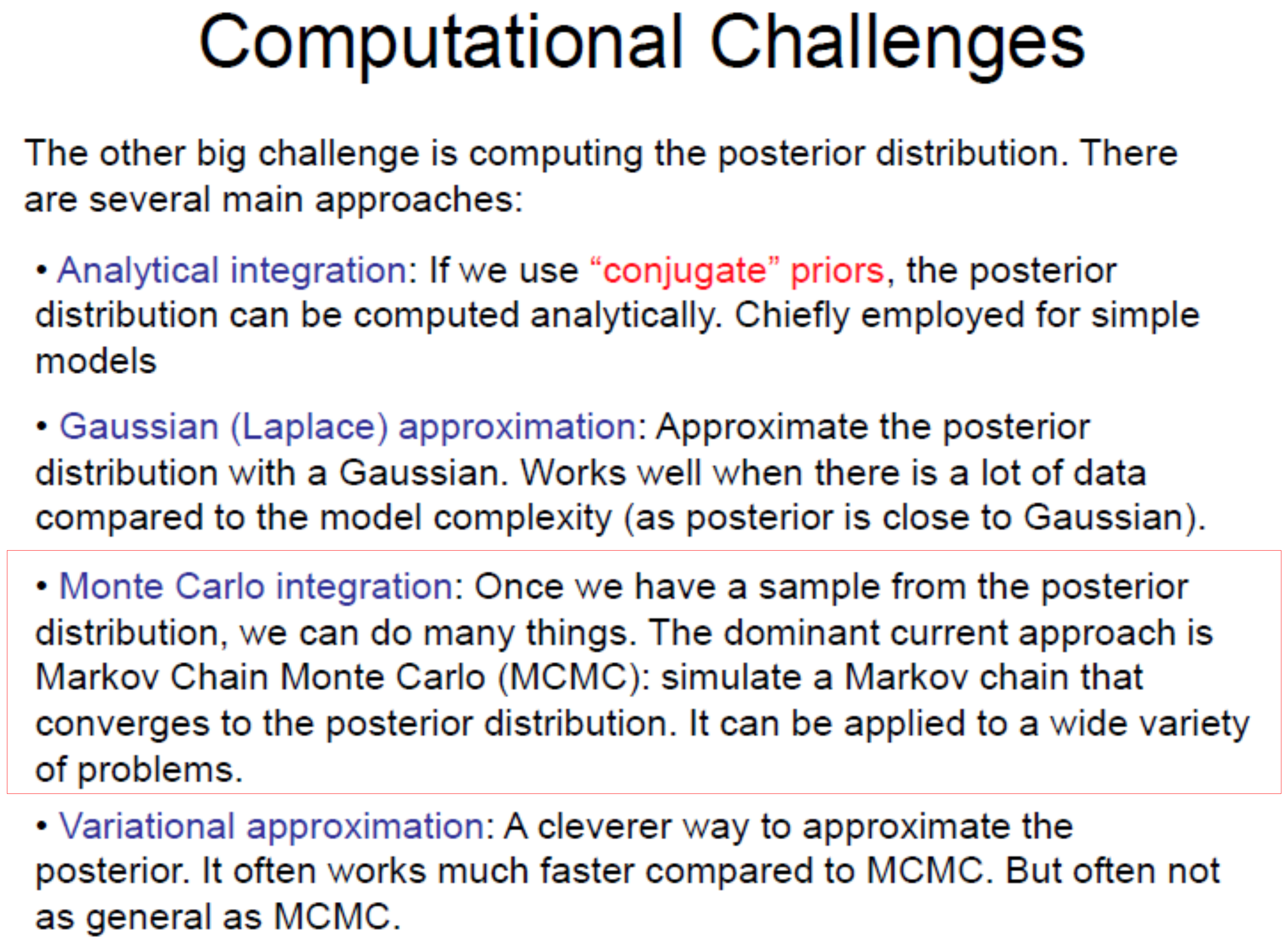
Recall that MLE and MAP approximate $p(\mathcal{D} \mid \theta)$ and $p(\theta \mid \mathcal{D})$, respectively, as delta functions to resolve intractable integrals. Better alternatives come up with more accurate models of those pdfs that nonetheless allow integration to be performed.

One way to improve upon MAP and MLE is the Laplace approximation.

This approach fits an un-normalised Gaussian around the maximum of an integrand.



Recall this slide from Lecture 4, Bayesian methods:



Computational Challenges

The other big challenge is computing the posterior distribution. There are several main approaches:

- Analytical integration: If we use “conjugate” priors, the posterior distribution can be computed analytically. Chiefly employed for simple models
- Gaussian (Laplace) approximation: Approximate the posterior distribution with a Gaussian. Works well when there is a lot of data compared to the model complexity (as posterior is close to Gaussian).
- Monte Carlo integration: Once we have a sample from the posterior distribution, we can do many things. The dominant current approach is Markov Chain Monte Carlo (MCMC): simulate a Markov chain that converges to the posterior distribution. It can be applied to a wide variety of problems.
- Variational approximation: A cleverer way to approximate the posterior. It often works much faster compared to MCMC. But often not as general as MCMC.

Sampling (Monte Carlo methods)

- Useful in many settings, including:
  - Numerical integration
  - Function approximation
  - etc

You may recall from second year: **sampling** from distributions

How to convert samples from a Uniform[0,1] generator?

$$h(y) = \int_{-\infty}^{y} p(y')dy'$$

sample *u* ~ Uniform[0,1]

$$y(u) = h^{-1}(u)$$

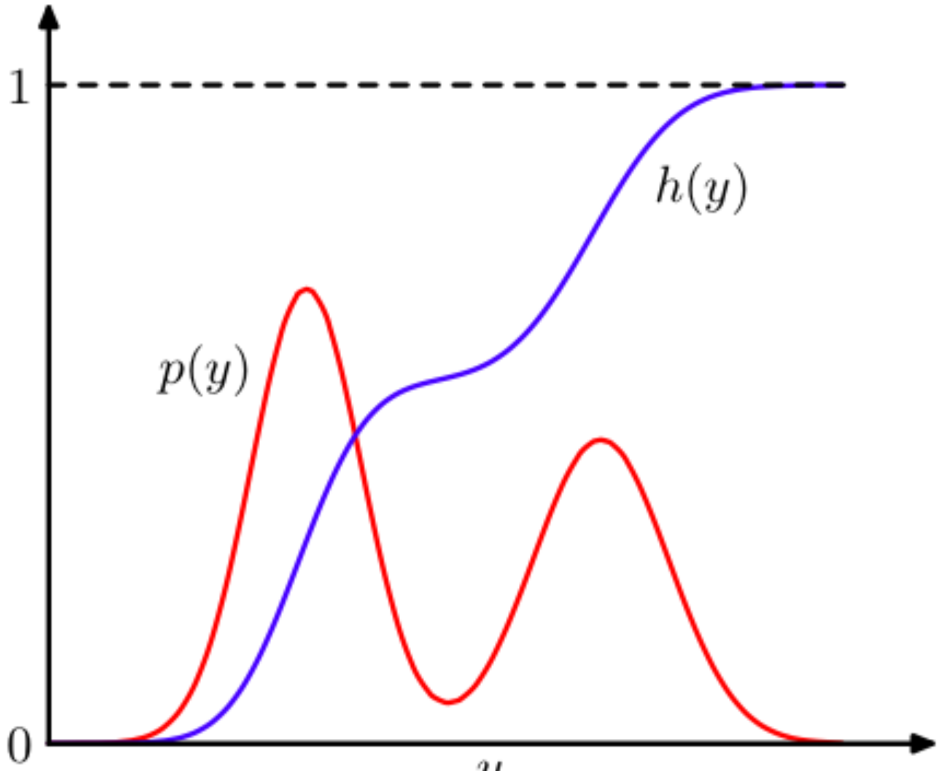
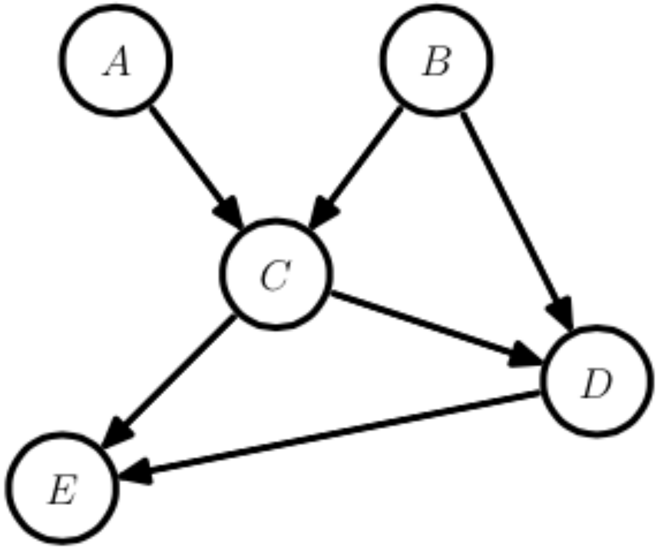


Figure from PRML, Bishop (2006)

But in many cases it's difficult to invert *h(y)*

Recall from week 9: **sampling** from a DGM

- *Ancestral pass* for directed graphical model
  - Sample each top level variable from its marginal
  - Sample each other node from its conditional once its parents have been sampled



Sample:

$$A \sim P(A)$$
$$B \sim P(B)$$
$$C \sim P(C \,|\, A, B)$$
$$D \sim P(D \,|\, B, C)$$
$$E \sim P(D \,|\, C, D)$$

$$P(A, B, C, D, E) = P(A)\, P(B)\, P(C \,|\, A, B)\, P(D \,|\, B, C)\, P(E \,|\, C, D)$$

Monte Carlo methods aim to solve the core problem in Bayesian inference.

That is, it estimates the expected value, $\mathbb{E}[a]$, of some function $a(\theta)$ over some density $p(\theta)$.

We assume that we can evaluate $a(\theta)$ for any given θ; however, we cannot usually evaluate $p(\theta)$ directly.

Instead, we have access only to $f(\theta) = {}^{p(\theta)}/_{c}$ for an unknown (normalising) constant c.

Hence the problem is to find

$$\mathbb{E}[a] = \frac{\int a(\theta) f(\theta) \mathrm{d}\theta}{\int f(\theta) \mathrm{d}\theta}$$

= 1 if f is a pdf. \int f(\theta) = 1

given $f(\cdot)$ and $a(\cdot)$ (black box functions that we can evaluate at points of our choice).

Monte Carlo uses a simple approximation for the integral.

Monte Carlo schemes generate a set of samples

$$\{\theta_i; i = 1, \ldots, N\}$$

from $p(\theta)$ (in a variety of ways.)

We evaluate $a(\cdot)$ and $f(\cdot)$ at those samples, giving $\{a(\theta_i); i = 1, \ldots, N\}$ and $\{f(\theta_i); i = 1, \ldots, N\}$, respectively.

We then approximate as

expectedf mean

$$\boxed{\mathbb{E}[a] \simeq \sum_{i=1}^{N} a(\theta_i)/N}$$

Monte Carlo methods aim to solve the core problem in Bayesian inference.

The problem of finding

$$p(f_\star \mid \mathcal{D}) = \int p(f_\star \mid \mathcal{D}, \theta)\, p(\theta \mid \mathcal{D})\, \mathrm{d}\theta.$$

turns into

$$\mathbb{E}[a] = \frac{\int a(\theta) f(\theta) \mathrm{d}\theta}{\int f(\theta) \mathrm{d}\theta}$$

for

1. $a(\theta) = p(f_\star \mid \mathcal{D}, \theta)$, the predictions, and
2. $f(\theta) = p(\mathcal{D} \mid \theta) p(\theta)$, the product of prior and likelihood of parameters, as

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta) p(\theta)}{\int p(\mathcal{D} \mid \theta) p(\theta) \mathrm{d}\theta}.$$

Properties of Simple Monte Carlo

- Estimator:

x are samples

f=a P=f

$$\int f(x)P(x)dx \approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}) \quad x^{(s)} \sim P(x)$$

- Estimator is unbiased

$$\mathbb{E}_{P(x^{(s)})}[\hat{f}] = \frac{1}{S}\sum_{s=1}^{S} \mathbb{E}_{P(x)}[f(x)] = \mathbb{E}_{P(x)}[f(x)]$$

- Variance shrinks $\propto 1/S$:

$$\mathrm{var}_{P(x^{(s)})}[\hat{f}] = \frac{1}{S^2}\sum_{s=1}^{S} \mathrm{var}_{P(x)}[f(x)] = \mathrm{var}_{P(x)}[f(x)]/S$$

Approximate inference

MLE, MAP, & Laplace approximation

Monte Carlo methods:

- Simple Monte Carlo
- **Rejection sampling**
- Importance sampling *(optional)*
- Two examples of MCMC:
    - Gibbs sampling *(optional)*
    - Metropolis-Hastings

Variational inference, Bayesian quadrature, etc …
(optional)

Rejection sampling uses a distribution $g(\theta)$ from which it is easy to draw samples to then draw samples from $p(\theta)$.

For many standard distributions, like the Gaussian, there exist optimised procedures for drawing samples.

Let's say that we know that $f(\theta) \leq cg(\theta)$, where c is a constant and g is one of those standard distributions.

To generate θ_i, rejection sampling:

1. draws a sample ν from $g(\cdot)$;
2. draws u from the uniform distribution over $[0, 1]$;
3. if $u < \frac{f(\nu)}{cg(\nu)}$, set $\theta_i = \nu$ (accept);
4. otherwise, go back to step 1 (reject) and try again.

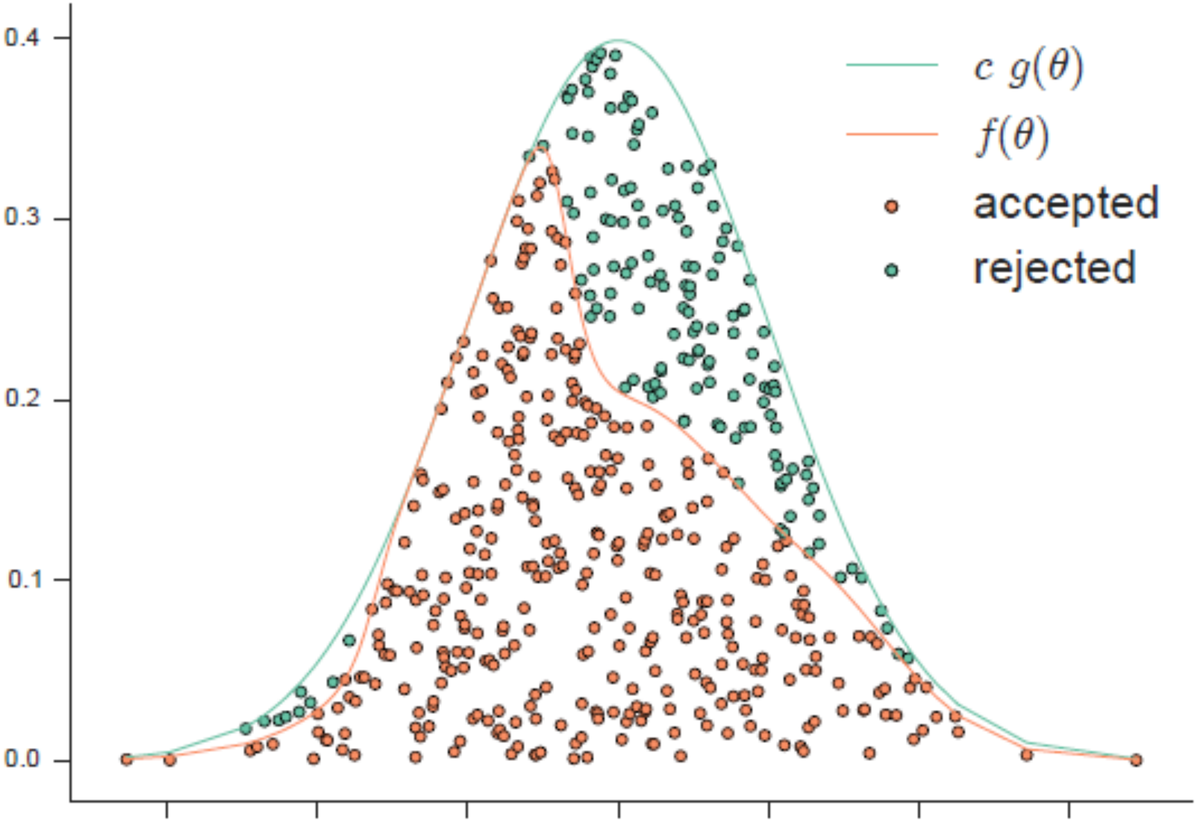
idea: v is randomly generated from g, u is a random point in the range [0, cg(v)]
reject sample v if f(v) <= u <= cg(v). Guarantees samples that is accepted follows f

1 draw a sample ν from $g(\cdot)$;

2 draw u from the uniform distribution over $[0, 1]$;

3 if $u < \frac{f(\nu)}{cg(\nu)}$, set $\theta_i = \nu$ (accept);

4 otherwise, go back to step 1 (reject) and try again.



Rejection sampling suffers from some practical difficulties.

Firstly, if the bound is not tight (if $f(\theta)$ is much lower than $cg(\theta)$), most samples will be wasted.

Secondly, it requires knowledge of the upper bounding factor c: usually we don’t know where the peaks of $f(\theta)$ are.

Rejection sampling is rarely useful for real, high-dimensional, problems, as the rate of accepting a sample reduces exponentially in the dimension of the space.

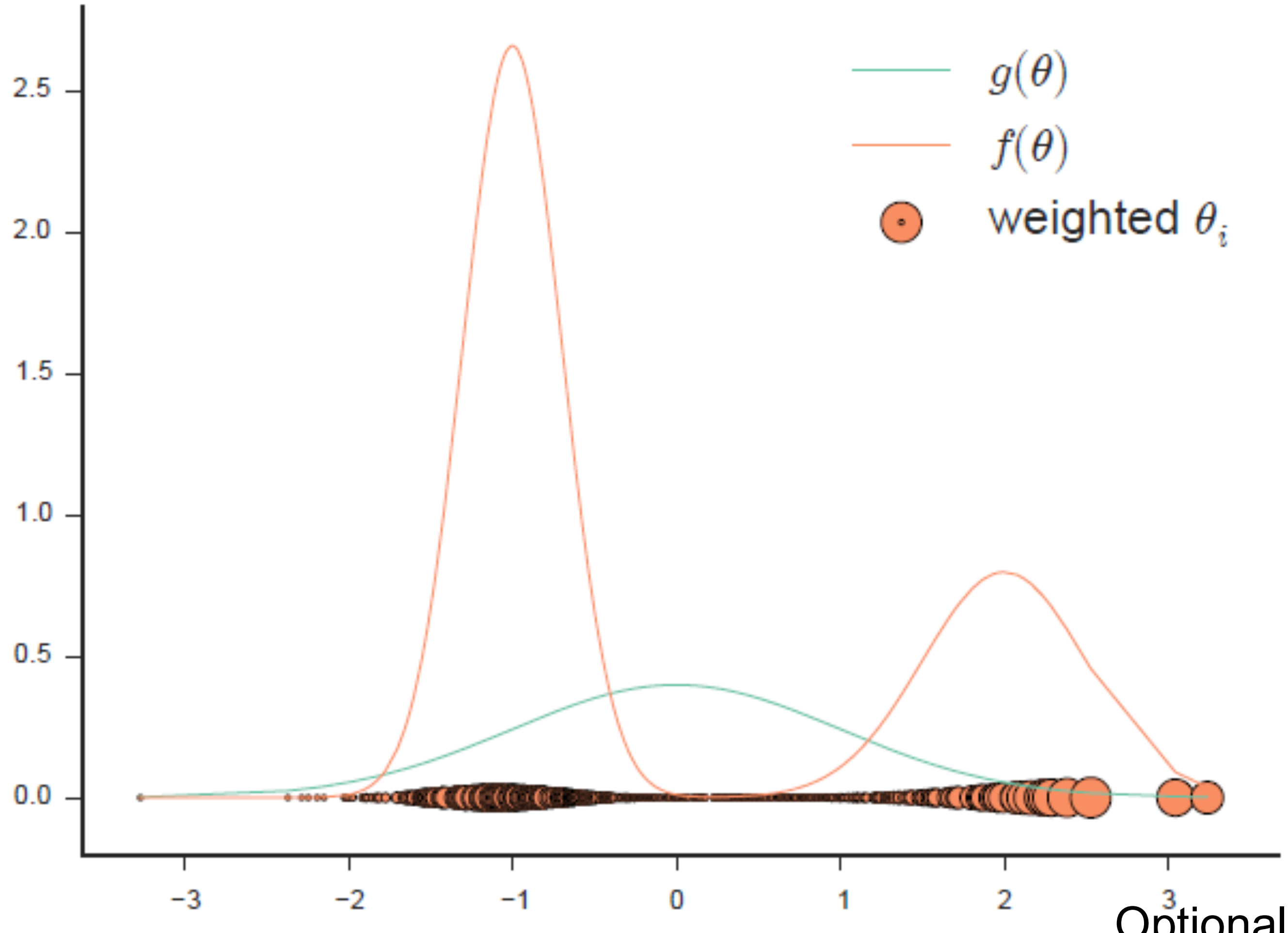
Importance sampling also uses a proposal distribution $g(\theta)$ from which it is easy to draw samples.

Importance sampling gives up trying to sample from $p(\theta)$ and instead simply draws $\{\theta_i; i = 1, \dots, N\}$ from $g(\theta)$.
We then approximate the expectation using

$$\begin{aligned}
\mathbb{E}[a] &= \frac{\int a(\theta) f(\theta) \mathrm{d}\theta}{\int f(\theta) \mathrm{d}\theta} \\
&= \frac{\int a(\theta) \frac{f(\theta)}{g(\theta)} g(\theta) \mathrm{d}\theta}{\int \frac{f(\theta)}{g(\theta)} g(\theta) \mathrm{d}\theta} \\
&\simeq \frac{\sum_{i=1}^{N} a(\theta_i) \frac{f(\theta_i)}{g(\theta_i)}}{\sum_{i=1}^{N} \frac{f(\theta_i)}{g(\theta_i)}}
\end{aligned}$$

Optional material

Importance sampling hence weights samples so that they appear more like draws from p.



Optional material

Importance sampling also suffers from practical difficulties.

Firstly, we must find a proposal g which is broad enough to cover all high values of f.

More importantly, if g is very dissimilar from f, we'll end up with mostly negligibly-weighted samples: not very useful.

Unfortunately, it's difficult to assess the similarity beforehand!

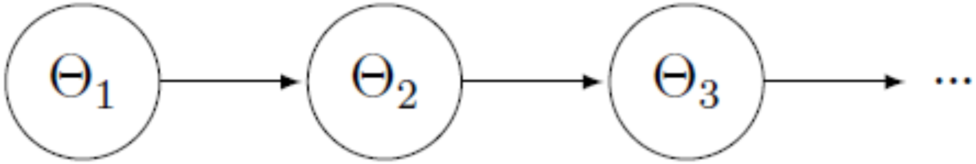
Optional material

Markov Chain Monte Carlo (MCMC) methods draw a sample conditional on the previous sample.

MCMC defines

1. an initial distribution $p(\Theta_1 = \theta_1)$ and
2. a transition density $p(\Theta_{i+1} = \theta_{i+1} \mid \Theta_i = \theta_i) = T_i(\theta_{i+1}; \theta_i)$ for drawing the sample θ_{i+1} given that the current sample is θ_i.

If T has no dependence on i, the chain is called homogenous.



Gibbs sampling is an MCMC scheme that sets $\theta_{i+1} = \theta_i$ and then changes a few elements.

This is particularly convenient when θ is very high-dimensional: typically, you may just change one of the elements of θ at a time.

Those elements that are changed are drawn from a specified distribution conditional on the previous value.

To change only the nth element, we often use

$$\boxed{T_i(\theta_{i+1}^{(n)}; \theta_i) = p(\theta_{i+1}^{(n)} \mid \theta_i^{(-n)})}$$

where $\theta^{(-n)}$ is all elements of θ aside from the nth.

Gibbs is hence useful when the conditionals of $p(\theta)$ have a nice form.

Optional material

Gibbs sampling changes a different set of variables at each iteration.

Usually, we just rotate through a pre-specified sequence of variables (e.g. change the first variable, followed by the second, etc.), but the sequence could also itself be determined randomly.

Gibbs is often slow to generate samples where $f(\theta)$ is large, as it can only explore the space with one small, axis-aligned, step at a time.

Optional material

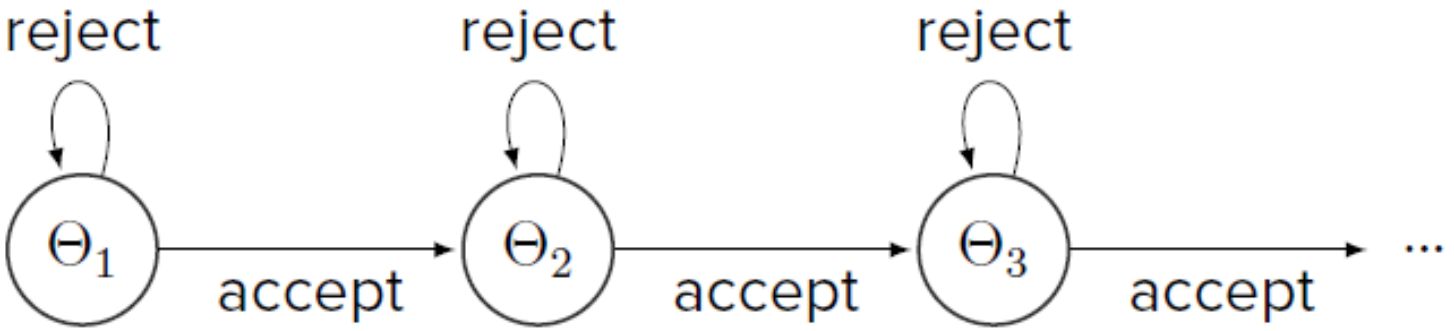
Metropolis-Hastings (MH) methods apply the idea of rejection to MCMC, taking a different proposal at every iteration.

To generate θ_{i+1}, Metropolis-Hastings:

1. draws a sample ν from a proposal distribution $Q(\cdot;\theta_i)$ (often as simple as a Gaussian centered at θ_i);
2. draws u from the uniform distribution over $[0,1]$;
3. if $u < \min\left(1, \frac{f(\nu)}{f(\theta_i)} \frac{Q(\theta_i;\nu)}{Q(\nu;\theta_i)}\right)$, we set $\theta_{i+1} = \nu$ (accept);

u is transition probability ?

4. else $\theta_{i+1} = \theta_i$ (reject).



Metropolis-Hastings accepts ν if and only if

$$u < \min\left(1, \frac{f(\nu)}{f(\theta_i)} \frac{Q(\theta_i; \nu)}{Q(\nu; \theta_i)}\right)$$

Metropolis-Hastings will tend to favour accepting samples for which:

1. $f(\nu) > f(\theta_i)$ i.e. where we've climbed to a higher value of f; and
2. $Q(\theta_i; \nu) > Q(\nu; \theta_i)$ i.e. where it's easier to get back to θ_i from ν than it is to have got to ν in the first place (so that we don't get stuck!).

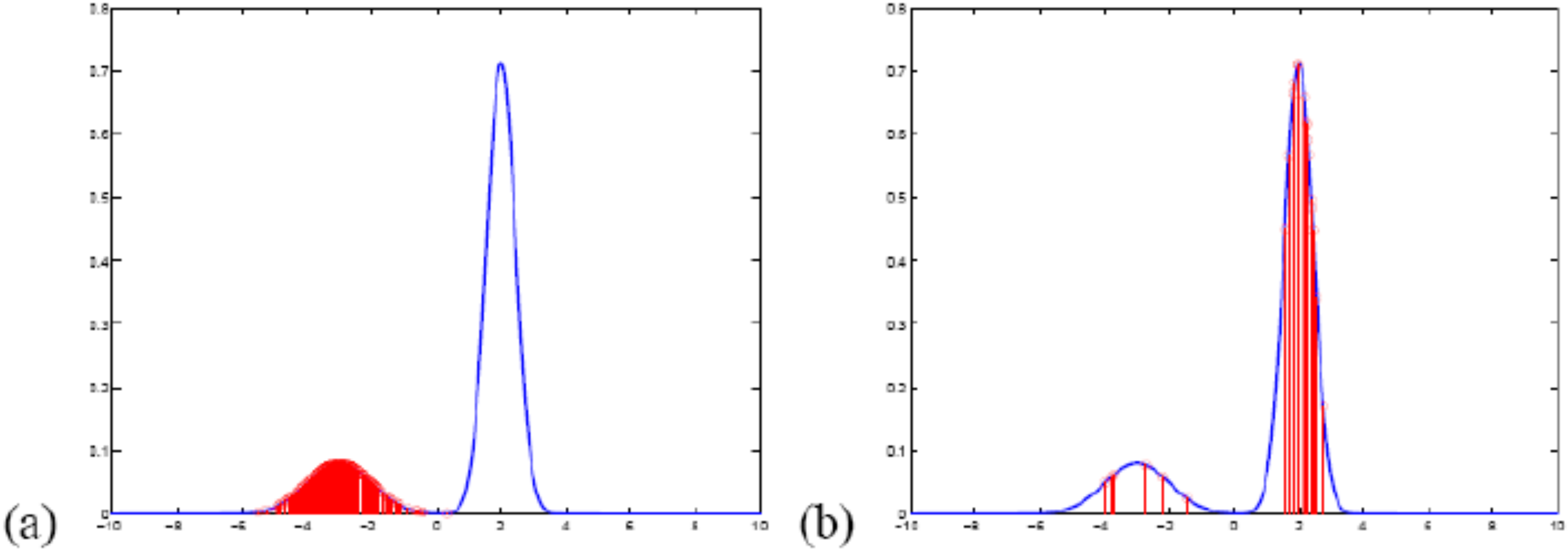
Rejection sampling suffers from the requirement of specifying a-priori a proposal g that matches f.

Metropolis-Hastings instead uses a proposal distribution (which need look nothing like f) as a means of (somewhat) local exploration.

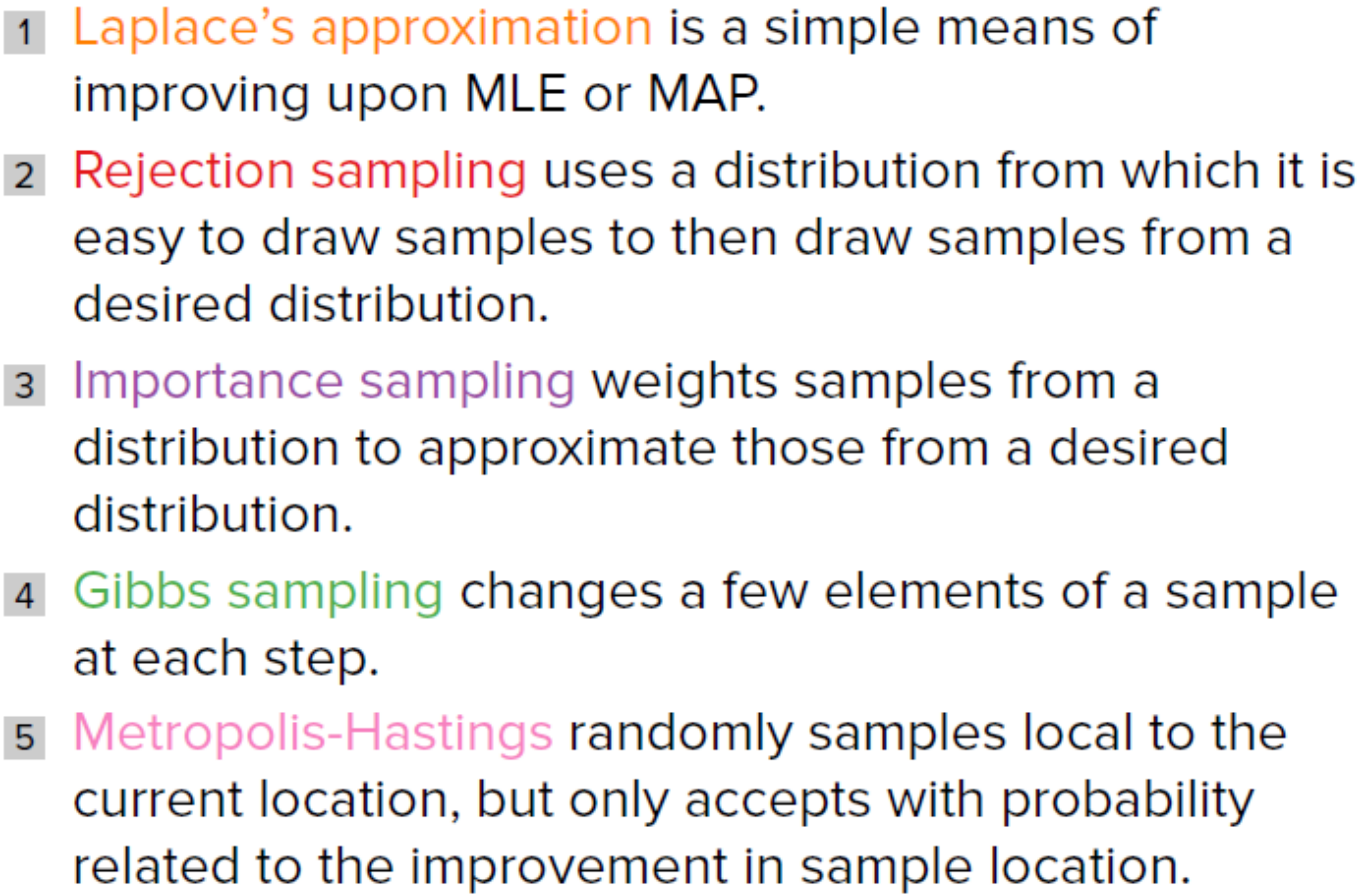
The degree of locality needs to be carefully selected: if our steps are too small, we'll never explore the whole space, but if our steps are too large, we'll never exploit local information about high f.

Like Gibbs, Metropolis-Hastings approaches may choose only to modify some subset of the elements of θ_i in moving to θ_{i+1}.

The degree of locality can be twofold: hybrid proposals in Metropolis-Hastings



Summary



1. Laplace's approximation is a simple means of improving upon MLE or MAP.
2. Rejection sampling uses a distribution from which it is easy to draw samples to then draw samples from a desired distribution.
3. Importance sampling weights samples from a distribution to approximate those from a desired distribution.
4. Gibbs sampling changes a few elements of a sample at each step.
5. Metropolis-Hastings randomly samples local to the current location, but only accepts with probability related to the improvement in sample location.

Concluding remarks

- The largest use of sampling methods is in the **sampling of parameters in model-inference problems**; these range from models of time series to neural networks and image analysis
- Another approach to approximate inference, variational methods, offers analytic tractability and in some important circumstances is preferred over sampling methods. We don't cover this in STA414
- Newer methods, such as Bayesian quadrature (not covered here), overcome a number of issues with Monte Carlo techniques
  NIPS 2015 paper: http://arxiv.org/abs/1506.02681
- Additional perspectives: Bishop 11.1-11.2, Murphy Ch 23-24