Lecture 4: Foundations of Bayesian Inference and Monte Carlo Methods

Advanced Topics in Machine Learning

Dr. Tom Rainforth

January 29th, 2020

rainforth@stats.ox.ac.uk

Lecture 4 Outline

In this lecture we will consider the problem of estimating and using Bayesian posteriors.

While previous lectures have focused on modeling, we will now be mostly concerned with computation instead; we will generally assume the model is given.

Particular topics:

- Why is Bayesian inference challenging?
- Deterministic Approximations
- Monte Carlo
- Rejection sampling
- Importance sampling

Why is Bayesian Inference Challenging?

Bayesian Inference is Hard!

- It might at first seem like Bayesian inference is a straightforward problem
  - By Bayes' rule we have that $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$ and so we already know the relative probability of any one value of θ compared to another.
- In practice, this could hardly be further from the truth
  - For non–trivial models, Bayesian inference is akin to calculating a high–dimensional integral
  - It is, in general, an NP–hard problem (the examples we have considered so far are special cases)

Why is Bayesian Inference Hard?

- We can break down Bayesian inference into two key challenges:
  - Calculating the normalization constant $p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)d\theta$
  - Providing a useful characterization of the posterior, for example, a set of approximate samples
- Each of these constitutes a somewhat distinct problem
- Many methods actually side–step the first problem and directly produce approximate samples

Why is Bayesian Inference Hard? (2)

- Optimization is like finding a needle in a haystack
- Inference is like finding **all** the needles in a haystack
  - This gets particularly hard when the space is high–dimensional

Image Credit: Metro

The Normalization Constant

If $p(\mathcal{D})$ is unknown, we lack scaling when evaluating a point

- We have no concept of how relatively significant that point is compared to the distribution as a whole
- We don't know how much mass is missing
- The larger the space of θ, the more difficult this becomes

Image Credit: www.theescapeartist.me

The Normalization Constant (2)

Consider a model where $\theta \in \{1, 2, 3\}$ with a corresponding uniform prior $P(\theta) = 1/3$ for each θ.

Now presume that for some reason we are only able to evaluate the likelihood at $\theta = 1$ and $\theta = 2$, giving $p(\mathcal{D}|\theta = 1) = 1$ and $p(\mathcal{D}|\theta = 2) = 10$ respectively.

Depending on the marginal likelihood $p(\mathcal{D})$, the posterior probability of $P(\theta = 2|\mathcal{D})$ will vary wildly:

- $p(\mathcal{D}) = 4$ gives $P(\theta = 2|\mathcal{D}) = 5/6$
- $p(\mathcal{D}) = 1000$ gives $P(\theta = 2|\mathcal{D}) = 1/100$

The Normalization Constant (3)

- If we manage to calculate $p(\mathcal{D})$ we can then calculate the posterior exactly through Bayes' rule
- One might reasonably assume this was enough to solve the inference problem
- In practice, even having an exact form for $p(\theta|\mathcal{D})$ is often not enough for many tasks we might want to carry out when θ is continuous or has a very large number of possible values

What Might we Use the Posterior For?

- To calculate the posterior probability or density for some particular θ
- To calculate the expected value of some function, $\mathbb{E}_{p(\theta|\mathcal{D})}\left[f(\theta)\right]$
- To make predictions using the posterior predictive distribution
- To find the most probable variable values $\theta^* = \arg\max_\theta p(\theta|\mathcal{D})$
- To produce a useful representation of the posterior for passing on to another part of a computational pipeline or to be directly observed by a user

Characterizing the Posterior

- Knowing is $p(\mathcal{D})$ is only sufficient for this first of these tasks
- The others require additional computation of some form
- In particular, it is knowing $p(\mathcal{D})$ is not sufficient (or even necessary!) for drawing **samples** from the posterior
- At its heart, the problem of Bayesian inference is a problem of where to concentrate our finite computational resources so that we can effectively characterize the posterior; being able to evaluate it piecewise is not always enough for this

Characterizing the Posterior: Example

Lets consider a simple example where we can easily calculate $p(\mathcal{D})$, and thus $p(\theta|\mathcal{D})$, numerically:

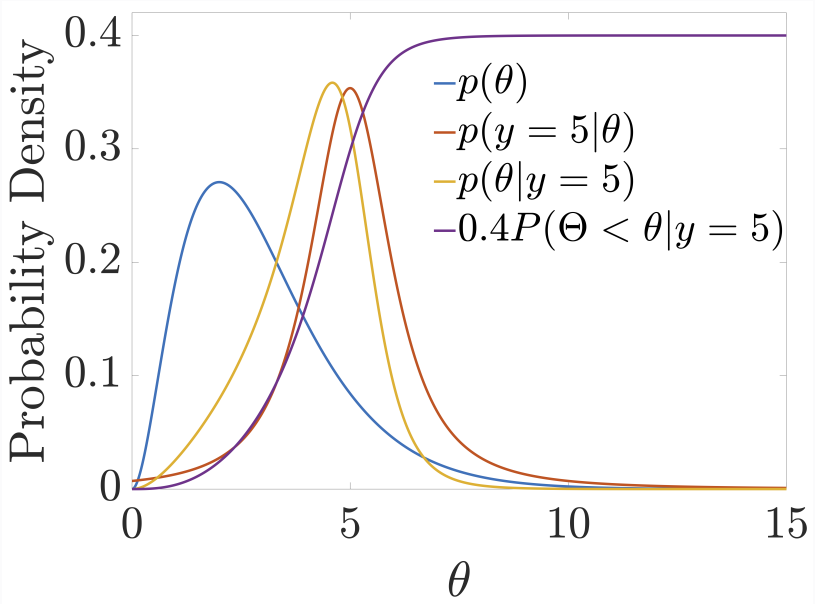
$$p(\theta) = \textsc{Gamma}\left(\theta; 3, 1\right) = \frac{\theta^2 \exp(-\theta)}{2} \quad \theta \in (0, \infty)\,,$$

$$p(y = 5|\theta) = \textsc{Student-t}\left(\theta - 5; 2\right) = \frac{\Gamma(1.5)}{\sqrt{2\pi}} \left(1 + \frac{(\theta - 5)^2}{2}\right)^{-3/2}$$

$$p(\theta|y = 5) \approx 5.348556\, \theta^2 \exp(-\theta) \left(2 + (5 - \theta)^2\right)^{-3/2}$$

Characterizing the Posterior: Example (2)

- Even though we have the posterior in closed form, it is not a standard distribution and so we don't know how to sample from it
- For higher dimensional problems, it will be very difficult to calculate expectations or the posterior predictive without being able to sample
- We'll return to how we might sample from this later



Deterministic Approximations

Point Estimates

- One of the simplest approaches is to effectively ignore the posterior computation problem completely and instead resort to a heuristic approximation
- The simplest such approach is to take a **point estimate** $\tilde{\theta}$ for θ and then approximate the posterior predictive distribution using only this value:

$$p(\mathcal{D}^*|\mathcal{D}) \approx p(\mathcal{D}^*|\tilde{\theta}). \tag{1}$$

- Finding $\tilde{\theta}$ requires only an **optimization** problem to be solved
  - This is far easier than the **integration** problem posed by full posterior inference

Maximum Likelihood

Maximum likelihood is a non-Bayesian, frequentist, approach for calculating a $\tilde{\theta}$ based on maximizing the likelihood:

$$\tilde{\theta}_{\mathsf{ML}} = \arg\max_{\theta \in \vartheta} p(\mathcal{D}|\theta). \tag{2}$$

This can be prone to overfitting and does not incorporate prior information leading to a host of issues we previously discussed (see Bayesian vs frequentist notes)

Maximum a Posteriori (MAP)

Maximum a Posteriori (MAP) estimation corresponds to choosing $\tilde{\theta}$ to maximize the posterior probability:

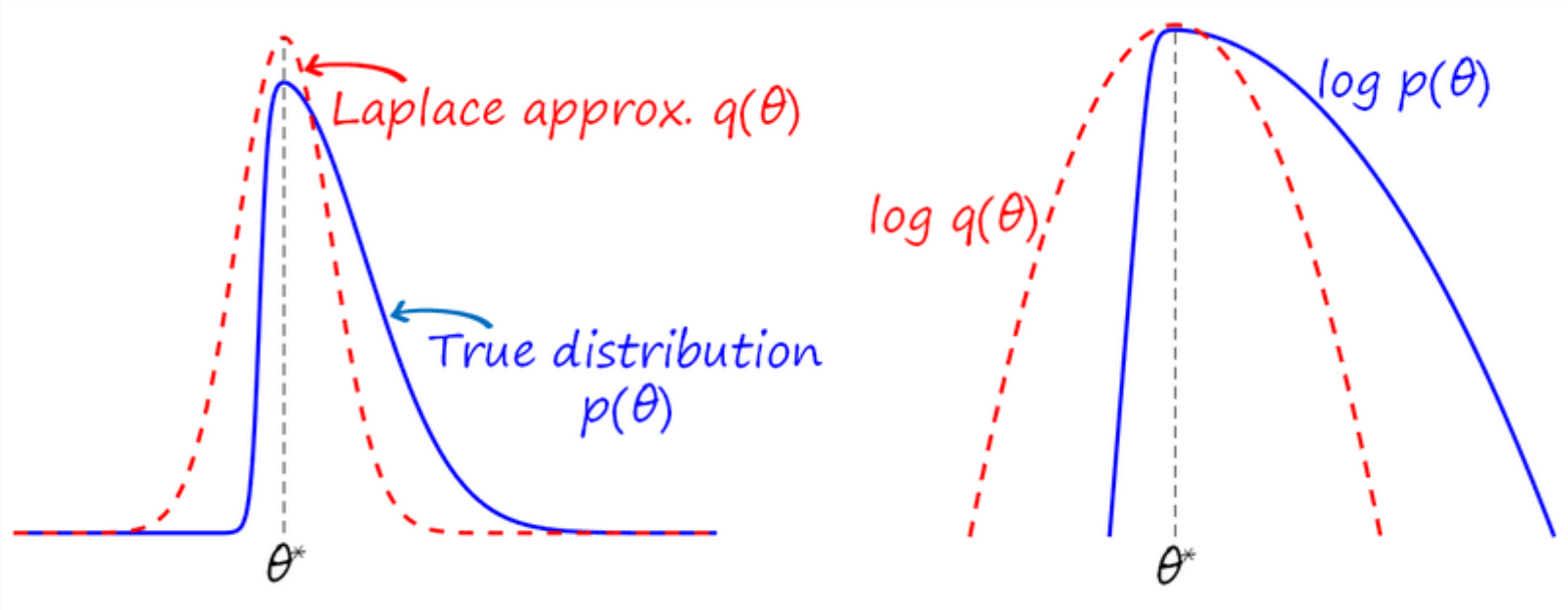
$$\tilde{\theta}_{\text{MAP}} = \arg\max_{\theta \in \vartheta} p(\theta|\mathcal{D}) = \arg\max_{\theta \in \vartheta} p(\mathcal{D}|\theta)p(\theta). \tag{3}$$

This provides regularization compared to ML estimation but still has a number of drawbacks compared to full inference:

- It incorporates less information into the predictive distribution
- The position of the MAP estimate is dependent of the parametrization of the problem (see notes on change of variables)

Laplace Approximation

The Laplace approximation refines the MAP estimate by approximating the full posterior with a Gaussian centered at the MAP estimate and covariance dictated by the curvature of the log density around this point



Images Credit: Luis Herranz

Laplace Approximation (2)

More formally, the Laplace approximation is given by

$$p(\theta|\mathcal{D}) \approx \mathcal{N}\left(\theta; \tilde{\theta}_{\text{MAP}}, (\Lambda_{\text{MAP}})^{-1}\right) \tag{4}$$

where Λ_{MAP} is the negative Hessian of the log joint density evaluated at the MAP, i.e.

$$\Lambda_{\text{MAP}} = -\nabla^2_\theta \log\left(p(\theta, \mathcal{D})\right)|_{\theta=\tilde{\theta}_{\text{MAP}}}. \tag{5}$$

Monte Carlo

Monte Carlo

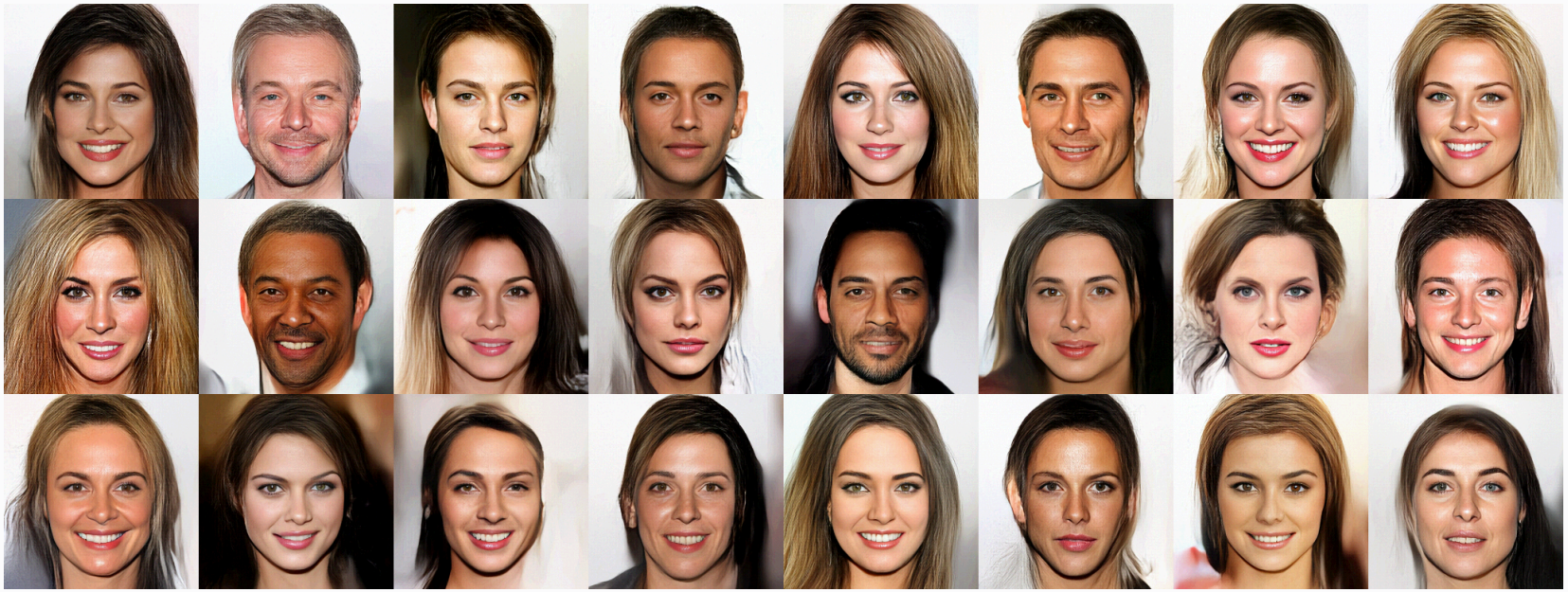
Definition

Monte Carlo is the characterization of a probability distribution through random sampling.

- It forms the underlying principle for all stochastic computation
  - It is the foundation for a huge array of methods for numerical integration, optimization, and Bayesian inference
- It provides us with a means of dealing with complex models and problems in a statistically principled manner.

Example: Visualizing a Generative Model Over Faces

Little can be understood from looking at the density of a generative model for generative faces: we need to draw samples to understand the model



[1] D P Kingma and P Dhariwal. "Glow: Generative flow with invertible 1x1 convolutions". In: *NeurIPS*. 2018.

Monte Carlo Estimators

Consider the problem of calculating the expectation of some function $f(\theta)$ under the distribution $\theta \sim \pi(\theta)$:

$$I := \mathbb{E}_{\pi(\theta)}[f(\theta)] = \int f(\theta)\pi(\theta)d\theta. \tag{6}$$

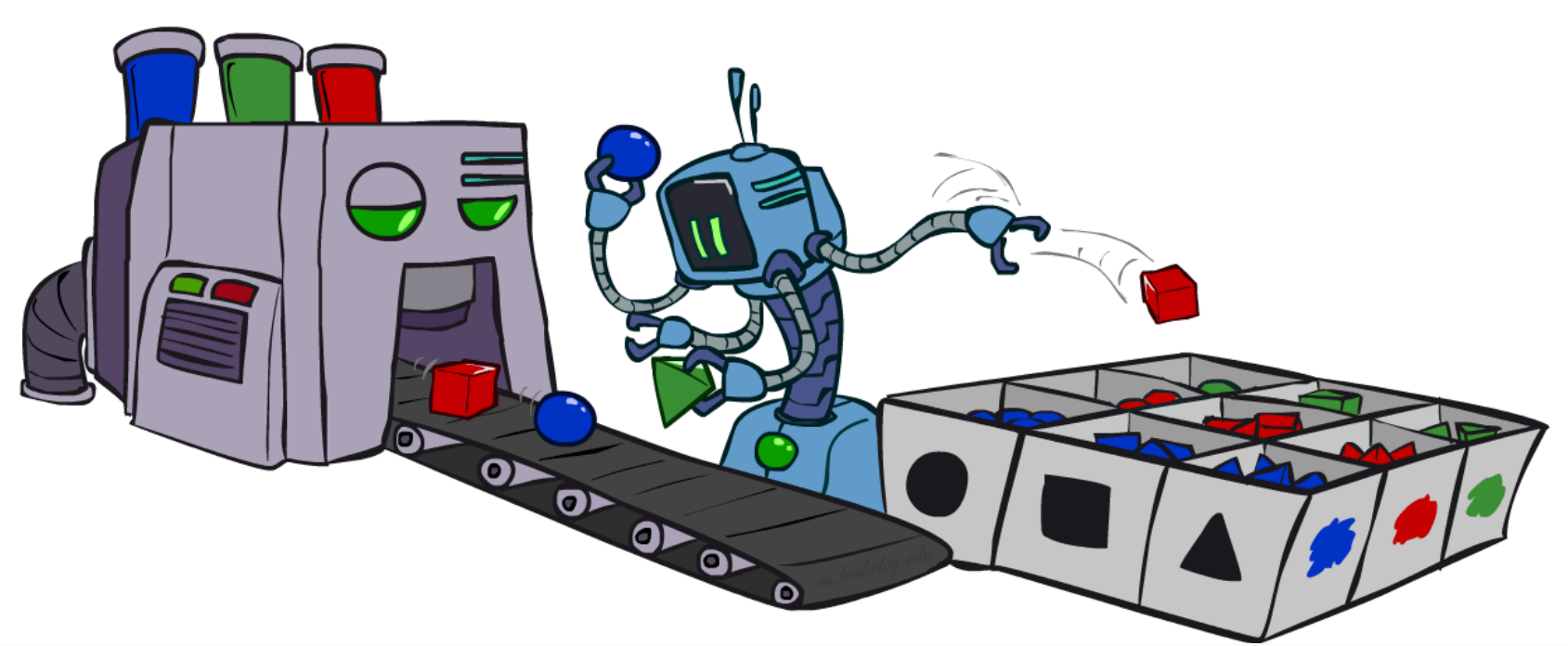
This can be approximated using the **Monte Carlo estimator** I_N:

$$I \approx I_N := \frac{1}{N}\sum_{n=1}^{N} f(\hat{\theta}_n) \quad \text{where} \quad \hat{\theta}_n \sim \pi(\theta) \tag{7}$$

are independent draws from $\pi(\theta)$.

Most of the tasks we laid out for Bayesian inference can be formulated as some form of (potentially implicit) expectation

Example: Production Line

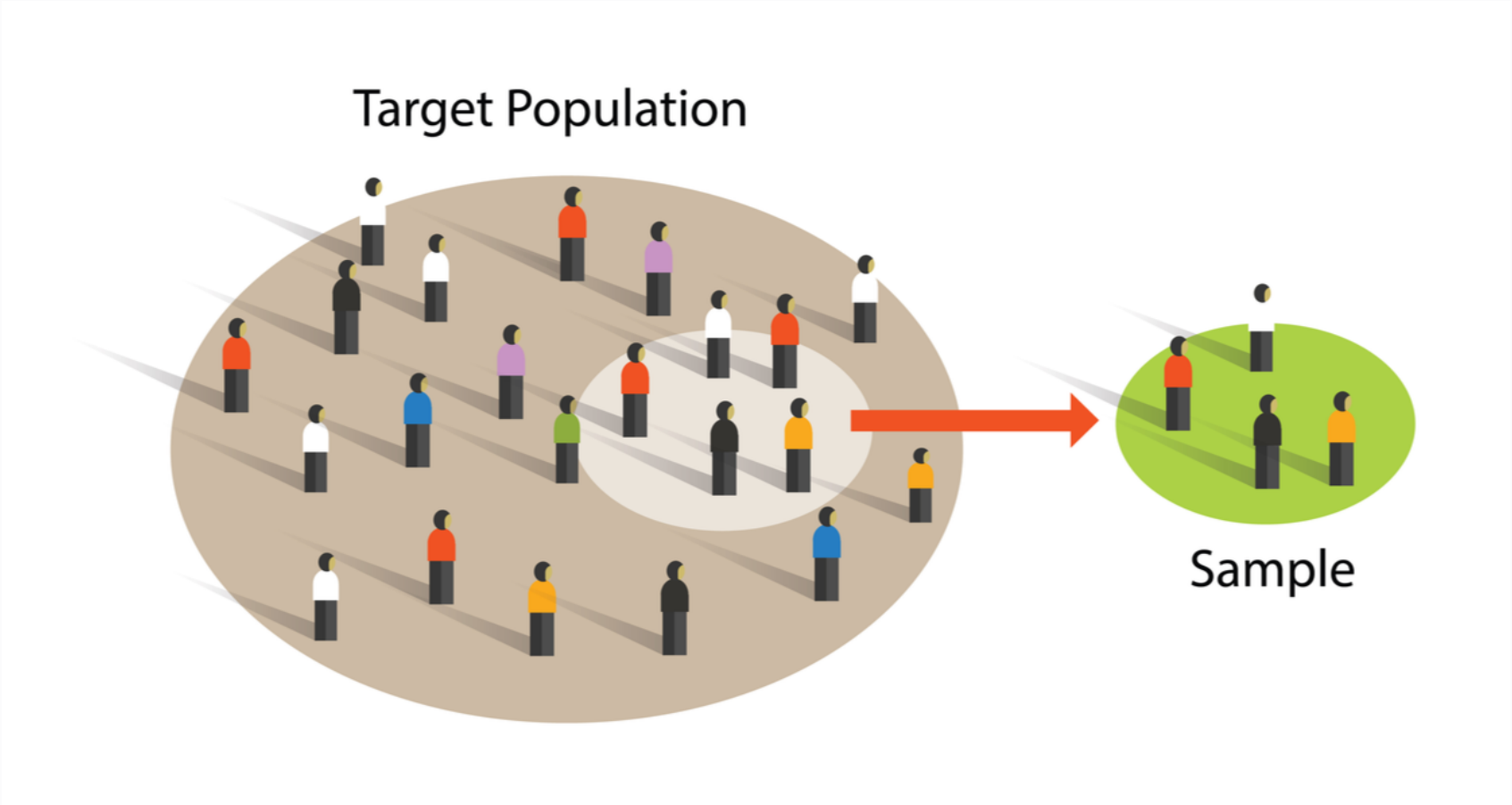


- The production machine randomly generates colored shapes from some distribution, a robot sorts them into bins
- The production machine is performing Monte Carlo sampling, the robot is constructing a Monte Carlo estimate

Images Credit: Pieter Abbeel

Example: Election Polling

We cannot query the full distribution over voters, so we poll instead



Images Credit: Anthony Figueroa

Unbiasedness

The Monte Carlo estimate is unbiased (for fixed N), i.e. $\mathbb{E}[I_N] = I$

$$\begin{aligned}
\mathbb{E}[I_N] &= \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N} f(\hat{\theta}_n)\right] \\
&= \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\left[f(\hat{\theta}_n)\right] \\
&= \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\left[f(\hat{\theta}_1)\right] \\
&= I
\end{aligned}$$

What Exactly Does Unbiasedness Mean?

- It means that Monte Carlo does not introduce any systematic error, i.e. **bias**, into the approximation
  - In expectation, it does not pathologically overestimate or underestimate the target
  - A biased estimator $\tilde{I}$ would have $\mathbb{E}[\tilde{I}] = I + B$ for some $B \neq 0$
  - Here we are implicitly using the frequentist definition of probability: the expectation is defined through repeating the sampling infinitely often
- It does **not** mean that it is equally likely to overestimate or underestimate
  - It may, for example, typically underestimate by a small amount and then rarely overestimate by a large amount

Consistency of an Estimator

- In general, we want an estimator to become arbitrarily good in the limit of using a large computation
  - For example, with our Monte Carlo estimator, we would like $I_N \to I$ as $N \to \infty$.
- This is know as **consistency** of an estimator
- It is not the same thing as unbiasedness
  - Unbiasedness is concerned with repeatedly constructing a finite estimator and averaging the results
  - Consistency is concerned with what happens when we increase the budget of a single estimator
  - Many estimators are biased in the finite regime but consistent (their bias decreases as N increases)

The Law of Large Numbers

The consistency of the standard Monte Carlo estimator is demonstrated by the **law of large numbers**.

Informally, the law of large numbers states that the empirical average of **independent and identically distributed** (i.i.d.) random variables converges to the true expected value of the underlying process as the number of samples increases

More formally we have:

The (Weak) Law of Large Numbers

$$\mathbb{E}\left[(I_N - I)^2\right] = \frac{\sigma_\theta^2}{N}$$

$$\text{where} \quad \sigma_\theta^2 := \mathbb{E}\left[\left(f(\hat{\theta}_1) - I\right)^2\right] = \text{Var}\left[f(\theta)\right]$$

The Law of Large Numbers (2)

There are two key consequences of the LLN:

- $I_N \to I$ as $N \to \infty$ such that the Monte Carlo estimate is consistent
- The rate of this convergence is such that $|I_N - I|$ is $O(1/\sqrt{N})$

Other more powerful results, like the **central limit theorem**, allow for the i.i.d. assumption of the LLN to be relaxed and give more information about the nature of this convergence.

- This is important if our samples are correlated (e.g. MCMC sampling)
- See the notes for more details

Composability of Monte Carlo

Another key property of Monte Carlo is that samples can be combined and deconstructed:

- Monte Carlo samples from a joint distribution will also have the correct marginal distribution over any of its individual components
  - If $\{\hat{\theta}_1, \hat{\theta}_2\} \sim p(\theta_1, \theta_2)$ then $\hat{\theta}_1 \sim p(\theta_1)$ and $\hat{\theta}_2 \sim p(\theta_2)$
- Sampling from the marginal distribution then sampling from the conditional distribution given these samples will give samples distributed according to the joint.
  - If $\hat{\theta}_1 \sim p(\theta_1)$ and $\hat{\theta}_2 \sim p(\theta_2|\theta_1 = \hat{\theta}_1)$, then $\{\hat{\theta}_1, \hat{\theta}_2\} \sim p(\theta_1, \theta_2)$

These mean that Monte Carlo can be used as a mechanism for **unbiasedly** propagating information

The Flaw of Averages

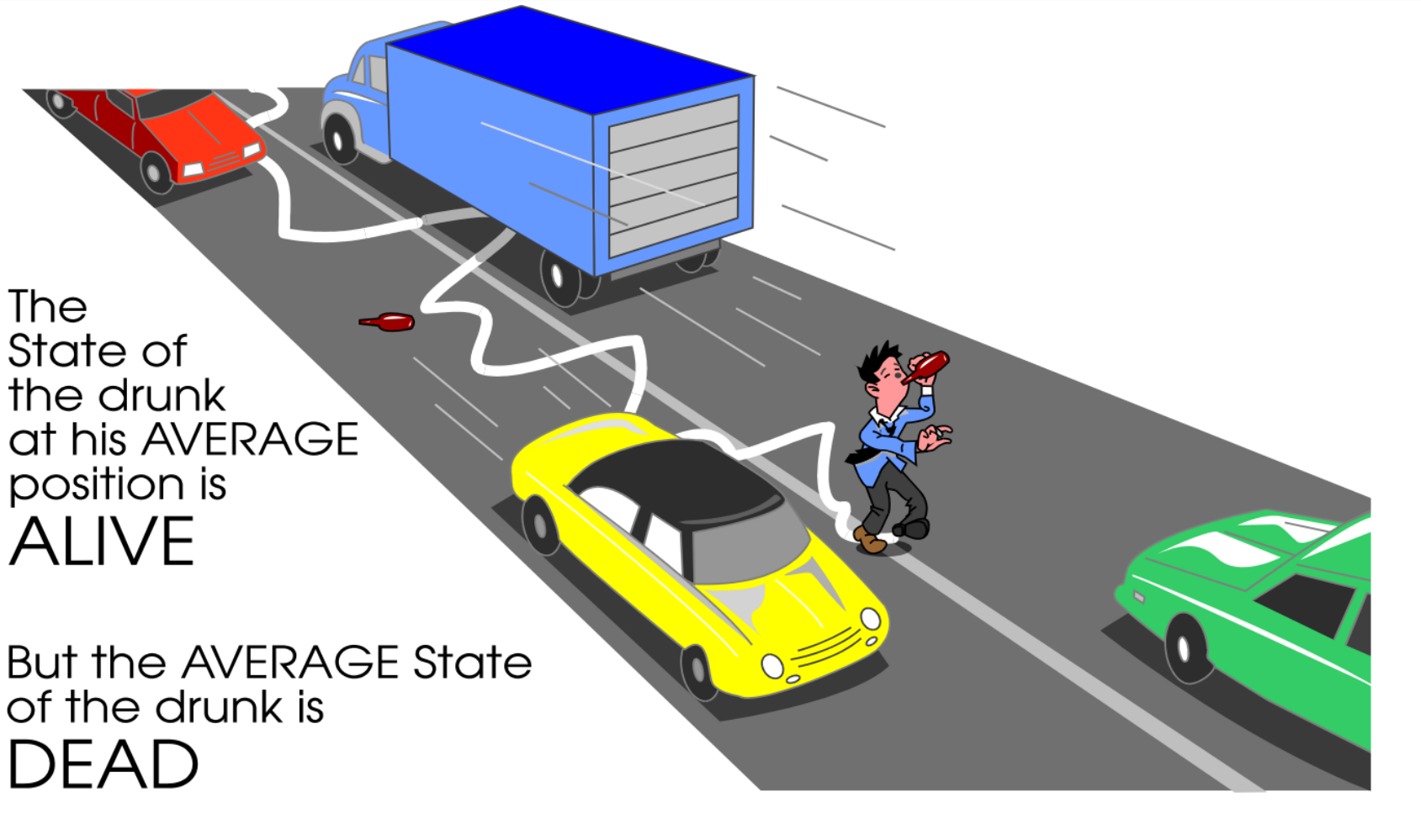


Image Credit: Sam L. Savage

The Flaw of Averages

- In general, $f(\mathbb{E}[\theta]) \neq \mathbb{E}[f(\theta)]$
- Avoid taking expectations in a computational pipeline until you absolutely have to
- Monte Carlo instead allows us to pass information through samples that we can average over at a later time
  - We can draw samples **autoregressively**. For example, we can sample $\hat{\theta}_1 \sim p(\theta_1)$, then $\hat{\theta}_2 \sim p(\theta_2|\theta_1 = \hat{\theta}_1)$ and so forth
  - We then only take the average of these when we actually need to calculate an expectation

Monte Carlo vs Classical Integration Schemes

- Classical integration approaches like Simpson's rule can offer far better convergence rates in low dimensions that the $O(1/\sqrt{N})$ of Monte Carlo
- But these rates break down (typically exponentially) as the dimension increases
- In high–dimensions, Monte Carlo estimates are one of the **only** approaches that can remain accurate

Drawing Samples

Drawing Samples

- We have shown how to use samples to characterize distributions and estimate expectations
- But how to we draw these samples in the first place?
- We'll now introduce a number of sampling schemes
- Note that most (with the exception of our first example) will not require us to know the normalization constant $p(\mathcal{D})$: they can operate on $p(\theta, \mathcal{D})$ directly

Sampling Using the Inverse CDF

If we know the cumulative density function (CDF) of the posterior

$$P(\Theta \leq \theta|\mathcal{D}) := \int_{\Theta=-\inf}^{\Theta=\theta} p(\theta = \Theta|\mathcal{D})d\Theta, \tag{8}$$

along with its inverse P^{-1} (we rarely do in practice), then we can draw exact samples by first sampling $\hat{u} \sim \text{UNIFORM}(0, 1)$ and then taking $\hat{\theta} = P^{-1}(\hat{u})$, noting that $\hat{u} = P(\Theta \leq \hat{\theta}|y = 5)$

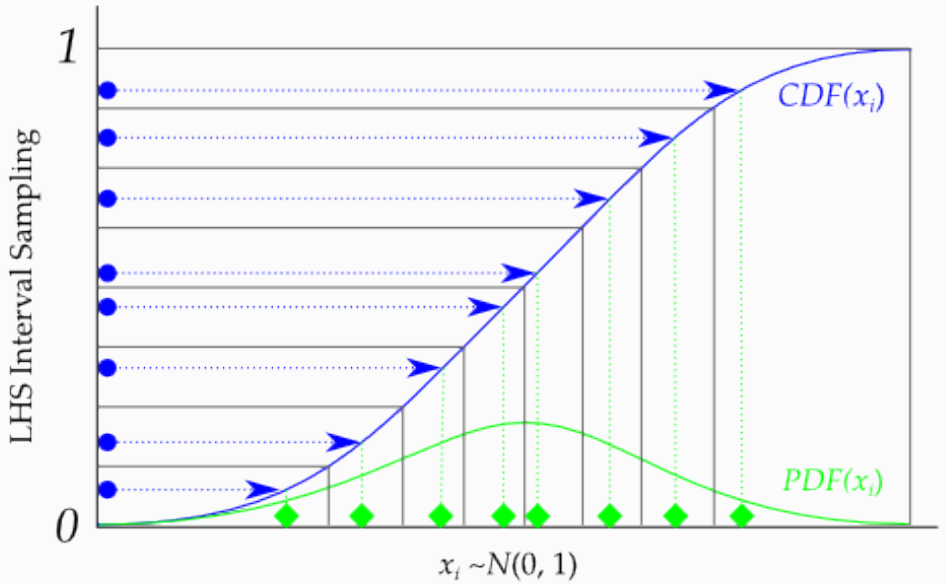
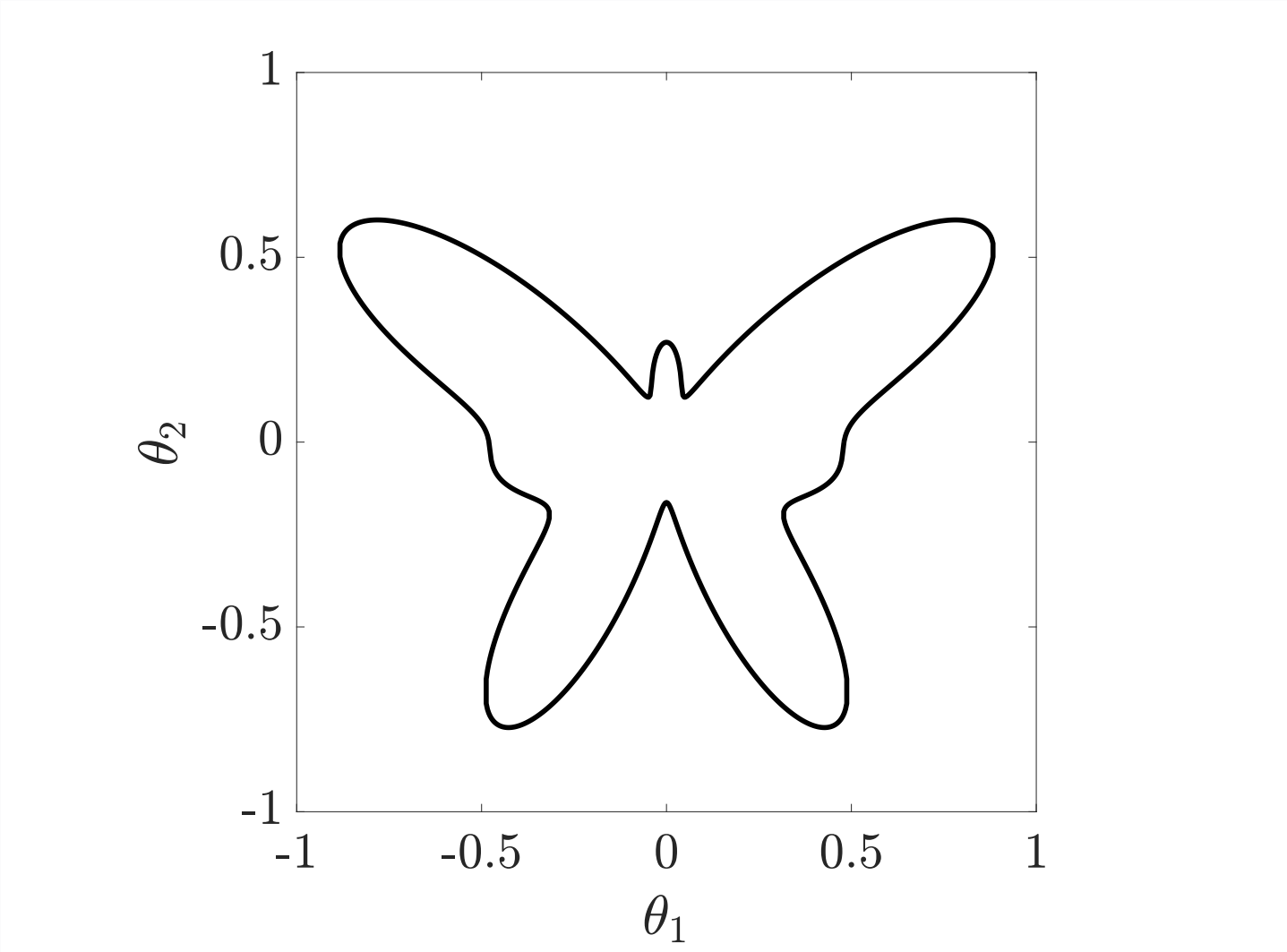


Image credit: pythonhosted.org

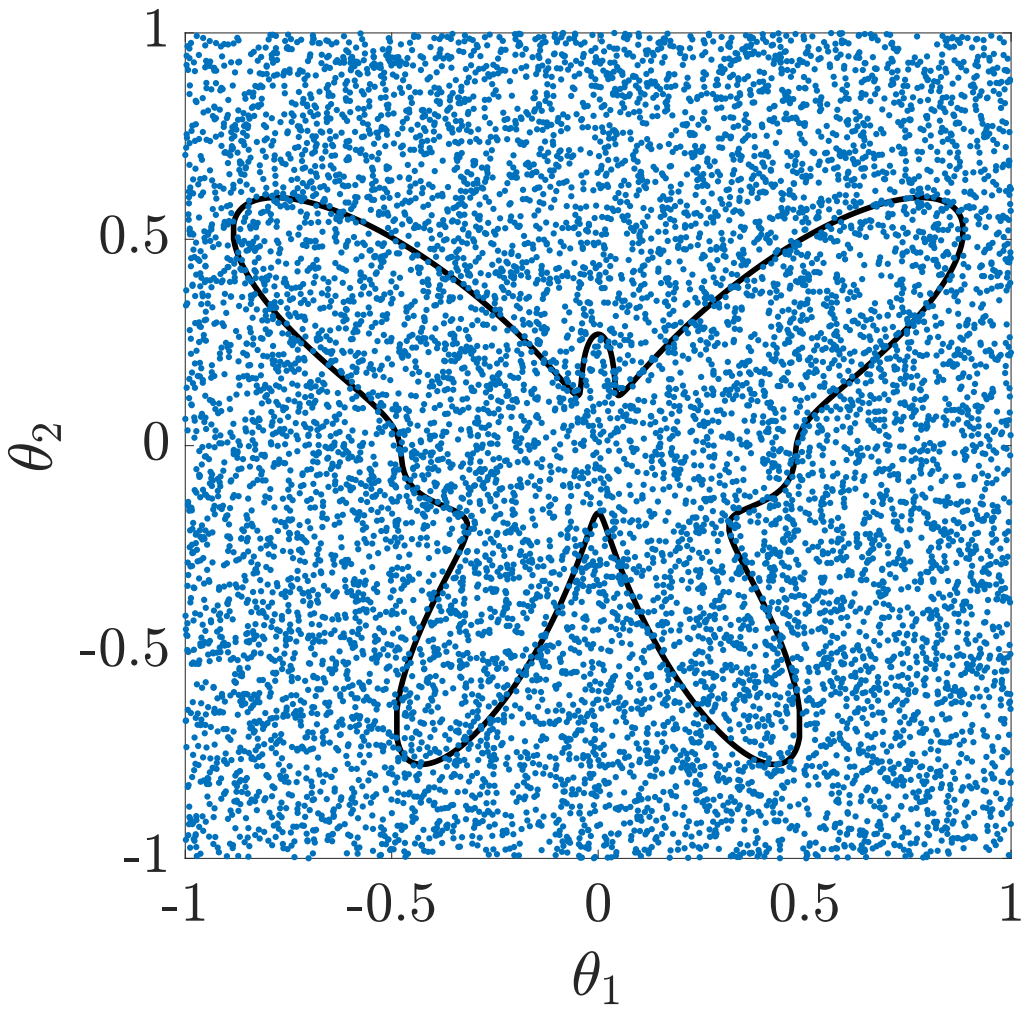
Sampling By Rejection

How might we draw samples from within this butterfly shape?



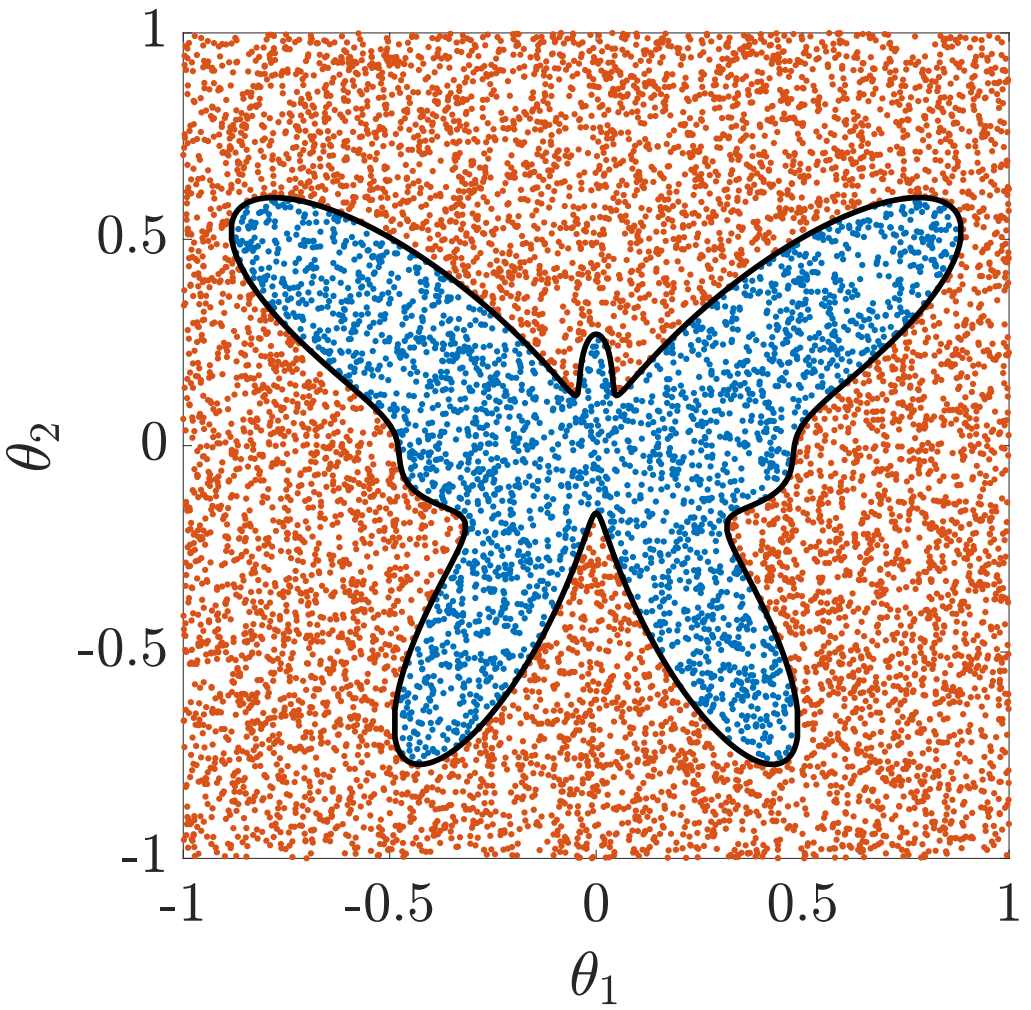
Sampling By Rejection

We can draw samples uniformly from a surrounding box



Sampling By Rejection

Then **reject** those not falling within the shape



Sampling By Rejection (2)

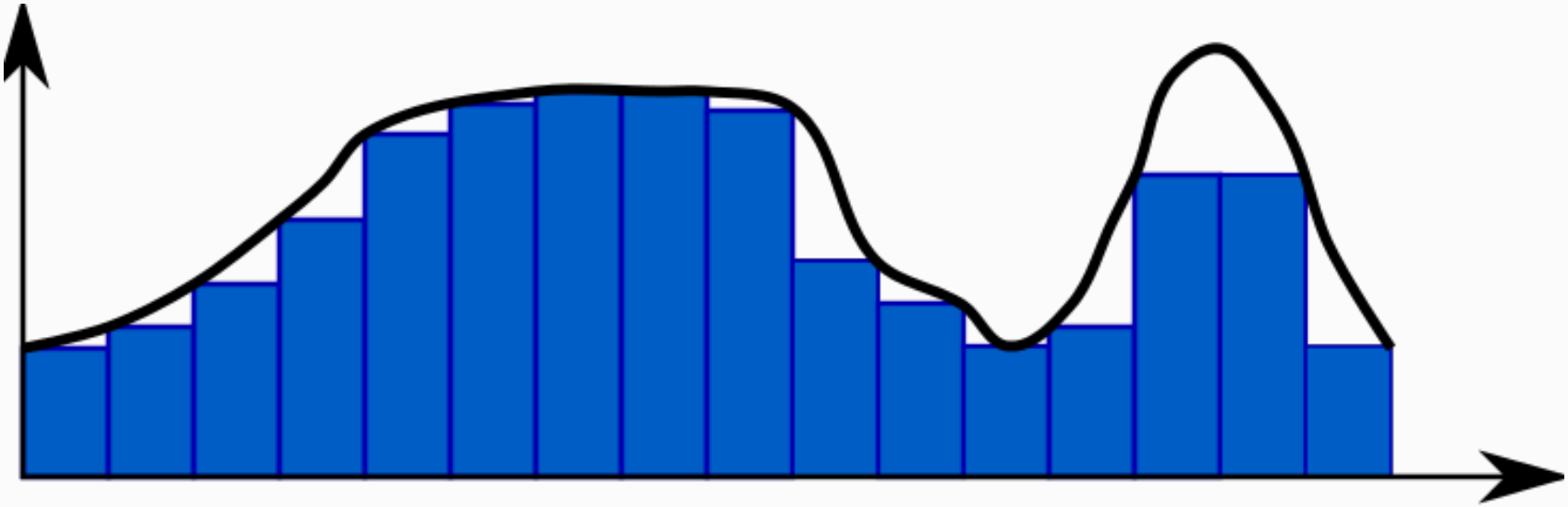
- We can also use this method to estimate the area of the shape
- The probability of any one sample falling within the shape is equal to the ratio of the areas of the shape and bounding box:

$$A_{\text{shape}} = A_{\text{box}} P(\theta \in \text{shape})$$
$$\approx \frac{A_{\text{box}}}{N} \sum_{n=1}^{N} \mathbb{I}(\hat{\theta}_n \in \text{shape}) \quad \text{where} \quad \hat{\theta}_n \sim \textsc{Uniform}(\text{box})$$

- Here we have used a Monte Carlo estimator for $P(\theta \in \text{shape})$
- Note that the value of $P(\theta \in \text{shape})$ will dictate the efficiency of our estimation as it represents the **acceptance rate** of our samples.

Sampling from Area Under Density

Sampling from the area under a density function is equivalent to sampling from that density itself.



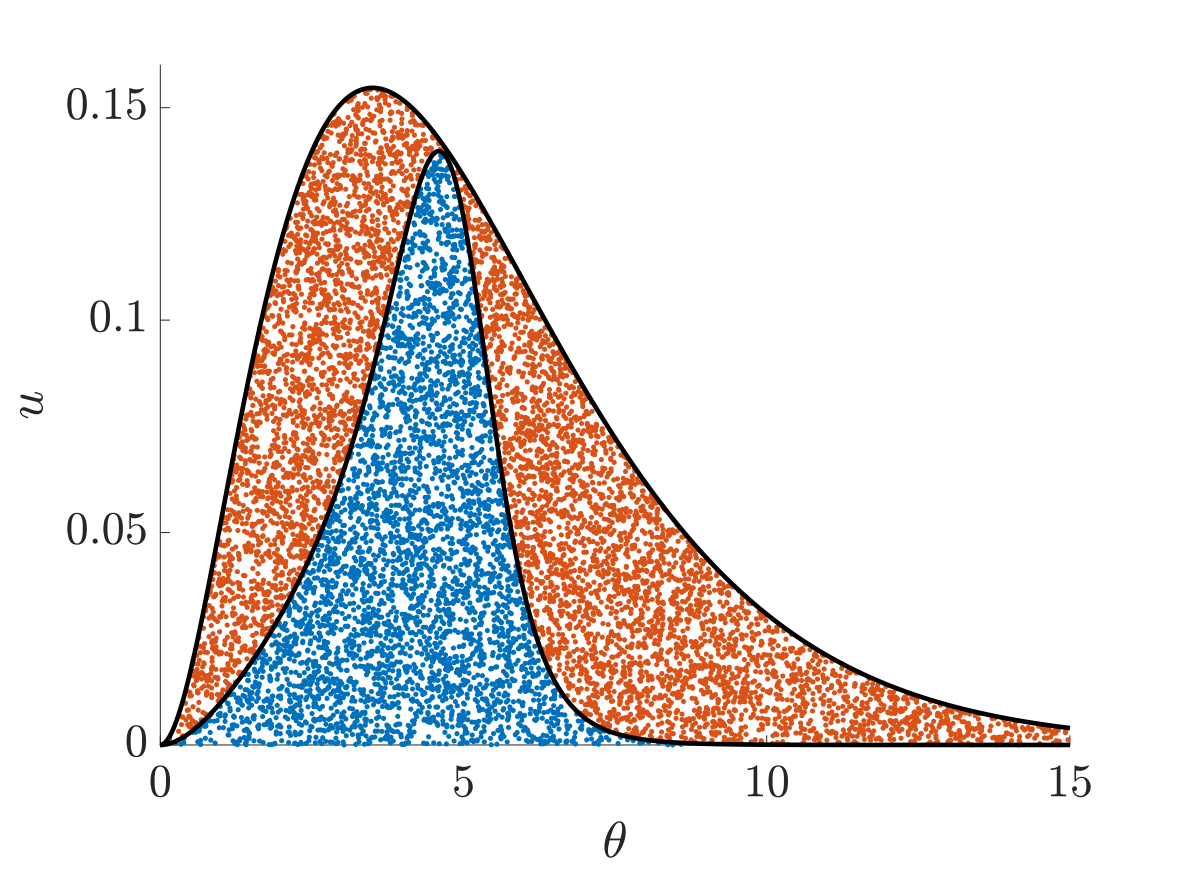
Think about sampling from a histogram with even width bins and then take the width of these bins to zero

---

Images Credit: Wikipedia

Rejection Sampling

Rejection sampling uses this idea to draw samples from a target by drawing samples from an area enveloping its density using an **auxiliary variable** u



Rejection Sampling (2)

More formally, we define a **proposal** distribution $q(\theta)$ which completely **envelopes** a scaled version of the unnormalized target distribution $Cp(\theta, \mathcal{D})$, for some fixed C, such that $q(\theta) \geq Cp(\theta, \mathcal{D})$ for all values of θ.

We then sample a pair $\{\hat{\theta}, \hat{u}\}$ by first sampling $\hat{\theta} \sim q(\theta)$ and then $\hat{u} \sim \textsc{Uniform}(0, q(\theta))$. The sample is accepted if

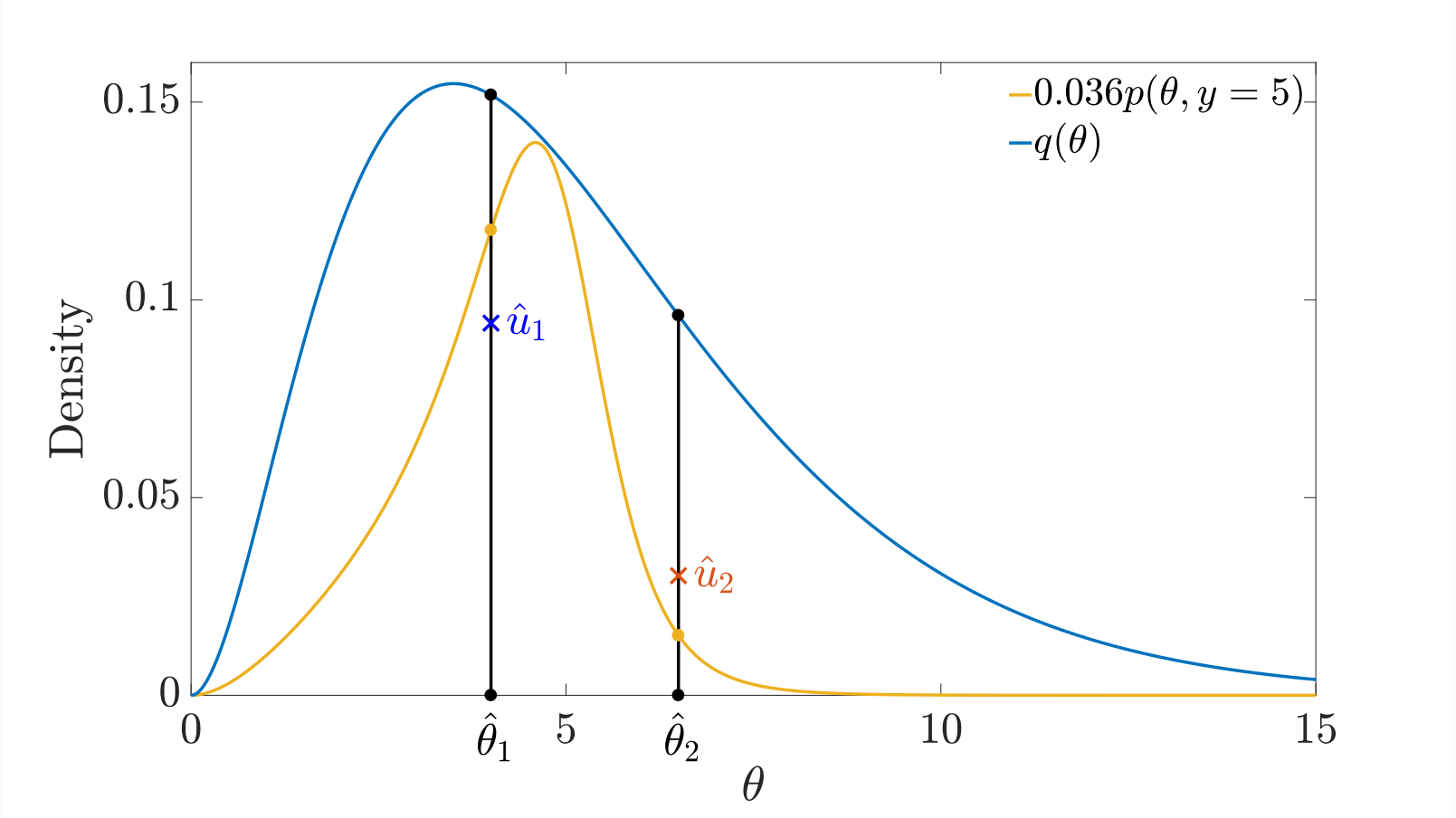
$$\hat{u} \leq Cp(\hat{\theta}, \mathcal{D}) \tag{9}$$

in which case $\hat{\theta}$ is an exact sample from $p(\theta|\mathcal{D})$

The acceptance rate of samples is $Cp(\mathcal{D})$, which thus provides and estimate for $p(\mathcal{D})$ by diving through by C

Rejection Sampling (3)

Rejection sampling in action for our earlier example:



Rejection Sampling (4)

Pros

- One of the only inference methods to produce exact samples
- Can be highly effective in low dimensions
- Works equally well for unnormalized targets (i.e. we there is no need to know $p(\mathcal{D})$
- Provides a marginal likelihood estimate via the acceptance rate

Rejection Sampling (5)

Cons

- Scales poorly to higher dimensions (more on this later)
- Requires carefully designed proposals
- Very dependent on the value of C
- Finding a valid C requires significant knowledge about the target density

Importance Sampling

Importance Sampling

- **Importance sampling** is a common sampling method that is also the cornerstone for many more advanced inference schemes
- It is closely related to rejection sampling in that it uses a proposal, i.e. $\hat{\theta} \sim q(\theta)$
- Instead of having an accept–reject step, it assigns an **importance weight** to each sample
- These importance weights act like correction factors to account for the fact that we sampled from $q(\theta)$ rather than our target $p(\theta|\mathcal{D})$

Importance Sampling Algorithm

Assume for now that we can evaluate $p(\theta|\mathcal{D})$ exactly. Here the algorithm is as follows:

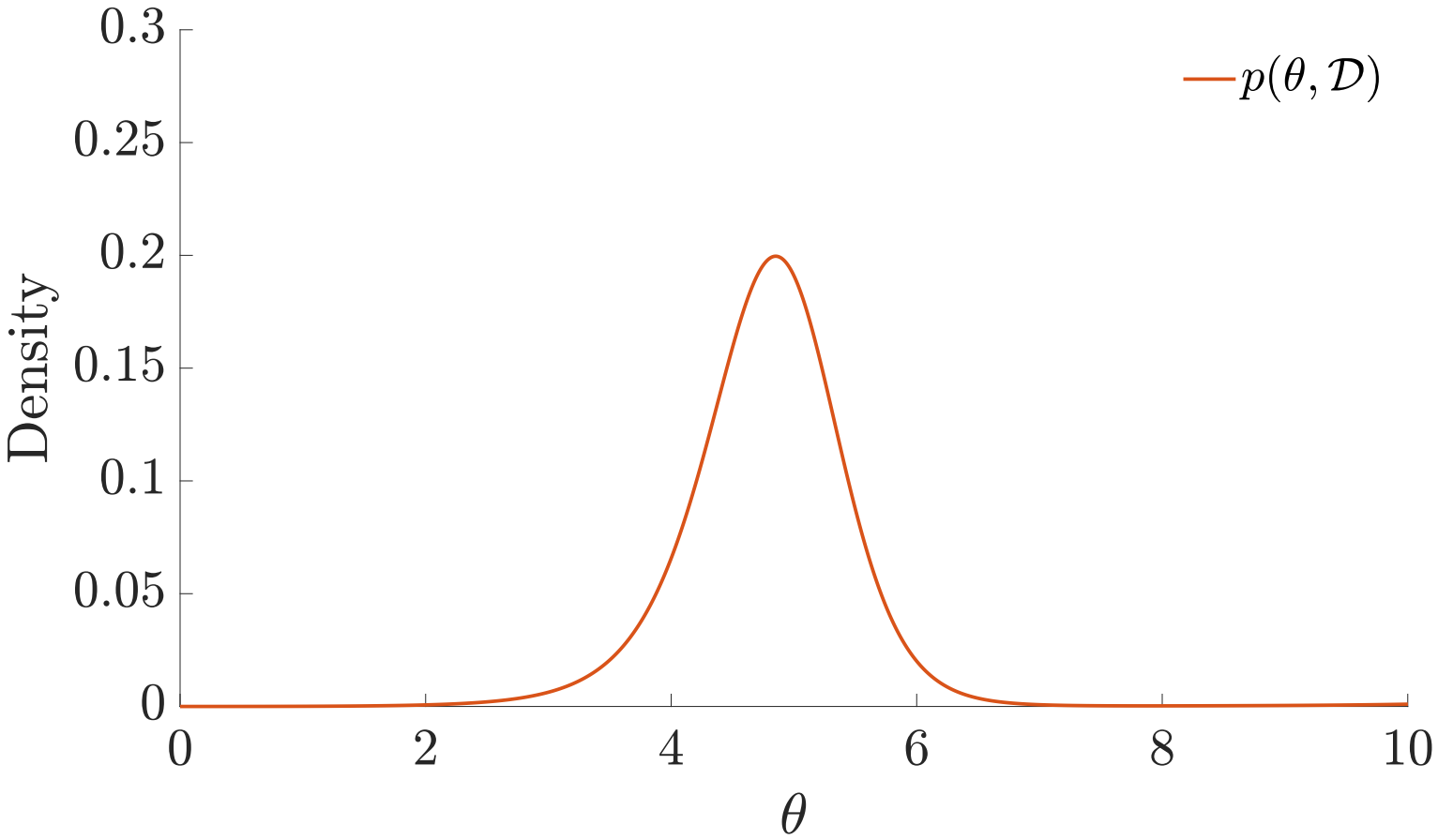
1. Define a proposal $q(\theta)$
2. Draw N i.i.d. samples $\hat{\theta}_n \sim q(\theta) \quad n = 1, \ldots, N$
3. Assign weight $w_n = \frac{p(\hat{\theta}_n|\mathcal{D})}{q(\hat{\theta}_n)}$ to each sample
4. Combine the samples to form the empirical measure

$$p(\theta|\mathcal{D}) \approx \hat{p}(\theta|\mathcal{D}) := \frac{1}{N}\sum_{n=1}^{N} w_n \delta_{\hat{\theta}_n}(\theta) \tag{10}$$

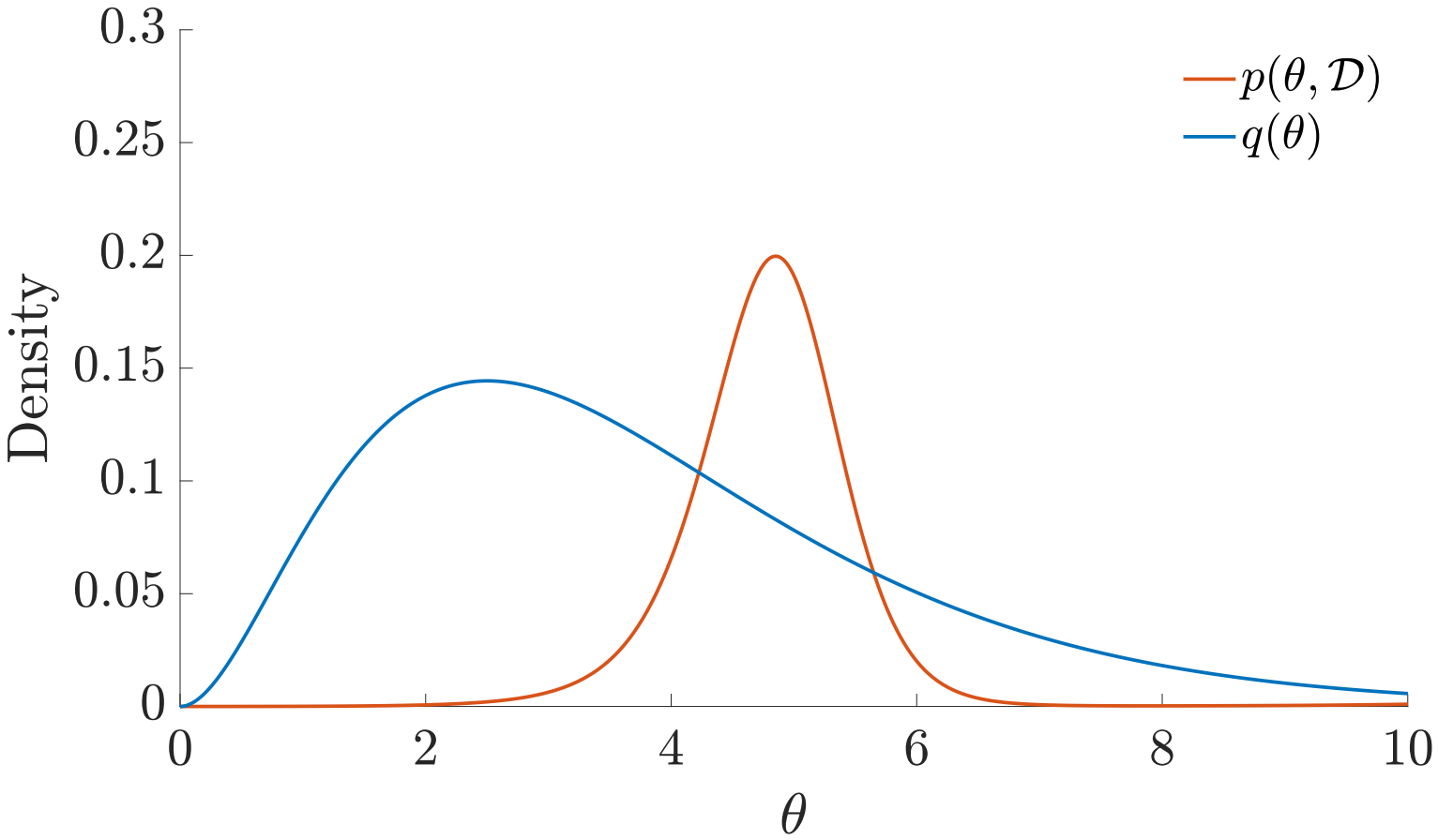
5. This can used to be estimate $\mathbb{E}_{p(\theta|\mathcal{D})}[f(\theta)]$ for any f using

$$\mathbb{E}_{p(\theta|\mathcal{D})}\left[f(\theta)\right] \approx \hat{\mu}_{\text{IS}} := \frac{1}{N}\sum_{n=1}^{N} w_n f(\hat{\theta}_n) \tag{11}$$

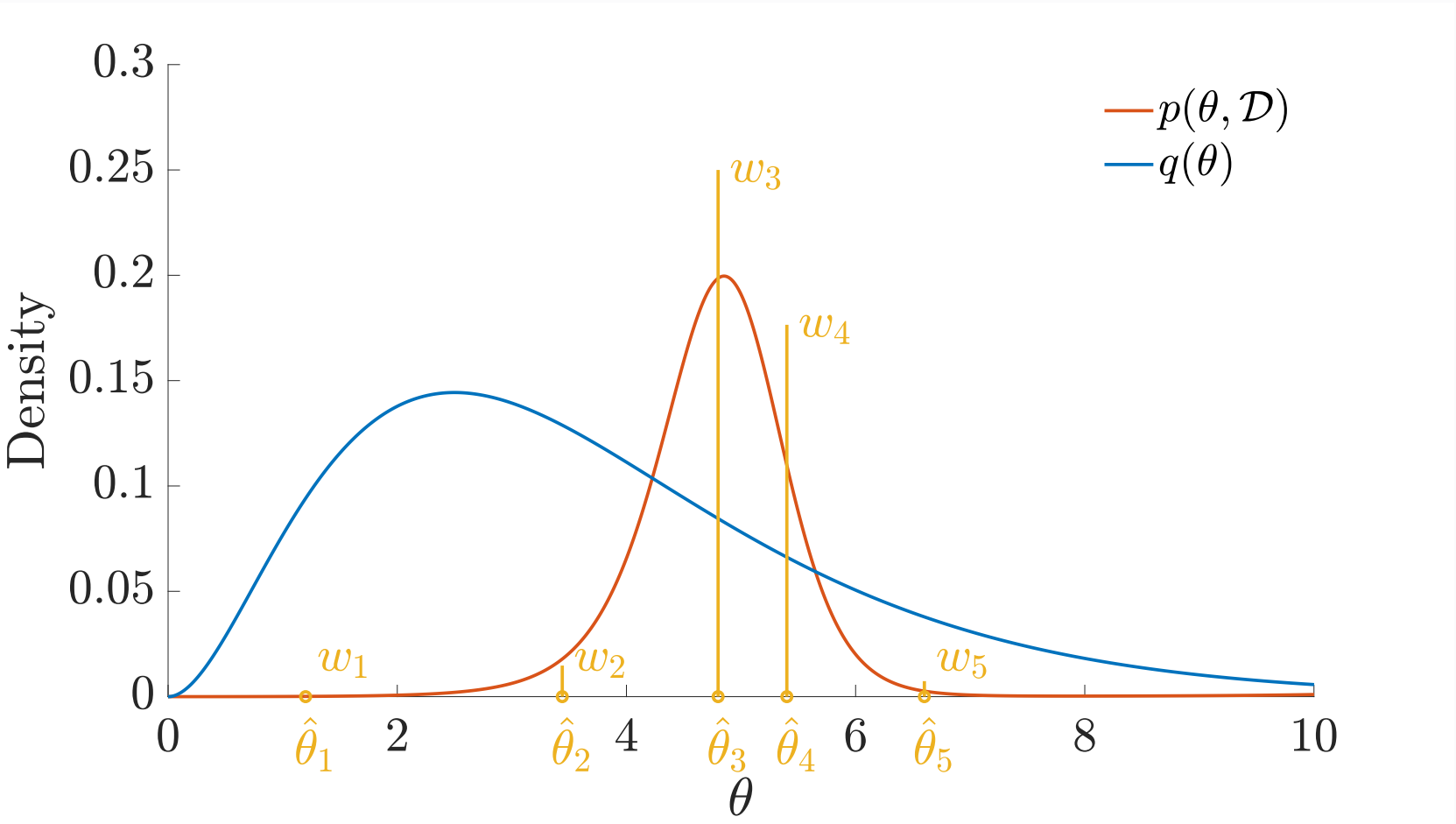
Importance Sampling Example



Importance Sampling Example



Importance Sampling Example



Importance Sampling Properties

Provided that $q(\theta)$ has **lighter tails** than $p(\theta|\mathcal{D})$, i.e. $q(\theta)/p(\theta|\mathcal{D}) > \epsilon,\ \forall\theta$ for some $\epsilon > 0$, then importance sampling provides an **unbiased** and **consistent** estimator for any integrable target function $f(\theta)$:

$$\mathbb{E}[\hat{\mu}_{\text{IS}}] = \mathbb{E}_{p(\theta|\mathcal{D})}\left[f(\theta)\right] \tag{12}$$

$$\text{Var}[\hat{\mu}_{\text{IS}}] = \frac{\text{Var}_{q(\theta)}[w\ f(\theta)]}{N} \tag{13}$$

Demonstrations of these results are given in the text

Self-Normalized Importance Sampling

- So far, we have assumed that we have access to a normalized version of the posterior $p(\theta|\mathcal{D})$
- Typically this will not be the case and we will only have access to an unnormalized target, namely the joint $p(\theta, \mathcal{D}) = p(\theta|\mathcal{D})p(\mathcal{D})$
- We can get around this by using $p(\theta, \mathcal{D})$ to define the weights, but then **self-normalizing** them
- The intuition is that doing this is using the importance samples to estimate both $p(\theta, \mathcal{D})$ and $p(\mathcal{D})$ (see the notes)
- Note that, unlike normal importance sampling, self-normalized importance sampling estimators are **biased**

Self-Normalized Importance Sampling (2)

The self-normalized importance sampling estimator is given by:

$$p(\theta|\mathcal{D}) \approx \hat{p}(\theta|\mathcal{D}) := \frac{\frac{1}{N}\sum_{n=1}^{N} w_n \delta_{\hat{\theta}_n}(\theta)}{\frac{1}{N}\sum_{n=1}^{N} w_n} = \sum_{n=1}^{N} \bar{w}_n \delta_{\hat{\theta}_n}(\theta)$$

$$\text{where} \quad \hat{\theta}_n \sim q(\theta), \quad w_n = \frac{p(\hat{\theta}_n, \mathcal{D})}{q(\hat{\theta}_n)}, \quad \bar{w}_n = \frac{w_n}{\sum_n w_n}$$

We can further use this to estimate expectations:

$$\mathbb{E}_{p(\theta|\mathcal{D})}\left[f(\theta)\right] \approx \sum_{n=1}^{N} \bar{w}_n f(\hat{\theta}_n)$$

Sequential Importance Sampling

- Importance weights are multiplicative when doing conditional sampling
- If we sample $\hat{\theta}_n \sim q_1(\theta)$ then $\hat{\phi}_n|\hat{\theta}_n \sim q_2(\phi|\hat{\theta}_n)$ when targeting $p(\theta|\mathcal{D})p(\phi|\theta,\mathcal{D})$ then the importance weight is

$$\frac{p(\hat{\theta}_n|\mathcal{D})p(\hat{\phi}_n|\hat{\theta}_n,\mathcal{D})}{q_1(\hat{\theta}_n)q_2(\hat{\phi}_n|\hat{\theta}_n)} = \frac{p(\hat{\theta}_n|\mathcal{D})}{q_1(\hat{\theta}_n)} \times \frac{p(\hat{\phi}_n|\hat{\theta}_n,D)}{q_2(\hat{\phi}_n|\hat{\theta}_n)} = w_{n,1} \times w_{n,2}.$$

- This is known as **sequential importance sampling** and means that we can propagate importance weighted samples through a computational system

Importance Sampling Pros and Cons

Many pros and cons are shared with rejection sampling

Pros

- By using all the samples from the proposal, can achieve lower variance estimates than rejection sampling from the same cost
- No need to find a constant scaling to bound the target (i.e. the C in rejection sampling)
- Can also be highly effective in low dimensions
- Self-normalization allows use with unnormalized targets
- Provides an unbiased marginal likelihood estimate by taking the average of the weights

Importance Sampling Pros and Cons (2)

Cons

- Also scales poorly to higher dimensions (more on this next lecture)
- Also requires a carefully designed proposals
- Samples are not exact
    - Self–normalization induces bias

Recap

- Bayesian inference is hard!
- Even if we can directly evaluate the posterior (which is rare), this may not be enough to characterize it and estimate expectations
- Monte Carlo methods give us a mechanism of representing distributions through samples
- Rejection sampling samples from an envelope of the target than only takes the samples that fall within it
- Importance sampling samples from a proposal and then assigns weights to the samples to account for them not being from the target

Next time: MCMC and variational methods

Further Reading

- The notes quite closely match the lecture with some extra details
- Chapters 1, 2, 7, and 9 of Art Owen's online book on Monte Carlo: `https://statweb.stanford.edu/~owen/mc/`
- Chapter 23 of K P Murphy. *Machine learning: a probabilistic perspective*. 2012
- M F Bugallo et al. "Adaptive importance sampling: the past, the present, and the future". In: *IEEE Signal Processing Magazine* (2017)
- David MacKay on Monte Carlo methods `http://videolectures.net/mackay_course_12/`