LECTURE 12: BAYESIAN INFERENCE AND MONTE CARLO METHODS

STAT 545: Intro. to Computational Statistics

Vinayak Rao

Purdue University

October 14, 2019

Given a set of observations X, MLE maximizes the likelihood:

$$\theta_{MLE} = \text{argmax}\ p(X|\theta)$$

Bayesian inference

Given a set of observations X, MLE maximizes the likelihood:

$$\theta_{MLE} = \text{argmax } p(X|\theta)$$

What if we believe θ is close to 0, is sparse, or is smooth?

Encode this with a 'prior' probability $p(\theta)$.

- Represents *a priori* beliefs about θ

Bayesian inference

Given a set of observations X, MLE maximizes the likelihood:

$$\theta_{MLE} = \text{argmax } p(X|\theta)$$

What if we believe θ is close to 0, is sparse, or is smooth?

Encode this with a 'prior' probability $p(\theta)$.

- Represents *a priori* beliefs about θ

Given observations, we can calculate the 'posterior':

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{P(X)}$$

Bayesian inference

Given a set of observations X, MLE maximizes the likelihood:

$$\theta_{MLE} = \operatorname{argmax}\, p(X|\theta)$$

What if we believe θ is close to 0, is sparse, or is smooth?

Encode this with a ‘prior’ probability $p(\theta)$.

- Represents *a priori* beliefs about θ

Given observations, we can calculate the ‘posterior’:

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{P(X)}$$

We can calculate the *maximum a posteriori* (MAP) solution:

$$\theta_{MAP} = \operatorname{argmax}\, p(\theta|X)$$

Given a set of observations X, MLE maximizes the likelihood:

$$\theta_{MLE} = \text{argmax } p(X|\theta)$$

What if we believe θ is close to 0, is sparse, or is smooth?

Encode this with a 'prior' probability $p(\theta)$.

- Represents *a priori* beliefs about θ

Given observations, we can calculate the 'posterior':

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{P(X)}$$

We can calculate the *maximum a posteriori* (MAP) solution:

$$\theta_{MAP} = \text{argmax } p(\theta|X)$$

Point estimate discards information about uncertainty in θ

Bayesian inference works with the entire distribution $p(\theta|X)$.

- Represents *a posteriori* beliefs about θ

Allows us to maintain and propagate uncertainty.

Bayesian inference

Bayesian inference works with the entire distribution $p(\theta|X)$.

- Represents *a posteriori* beliefs about θ

Allows us to maintain and propagate uncertainty.

E.g. consider the likelihood $p(X|\theta) = N(X|\theta, 1)$

- What is a good prior over θ?

Bayesian inference

Bayesian inference works with the entire distribution $p(\theta|X)$.

- Represents *a posteriori* beliefs about θ

Allows us to maintain and propagate uncertainty.

E.g. consider the likelihood $p(X|\theta) = N(X|\theta, 1)$

- What is a good prior over θ?
- What is a convenient prior over θ?

The posterior distribution $p(\theta|X) \propto p(X|\theta)p(\theta)$ summarizes all new information about θ provided by the data

In practice, these distributions are unwieldy.

Need approximations.

Bayesian inference

The posterior distribution $p(\theta|X) \propto p(X|\theta)p(\theta)$ summarizes all new information about θ provided by the data

In practice, these distributions are unwieldy.

Need approximations.

An exception: 'Conjugate priors' for exponential family distributions.

Conjugate exponential family priors

Let observations come from an exponential-family:

$$\begin{aligned} p(x|\theta) &= \frac{1}{Z(\theta)} h(x) \exp(\theta^\top \phi(x)) \\ &= h(x) \exp(\theta^\top \phi(x) - \zeta(\theta)) \quad \text{with } \zeta(\theta) = \log(Z(\theta)) \end{aligned}$$

Place a prior over θ:

Conjugate exponential family priors

Let observations come from an exponential-family:

$$p(x|\theta) = \frac{1}{Z(\theta)} h(x) \exp(\theta^\top \phi(x))$$
$$= h(x) \exp(\theta^\top \phi(x) - \zeta(\theta)) \quad \text{with } \zeta(\theta) = \log(Z(\theta))$$

Place a prior over θ:

$$p(\theta|a, b) \propto \eta(\theta) \exp(\theta^\top a - \zeta(\theta) b)$$

Conjugate exponential family priors

Let observations come from an exponential-family:

$$p(x|\theta) = \frac{1}{Z(\theta)} h(x) \exp(\theta^\top \phi(x))$$
$$= h(x) \exp(\theta^\top \phi(x) - \zeta(\theta)) \quad \text{with } \zeta(\theta) = \log(Z(\theta))$$

Place a prior over θ:

$$p(\theta|a, b) \propto \eta(\theta) \exp(\theta^\top a - \zeta(\theta) b)$$

Given a set of observations $X = \{x_1, \ldots, x_N\}$

$$p(\theta|X) \propto \left(\prod_{i=1}^{N} h(x_i) \exp(\theta^\top \phi(x_i) - \zeta(\theta)) \right) \eta(\theta) \exp(\theta^\top a - \zeta(\theta) b)$$

Conjugate exponential family priors

Let observations come from an exponential-family:

$$p(x|\theta) = \frac{1}{Z(\theta)} h(x) \exp(\theta^\top \phi(x))$$
$$= h(x) \exp(\theta^\top \phi(x) - \zeta(\theta)) \quad \text{with } \zeta(\theta) = \log(Z(\theta))$$

Place a prior over θ:

$$p(\theta|a, b) \propto \eta(\theta) \exp(\theta^\top a - \zeta(\theta) b)$$

Given a set of observations $X = \{x_1, \ldots, x_N\}$

$$p(\theta|X) \propto \left(\prod_{i=1}^{N} h(x_i) \exp(\theta^\top \phi(x_i) - \zeta(\theta)) \right) \eta(\theta) \exp(\theta^\top a - \zeta(\theta) b)$$
$$\propto \eta(\theta) \exp\left(\theta^\top \left(a + \sum_{i=1}^{N} \phi(x_i) \right) - \zeta(\theta)(b + N) \right)$$

Conjugate priors (contd.)

Prior over θ: exp. fam. distribution with parameters (a, b).

Posterior: same family with parameters $(a + \sum_{i=1}^{N} \phi(x_i), b + N)$.

Rare instance where analytical expressions for posterior exists.

In most cases a simple prior quickly leads to a complicated posterior, requiring Monte Carlo methods.

Conjugate priors (contd.)

Prior over θ: exp. fam. distribution with parameters (a, b).

Posterior: same family with parameters $(a + \sum_{i=1}^{N} \phi(x_i), b + N)$.

Rare instance where analytical expressions for posterior exists.

In most cases a simple prior quickly leads to a complicated posterior, requiring Monte Carlo methods.

Note the conjugate prior is an entire family of distributions.

- Actual distribution is chosen by setting the parameters (a, b) (a has the same dimension as ϕ, b is a scalar)
- These might be set by e.g. talking to a domain expert.

Let $x_i \in \{0, 1\}$ indicate if a new drug works for subject i.

The unknown probability of success is π: $x \sim \text{Bern}(\pi)$.

Let $x_i \in \{0, 1\}$ indicate if a new drug works for subject i.

The unknown probability of success is π: $x \sim \text{Bern}(\pi)$.

$$\begin{aligned}
p(x|\pi) &= \pi^{\mathbb{1}(x=1)}(1-\pi)^{\mathbb{1}(x=0)} \\
&= \exp\left(\mathbb{1}(x=1)\log(\pi) + (1-\mathbb{1}(x=1))\log(1-\pi)\right) \\
&= (1-\pi)\exp\left(\mathbb{1}(x=1)\log\frac{\pi}{1-\pi}\right) \\
&= \frac{1}{1+\exp(\theta)}\exp\left(\phi(x)\theta\right)
\end{aligned}$$

Let $x_i \in \{0, 1\}$ indicate if a new drug works for subject i.

The unknown probability of success is π: $x \sim \text{Bern}(\pi)$.

$$\begin{aligned} p(x|\pi) &= \pi^{\mathbb{1}(x=1)}(1-\pi)^{\mathbb{1}(x=0)} \\ &= \exp\left(\mathbb{1}(x=1)\log(\pi) + (1-\mathbb{1}(x=1))\log(1-\pi)\right) \\ &= (1-\pi)\exp\left(\mathbb{1}(x=1)\log\frac{\pi}{1-\pi}\right) \\ &= \frac{1}{1+\exp(\theta)}\exp\left(\phi(x)\theta\right) \end{aligned}$$

This is an exponential family distrib., with
$\theta = \log\frac{\pi}{1-\pi}, \phi(x) = \mathbb{1}(x=1), h(x) = 1, Z(\theta) = (1+\exp(\theta))$.

Conjugate priors: Beta-Bernoulli example

Let $x_i \in \{0, 1\}$ indicate if a new drug works for subject i.

The unknown probability of success is π: $x \sim \text{Bern}(\pi)$.

$$\begin{aligned} p(x|\pi) &= \pi^{\mathbb{1}(x=1)}(1-\pi)^{\mathbb{1}(x=0)} \\ &= \exp\left(\mathbb{1}(x=1)\log(\pi) + (1-\mathbb{1}(x=1))\log(1-\pi)\right) \\ &= (1-\pi)\exp\left(\mathbb{1}(x=1)\log\frac{\pi}{1-\pi}\right) \\ &= \frac{1}{1+\exp(\theta)}\exp\left(\phi(x)\theta\right) \end{aligned}$$

This is an exponential family distrib., with
$\theta = \log\frac{\pi}{1-\pi}, \phi(x) = \mathbb{1}(x=1), h(x) = 1, Z(\theta) = (1+\exp(\theta))$.
Defining $\zeta(\theta) = \log Z(\theta)$ as in the previous slide,

$$p(x|\theta) = \exp\left(\phi(x)\theta - \zeta(\theta)\right)$$

Conjugate priors: Beta-Bernoulli example

When $\theta = \log \frac{\pi}{1-\pi}$ is unknown, a Bayesian places a prior on it.

As before, define an exp. fam. prior with parameters $\vec{a}$:

$$p(\theta|\vec{a}) \propto \exp(a_1\theta + a_2\zeta(\theta))$$

When $\theta = \log \frac{\pi}{1-\pi}$ is unknown, a Bayesian places a prior on it.

As before, define an exp. fam. prior with parameters $\vec{a}$:

$$p(\theta|\vec{a}) \propto \exp(a_1\theta + a_2\zeta(\theta))$$

Then given data $X = (x_1, \ldots, x_N)$,

$$\begin{aligned} p(\theta|\vec{a}, X) &\propto p(\theta, X|\vec{a}) \\ &\propto \exp\left(\left(a_1 + \sum_{i=1}^{N} \mathbb{1}(x_i = 1)\right)\theta + (a_2 - N)\zeta(\theta)\right) \end{aligned}$$

When $\theta = \log \frac{\pi}{1-\pi}$ is unknown, a Bayesian places a prior on it.

As before, define an exp. fam. prior with parameters $\vec{a}$:

$$p(\theta|\vec{a}) \propto \exp(a_1\theta + a_2\zeta(\theta))$$

Then given data $X = (x_1, \ldots, x_N)$,

$$\begin{aligned} p(\theta|\vec{a}, X) &\propto p(\theta, X|\vec{a}) \\ &\propto \exp\left(\left(a_1 + \sum_{i=1}^{N} \mathbb{1}(x_i = 1)\right)\theta + (a_2 - N)\zeta(\theta)\right) \end{aligned}$$

Thus, the posterior is in the same family as the prior, but with updated parameters $\left(a_1 + \sum_{i=1}^{N} \mathbb{1}(x_i = 1), a_2 - N\right)$.

Looking at the prior more carefully, we see:

$$\begin{aligned} p(\theta|\vec{a}) &\propto \exp(a_1\theta + a_2\zeta(\theta)) \\ &\propto \exp\left(a_1 \log \frac{\pi}{1-\pi} + a_2 \log(1-\pi)\right) \\ &\propto \pi^{a_1}(1-\pi)^{(a_2-a_1)} \\ &= \pi^{b_1-1}(1-\pi)^{(b_2-1)} \end{aligned}$$

This is just the Beta(b_1, b_2) distribution, and you can check that the posterior is Beta$\left(b_1 + \sum_{i=1}^N \mathbb{1}(x_i = 1), b_2 + \sum_{i=1}^N \mathbb{1}(x_i = 0)\right)$.

Conjugate priors: Beta-Bernoulli example

Looking at the prior more carefully, we see:

$$\begin{aligned} p(\theta|\vec{a}) &\propto \exp(a_1\theta + a_2\zeta(\theta)) \\ &\propto \exp\left(a_1 \log\frac{\pi}{1-\pi} + a_2\log(1-\pi)\right) \\ &\propto \pi^{a_1}(1-\pi)^{(a_2-a_1)} \\ &= \pi^{b_1-1}(1-\pi)^{(b_2-1)} \end{aligned}$$

This is just the Beta(b_1, b_2) distribution, and you can check that the posterior is Beta$\left(b_1 + \sum_{i=1}^N \mathbb{1}(x_i = 1), b_2 + \sum_{i=1}^N \mathbb{1}(x_i = 0)\right)$.

b_1 and b_2 are sometimes called pseudo-observations, and capture our prior beliefs: before seeing any x's our prior is as if we saw b_1 successes and b_2 failures. After seeing data, we factor actual observations into the pseudo-observations.

MONTE CARLO METHODS

What about the situation when the posterior $p(\theta|X)$ is no longer simple/available in closed form?

What information about $p(\theta|X)$ do we really need?

Typically, expectations of different functions g:

$$\mathbb{E}_{\theta|X}[g] = \int \mathrm{d}\theta g(\theta) p(\theta|X)$$

What about the situation when the posterior $p(\theta|X)$ is no longer simple/available in closed form?

What information about $p(\theta|X)$ do we really need?

Typically, expectations of different functions g:

$$\mathbb{E}_{\theta|X}[g] = \int \mathrm{d}\theta g(\theta) p(\theta|X)$$

What is g for to calculate 1) mean, 2) variance, 3) $p(\theta > 10|X)$?

Monte Carlo integration

Let us forget the posterior distribution $p(\theta|X)$, and consider some general probability distribution $p(x)$. We want

$$\mu := \mathbb{E}_p[g] = \int_{\mathcal{X}} g(x)p(x)\mathrm{d}x$$

Let us forget the posterior distribution $p(\theta|X)$, and consider some general probability distribution $p(x)$. We want

$$\mu := \mathbb{E}_p[g] = \int_{\mathcal{X}} g(x)p(x)\mathrm{d}x$$

Sampling approximation: rather than visit all points in $\mathcal{X}$, calculate a summation over a finite set.

$$\mu \approx \frac{1}{N}\sum_{i=1}^{N} g(x_i) := \hat{\mu}$$

Monte Carlo integration

Let us forget the posterior distribution $p(\theta|X)$, and consider some general probability distribution $p(x)$. We want

$$\mu := \mathbb{E}_p[g] = \int_{\mathcal{X}} g(x)p(x)\mathrm{d}x$$

Sampling approximation: rather than visit all points in $\mathcal{X}$, calculate a summation over a finite set.

$$\mu \approx \frac{1}{N}\sum_{i=1}^{N} g(x_i) := \hat{\mu}$$

Monte Carlo approximation:

- Obtain points by sampling from $p(x)$: $x_i \sim p$
- Approximate integration with summation

$$\hat{\mu} \approx \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

Monte Carlo integration

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} g(x_i)$$

If $x_i \sim p$,

$$\mathbb{E}_p[\hat{\mu}] = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_p[g] = \mu$$

Unbiased estimate

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

If $x_i \sim p$,

$$\mathbb{E}_p[\hat{\mu}] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_p[g] = \mu$$ Unbiased estimate

$$\mathrm{Var}_p[\hat{\mu}] = \frac{1}{N}\mathrm{Var}_p[g],$$ Error = StdDev $\propto N^{-1/2}$

$$\frac{1}{N}\sum_{i=1}^{N} f \to \mathbb{E}_p(g) = \mu \quad \text{as } N \to \infty$$ Consistent estimate (LLN)

Is this a good idea?

- In low-dims, worth considering numerical methods like quadrature. In high-dims, these quickly become infeasible.

Monte Carlo Sampling

Is this a good idea?

- In low-dims, worth considering numerical methods like quadrature. In high-dims, these quickly become infeasible. Simpson's rule in d-dimensions, with N grid points:

$$\text{error} \propto N^{-4/d}$$

  Monte Carlo integration:

$$\text{error} \propto N^{-1/2}$$

Is this a good idea?

- In low-dims, worth considering numerical methods like quadrature. In high-dims, these quickly become infeasible. Simpson's rule in d-dimensions, with N grid points:

$$\text{error} \propto N^{-4/d}$$

Monte Carlo integration:

$$\text{error} \propto N^{-1/2}$$

Independent of dimensionality!

Monte Carlo Sampling

Is this a good idea?

- In low-dims, worth considering numerical methods like quadrature. In high-dims, these quickly become infeasible. Simpson's rule in d-dimensions, with N grid points:

  $$\text{error} \propto N^{-4/d}$$

  Monte Carlo integration:

  $$\text{error} \propto N^{-1/2}$$

  Independent of dimensionality!
- If unbiasedness is important to you.
- Very simple.
- Very modular: easily incorporated into more complex models (Gibbs sampling)

An aside: Monte Carlo should be your method of last resort!

Don't hesitate using numerical integration

- Numerical integration can be much faster and more accurate

Contrast

```
> integrate(function(x) x * exp(-x), lower = 0, upper = Inf)
```

with

```
> mean(rexp(1000))
```

Generating random variables

- The simplest useful probability distribution $\texttt{Unif}(0, 1)$.
- In theory, can be used to generate any other RV.
- Easy to generate uniform RVs on a deterministic computer?

Generating random variables

- The simplest useful probability distribution `Unif`$(0, 1)$.
- In theory, can be used to generate any other RV.
- Easy to generate uniform RVs on a deterministic computer?

No!

Generating random variables

- The simplest useful probability distribution `Unif`$(0, 1)$.
- In theory, can be used to generate any other RV.
- Easy to generate uniform RVs on a deterministic computer?

No!

- Instead: *pseudorandom* numbers.

Generating random variables

- The simplest useful probability distribution `Unif`$(0, 1)$.
- In theory, can be used to generate any other RV.
- Easy to generate uniform RVs on a deterministic computer?

No!

- Instead: *pseudorandom* numbers.
- Map a seed to a 'random-looking' sequence.

Generating random variables

- The simplest useful probability distribution `Unif`$(0, 1)$.
- In theory, can be used to generate any other RV.
- Easy to generate uniform RVs on a deterministic computer?

No!

- Instead: *pseudorandom* numbers.
- Map a seed to a 'random-looking' sequence.
- Downside: http://boallen.com/random-numbers.html

Generating random variables

- The simplest useful probability distribution `Unif`$(0, 1)$.
- In theory, can be used to generate any other RV.
- Easy to generate uniform RVs on a deterministic computer?

No!

- Instead: *pseudorandom* numbers.
- Map a seed to a 'random-looking' sequence.
- Downside: http://boallen.com/random-numbers.html
- Upside: Can use seeds for reproducibility or debugging

```
> set.seed(1)
```

Generating random variables

- The simplest useful probability distribution `Unif`$(0, 1)$.
- In theory, can be used to generate any other RV.
- Easy to generate uniform RVs on a deterministic computer?

No!

- Instead: *pseudorandom* numbers.
- Map a seed to a 'random-looking' sequence.
- Downside: http://boallen.com/random-numbers.html
- Upside: Can use seeds for reproducibility or debugging

```
> set.seed(1)
```

- Careful with batch/parallel processing.

Generating random variables

`R` has a bunch of random number generators.

`rnorm`, `rgamma`, `rbinom`, `rexp`, `rpoiss` etc.

What if we want samples from some other distribution?

Inverse transform sampling

Let X have pdf $p(x)$, and cdf $F(x) = P(X \leq x) = \int_{-\infty}^{x} p(u)\mathrm{d}u$

Let:

$$X \sim p(\cdot)$$

$$U = F(X)$$

Inverse transform sampling

Let X have pdf $p(x)$, and cdf $F(x) = P(X \leq x) = \int_{-\infty}^{x} p(u) \mathrm{d}u$

Let:

$$X \sim p(\cdot)$$
$$U = F(X)$$

Then U is $\texttt{Unif}(0, 1)$

Inverse transform sampling

Let X have pdf $p(x)$, and cdf $F(x) = P(X \leq x) = \int_{-\infty}^{x} p(u)\mathrm{d}u$

Let:

$$X \sim p(\cdot)$$
$$U = F(X)$$

Then U is $\texttt{Unif}(0, 1)$

Equivalently, sample $U \sim \texttt{Unif}(0, 1)$, and let $X = F^{-1}(U)$
Then $X \sim p(\cdot)$ (see wikipedia for proof)

Inverse transform sampling

Let X have pdf $p(x)$, and cdf $F(x) = P(X \leq x) = \int_{-\infty}^{x} p(u)\mathrm{d}u$

Let:

$$X \sim p(\cdot)$$
$$U = F(X)$$

Then U is $\texttt{Unif}(0,1)$

Equivalently, sample $U \sim \texttt{Unif}(0,1)$, and let $X = F^{-1}(U)$
Then $X \sim p(\cdot)$ (see wikipedia for proof)

E.g. $-\log(U)$ is $\texttt{Exponential}(1)$.

Inverse transform sampling

Let X have pdf $p(x)$, and cdf $F(x) = P(X \leq x) = \int_{-\infty}^{x} p(u)\mathrm{d}u$

Let:

$$X \sim p(\cdot)$$
$$U = F(X)$$

Then U is $\texttt{Unif}(0,1)$

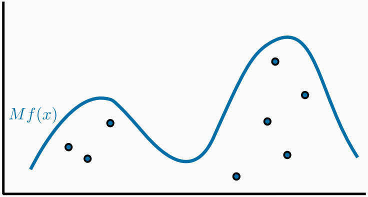
Equivalently, sample $U \sim \texttt{Unif}(0,1)$, and let $X = F^{-1}(U)$
Then $X \sim p(\cdot)$ (see wikipedia for proof)

E.g. $-\log(U)$ is $\texttt{Exponential}(1)$.
Usually hard to compute F^{-1}.

Rejection sampling

Let $p(x) = \frac{f(x)}{Z}$.

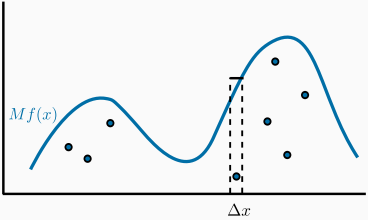
Probability of a sample in $[x_0, x_0 + \Delta x] = p(x_0)\Delta x$.



If we sample points uniformly below the curve $Mf(x)$:

Rejection sampling

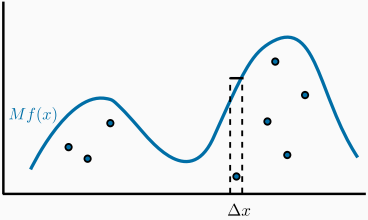
Let $p(x) = \frac{f(x)}{Z}$.
Probability of a sample in $[x_0, x_0 + \Delta x] = p(x_0)\Delta x$.



If we sample points uniformly below the curve $Mf(x)$:
Probability of a sample in $[x_0, x_0 + \Delta x] = \frac{Mf(x_0)\Delta X}{\int_X Mf(x_0)\mathrm{d}x} = p(x_0)\Delta x$.

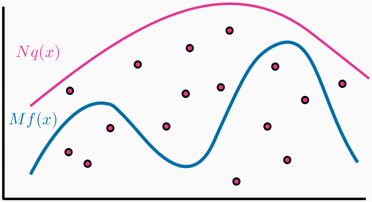
Rejection sampling

Let $p(x) = \frac{f(x)}{Z}$.
Probability of a sample in $[x_0, x_0 + \Delta x] = p(x_0)\Delta x$.



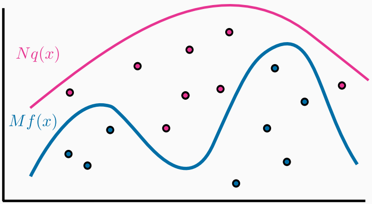
If we sample points uniformly below the curve $Mf(x)$:
Probability of a sample in $[x_0, x_0 + \Delta x] = \frac{Mf(x_0)\Delta X}{\int_X Mf(x_0)\mathrm{d}x} = p(x_0)\Delta x$.
How to do this (without sampling from p)?

Let $p(x) = \frac{f(x)}{Z}$.
Probability of a sample in $[x_0, x_0 + \Delta x] = p(x_0)\Delta x$.



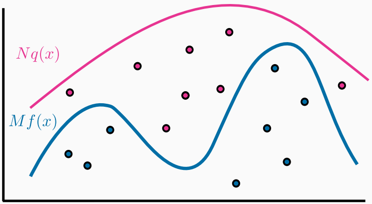
If $Mf(x) \leq Nq(x)\ \forall x$ for constant N and distribution $q(\cdot)$
Sample points uniformly under $Nq(x)$.
(sample $x_0 \sim q(\cdot)$, and assign it a uniform height in $[0, Nq(x_0)]$

Let $p(x) = \frac{f(x)}{Z}$.
Probability of a sample in $[x_0, x_0 + \Delta x] = p(x_0)\Delta x$.



If $Mf(x) \leq Nq(x)\ \forall x$ for constant N and distribution $q(\cdot)$
Sample points uniformly under $Nq(x)$.
Keep only points under $Mf(x)$.

Let $p(x) = \frac{f(x)}{Z}$.
Probability of a sample in $[x_0, x_0 + \Delta x] = p(x_0)\Delta x$.



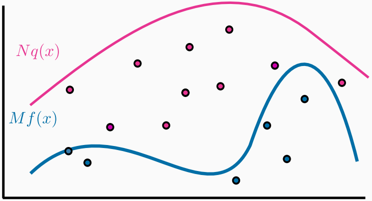
Equivalent algorithm: (convince yourself)

- Propose $x^* \sim q(\cdot)$
- Accept with probability $Mf(x^*)/Nq(x^*)$

Rejection sampling

Let $p(x) = \frac{f(x)}{Z}$.

Probability of a sample in $[x_0, x_0 + \Delta x] = p(x_0)\Delta x$.



We need a bound on $f(x)$.

A loose bound leads to lots of rejections.

Probability of acceptance = $\frac{MZ}{N}$.

Intractable normalization constants

A probability density takes the form $p(x) = \frac{f(x)}{Z}$

- $Z = \int_{\mathcal{X}} f(x) \mathrm{d}x$ is the normalization contant
- Ensures probability integrates to 1

Intractable normalization constants

A probability density takes the form $p(x) = \frac{f(x)}{Z}$

- $Z = \int_{\mathcal{X}} f(x) \mathrm{d}x$ is the normalization contant
- Ensures probability integrates to 1

Often Z is difficult to calculate (intractable integral over $f(x)$)

Consequently, evaluating $p(x)$ is hard

Intractable normalization constants

A probability density takes the form $p(x) = \frac{f(x)}{Z}$

- $Z = \int_{\mathcal{X}} f(x) \mathrm{d}x$ is the normalization contant
- Ensures probability integrates to 1

Often Z is difficult to calculate (intractable integral over $f(x)$)

Consequently, evaluating $p(x)$ is hard

However, rejection sampling doesn't need Z or $p(x)$

Example 1:

$$p(x) \propto \exp(-x^2/2)|\sin(x)|$$

Example 2 (truncated normal):

$$p(x) \propto \exp(-x^2/2)1_{\{x>c\}}$$

What is M for each case? What can we say about efficiency?

Importance Sampling

Rather that accept/reject, assign weights to samples.

Importance Sampling

Rather that accept/reject, assign weights to samples. Observe:

$$\mathbb{E}_p[g] = \int g(x)p(x)\mathrm{d}x = \int g(x)\frac{p(x)}{q(x)}q(x)\mathrm{d}x = \mathbb{E}_q\left[\frac{g(x)p(x)}{q(x)}\right]$$

Importance Sampling

Rather that accept/reject, assign weights to samples. Observe:

$$\mathbb{E}_p[g] = \int g(x)p(x)\mathrm{d}x = \int g(x)\frac{p(x)}{q(x)}q(x)\mathrm{d}x = \mathbb{E}_q\left[\frac{g(x)p(x)}{q(x)}\right]$$

Use Monte Carlo approximation to the latter expectation:

- Draw proposal x from $q(\cdot)$ and calculate weight $w(x) = \frac{p(x)}{q(x)}$.

$$\int g(x)p(x)\mathrm{d}x \approx \frac{1}{N}\sum_{s=1}^{N} w(x_s)g(x_s)$$

Importance Sampling

Rather that accept/reject, assign weights to samples. Observe:

$$\mathbb{E}_p[g] = \int g(x)p(x)\mathrm{d}x = \int g(x)\frac{p(x)}{q(x)}q(x)\mathrm{d}x = \mathbb{E}_q\left[\frac{g(x)p(x)}{q(x)}\right]$$

Use Monte Carlo approximation to the latter expectation:

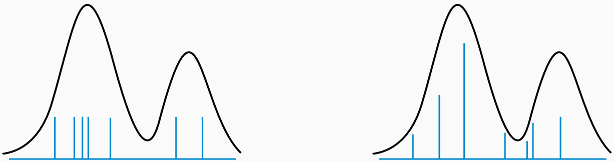
- Draw proposal x from $q(\cdot)$ and calculate weight $w(x) = \frac{p(x)}{q(x)}$.

$$\int g(x)p(x)\mathrm{d}x \approx \frac{1}{N}\sum_{s=1}^{N} w(x_s)g(x_s)$$

Since $w(x) = p(x)/q(x) = \frac{f(x)}{Zq(x)}$:

- We don't need a bounding envelope.
- We need normalizn constant Z (but see later).

Importance sampling vs simple Monte Carlo



Simple Monte Carlo/MCMC (left) uses sampling approximation

Importance sampling (right) weights the samples

Importance Sampling (contd)

When does this make sense?

Sometimes it's easier to simulate from $q(x)$ than $p(x)$.

Importance Sampling (contd)

When does this make sense?
Sometimes it's easier to simulate from $q(x)$ than $p(x)$.

Sometimes it's better to simulate from $q(x)$ than $p(x)$!

To reduce variance. E.g. rare event simulation.

Importance Sampling (contd)

When does this make sense?

Sometimes it's easier to simulate from $q(x)$ than $p(x)$.

Sometimes it's better to simulate from $q(x)$ than $p(x)$!

To reduce variance. E.g. rare event simulation.

Let $x \sim (0, 1)$

- What is $P(X > 5)$?

Let $X = (x_1, \ldots, x_{100})$ be a hundred dice.
What is $p(\sum x_i \geq 550)$?

Let $X = (x_1, \ldots, x_{100})$ be a hundred dice.
What is $p(\sum x_i \geq 550)$?

Rejection sampling (from $p(x)$) leads to high rejection.

Let $X = (x_1, \dots, x_{100})$ be a hundred dice.
What is $p(\sum x_i \geq 550)$?

Rejection sampling (from $p(x)$) leads to high rejection.

A better choice might be to bias the dice.
E.g. $q(x_i = v) \propto v$ (for $v \in \{1, \dots 6\}$)

Importance sampling:

Define $S_X = \sum x_i$

$$p(S \geq 550) = \sum_{y \in \text{ all configs of 100 dice}} \delta(\sum y \geq 550) p(y)$$

$$= \sum_{y \in \text{ all configs of 100 dice}} \frac{p(y)}{q(y)} \delta(\sum y \geq 550) q(y)$$

For a proposal $X^* \sim q$,

$$w(X^*) = \frac{p(X^*}{q(X^*)} = \frac{(1/6)^{100}}{\prod_i q(x_i^*)}$$

Use approximation $p(S \geq 550) \approx \sum_{j=1}^{N} w(X_j) \delta(\sum x_i^j \geq 550)$

Importance sampling (contd)

What is the variance of the estimate?

Importance sampling (contd)

What is the variance of the estimate?

$$\begin{aligned}\text{Var}[\mu_{imp}] &= \mathbb{E}[\mu_{imp}^2] - \mu^2 \\ &= \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N} w_i g(x_i)\right)^2\right] - \mu^2\end{aligned}$$

Importance sampling (contd)

What is the variance of the estimate?

$$\begin{aligned}\text{Var}[\mu_{imp}] &= \mathbb{E}[\mu_{imp}^2] - \mu^2 \\ &= \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N} w_i g(x_i)\right)^2\right] - \mu^2 \\ &= \mathbb{E}\left[\left(\frac{p(x)g(x)}{q(x)}\right)^2\right] - \mu^2\end{aligned}$$

What is the variance of the estimate?

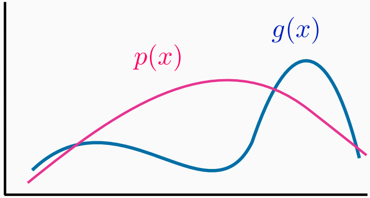
$$\begin{aligned}
\mathrm{Var}[\mu_{imp}] &= \mathbb{E}[\mu_{imp}^2] - \mu^2 \\
&= \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N} w_i g(x_i)\right)^2\right] - \mu^2 \\
&= \mathbb{E}\left[\left(\frac{p(x)g(x)}{q(x)}\right)^2\right] - \mu^2 \\
&= \int_{\mathcal{X}} q(x)\left(\frac{p(x)g(x)}{q(x)}\right)^2 \mathrm{d}x - \mu^2
\end{aligned}$$

What is the variance of the estimate?

$$\begin{aligned}
\mathrm{Var}[\mu_{imp}] &= \mathbb{E}[\mu_{imp}^2] - \mu^2 \\
&= \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N} w_i g(x_i)\right)^2\right] - \mu^2 \\
&= \mathbb{E}\left[\left(\frac{p(x)g(x)}{q(x)}\right)^2\right] - \mu^2 \\
&= \int_{\mathcal{X}} q(x)\left(\frac{p(x)g(x)}{q(x)}\right)^2 \mathrm{d}x - \mu^2 \\
&\geq \left(\int_{\mathcal{X}} q(x)\frac{p(x)g(x)}{q(x)}\mathrm{d}x\right)^2 - \mu^2
\end{aligned}$$

What is the variance of the estimate?

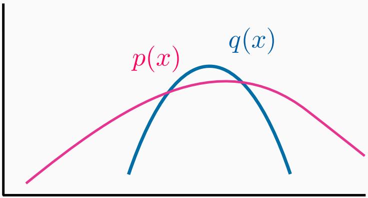
$$\begin{aligned}
\mathrm{Var}[\mu_{imp}] &= \mathbb{E}[\mu_{imp}^2] - \mu^2 \\
&= \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N} w_i g(x_i)\right)^2\right] - \mu^2 \\
&= \mathbb{E}\left[\left(\frac{p(x)g(x)}{q(x)}\right)^2\right] - \mu^2 \\
&= \int_{\mathcal{X}} q(x)\left(\frac{p(x)g(x)}{q(x)}\right)^2 \mathrm{d}x - \mu^2 \\
&\geq \left(\int_{\mathcal{X}} q(x)\frac{p(x)g(x)}{q(x)}\mathrm{d}x\right)^2 - \mu^2 \\
&= 0 \qquad\qquad (!)
\end{aligned}$$



We achieve this lower bound when $q(x) \propto p(x)g(x)$.
A slightly useless result, because

$$q(x) = \frac{p(x)g(x)}{\int_{\mathcal{X}} p(x)g(x)\mathrm{d}x}$$

requires solving the integral we care about.



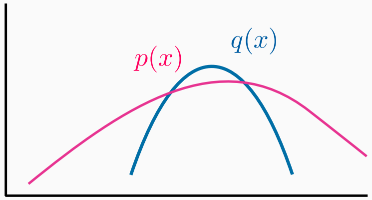
We want a small variance in the weights $w(x_i)$.

Easy to check at $\mathbb{E}_q[w(x)] = 1$.

$$\mathrm{Var}_q[w(x)] = \mathbb{E}_q[w(x)^2] - \mathbb{E}_q[w(x)]^2$$
$$= \int_{\mathcal{X}} \left(\frac{p(x)}{q(x)}\right)^2 q(x)\mathrm{d}x - 1 \qquad = \int_{\mathcal{X}} \frac{p(x)^2}{q(x)}\mathrm{d}x - 1$$

Can be unbounded. E.g. $p = \mathcal{N}(0, 2)$ and $q = \mathcal{N}(0, 1)$.

Importance sampling (contd)



A popular diagnosis statistic: effective sample size (ESS).

$$ESS = \frac{\left(\sum_{i=1}^{N} w(x_i)\right)^2}{\sum_{i=1}^{N} w(x_i)^2}$$

Small ESS $\rightarrow$ Large variability in w's $\rightarrow$ bad estimate.

Large ESS promises you nothing!

Importance weights are $w(x) = p(x)/q(x)$, where $p(x) = f(x)/Z$.

Importance sampling when Z is unknown

Importance weights are $w(x) = p(x)/q(x)$, where $p(x) = f(x)/Z$.

How can we estimate $Z = \int f(x) \mathrm{d}x$?

Importance sampling when Z is unknown

Importance weights are $w(x) = p(x)/q(x)$, where $p(x) = f(x)/Z$.

How can we estimate $Z = \int f(x)\mathrm{d}x$?

$$Z = \int f(x)\mathrm{d}x = \int \frac{f(x)q(x)}{q(x)}\mathrm{d}x$$

Importance sampling when Z is unknown

Importance weights are $w(x) = p(x)/q(x)$, where $p(x) = f(x)/Z$.

How can we estimate $Z = \int f(x)\mathrm{d}x$?

$$Z = \int f(x)\mathrm{d}x = \int \frac{f(x)q(x)}{q(x)}\mathrm{d}x$$

Reuse samples from the proposal distribution $q(x)$:

$$\hat{Z} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{q(x_i)} = \frac{1}{N}\sum_{i=1}^{N}\tilde{w}(x_i)$$

Can use to approximate importance sampling weights $w(x_i)$:

$$w(x_i) = \frac{p(x_i)}{q(x_i)} = \frac{f(x_i)}{Zq(x_i)} \approx \frac{1}{\hat{Z}}\tilde{w}(x_i)$$

Importance sampling when Z is unknown

Importance weights are $w(x) = p(x)/q(x)$, where $p(x) = f(x)/Z$.

How can we estimate $Z = \int f(x)\mathrm{d}x$?

$$Z = \int f(x)\mathrm{d}x = \int \frac{f(x)q(x)}{q(x)}\mathrm{d}x$$

Reuse samples from the proposal distribution $q(x)$:

$$\hat{Z} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{q(x_i)} = \frac{1}{N}\sum_{i=1}^{N}\tilde{w}(x_i)$$

Can use to approximate importance sampling weights $w(x_i)$:

$$w(x_i) = \frac{p(x_i)}{q(x_i)} = \frac{f(x_i)}{Zq(x_i)} \approx \frac{1}{\hat{Z}}\tilde{w}(x_i)$$

Use $\tilde{w}(x)$ instead of $w(x)$ in the Monte Carlo approximation.

Importance sampling when Z is unknown

Importance weights are $w(x) = p(x)/q(x)$, where $p(x) = f(x)/Z$.

How can we estimate $Z = \int f(x)\mathrm{d}x$?

$$Z = \int f(x)\mathrm{d}x = \int \frac{f(x)q(x)}{q(x)}\mathrm{d}x$$

Reuse samples from the proposal distribution $q(x)$:

$$\hat{Z} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{q(x_i)} = \frac{1}{N}\sum_{i=1}^{N}\tilde{w}(x_i)$$

Can use to approximate importance sampling weights $w(x_i)$:

$$w(x_i) = \frac{p(x_i)}{q(x_i)} = \frac{f(x_i)}{Zq(x_i)} \approx \frac{1}{\hat{Z}}\tilde{w}(x_i)$$

Use $\tilde{w}(x)$ instead of $w(x)$ in the Monte Carlo approximation.

Is biased for finite N, but consistent as $N \to \infty$.