Sampling Methods

Nipun Batra

August 31, 2023

IIT Gandhinagar

Topics

1. Discovery that transformed Pi
2. Monte Carlo Simulation
3. Rejection Sampling
4. Inverse CDF
5. Importance Sampling
6. Gibbs Sampling
7. Markov Chain Monte Carlo

Discovery that transformed Pi

The Discovery That Transformed Pi

Monte Carlo Simulation

The idea behind MC Simulation

- We often want to compute expected value of some function of a random variable, which turns into the integral,

$$\mathbb{E}\left[f(x)\right] = \int f(x)p(x)dx$$

where $x \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R}^m$ and $p(x)$ is the target distribution.

- In low dimensions, we can use numerical integration techniques to compute the above integral. However, in high dimensions, this is not feasible.
- Alternative approach is to draw multiple random samples, $x_i \sim p(x)$ and compute

$$\mathbb{E}\left[f(x)\right] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

General Form

The general form of Monte Carlo methods is: The expectation of a function $f(x)$ with respect to a distribution $p(x)$ is given by:

$$\mathbb{E}_{x\sim p(x)}[f(x)] = \int f(x)p(x)dx \tag{1}$$

Using Monte Carlo methods, we can estimate the above expectation by sampling x_i from $p(x)$ and computing the average of $f(x_i)$.

$$\mathbb{E}_{x\sim p(x)}[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{2}$$

where $x_i \sim p(x)$.

Unbiased Estimator?

Is Monte Carlo Sampling a biased or unbiased estimator?

We know:

$$\mathbb{E}_{x\sim p(x)}[f(x)] = \int f(x)p(x)dx = \phi \tag{3}$$

Let $x_i \in 1, \ldots, N$ be i.i.d samples:

$$\hat{\phi} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

$$\mathbb{E}(\hat{\phi}) = \int \frac{1}{N}\sum_{i=1}^{N} f(x_i)p(x_i)dx = \frac{1}{N}\sum_{i=1}^{N}\int f(x_i)p(x_i)dx$$

$$= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}(f(x_i)) = \phi$$

Thus, it is an unbiased estimator!

Estimating Pi using Monte Carlo (Part 1)

We can estimate the value of pi using Monte Carlo methods by considering a unit square with a quarter circle inscribed within it.

- Let $p(x)$ be defined over the unit square using the uniform distribution in two dimensions, i.e., $p(x) = U(x) = 1$ for $x \in [0, 1]^2$.
- Let $f(x)$ be the indicator function defined as follows:

$$f(x) = \begin{cases} \text{Green}(1), & \text{if } x \text{ falls inside the quarter circle,} \\ \text{Red}(0), & \text{otherwise.} \end{cases}$$

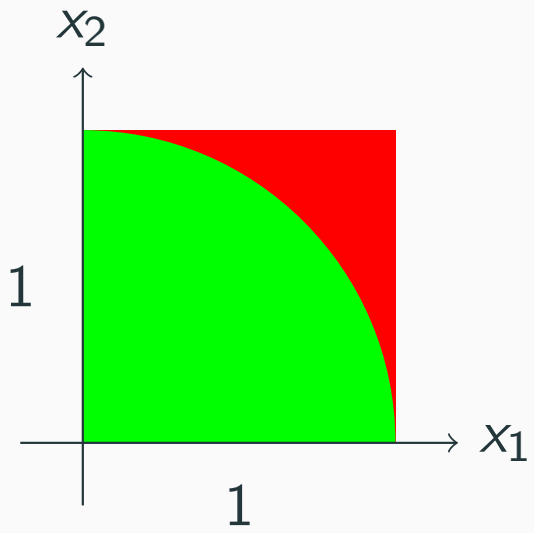
Estimating Pi using Monte Carlo (Part 1)

- Or, we can write $f(x)$ to be the following:

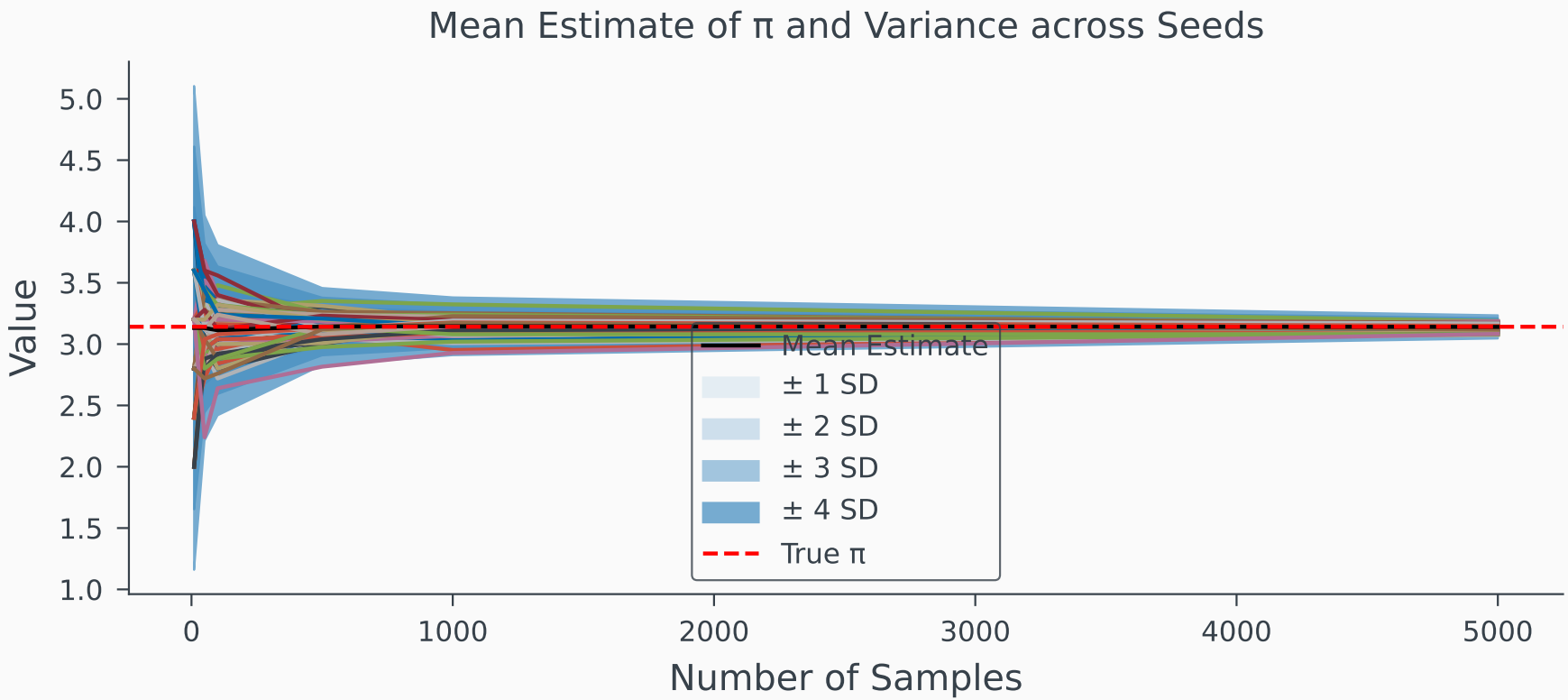
$$f(x) = \begin{cases} 1, & \text{if } x_1^2 + x_2^2 \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

- Or, using the indicator function, we can write $f(x)$ to be the following:

$$f(x) = \mathbb{I}(x_1^2 + x_2^2 \leq 1)$$

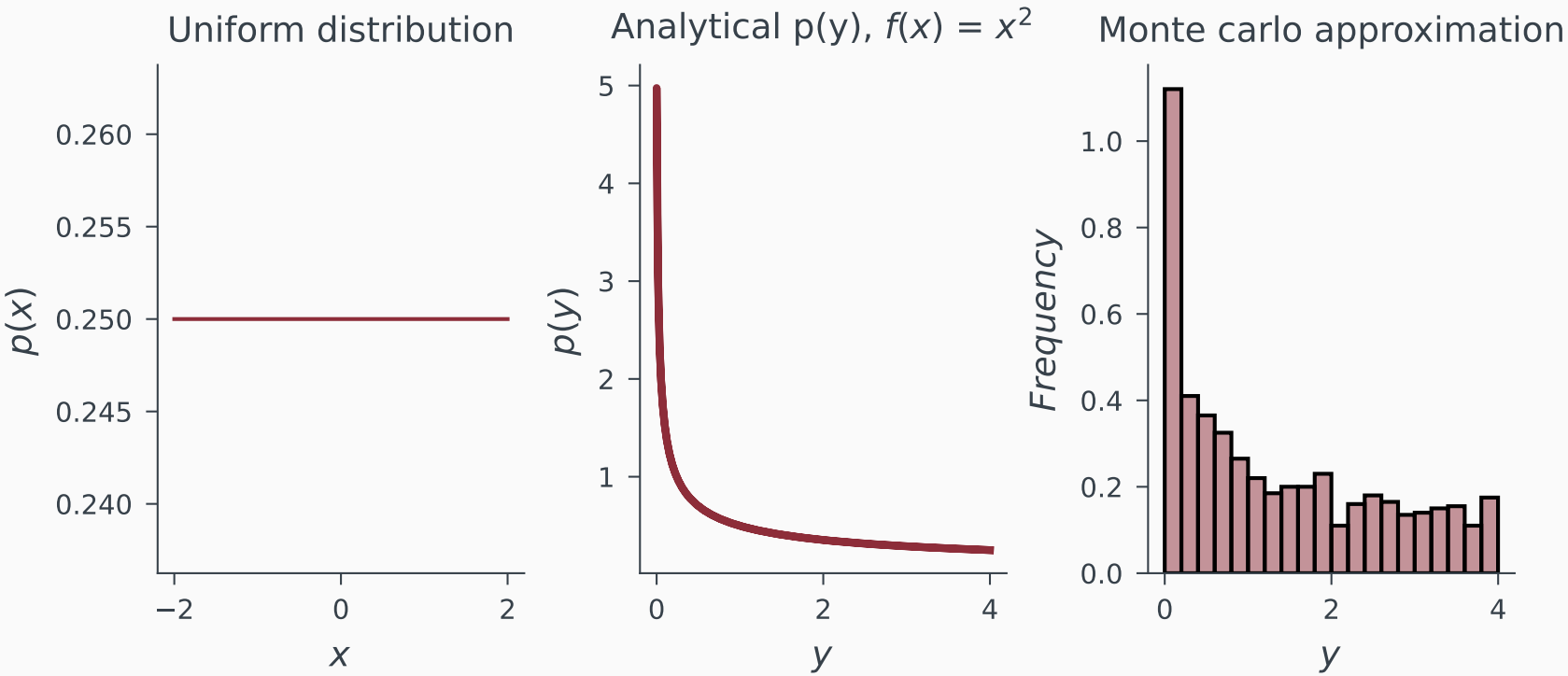


$$\frac{\pi}{4} \approx \frac{\text{Green area}}{\text{Green area} + \text{Red area}}$$


Mean Estimate of π and Variance across Seeds
Value
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0
1000
2000
3000
4000
5000
Number of Samples
Mean Estimate
± 1 SD
± 2 SD
± 3 SD
± 4 SD
True π

Estimating a function using Monte Carlo

Let $x \in \mathcal{U}(-1, 1)$ and $y = f(x) = x^2$.



Sampling converges slowly

The expected square error of the Monte Carlo estimate is given by:

$$\mathbb{E}\left(\hat{\phi} - \mathbb{E}(\hat{\phi})\right)^2 = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}(f(x_i) - \phi)\right]^2$$

$$= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbb{E}(f(x_i)f(x_j)) - \phi\mathbb{E}(f(x_i)) - \mathbb{E}(f(x_j))\phi + \phi^2$$

$$= \frac{1}{N^2}\sum_{i=1}^{N}\left(\left(\sum_{i\neq j}\phi^2 - 2\phi^2 + \phi^2\right) + \mathbb{E}(f^2) - \phi^2\right) = \frac{1}{N}\mathbb{V}(f)$$

$$\therefore \mathbb{E}\left(\hat{\phi} - \mathbb{E}(\hat{\phi})\right)^2 = \mathcal{O}(N^{-1})$$

Thus, the expected error drops as $\mathcal{O}(N^{-\frac{1}{2}})$.

Pop Quiz

How many samples (N) do we need to reach single-precision (i.e., $\sim 10^{-7}$)?

Is sampling easy?

Many reasons contribute to sampling not always being easy in higher dimensions. For example,

- need a global description of the entire function
- need to know probability densities everywhere
- need to know regions of high density

Estimating prior predictive distribution

- Let $p(\theta)$ be the prior distribution of parameter $\theta \in \mathbb{R}^2$. Say, for example, $p(\theta_i) = \mathcal{N}(0, 1) \forall i$.
- Let $p(y|\theta, x)$ be the likelihood function. Say, for example, $p(y|\theta, x) = \mathcal{N}(\theta_0 + \theta_1 x, 1)$.
- Then, the prior predictive distribution is given by:

$$p(y|x) = \int p(y|\theta, x)p(\theta)d\theta \tag{4}$$

$$p(y|x) \approx \frac{1}{N} \sum_{i=1}^{N} p(y|\theta_i, x) \tag{5}$$

where $\theta_i \sim p(\theta)$.

Estimating posterior predictive distribution

Extending for posterior predictive distribution, we have:

$$p(y|x, D) = \int p(y|\theta, x)p(\theta|D)d\theta \tag{6}$$

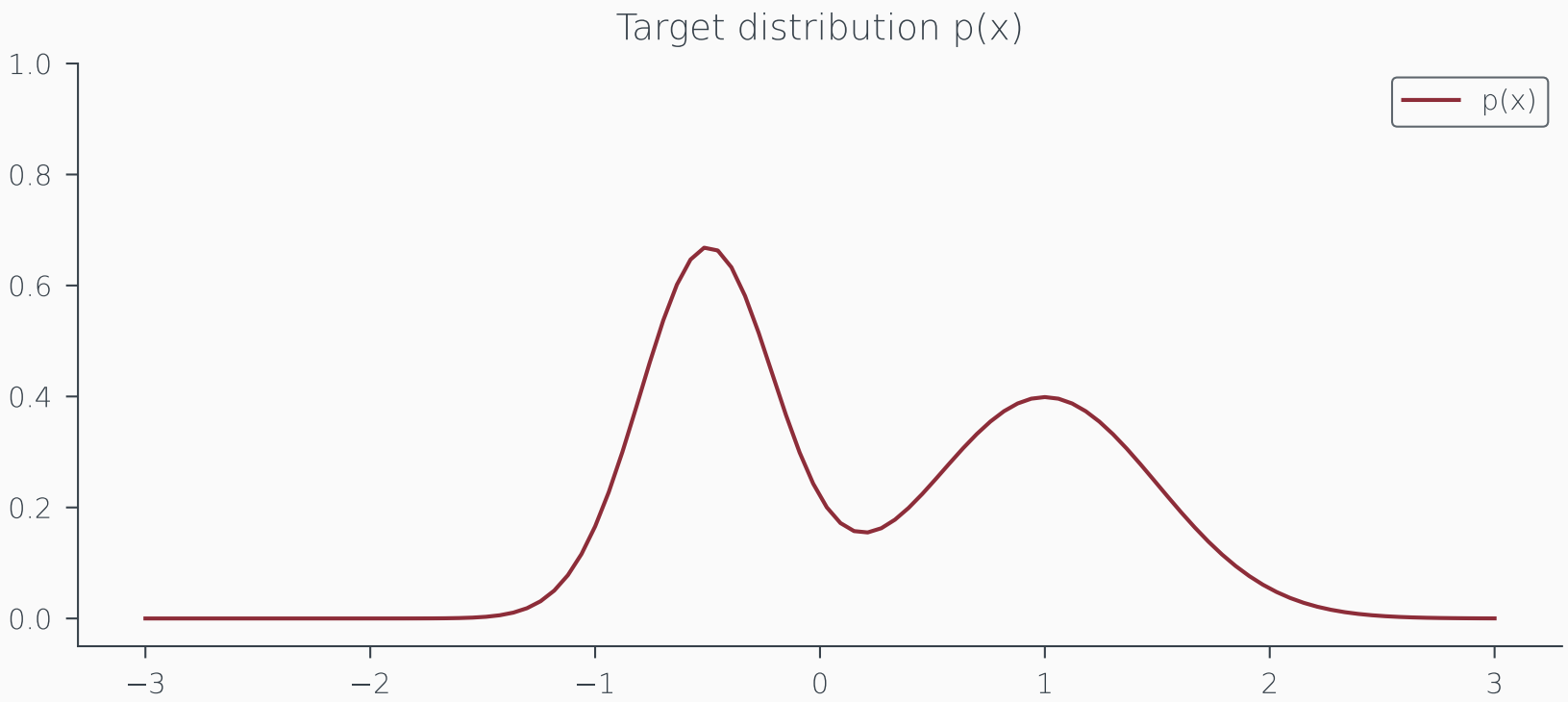
$$p(y|x, D) \approx \frac{1}{N}\sum_{i=1}^{N} p(y|\theta_i, x) \tag{7}$$

Rejection Sampling

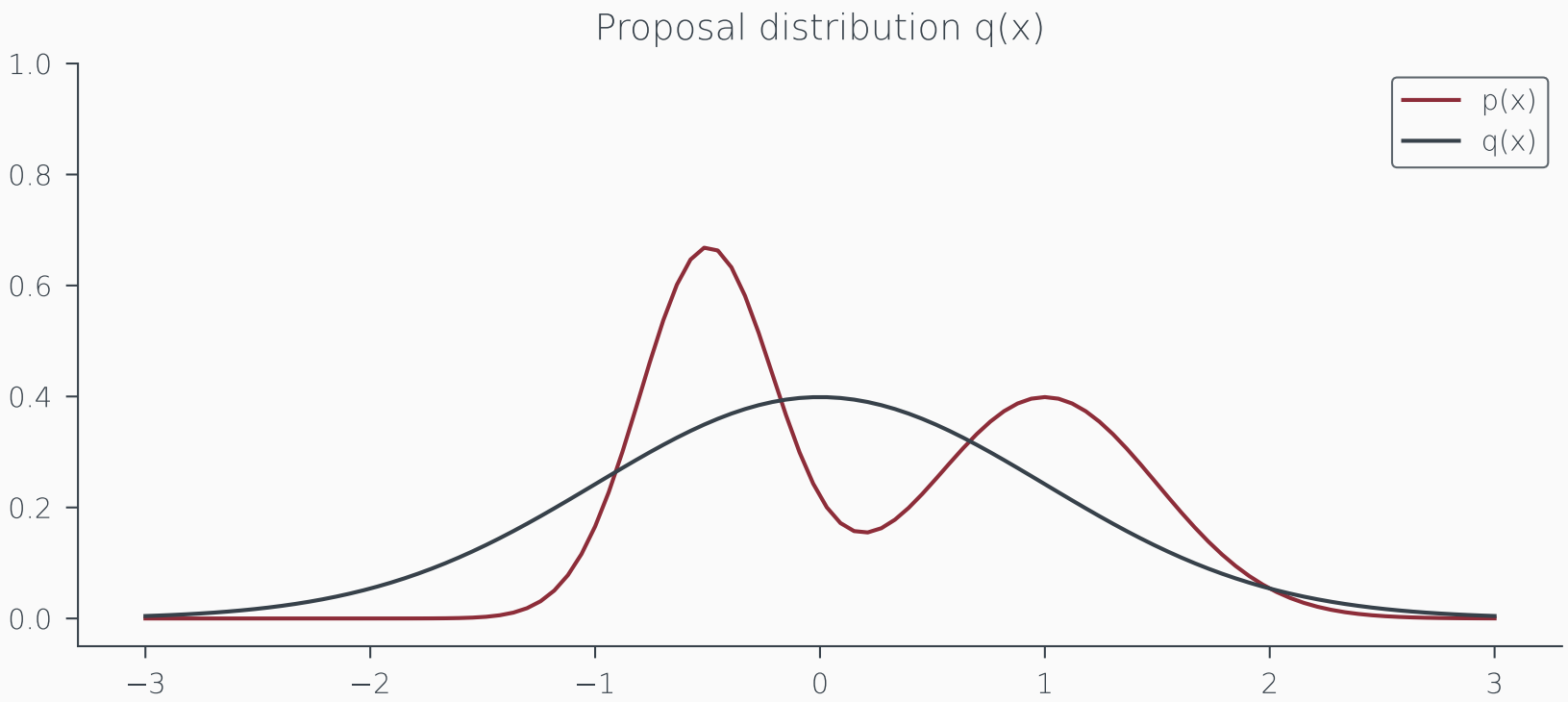
Rejection Sampling

- Let $p(x)$ be the target distribution from which we want to sample.
- Let $q(x)$ be a proposal distribution from which we can sample.
- Let M be a constant such that $M \geq \frac{p(x)}{q(x)} \forall x$.
- Then, we can sample from $p(x)$ by sampling from $q(x)$ and accepting the sample with probability $\frac{p(x)}{Mq(x)}$.

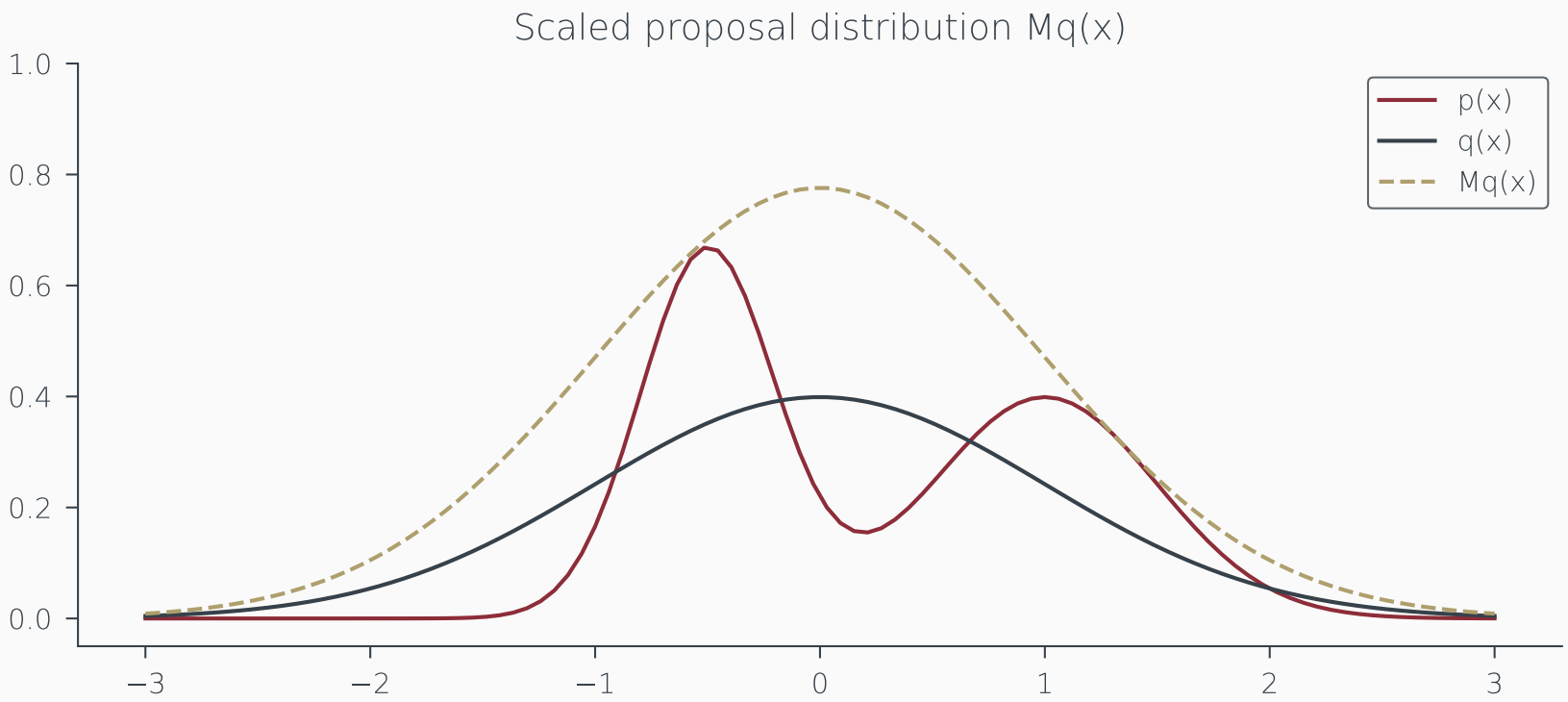
Rejection Sampling



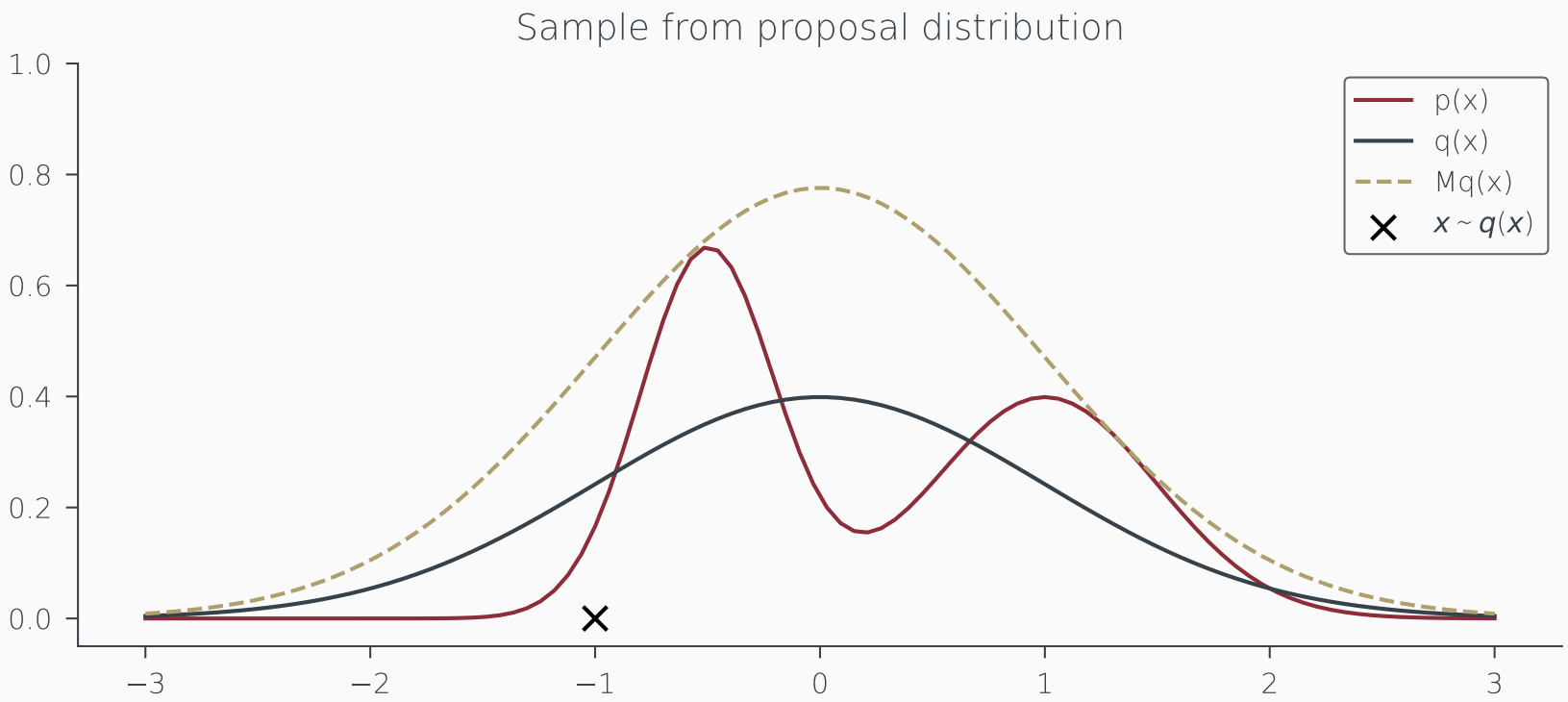
Rejection Sampling



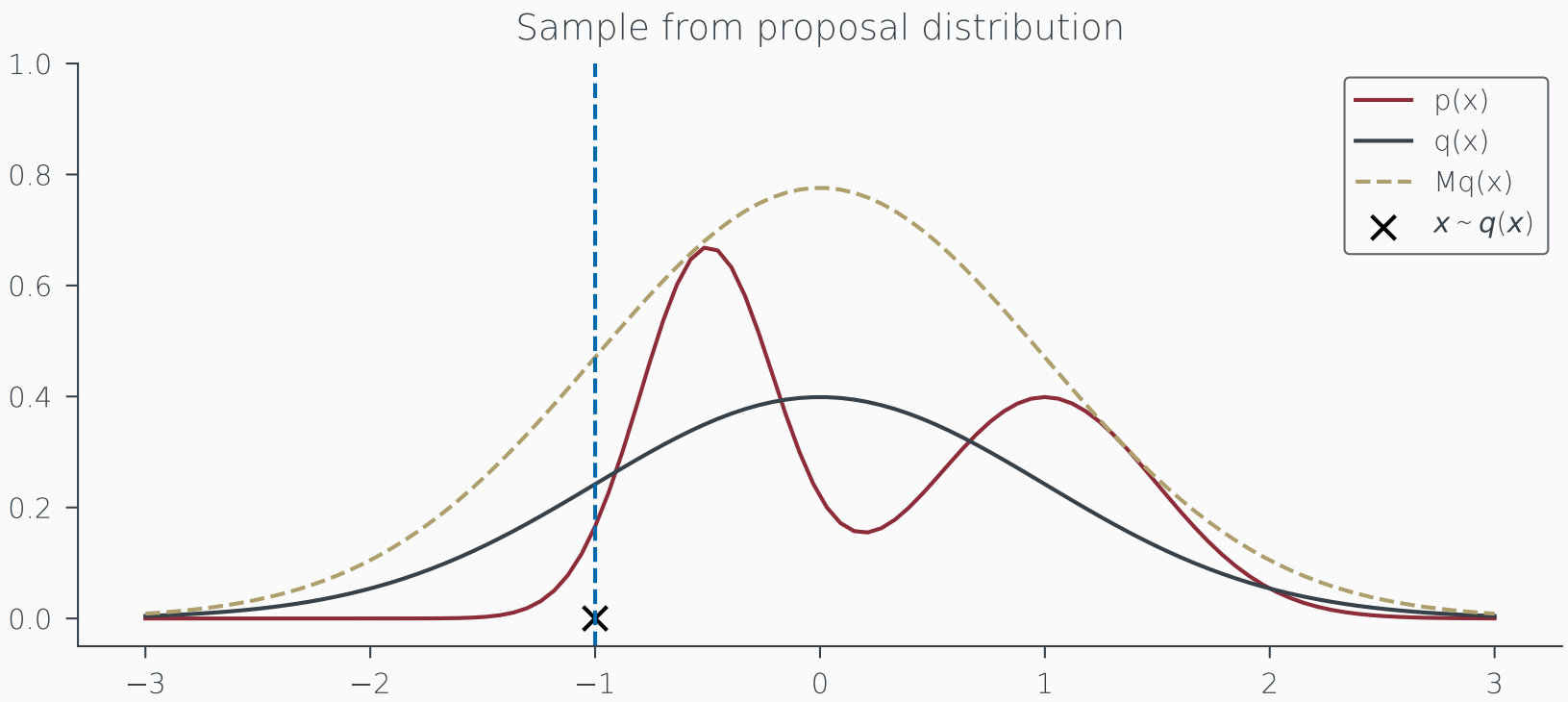
Rejection Sampling



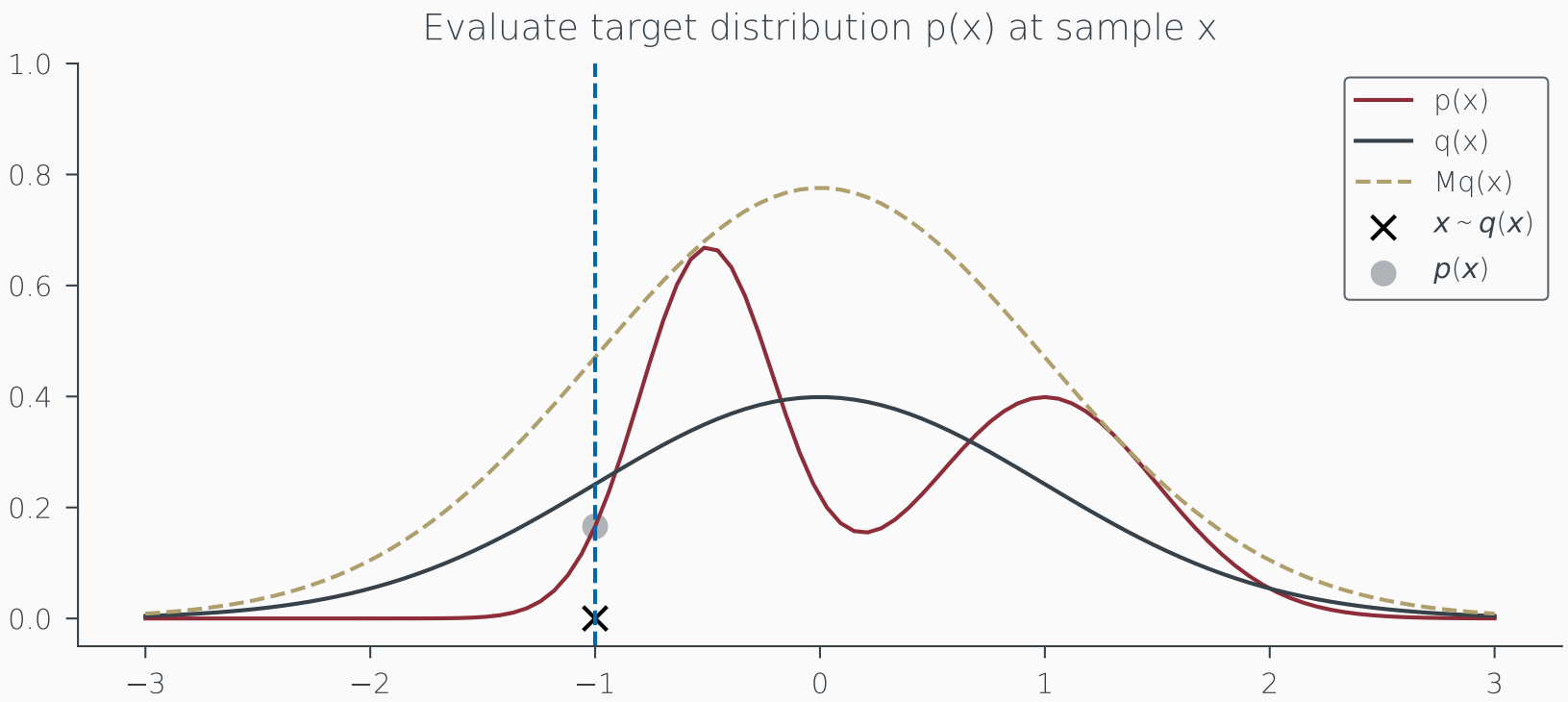
Rejection Sampling



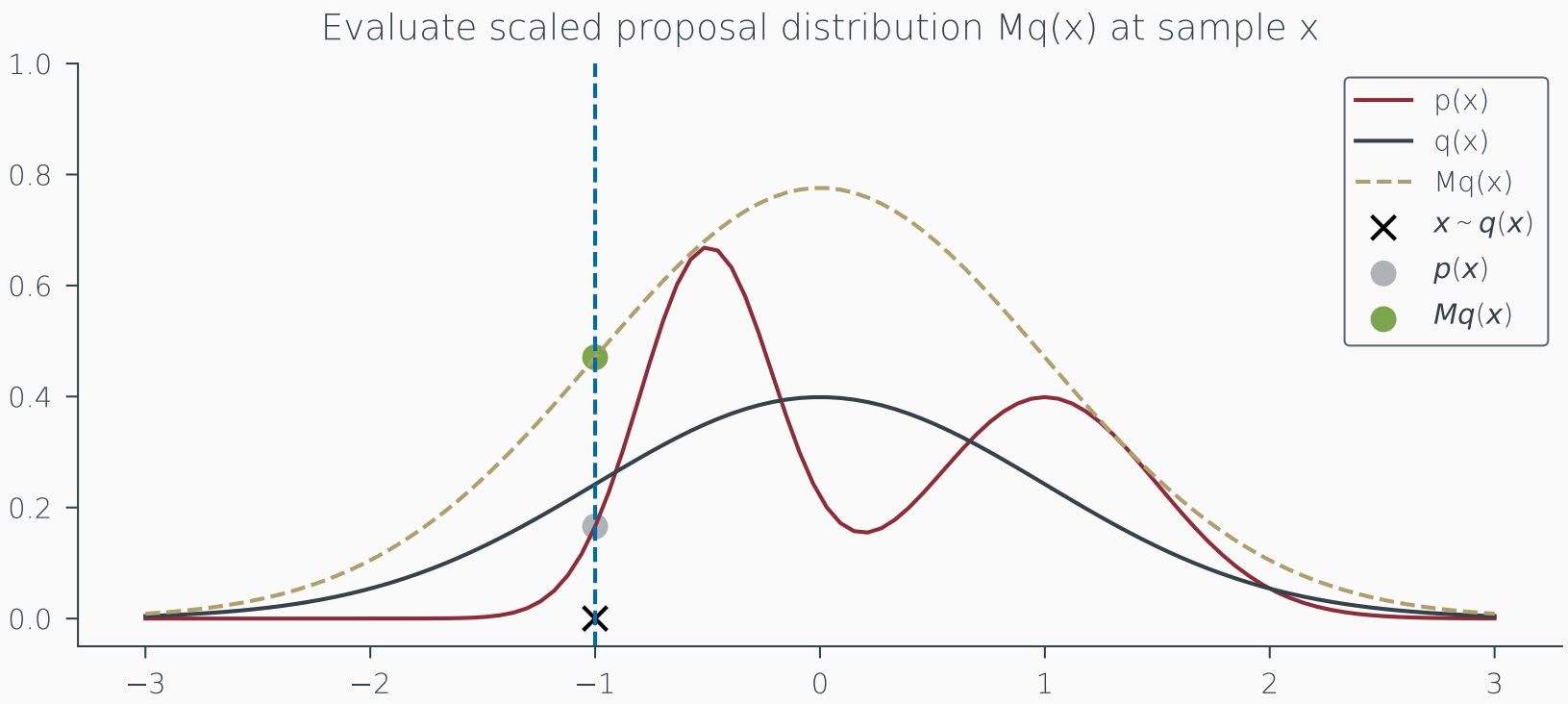
Rejection Sampling



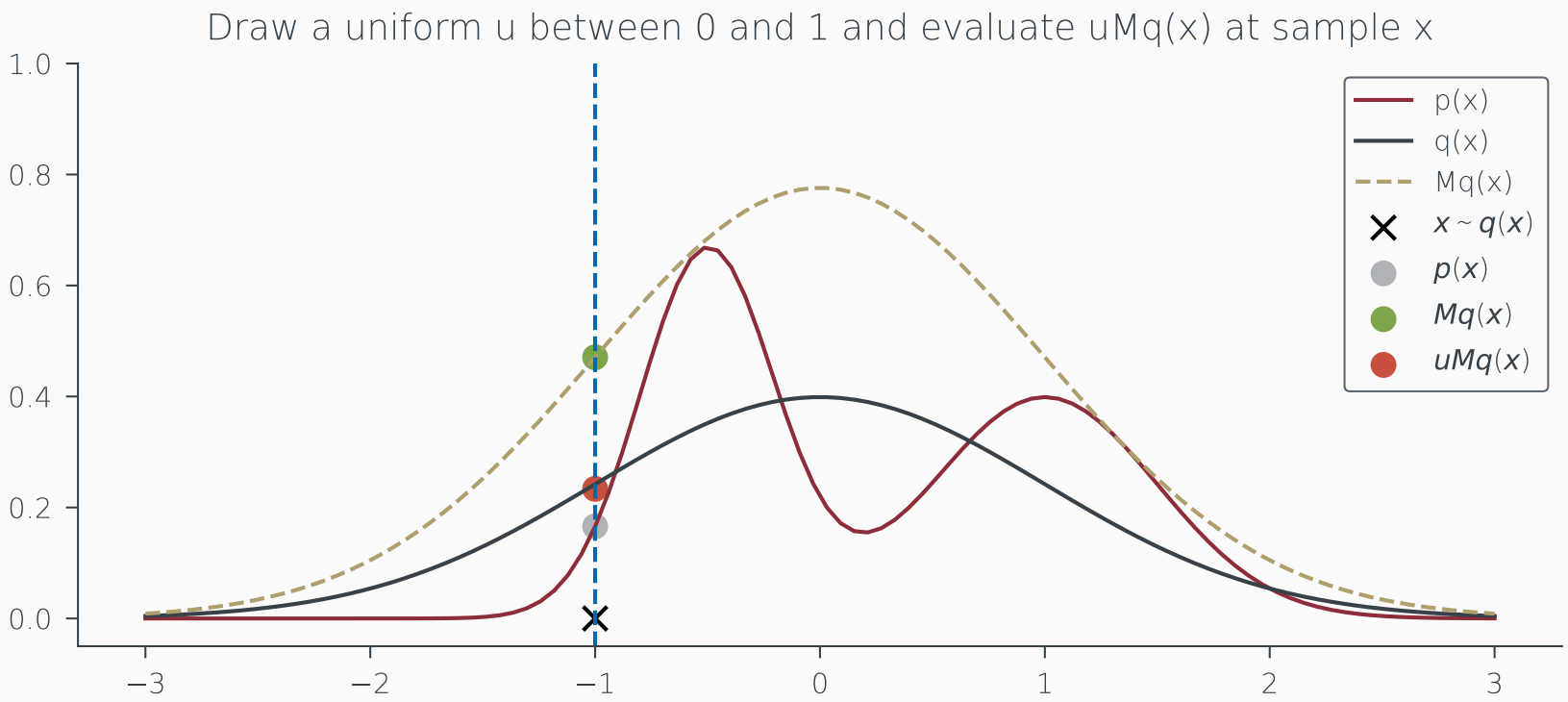
Rejection Sampling



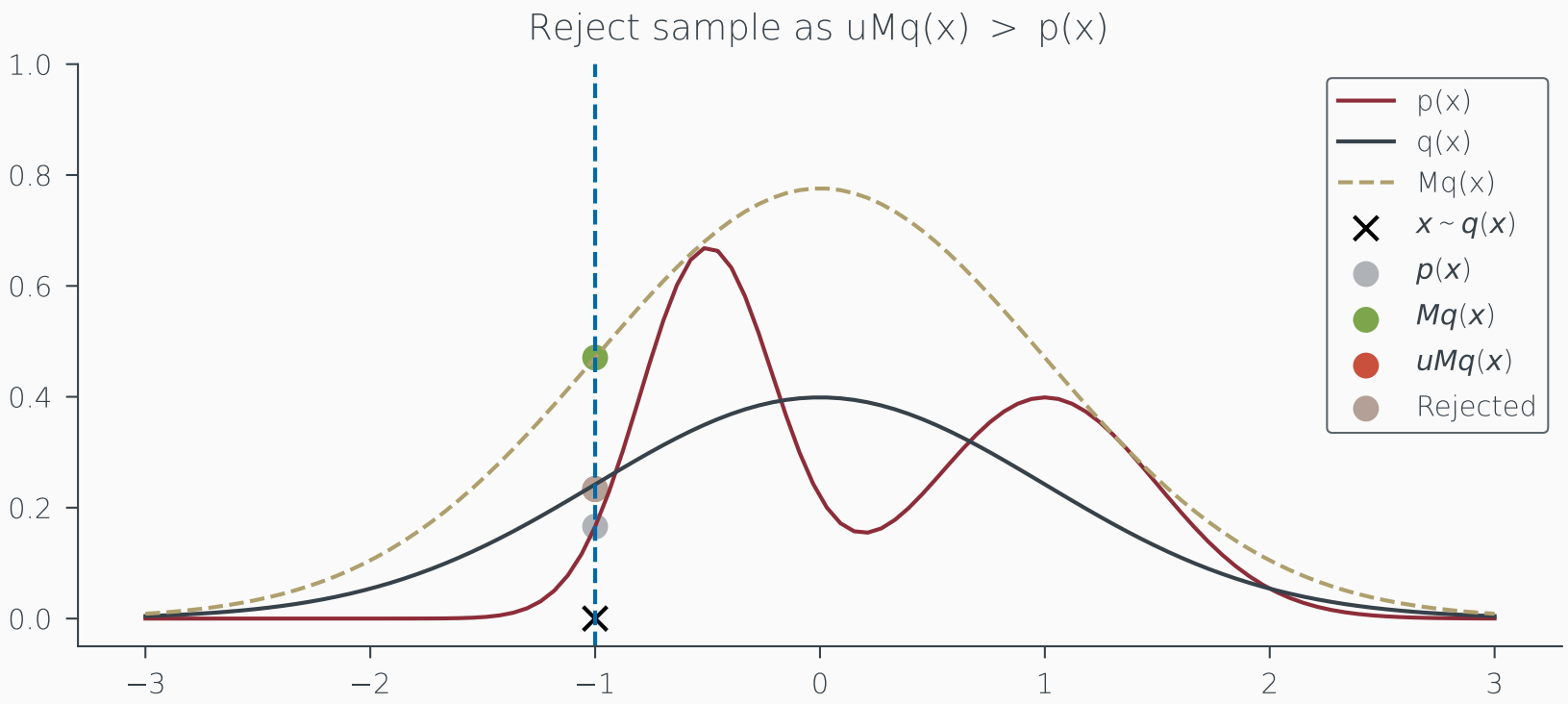
Rejection Sampling



Rejection Sampling



Rejection Sampling



Proof of Rejection Sampling

Acceptance Probability $\alpha(x)$

$$\alpha(x) = \frac{p(x)}{Mq(x)} \tag{8}$$

Bayes Rule for Acceptance

$$P(Sample|Accept) = \frac{P(Accept|Sample)P(Sample)}{P(Accept)} \tag{9}$$

P(Sample)

We draw samples from $q(x)$, so $P(Sample) = q(x)$.

Proof of Rejection Sampling

Further, $P(Accept|Sample) = \alpha(x) = \frac{p(x)}{Mq(x)}$.

Finally, $P(Accept) = \int P(Accept|Sample)P(Sample)dSample = \int \alpha(x)q(x)dx = \frac{1}{M}\int p(x)dx = \frac{1}{M}$.

P(Accept)

$$P(Accept) = \frac{1}{M} \tag{10}$$

Thus, $P(Sample|Accept) = \frac{p(x)}{Mq(x)} \times \frac{q(x)}{1/M} = p(x)$.

Thus, we have shown that the samples we accept are distributed according to $p(x)$.

Rejection Sampling Completed Example

Note: Figures not on github.

Challenges with Rejection Sampling

- Rejection sampling is inefficient when the target distribution is very different from the proposal distribution.
- In this case, we will reject a lot of samples.
- This is a problem when sampling from high-dimensional distributions.
- Acceptance probability $\alpha(x)$ is very low.

Inverse CDF

General Form

Inverse Cumulative Distribution Function (Inverse CDF) sampling is a technique used to generate random numbers from a given probability distribution.

Particularly useful when sampling from distributions lacking a straightforward analytical method for direct sampling.

A method of sampling from the distribution is sampling $u \in \mathcal{U}(0,1)$ and find $x = F_X^{-1}(u)$. The cumulative probability distribution (cdf) of X is:

$$F_X(x) = \mathbb{P}(X <= x) = \int_{-\infty}^{\infty} \pi(u) I(u <= x) du = \int_{-\infty}^{\infty} \pi(u) du \tag{11}$$

Thus, Sample $u \in \mathcal{U}(0,1)$ and set $Y = F_\pi^{-1}(u)$.

Proof

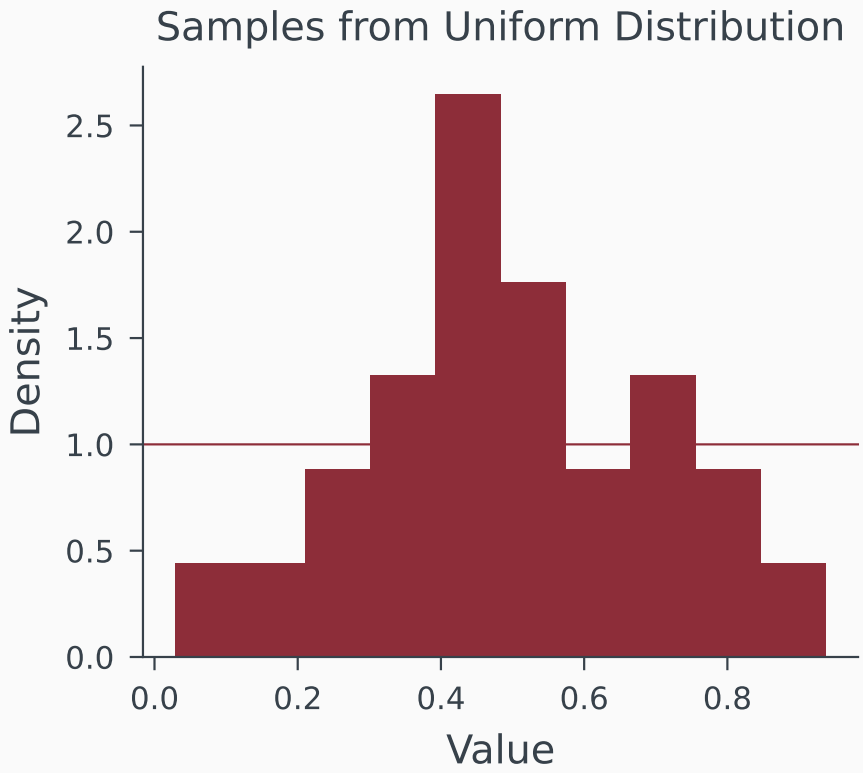
We need to prove that the algorithm mentioned above produces samples from π. We calculate the cdf of X produced by the algorithm above. For any $y \in X$ we have:

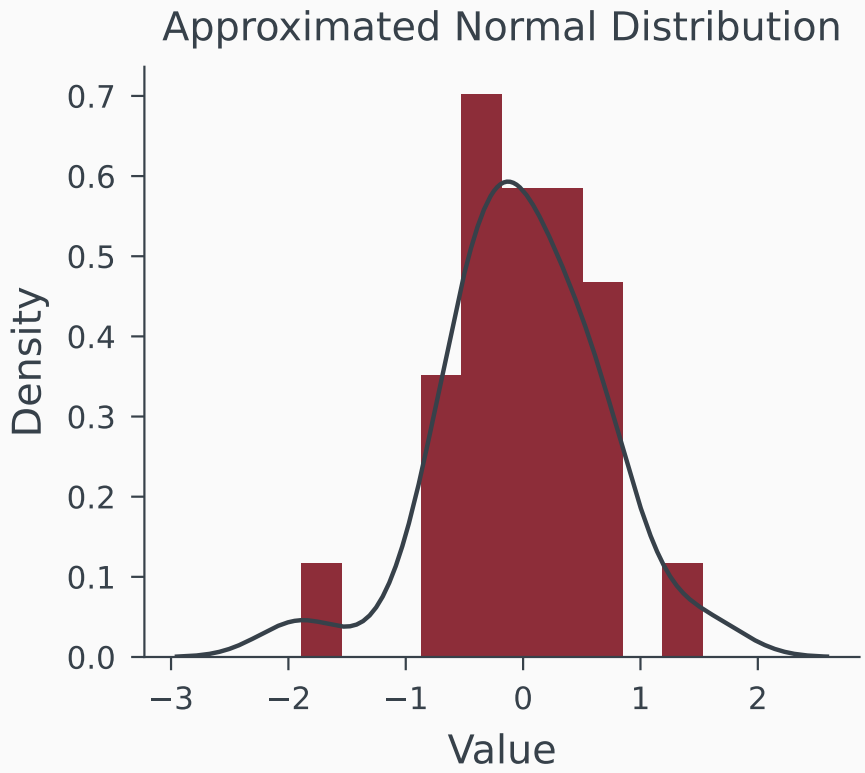
$$\begin{aligned}
\mathbb{P}(Y <= y) &= \mathbb{P}(Y = F_X^{-1}(u) <= y) \\
&= \mathbb{P}(u <= F_X(y)) \\
&= \int_0^1 I(u <= F_X(y)).1 du = \int_0^{F_X(y)} du = F_X(y)
\end{aligned}$$

This shows that the cdf of Y produced by the algorithm is the same as cdf of $X \sim \pi$.

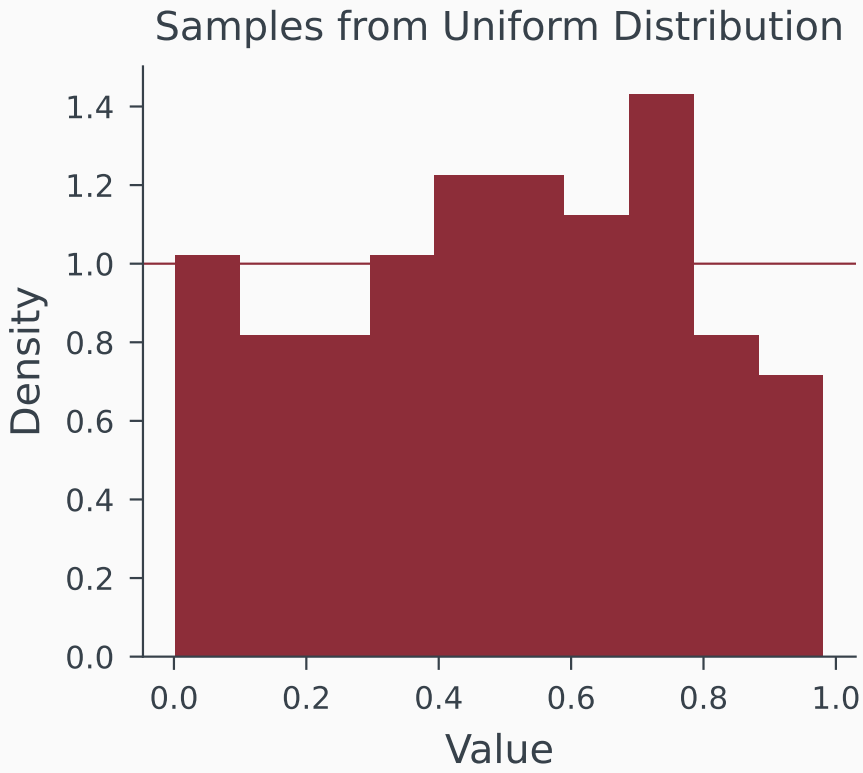
Example of Normal distribution

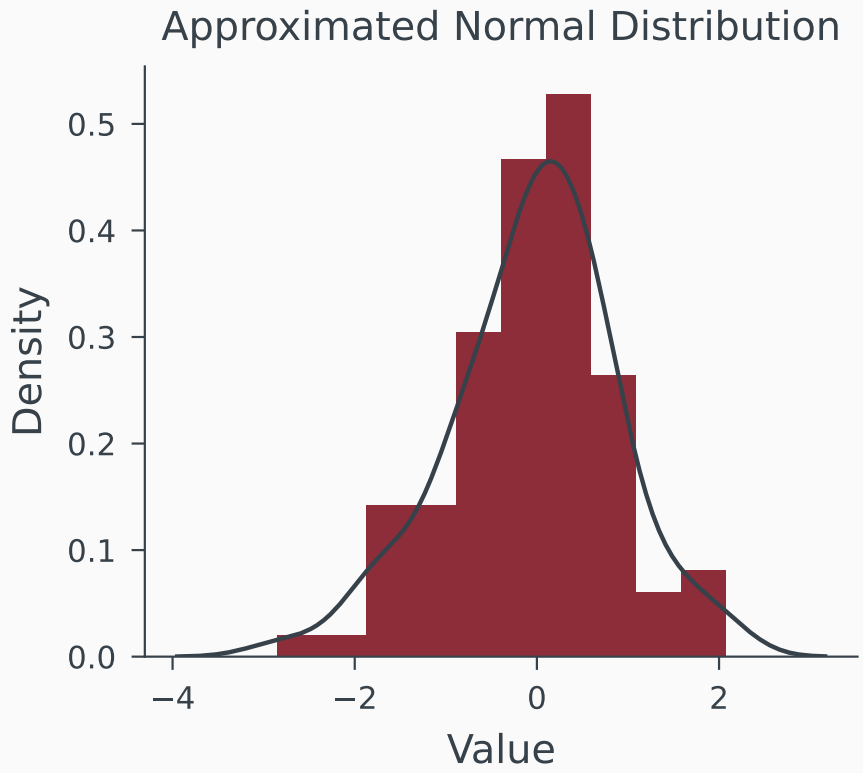
Number of samples = 25:



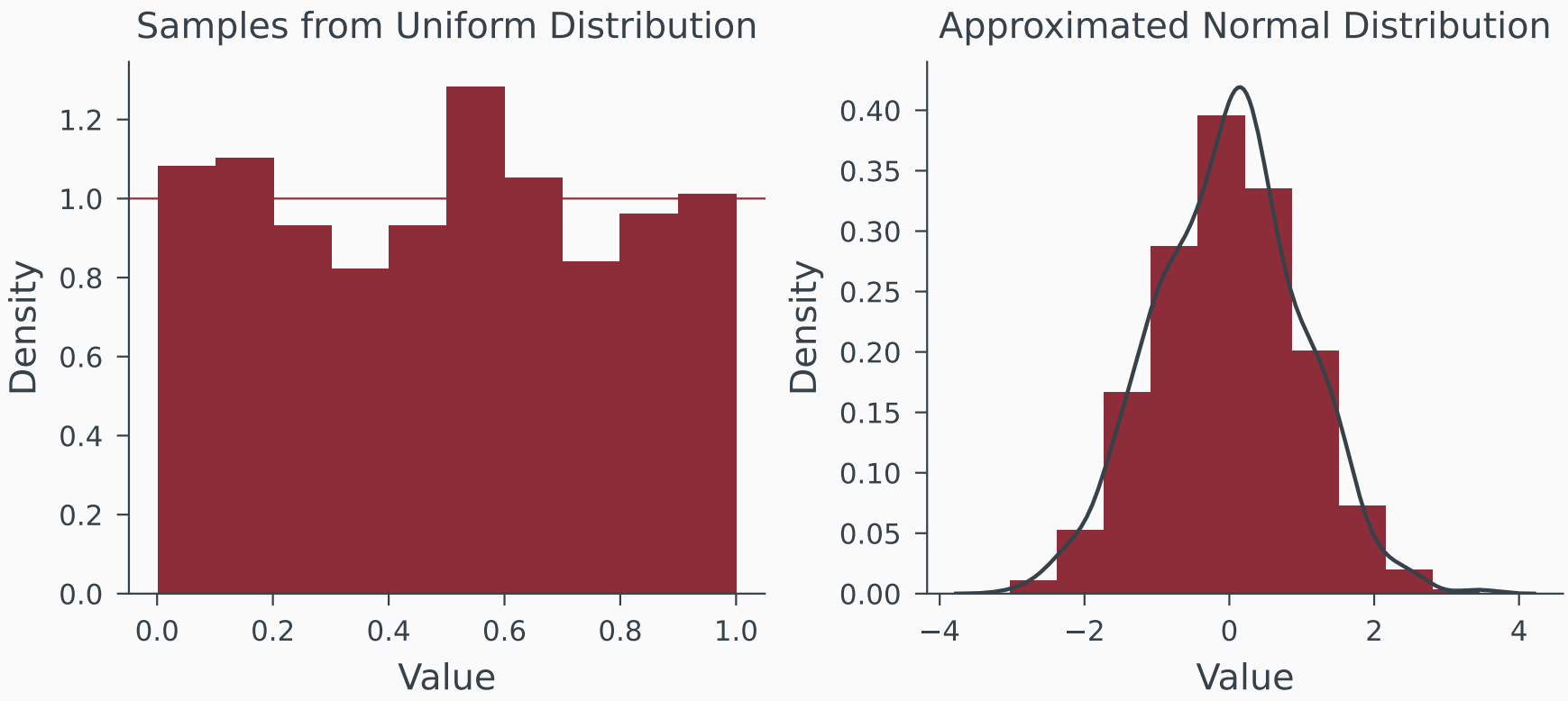


Number of samples = 100:





Number of samples = 1000:



We see that as the number of samples increases, we are able to approximate the induced distribution which is the normal distribution for this example.

Limitations:

- Limited Distribution Complexity: It relies on having an analytically calculable cumulative distribution function (CDF) and an invertible CDF function.
- Numerical Inversion Challenges: When the inverse of the CDF cannot be expressed analytically, numerical methods introduce numerical errors and slow down the sampling process.
- Efficiency and Multivariate Distributions: It can be resource-intensive for high-dimensional multivariate distributions.

Importance Sampling

General Form

In rejection sampling, we saw that due to less acceptance probability, a lot of samples were wasted leading to more time and higher complexity to approximate a distribution.

Computing $p(x), q(x)$ thus seems wasteful. Let us rewrite the equation as:

$$\phi = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx$$
$$\sim \frac{1}{N}\sum_{i=1}^{N} f(x_i)\frac{p(x_i)}{q(x_i)} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)w_i$$

Here, $x_i \sim q(x)$. w_i is known as the importance(weight) of sample i.

However the normalization constant Z is generally not known to us. Thus writing:

$$p(x) = \frac{\tilde{p}(x)}{Z} \tag{12}$$

Now inserting this in earlier equations, we get:

$$\begin{aligned}\phi &= \frac{1}{Z}\int f(x)\tilde{p}(x)dx = \frac{1}{Z}\int f(x)\frac{\tilde{p}(x)}{q(x)}q(x)dx \\ &\sim \frac{1}{NZ}\sum_{i=1}^{N} f(x_i)\frac{\tilde{p}(x_i)}{q(x_i)} = \frac{1}{NZ}\sum_{i=1}^{N} f(x_i)w_i\end{aligned}$$

We know that:

$$\begin{aligned}Z &= \int_{\infty}^{\infty}\tilde{p}(x)dx = \int_{\infty}^{\infty}\frac{\tilde{p}(x)}{q(x)}q(x)dx \\ &= \frac{1}{N}\sum_{i=1}^{N} w_i\end{aligned}$$

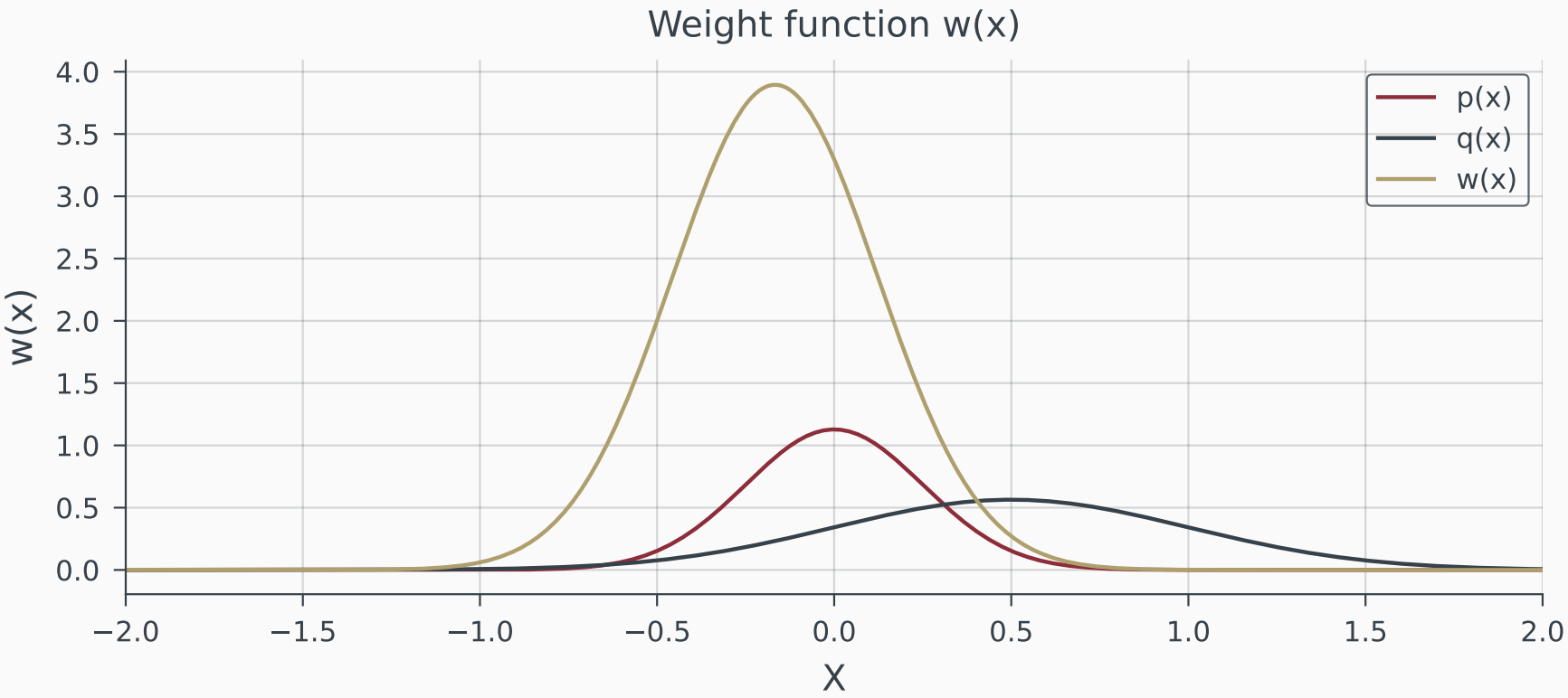
Substuting this value of Z in the equation above, we get:

$$\phi = \frac{1}{N}\sum_{i=1}^{N} f(x_i)w_i = \frac{\sum_{i=1}^{N} f(x_i)w_i}{\sum_{i=1}^{N} w_i}$$
$$= \sum_{i=1}^{N} f(x_i)W_i$$

Here $W_i = \frac{w_i}{\sum_{i=1}^{N} w_i}$ are the normalized weights.

Limitations

- Recall that Var $\hat{\phi} = \frac{var(f)}{N}$. Importance sampling replaces $var(f)$ with $var(f\frac{p}{q})$. At positions where $p >>> q$, the weight can tend to ∞!



Gibbs Sampling

General Form

Suppose we wish to sample $\theta_1, \theta_2 \sim p(\theta_1, \theta_2)$, but cannot use:

- direct simulation
- accept-reject method
- Metropolis-Hasting

But we can sample using the conditionals i.e.:

- $p(\theta_1|\theta_2)$ and
- $p(\theta_2|\theta_1)$,

then we can use Gibbs sampling.

Suppose $\theta_1, \theta_2 \sim p(\theta_1, \theta_2)$ and we can sample from $p(\theta_1, \theta_2)$. We begin with an initial value (θ_1^0, θ_2^0), the workflow for Gibbs algorithm is:

1. sample $\theta_1^j \sim p(\theta_1 | \theta_2^{j-1})$ and then
2. sample $\theta_2^j \sim p(\theta_2 | \theta_1^j)$.

One thing to note here is that the sequence in which the theta's are sampled are not independent!

Bivariate Normal Example

Suppose

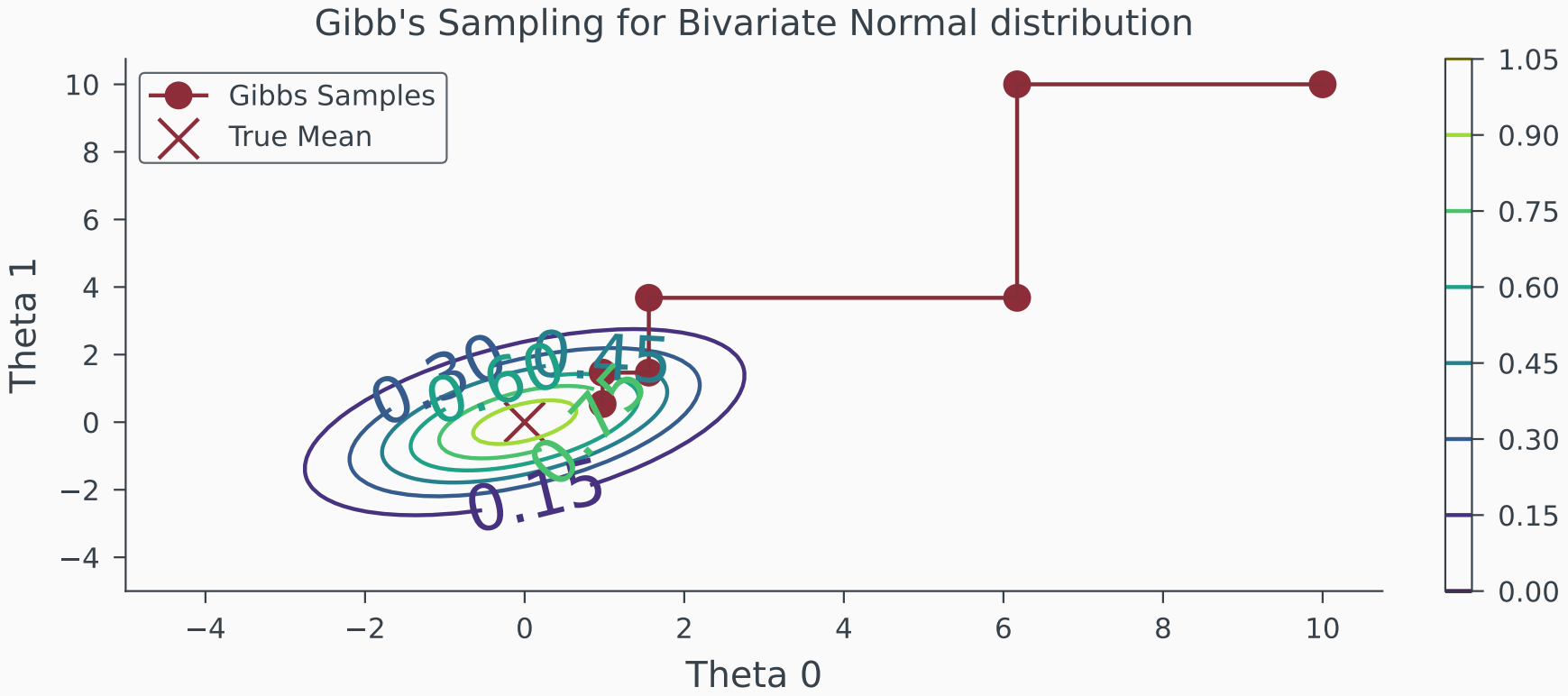
$\theta \sim N_2(0, \Sigma)$ and $\Sigma = \begin{matrix} 1 & \rho \\ \rho & 1 \end{matrix}$

Then, we have:

$\theta_1|\theta_2 \sim N(\rho\theta_2, [1-\rho^2])$

$\theta_2|\theta_1 \sim N(\rho\theta_1, [1-\rho^2])$ are the conditional distributions. The Gibbs sampling proceeds as follows:

Iteration	Sample θ_1	Sample θ_2
1	$\theta_1 \sim N(\rho\theta_2^0, [1-\rho^2])$	$\theta_2 \sim N(\rho\theta_1^1, [1-\rho^2])$
	.	
	.	
k	$\theta_1 \sim N(\rho\theta_2^{k-1}, [1-\rho^2])$	$\theta_2 \sim N(\rho\theta_1^k, [1-\rho^2])$


Gibb's Sampling for Bivariate Normal distribution
Gibbs Samples
True Mean
Theta 1
Theta 0
10
8
6
4
2
0
−2
−4
−4
−2
0
2
4
6
8
10
1.05
0.90
0.75
0.60
0.45
0.30
0.15
0.00

Multivariate case

Suppose $\theta = (\theta_1, \theta_2, \ldots, \theta_K)$, the Gibbs workflow is as follows:

$\theta_1^j = p(\theta_1 | \theta_2^{j-1}, \ldots, \theta_K^{j-1})$

$\theta_2^j = p(\theta_2 | \theta_1^j, \theta_3^{j-1}, \ldots, \theta_K^{j-1})$

.

.

$\theta_k^j = p(\theta_k | \theta_1^j, \ldots, \theta_{k-1}^j, \theta_{k+1}^{j-1}, \ldots, \theta_K^{j-1})$

.

.

$\theta_K^j = p(\theta_K | \theta_1^j, \ldots, \theta_{K-1}^j)$

The distributions above are call the full conditional distributions.

Advantages

Gibbs sampling can be used to draw samples from $p(\theta)$ when:

- Other methods don't work quite well in higher dimensions.
- Draw samples from the full conditional distributions is easy, $p(\theta_k|\theta_{-k})$.

Markov Chain Monte Carlo

Limitations of basic sampling methods

- *Transformation based methods*: Usually limited to drawing from standard distributions.
- *Rejection and Importance sampling*: Require selection of good proposal distirbutions.

In high dimensions, usually most of the density $p(x)$ is concentrated within a tiny subspace of x. Moreover, those subspaces are difficult to be known a priori.

A solution to these are MCMC methods.

Markov Chain

- **Markov Chain**: A joint distribution $p(X)$ over a sequence of random variables $X = \{X_1, X_2, \ldots, X_n\}$ is said to have the Markov property if

$$p(X_i|X_1, \ldots, X_{i-1}) = p(X_i|X_{i-1})$$

  The sequence is then called a Markov chain.
- The idea is that the estimates contain information about the shape of the target distribution p.

Metropolis Hastings

- The basic idea is propose to move to a new state x_{i+1} from the current state x_i with probability $q(x_{i+1}|x_i)$, where q is called the proposal distribution and our target density of interest is $p(= \frac{1}{Z}\tilde{p})$.
- The new state is accepted with probability $\alpha(x_i, x_{i+1})$.
  - If $p(x_{i+1}|x_i) = p(x_i|x_{i+1})$, then $\alpha(x_i, x_{i+1}) = \min(1, \frac{p(x_{i+1})}{p(x_i)})$.
  - If $p(x_{i+1}|x_i) \neq p(x_i|x_{i+1})$, then $\alpha(x_i, x_{i+1}) = \min(1, \frac{p(x_{i+1})q(x_i|x_{i+1})}{p(x_i)q(x_{i+1}|x_i)}) = \min(1, \frac{\tilde{p}(x_{i+1})q(x_i|x_{i+1})}{\tilde{p}(x_i)q(x_{i+1}|x_i)})$
- Evaluating α, we only need to know the target distribution up to a constant of proportionality or without normalization constant.

Algorithm: Metropolis Hastings

1. Initialize x_0.
2. for $i = 1, \ldots, N$ do:
3. Sample $x^* \sim q(x^*|x_{i-1})$.
4. Compute $\alpha = \min(1, \frac{\tilde{p}(x^*)q(x_{i-1}|x^*)}{\tilde{p}(x_{i-1})q(x^*|x_{i-1})})$
5. Sample $u \sim \mathcal{U}(0, 1)$
6. if $u \leq \alpha$:

   $x_i = x^*$

   else:

   $x_i = x_{i-1}$

Pop Quiz

How do we choose the initial state x_0?

Pop Quiz

How do we choose the initial state x_0?

1. Start the Markov Chain at an initial x_0.
2. Using the proposal $q(x|x_i)$, run the chain long enough, say N_1 steps.
3. Discard the first $N_1 - 1$ samples (called 'burn-in' samples).
4. Treat x_{N_1} as first sample from $p(x)$.

MCMC demo

https://chi-feng.github.io/mcmc-demo/app.html