Thermodynamic Integration

Using MCMC to find Normalization Constants

Sachin Shanbhag

Department of Scientific Computing
Florida State University,
Tallahassee, FL 32306.

Contents

Motivation

MCMC enables us to sample $x \sim q(x)$ where $q(x)$ is an unnormalized PDF (uPDF)

Sometimes we are interested in the normalization constant

$$Z = \int q(x)\, dx \tag{1}$$

The normalized PDF

$$p(x) = \frac{q(x)}{Z} \qquad \int p(x)\, dx = 1 \tag{2}$$

Thermodynamic integration (TI) is a method that uses MCMC to find normalization constants.

Motivation: Physics

In statistical physics, the probability of a state $\boldsymbol{x}$ at a given temperature T is given by the Boltzmann factor,

$$q(\boldsymbol{x}) = \exp\left(-\frac{U(\boldsymbol{x})}{k_B T}\right) \tag{3}$$

where $\boldsymbol{x}$ may represent the positions and velocities of all the particles in the system.

The **free energy** of the system $F(T)$, an important property, is related to the normalization constant which is also called the partition function.

$$F(T) = -k_B T \log Z(T) \tag{4}$$

Motivation: Bayesian Statistics

It is also important for **model selection** in a Bayesian setting.

$$\Pr(M|\text{data}) = \frac{\Pr(\text{data}|M)\Pr(M)}{\Pr(\text{data})} \tag{5}$$

For example, given two models M_1 and M_2 representing competing hypothesis, with equal priors, the ratio,

$$\text{Bayes factor} = \frac{\Pr(M_1|\text{data})}{\Pr(M_2|\text{data})}$$

tells us how much to favor M_1 relative to M_2

This requires us to determine the normalization constant of the posterior distribution.

Computing Normalization Constants

(i) Analytical Approximation

We can approximate the $q(x)$ and find the normalization constant from the approximate distribution.

Ex. Laplace approximation

(ii) Numerical Approximation

When x is low-dimensional (< 4, say), we can use numerical quadrature to directly compute the integral

This method fails in higher dimensions due to the curse of dimensionality

(iii) MCMC Approximation

TI is an example of MCMC approximation.

It works particularly well in high dimensions.

Basic Idea

The basic concept is to smoothly interpolate between two probability distributions.

One of the distributions is given to us. We call this the *target* distribution.

We get to choose one of them. We call this the *reference* distribution.

Two factors for choosing the reference PDF

- the normalization constant is known
- it has the same domain Ω as the target distribution

TI then makes it possible to compute the normalization constant for the target distribution.

Relation to Area of Circle

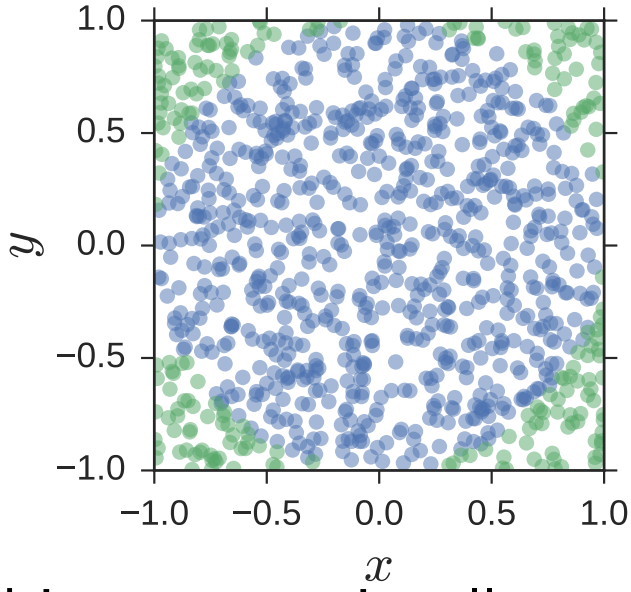
This principle is related to an early example we considered

"Find the area of a circle by inscribing it in a square"

square	reference	$A_s = a^2$
circle	target	$A_c = ?$

We knew the area of the square, and we could compute the ratio of their areas by rejection sampling.

$$A_c \approx \left(\frac{A_c}{A_s}\right)_{\text{MC}} \cdot A_s$$



However, rejection sampling of this kind is computationally inefficient in high dimensions.

Setup

Suppose the target uPDF $q_1(x)$ is defined by a potential energy function $U_1(x)$,

$$q_1(x) = e^{-U_1(x)} \qquad x \in \Omega \tag{6}$$

We select a reference distribution $q_0(x)$ defined by potential energy $U_0(x)$, for $x \in \Omega$.

The normalization constants,

$$Z_i = \int_\Omega q_i(x)\, dx \qquad i \in \{0, 1\} \tag{7}$$

The normalized PDFs,

$$p_i(x) = \frac{q_i(x)}{Z_i} \qquad i \in \{0, 1\} \tag{8}$$

Setup

We know Z_0, since we get to choose p_0.

In TI we define a family of distributions parameterized by λ.

A common choice is a linear interpolation of the energies,

$$U_\lambda(x) = (1-\lambda)U_0(x) + \lambda U_1(x), \qquad \lambda \in [0,1]$$

When $\lambda = 0$, $U_\lambda(x) = U_0(x)$, and when $\lambda = 1$, $U_\lambda(x) = U_1(x)$

This implies a geometric combination of the uPDFs.

With $q_\lambda(x) = e^{-U_\lambda(x)}$, we have

$$q_\lambda(x) = q_0^{1-\lambda}(x) \cdot q_1^\lambda(x), \qquad \lambda \in [0,1]. \tag{9}$$

The associated partition function and normalized PDF are,

$$Z_\lambda = \int_\Omega q_\lambda(x)\, dx, \qquad p_\lambda = \frac{q_\lambda(x)}{Z_\lambda}$$

Derivation

$$\log \frac{Z_1}{Z_0} = \int_0^1 \frac{\partial \log Z_\lambda}{\partial \lambda} d\lambda = \int_0^1 \frac{1}{Z_\lambda} \frac{\partial Z_\lambda}{\partial \lambda} d\lambda \qquad \text{fundamental theorem}$$

$$= \int_0^1 \frac{1}{Z_\lambda} \frac{\partial}{\partial \lambda} \int_\Omega q_\lambda(x)\, dx\, d\lambda \qquad \text{definition of } Z_\lambda$$

$$= \int_0^1 \frac{1}{Z_\lambda} \int_\Omega \frac{\partial \log q_\lambda(x)}{\partial \lambda} q_\lambda(x)\, dx\, d\lambda \qquad \text{order dvt; log}$$

$$= \int_0^1 E_{p_\lambda} \left[\frac{\partial \log q_\lambda(x)}{\partial \lambda} \right] d\lambda \qquad \mathsf{E}[\cdot], p_\lambda = q_\lambda / Z_\lambda$$

$$= \int_0^1 E_{q_\lambda} \left[\log \frac{q_1}{q_0} \right] d\lambda \qquad \text{eq } 9, q_\lambda \sim p_\lambda$$

$$= \int_0^1 E_{q_\lambda} \left[U_0 - U_1 \right] d\lambda \qquad \text{convert to energy}$$

Derivation

Thus,

$$\log \frac{Z_1}{Z_0} = \int_0^1 E_{q_\lambda} \left[U_0 - U_1 \right] d\lambda \tag{10}$$

The integrand $f(\lambda) = E_{q_\lambda} \left[U_0 - U_1 \right]$ is a function of λ.

It is an expected value which can be estimated using MCMC by sampling $x \sim q_\lambda(x)$.

The integral can then be approximated numerically by evaluating $f(\lambda)$ at different gridpoints for $\lambda \in [0, 1]$.

A method like the trapezoidal method is often sufficient to integrate

$$\log \frac{Z_1}{Z_0} = \int_0^1 f(\lambda) \, d\lambda$$

Algorithm

- select a set of $\lambda \in [0, 1]$
- for each λ:
  - generate N samples $x_i \sim q_\lambda(x)$ using MCMC
  - compute the mean energy difference

$$f(\lambda) = E_{q_\lambda}\left[U_0 - U_1\right] \approx \frac{1}{N}\sum_{i=1}^{N}\left(U_0(x_i) - U_1(x_i)\right)$$

  Since both distributions share Ω the energy difference is finite

- Use trapezoidal rule to evaluate the integral

$$\log \frac{Z_1}{Z_0} = \int_0^1 f(\lambda)\, d\lambda$$

Example

Consider a target uPDF defined as $q(x) = e^{-U_1(x)}$ where

$$U_1(x) = \left(x^2 - 1\right)^2, \qquad x \in [-\infty, \infty]$$

This double-well potential has two minima at $x = \pm 1$ and a barrier between the wells.

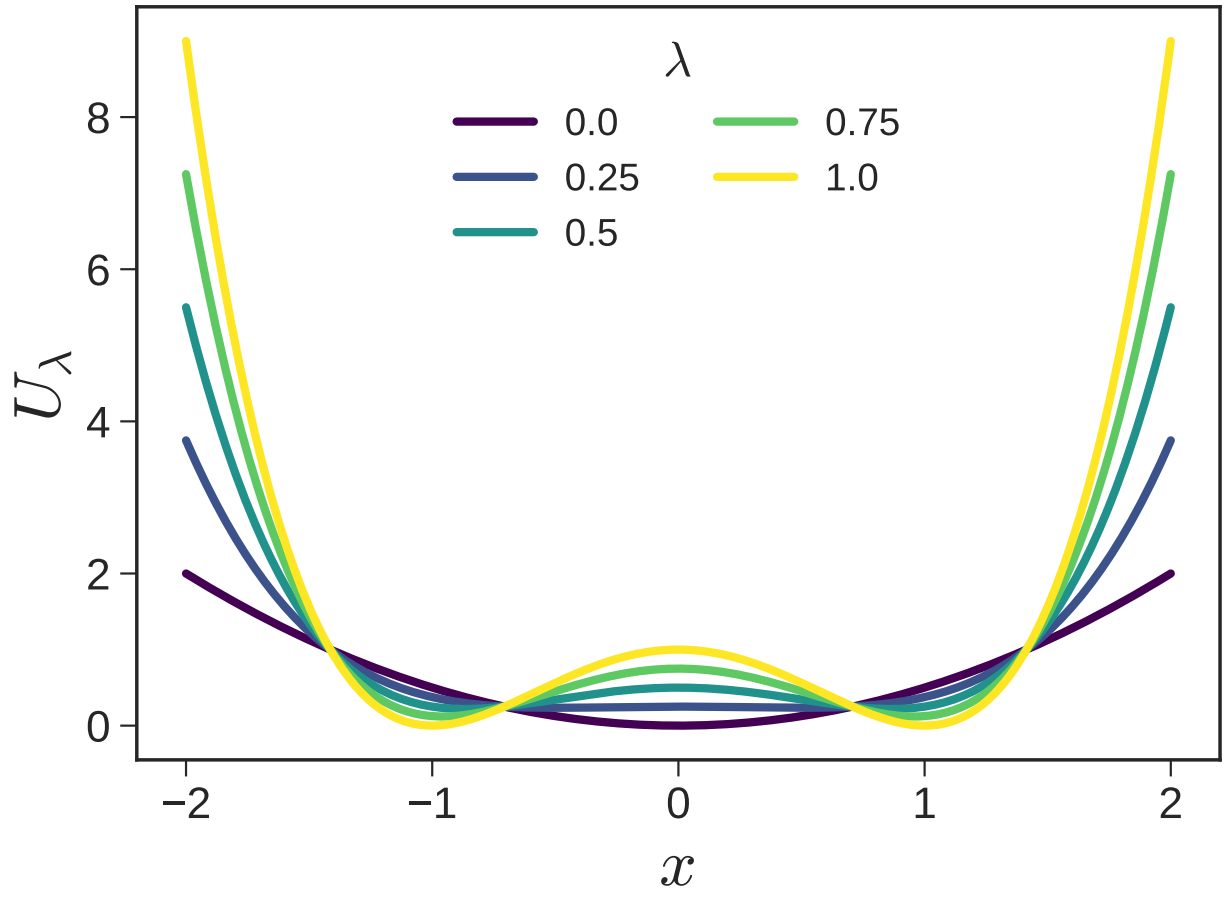
As the reference potential let us consider the quadratic potential

$$U_0(x) = \frac{1}{2}x^2$$

The interpolation between $U_0(x)$ and $U_1(x)$ is

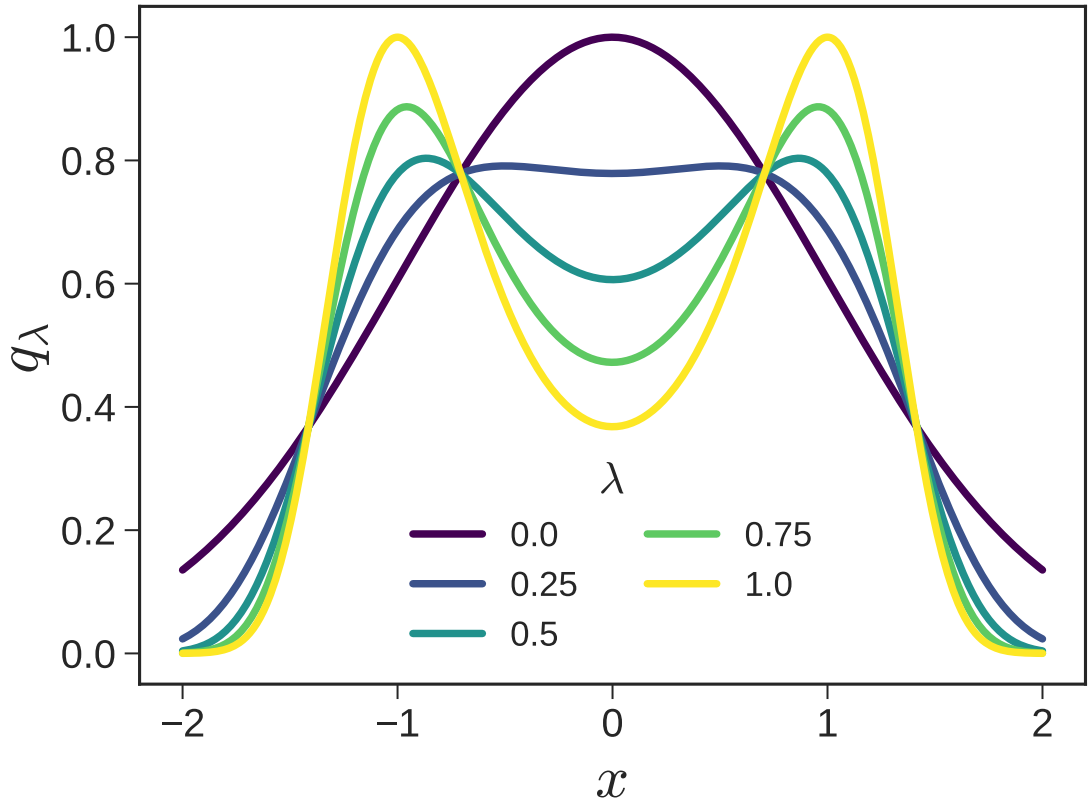
$$U_\lambda(x) = \lambda U_0(x) + (1 - \lambda)U_1(x), \qquad \lambda \in [0, 1]$$

U_λ



The target is $\lambda = 1$. The reference is $\lambda = 0$.

$$q_\lambda(x) = e^{-U_\lambda(x)}$$



Since $q_0 = e^{-x^2/2}$ is the unnormalized Gaussian distribution, the normalization constant is $Z_0 = \sqrt{2\pi}$

Solution

Since $q_1(x)$ is a 1D distribution, we can use quadrature to compute the normalization constant for later verification.

Using `scipy.integrate.quad`,

$$\int_{-\infty}^{\infty} q_1(x)\,dx \approx 1.9737$$

For TI, we need an MCMC method to compute the integrand $f(\lambda)$ at different values of λ.

We could easily fire up a Metropolis MCMC method.

I will be lazy and use pyMC. It requires $\log q_\lambda$.

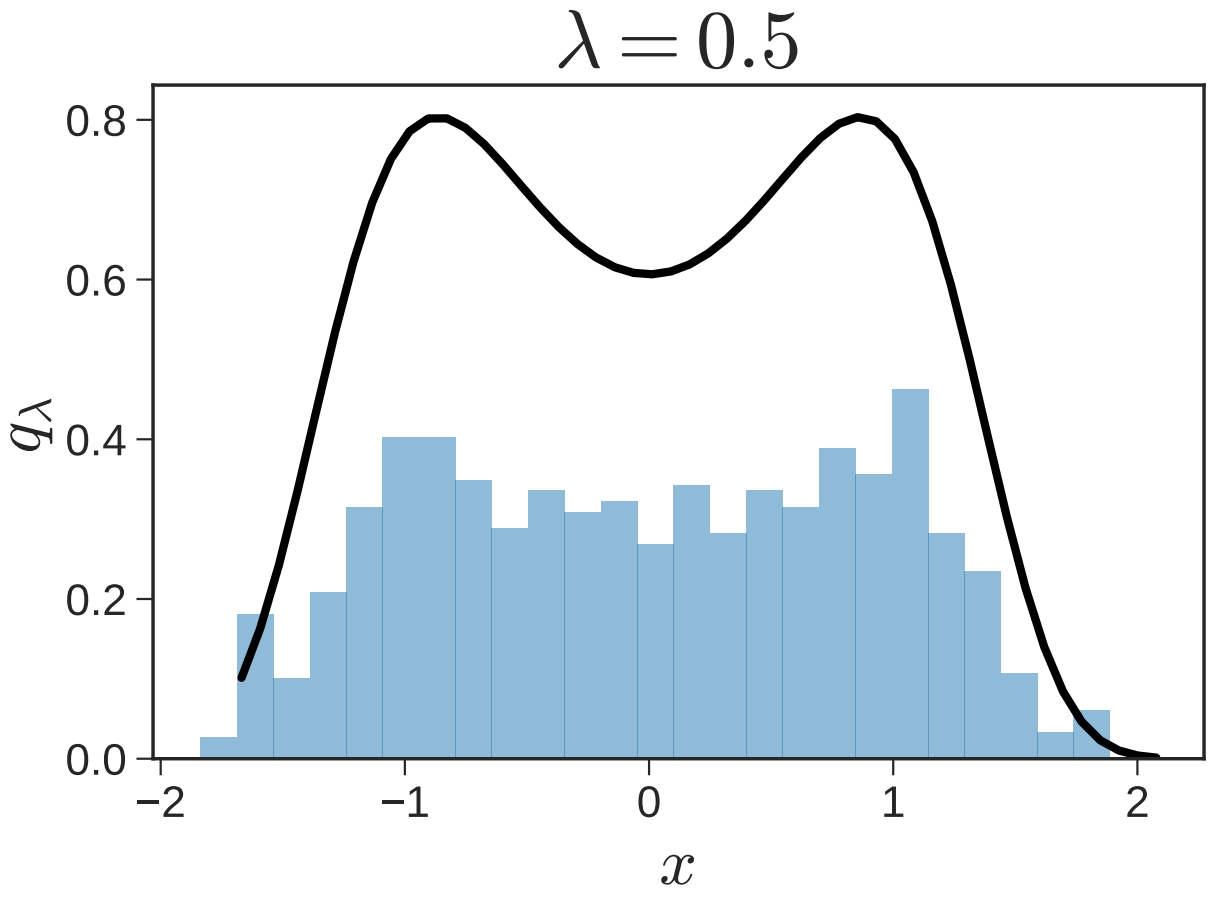
$$\log q_\lambda = -U_\lambda(x) = -\left[\lambda\frac{x^2}{2} + (1-\lambda)\left(x^2-1\right)^2\right]$$

pyMC Implementation

```
import pymc as pm
from pytensor.tensor import TensorVariable

# define distribution
def logp_qlambda(x: TensorVariable, lam : TensorVariable)
            -> TensorVariable:
    return -((1-lam) * x**2/2 + lam * (x**2 - 1)**2)

# sample from qlambda
with pm.Model():
    pm.CustomDist('x', 0.5, logp=logp_qlambda)
    idata = pm.sample(250)

# put samples into a numpy column
xs = idata.posterior['x'].values.reshape(-1, 1)
```

The second argument to `pm.CustomDist()` is λ.

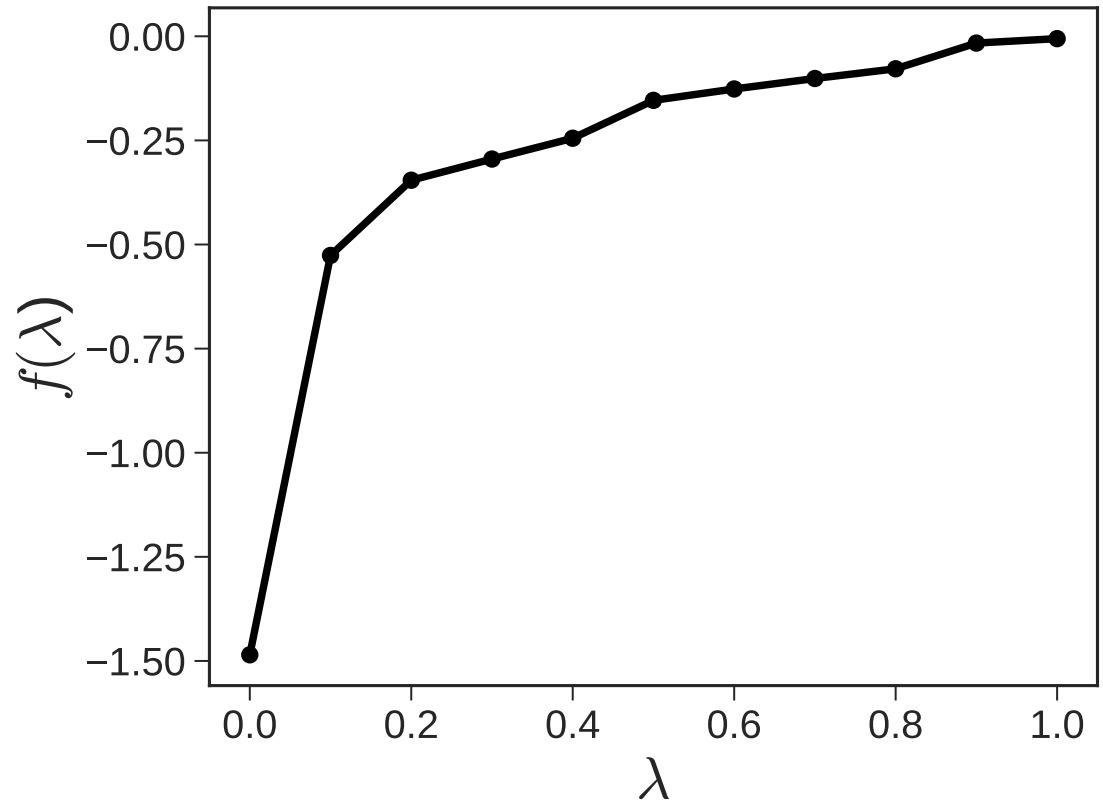
The histogram is normalized. It is clear that q_λ is not.



$$f(\lambda = 0.5) \approx \frac{1}{N} \sum_{i=1}^{N} (U_0(x_i) - U_1(x_i)) = -0.1803$$

Solution

We set $\lambda_i = 0.1i$ for $i = 0, \cdots, 10$.



Using the trapezoidal rule to integrate

$$\log \frac{Z_1}{Z_0} = \int_0^1 f(\lambda)\, d\lambda \approx -0.2633$$

Z_1

Thus,

$$Z_1 \approx Z_0 e^{-0.2633} = \sqrt{2\pi} e^{-0.2633} = 1.9264$$

This compares quite well with the quadrature solution 1.9737.

Exercise: Use a different U_0, perhaps a combination of two normal distribution peaked at ± 1.

Exercise: How would you analyze the propagation of error?

Exercise: What are different ways you could make this error smaller?

Perspective

This example shows how TI works for this simple 1D example.

The true power of TI is for multidimensional distributions, where quadrature is unfeasible.

Note MCMC methods work fine with unnormalized high-dimensional distributions

TI harnesses this power to compute normalization constants

In physical systems, the ideal gas is often used as a reference state

Multidimensional normal distributions can be used as reference state, when the domain is infinite

Uniform distributions can be used when the domain is finite; but better choices may be available.