

Assignment 7

1. Brownian bridge and control variates.

The Brownian bridge is a probability distribution involving d components $x = (x_1, \dots, x_d)$. It is convenient to invent two “ghost” components $x_0 = 0$ and $x_{d+1} = 0$. The “energy” of a Brownian bridge is

$$U(x) = \frac{1}{2} \sum_{k=0}^d (x_{k+1} - x_k)^2 .$$

The $k = 0$ term in this sum is x_1^2 because $x_0 = 0$ and the $k = d$ term is x_d^2 . The *boundary conditions* $x_0 = x_{d+1} = 0$ “tie” the Brownian “path” to zero at its ends. The path is a “bridge” starting and ending at zero.

The *Gibbs Boltzmann*¹ distribution for this energy function is a multivariate joint PDF given by

$$f(x) = \frac{1}{Z} e^{-U(x)} .$$

Typically, the energy function is known but the normalization constant Z is unknown. The Gibbs Boltzmann distribution for any energy function makes “states” (vectors x) with low energy more likely than states with higher energy. However, there may be more high energy than low energy states. Therefore, even though an individual high energy state have less probability, there are so many more of them that the expected energy may be high. This is a subtle tradeoff physicists are still struggling with in specific models today.

We want to know

$$P(d) = \Pr(X_k > 0 , \text{ for all } k = 1, \dots, d) .$$

This can be expressed as

$$P(d) = \frac{1}{Z} \int_{x_1=0}^{\infty} \dots \int_{x_d=0}^{\infty} e^{-U(x_1, \dots, x_d)} dx_1 \dots dx_d .$$

This formula isn’t useful for find the value of $P(d)$ because Z is not known and because evaluating high dimensional ($d > 4$, say) integrals by quadrature is impractical. Instead, we will estimate $P(d)$ using Monte Carlo sampling of the Gibbs Boltzmann distribution.

¹W. Gibbs and L. Boltzmann were physicists from around 1900 who contributed to the field now called *statistical mechanics*. This involves, among other things, using probability models to describe the behavior of systems of atoms or molecules that seem to move “at random” That means modeling the mechanics ($F = ma$) of these systems using statistical (probability) methods.

- (a) Find the symmetric (precision) matrix H so that $U(x) = \frac{1}{2}x^T H x$.
- (b) Write Python code that (1) calls the appropriate `numpy` routines to accomplish the following tasks:
- Create H and calculate the Cholesky factor L so that $LL^T = H$
 - Given L and $z \in \mathbb{R}^d$ finds x so that $Lx = z$. Note that this is equivalent to $x = L^{-1}z$.
 - Generates an d component $Z \sim \mathcal{N}(0, I)$ and the corresponding Brownian bridge path $X = L^{-1}Z$.
 - Generates n independent Brownian bridge paths $X_1, \dots, X_n$. This should create L only once but use it n times with n independent vectors $Z_1, \dots, Z_n$.
- (c) Verify that the algorithm of part (b) generates paths (samples) X whose PDF is the Gibbs Boltzmann distribution with this quadratic energy function. *Hint.* One way to do this is to use the general fact that the covariance matrix Σ is the inverse of the precision matrix H , and the Cholesky factorization of Σ involves L^{-1} .
- (d) Suppose H is any symmetric and positive definite matrix and

$$X \sim \frac{1}{Z} e^{-\frac{1}{2}x^T H x}.$$

Show that the covariance matrix of X is

$$\Sigma = \mathbb{E}[XX^T] = H^{-1}.$$

[This should come before part (c), but the Cholesky factorization is a good tool for solving this, along with the multi-variate change of variables in the integral $x = L^{-1}z$, $dx = \det(L)dz$.

- (e) Let $Y(X)$ be the *indicator function* saying whether X satisfies the constraints $X_k > 0$ for all k . That is

$$Y = \begin{cases} 1 & \text{if } X_k > 0 \text{ for all } k \\ 0 & \text{otherwise} \end{cases}$$

Let Y_j be the indicator for path $X_j \in \mathbb{R}^d$. Use the samples X_j and the estimator

$$\hat{P}(d) = \frac{1}{n} \sum_{j=1}^n Y_j = \frac{1}{n} \# \{ X_j \text{ with } Y(X_j) = 1 \}.$$

- (f) Write code to find the one standard deviation error bar for $\hat{P}(d)$ using the data from part (e). Make a plot showing $\hat{P}(d)$ as a dot and the error bar as a vertical bar for each dot for d values as large as the estimator gives any information for the number of samples practical for you (think of runs taking a few hours once the code is ready).

- (g) Theory and experience suggest that $P(d)$ behaves like a power of d for large d . That means $P(d) \approx Cd^{-\beta}$ for some exponent β . Try to see whether there is a power law relation for large d and, if possible, estimate β . One way to do this is to make a log/log plot, where power lines become straight lines and the slope is the power. If you do this, include the error bars as vertical lines in the log variable. You will see that the error bars are larger when P is smaller, indicating that it is hard (inaccurate) to estimate the probability of rare events ($Y(X) = 1$ is rare). This will be frustrating because you will want to increase d to see the power, but those estimates require large sample size to be accurate.
- (h) Use the method of control variates to improve the accuracy of the basic estimator of P . For a control variate, take a linear function of the path X , such as

$$W(X) = \sum_{j=1}^d X_j .$$

You know that $E(W) = 0$ because the probability distribution is symmetric. You suspect that W is correlated with Y because paths that have $Y = 1$ (always positive) probably have large positive path sums W . See how much variance reduction you can get this way. Does it make the computations of part (g) more accurate?

- (i) (*time permitting, extra credit, possibly thankless*) Experiment with other linear control variates such as

$$W(X) = \sum_{j=1}^d a_j X_j .$$

They all have mean value zero. It might be that weighting (large a_j) only near the ends is a better control variate because a path that never touches $x \leq 0$ has to start and end by going in the positive direction.

- (j) Plot a few (three or four at most) paths with $Y = 1$ for the largest d where you can find them. It might be that looking at these will help you design a good control variate.

2. Importance sampling.

Let $X \sim \mathcal{N}(0, 1)$ be a standard normal. Write code to estimate the even power moments for $s = 1, 2, \dots$.

$$\mu_s = E[X^{2s}] .$$

Use the standard estimator

$$\hat{\mu}_s = \frac{1}{n} \sum_{j=1}^n X_j^{2s} .$$

- (a) Show that $\mu_2 = 3$, $\mu_3 = 5 \cdot 3$, $\dots$, $\mu_s = (2s - 1)(2s - 3) \dots$. *Hint.* This may be done using integration by parts, using $x^4 = x^3 \cdot x$, etc., as we did in class for the variance. Clearly, μ_s is very large when s is large.

- (b) Make a plot (as a function of s for practical s values) with error bars of

$$\frac{\hat{\mu}_s}{\mu_s}.$$

Make sure to scale the error bars with μ_s also. This should show that the relative accuracy of $\hat{\mu}_s$ gets worse for large s . Here “large” s could be as small as $s = 4$.

- (c) μ_s is given by the integral formula

$$\mu_s = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x^{2s} e^{-\frac{1}{2}x^2} dx.$$

Make a figure with the integrand (normalized by its maximum value) and the PDF part $e^{-\frac{1}{2}x^2}$ on the same plot. This should show (if you choose s large but not too large) that most of the “mass” of the integral is over parts of the x axis that have low probability. Thus, most of the samples $X \sim \mathcal{N}(0, 1)$ contribute only slightly to the actual answer (which is large). This explains why even a large number of samples can give an inaccurate estimate of μ_s .

- (d) Experiment with an importance sampling strategy using the stretched distribution $g = \mathcal{N}(0, \sigma^2)$ for $\sigma > 1$. Write the PDF for g and the likelihood ratio

$$L(x, \sigma) = \frac{f(x)}{g(x, \sigma)}.$$

Estimate μ_s using (together with samples $X_j \sim g$)

$$\mu_s = \mathbb{E}_g [L(X, \sigma) X^{2s}].$$

Make a plot of the error bar size as a function of σ and some $s > 1$ to see that an $s > 1$ can reduce the variance.

- (e) Playing around a (possibly long) while, it is possible to see that the most effective σ is $\sigma \approx \sqrt{2s}$. Use this to see what s values you can estimate μ_s for. The relative error still grows with s , but slower than before.