COMP 480/580 — Probabilistic Algorithms and Data Structure Feb 28, 2019

Lecture 14: Basic Sampling

Lecturer: Anshumali Shrivastava *Scribe By: Maomao Ding and Erik Yamada*

1 What is Sampling?

In this section we address practical methods for generating samples and estimating properties of probability distributions. In most computers, we only have access to a *uniform* pseudo-random number generator $U[0,1]$. We often with to draw samples from some other distribution $f(\cdot)$: We would like to generate **s** such that $x_1, x_2, x_3, ..., x_n \sim f$. How can we do this?

Since we only have access to a uniform distribution, we need some way to transform samples or properties of the uniform distribution so that we get the distribution we want ($f(\cdot)$). We will discuss three methods.

1. Inversion Method
2. Importance Sampling
3. Rejection Sampling

2 Inversion Method

The inversion method requires that we find the inverse of the cumulative distribution function (CDF) of $f(\cdot)$. Here we will use capital $F(\cdot)$ to denote the CDF and lowercase $f(\cdot)$ to denote the probability density function (PDF). To use the inversion method, we find the inverse function $F^{-1}(\cdot)$ and apply $F^{-1}(\cdot)$ to $X \sim U[0,1]$. It should be noted that $F(\cdot)$ is monotone increasing by the definition of the CDF. Therefore, $F^{-1}(\cdot)$ exists. However, $F^{-1}(\cdot)$ may not be easy to find - to use the inverse method one must be able to find or approximate $F^{-1}(\cdot)$.

The inversion method proceeds as follows:

Algorithm 1 Inversion Method

Draw $y_k \sim U[0,1]$, $k = 1, \ldots, n$
Set $x_k = F^{-1}(y_k)$

To show that $X \sim f$, suppose we have a uniform random variable $Y \sim U[0,1]$. Then we wish to show that $F^{-1}(Y) \sim f$. Note that this is true if and only if the CDFs are the same. That is, if $P(F^{-1}(Y) \leq x) = F(x)$. To prove this, start with

$$P(F^{-1}(Y) \leq x)$$

Now apply $F(\cdot)$ to both sides of the inequality. Since $F(\cdot)$ is monotone, the inequality is still valid. Since $F(\cdot)$ is increasing, the direction (sign) of the inequality does not change.

$$P(F(F^{-1}(Y)) \leq F(x))$$

$F(F^{-1}(Y)) = Y$, so we have that

$$P(Y \leq F(x))$$

Y is uniformly distributed, and so $P(Y \leq F(x)) = F(x)$ by the cumulative distribution of the uniform distribution. Therefore we have

$$P(F^{-1}(Y) \leq x) = F(x)$$

as desired.

3 Importance Sampling

Importance sampling is a way to estimate the properties (but not obtain samples) of f when we only have access to $U[0,1]$. In importance sampling we consider the integral

$$I = \int w(x)f(x)dx$$

where $w(x)$ is some function. Suppose we wish to estimate the mean of $f(x)$, but we do not have access to $f(x)$ - instead we have access to $g(x)$. In our discussion here, $g(x)$ is the standard uniform PDF. We can multiply the integrand by $1 = g(x)/g(x)$ and obtain

$$I = \int \frac{w(x)f(x)}{g(x)} g(x)dx$$

which we recognize as the expectation of $\frac{w(x)f(x)}{g(x)}$:

$$I = \mathbb{E}_g \left[\frac{w(x)f(x)}{g(x)} \right]$$

To actually estimate this expectation, compute an estimator $\hat{I}$ of I, for $x_1, x_2, ...x_N \sim g$,

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

Or, put more succinctly,

Algorithm 2 Importance Sampling

Draw $x_k \sim g$, $k = 1, \ldots, n$
Report $\hat{I} = \frac{1}{n} \sum_{k=1}^{n} \frac{f(x_k)}{g(x_k)} w(x_k)$

Here, $\frac{f(x_k)}{g(x_k)}$ is called importance weight. It is easy to see that $\hat{I}$ is an unbiased estimator. Since we want a good estimate of I, we are also interested in the variance. The variance of $\hat{I}$ is

$$Var(\hat{I}) = \frac{1}{n} \left[\int_0^1 \frac{w^2(x)f^2(x)}{g(x)} dx - I^2 \right]$$

Using the Cauchy-Schwarz inequality

$$\int_0^1 \frac{w^2(x)f^2(x)}{g(x)} dx = \int_0^1 \frac{w^2(x)f^2(x)}{g(x)} dx \cdot \int_0^1 g(x)dx \geq I^2$$

It should be noted that g cannot be arbitrary chosen - some choices for g will not work. In general the support of g must cover that of f and g must have a heavier tail than f.

To reduce the variance of $\hat{I}$, by the equality condition of Cauchy-Schwarz inequality, we need the curve $\frac{w^2(x)f^2(x)}{g(x)}$ and $g(x)$ to be as "well-aligned" as possible. That is, we require that the proportion between $\frac{w^2(x)f^2(x)}{g(x)}$ and $g(x)$ be nearly constant.

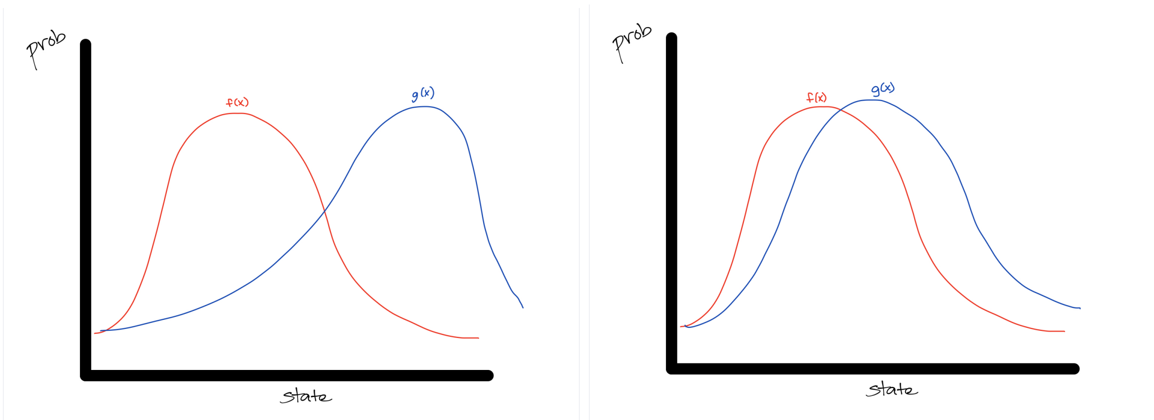

Figure 1: Choice of g given f for importance sampling. For low-variance estimates, we need a g that is similar to f. The plot on the left shows a bad choice for g, while the plot on the right shows a good choice.

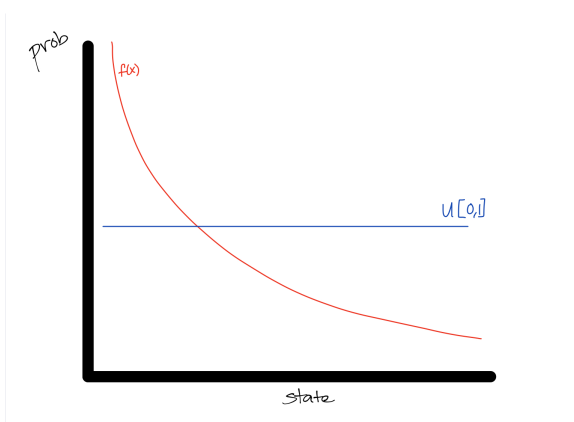

Figure 2: A situation where a uniform $g(x)$ is not the best choice. In general, one should try to match the shape of $g(x)$ to $f(x)$ as far as is possible

Why do we need importance sampling? In general, we know that $\int w(x)f(x)dx = E[w(x)]$ when sampling over the distribution $f(x)$. It is trivial to come up with an estimator for this quantity without using importance sampling when we can sample from $f(x)$. We do this by simply drawing $x_1, x_2, ..., x_n \sim f(x)$ and reporting $\frac{1}{n}\sum_i w(x_i)$. However it is not always possible (or efficient) to sample from $f(x)$. In this case, we use importance sampling to sample from $g(x)$ and push the dependence on f into the importance weight.

4 Monte Carlo Estimation

Monte Carlo estimation is a special case of importance sampling, when we use the uniform distribution for $g(x)$. Suppose we wish to evaluate an integral with no closed-form solution, such as

$$I = \int_0^1 e^{-x^3} dx$$

We can use importance sampling to solve this problem. Let $g(x) = f(x) = 1$ on the interval $[0, 1]$ - that is, $g, f = U[0, 1]$. Now let $w(u) = e^{-u^3}$, where $u \sim U[0, 1]$, and note that

$$I = \mathbb{E}_u[e^{-u^3}]$$

Using our result from importance sampling, we may estimate I using $\hat{I}$.

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} e^{-u_i^3}$$

where $u_1, u_2, ...u_N \sim U[0, 1]$.

5 Rejection Sampling

Importance sampling and Monte Carlo estimation were able to find the values of integrals involving f, but were not able to return actual samples from f. In this section we provide a method to generate actual samples from $f(x)$. Our goal is to sample from some complex distribution f, where f is the PDF of a random variable X. We know how to compute $f(x)$ for every x, but we are given a sampler for another PDF, $g(x)$. Additionally, we know that there is some value M such that $f(x) \leq Mg(x), \forall x$. Rejection sampling works as follows.

Algorithm 3 Rejection Sampling

```
Draw x ~ g
Draw u ~ U[0,1]
if u ≤ f(x)/(M·g(x)) then
    Accept x
else
    Loop
end if
```

Optimizing rejection sampling This procedure guarantees that every sample that you generate will be from the distribution $f(x)$, but $g(x)$ and M must be carefully chosen. Otherwise, a large portion of x will be rejected and the sampling procedure will be extremely slow. To make the sampling procedure fast, we want the following items:

1. Minimize M. We want to pick the smallest possible M such that $\forall x, f(x) \leq M * g(x)$. A smaller M implies a higher probability of accepting samples, which will mean that the algorithm will take less time to collect samples.

2. Choose a $g(x)$ that is similar to $f(x)$. Choosing a good $g(x)$ will result in choosing a smaller M.

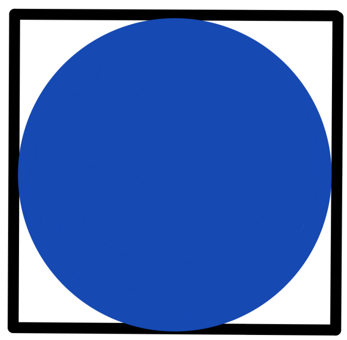

Figure 3: Dart throwing example of rejection sampling. First, we randomly generate points within the square. Then, we accept the points within the circle and reject the points outside the circle. Increasing M is equivalent to increasing the size of the square (fewer randomly sampled points will overlap with the circle).