Importance Sampling

Max Turgeon

STAT 3150—Statistical Computing

Lecture Objectives

- Estimate integrals using importance sampling.
- Learn strategies for choosing an appropriate importance function.
- Understand how importance sampling is a form of variance reduction.

Motivation

- In the last module, we talked about Monte Carlo integration, and how we could estimate integrals by rewriting them as an expectation.
    - It gave us a powerful method where we sample from a distribution X and transform through a function g to estimate $E(g(X))$.
- **Importance sampling** is a different way to tackle the same problem, by re-weighting samples from one distribution so that it matches a different distribution.
    - *Why?* Because it gives us another way to reduce the variance of our estimate.

- The setup is the same as earlier: suppose we want to estimate an integral of the form

$$\theta = \int_A g(x)f(x)dx,$$

where $f(x)$ is a density supported on A.
- If we have a function $\phi(x)$ that is positive on A, i.e. $\phi(x) > 0$ for all $x \in A$, we can also write

$$\theta = \int_A g(x)\frac{f(x)}{\phi(x)}\phi(x)dx.$$

Importance sampling ii

- **Why?** If ϕ is a density, we have just found a relationship between two expectations:

$$E_f(g(X)) = E_\phi\left(\frac{g(X)f(X)}{\phi(X)}\right).$$

- The goal would then be to choose a density ϕ such that:
    - It is (relatively) easy to sample from ϕ.
    - We can minimize the variance of $Y = \frac{g(X)f(X)}{\phi(X)}$.

Example i

- We will look at the following integral:

$$\int_0^1 \frac{e^{-x}}{1+x^2}dx.$$

- One way to write this integral as an expectation is by using a uniform on $(0,1)$:

$$\int_0^1 \frac{e^{-x}}{1+x^2}dx = E\left(\frac{e^{-X}}{1+X^2}\right), \quad X \sim U(0,1).$$

- We will look at $\phi(x) = e^{-x}$, i.e. the exponential density.
    - But note that the density is supported on a *larger* set than $(0,1)$.

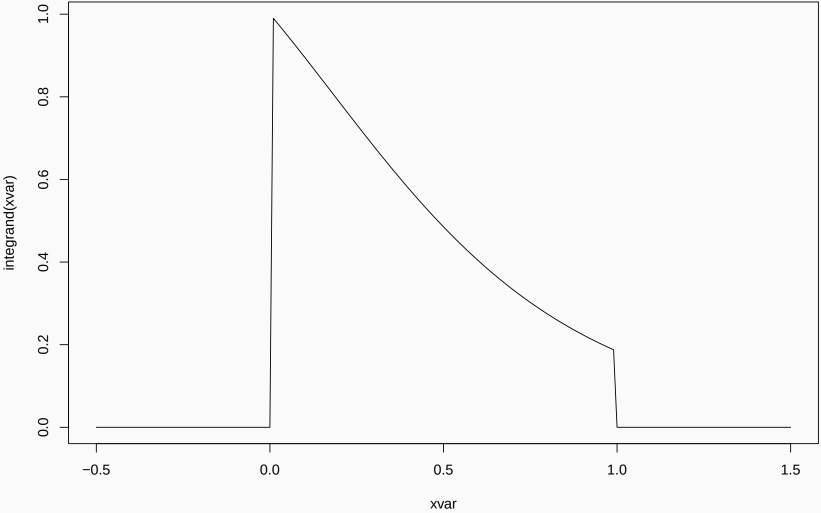
Example ii

```r
# Sample size
n <- 5000
# Define a function for integrand
integrand <- function(x) {
  # We want to multiply by zero if outside the range
  supp_ind <- as.numeric(x > 0 & x < 1)
  return(supp_ind * exp(-x)/(1 + x^2))
}
```

Example iii

```
# Look at the graph of the function
xvar <- seq(-0.5, 1.5, by = 0.01)
plot(xvar, integrand(xvar), type = "l")
```

Example iv



Example v

```r
# 1. Basic MC integration
unif_vars <- runif(n)

theta1 <- mean(integrand(unif_vars))
sd1 <- sd(integrand(unif_vars))
```

```r
# 2. Exponential density
exp_vars <- -log(unif_vars)

theta2 <- mean(integrand(exp_vars)/dexp(exp_vars))
sd2 <- sd(integrand(exp_vars)/dexp(exp_vars))
```

Example vi

```
# Compare results
c(theta1, theta2)
```

```
## [1] 0.5289111 0.5187084
```

```
c(sd1, sd2)/sqrt(n)
```

```
## [1] 0.003445435 0.005892648
```

- So the importance sampling algorithm seems to work, but the standard error is about the same as basic Monte Carlo integration. Can we do better?

Example vii

- **Key observation**: because some exponential samples fall outside the interval $(0, 1)$, they don't actually contribute to the estimate...

```
# How many are zeros?
sum(integrand(exp_vars) == 0)
```

```
## [1] 1853
```

- Therefore, we should probably restrict the domain of the exponential to $(0, 1)$.
- **Check**: $\int_0^1 e^{-x}dx = 1 - e^{-1}$.

Example viii

- We will use the following density:

$$\phi_2(x) = \frac{e^{-x}}{1 - e^{-1}}.$$

- **How can we generate from this density?** Inverse-transform!
- First, note that for $x \in (0, 1)$:

$$\begin{aligned} F(x) &= \int_0^x \frac{e^{-y}}{1 - e^{-1}} dy \\ &= \frac{1 - e^{-x}}{1 - e^{-1}}. \end{aligned}$$

Example ix

- We can then get the quantile function through inversion:

$$
\begin{aligned}
p = \frac{1 - e^{-x}}{1 - e^{-1}} &\Leftrightarrow p(1 - e^{-1}) = 1 - e^{-x} \\
&\Leftrightarrow e^{-x} = 1 - p(1 - e^{-1}) \\
&\Leftrightarrow x = -\log\left(1 - p(1 - e^{-1})\right).
\end{aligned}
$$

Example x

```r
# 3. Truncated exponential density
unif_vars <- runif(n)
truncexp_vars <- -log(1 - unif_vars*(1 - exp(-1)))

# Evaluate the density at those points
phi_vars <- exp(-truncexp_vars)/(1 - exp(-1))

theta3 <- mean(integrand(truncexp_vars)/phi_vars)
sd3 <- sd(integrand(truncexp_vars)/phi_vars)
```

Example xi

```r
# Compare results
c(theta1, theta2, theta3)
```

```
## [1] 0.5289111 0.5187084 0.5265824
```

```r
c(sd1, sd2, sd3)/sqrt(n)
```

```
## [1] 0.003445435 0.005892648 0.001355448
```

Exercise

Suppose that $f(x)$ is the density of a standard normal distribution, and that $g(x) = \exp\left(-\frac{1}{2}(x-2)^2\right)$. Use important sampling to estimate $E_f(g(X))$ using 1) $phi(x)$ is the density of a standard normal; 2) $phi(x)$ is the density of $N(2, 1)$.

Solution i

- First, we sample from $N(0, 1)$, i.e. normal MC integration.

```
n <- 3150
integrand <- function(x) exp(-0.5*(x - 2)^2)
norm_vars <- rnorm(n)
theta1 <- mean(integrand(norm_vars))
std_er1 <- sd(integrand(norm_vars))/sqrt(n)

c(theta1, std_er1)
```

```
## [1] 0.254829449 0.005124054
```

Solution ii

- Next we sample from $N(2, 1)$. We can simply shift our previous sample.

```
norm_vars2 <- norm_vars + 2
phi_vars <- dnorm(norm_vars2, mean = 2)
theta2 <- mean(integrand(norm_vars2)*dnorm(norm_vars2)/p
std_er2 <- sd(integrand(norm_vars2)*dnorm(norm_vars2)/ph

c(theta2, std_er2)
```

```
## [1] 0.258526360 0.005075798
```

Variance comparison i

- In the example above, we looked at three different approaches:
  - $E\left(\frac{e^{-X}}{1+X^2}\right)$, where $X \sim U(0,1)$;
  - Sampling from $Exp(1)$ and throwing away samples that fall outside $(0,1)$;
  - Sampling from an $Exp(1)$ truncated to the interval $(0,1)$.
- It's easy to see why the first and third approach were better than the second:
  - They used all the samples.
- But why was the third approach better than the first?

Variance comparison ii

Theorem

The best density ϕ, i.e. the one that minimizes variance, is given by

$$\phi^*(x) = \frac{|g(x)|f(x)}{\int_A |g(t)|f(t)dt}.$$

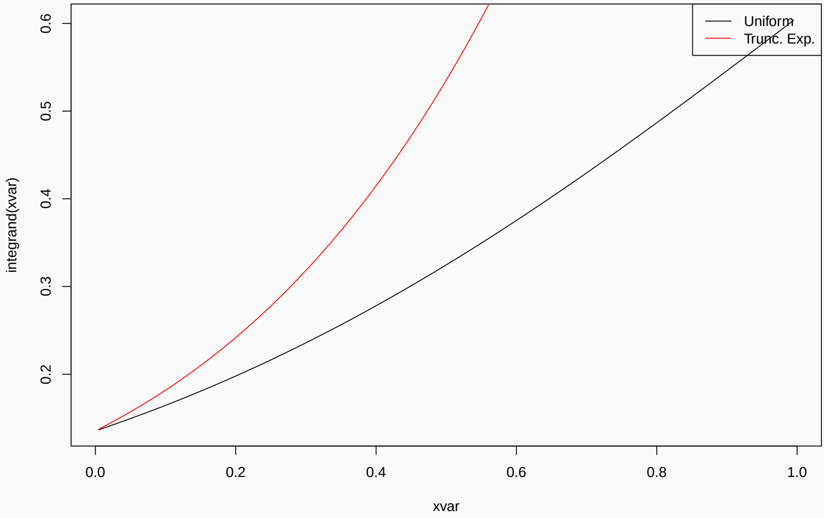
- Of course, we typically can't compute the denominator, otherwise we wouldn't need to estimate it!
- But the general idea is **we want ϕ to look like** $|g(x)|f(x)$.
- In our example above, $\phi = \frac{e^{-x}}{1-e^{-1}}$ looks more like $|g(x)|f(x)$ than $\phi(x) = 1$.

Visualization i

- We can check by plotting the ratio $\frac{|g(x)|f(x)}{\phi(x)}$.
  - We want it to be *almost constant*, i.e. close to horizontal.

```
# Points between 0 and 1 without boundary
xvar <- ppoints(100)
plot(xvar, integrand(xvar), type = "l")
lines(xvar, integrand(xvar)/exp(-xvar), col = "red")
legend(x = "topright",
       legend = c("Uniform", "Trunc. Exp."),
       col = c("black", "red"), lty = 1)
```

Visualization ii



Example i

- Suppose we want to estimate a tail probability of a standard normal variable $X \sim N(0, 1)$. Specifically, we want to estimate $P(X > 5)$.
- We will explore a few different ways of estimating this quantity, trying to find the most efficient estimate.
- First, we can use the “hit-or-miss” approach, i.e. sample from a standard normal and count the proportion of samples that are greater than 5.

Example ii

```r
n <- 5000
norm_vars <- rnorm(n)
# Average of 0s and 1s gives proportion of 1s
mean(norm_vars > 5)
```

```
## [1] 0
```

- This tail probability is so small that we didn't generate any value greater than 5... let's increase the sample size.

Example iii

```r
n <- 10000000
norm_vars <- rnorm(n)
# Average of 0s and 1s gives proportion of 1s
mean(norm_vars > 5)
```

```
## [1] 3e-07
```

- So we had 3 out of 10 million samples! But we can use the symmetry of the standard normal to do slightly better.

Example iv

```
# Check if > 5 in absolute value, and divide by 2
0.5*mean(abs(norm_vars) > 5)
```

```
## [1] 3e-07
```

```
# Compare both standard errors
c(sd(norm_vars > 5), 0.5*sd(abs(norm_vars) > 5))
```

```
## [1] 0.0005477225 0.0003872982
```

- Let’s see if we can do better using importance sampling.

Example v

- The main problem with our approach above is that most samples don't count towards tail probabilities.
- **Solution**: Sample from a distribution where *every* sample will count towards the tail probabilities.
    - E.g. a shifted exponential, with support $(5, \infty)$.
- **Exercise**: the density is given by $\phi(x) = \exp(-x + 5)$

Example vi

```r
# Shifted exponential density
unif_vars <- runif(n)
shiftexp_vars <- -log(unif_vars) + 5

# Evaluate the density at those points
phi_vars <- exp(-(shiftexp_vars - 5))

theta_est <- mean(dnorm(shiftexp_vars)/phi_vars)
sd_est <- sd(dnorm(shiftexp_vars)/phi_vars)
```

Example vii

```r
# Compare all three approaches
c("Method1" = mean(norm_vars > 5),
  "Method2" = 0.5*mean(abs(norm_vars) > 5),
  "Method 3" = theta_est)
```

```
##      Method1      Method2     Method 3
## 3.000000e-07 3.000000e-07 2.866245e-07
```

```r
c("Method1" = sd(norm_vars > 5),
  "Method2" = 0.5*sd(abs(norm_vars) > 5),
  "Method 3" = sd_est)
```

Example viii

```
##      Method1      Method2     Method 3
## 5.477225e-04 3.872982e-04 3.972687e-07
```

- This corresponds to a variance reduction of 975 times!
- In other words, with Method 3, we can achieve the same precision as Method 2 by using 31 times less samples.

- As we can see, the posterior is supported on $(0, 1)$ with a peak around $\pi = 0.1$.
- Recall that the beta distribution $\mathrm{Beta}(\alpha, \beta)$ is supported on $(0, 1)$ with a peak at $\frac{\alpha-1}{\alpha+\beta+2}$.
- This suggests using $\mathrm{Beta}(2, 10)$ for the density ϕ.

Example ix

```r
# Create a function for posterior and weight
post_fun <- function(pi) {
  1*choose(294, 32)*pi^32*(1 - pi)^262
}

weight <- function(pi) {
  post_fun(pi)/dbeta(pi, shape1 = 2,
                         shape2 = 10)
}
```

Example x

```r
# Assume we are interested in posterior mean
# so g(x) = x
n <- 5000
beta_vars <- rbeta(n, shape1 = 2, shape2 = 10)

denominator <- weight(beta_vars)
numerator <- weight(beta_vars)*beta_vars
(theta <- mean(numerator)/mean(denominator))
```

```
## [1] 0.1112311
```

Example xi

```r
# What about posterior variance?
denominator <- weight(beta_vars)
numerator <- weight(beta_vars)*beta_vars^2 # g(x) = x^2
theta2 <- mean(numerator)/mean(denominator)
# Var(X) = E(X^2) - E(X)^2
theta2 - theta^2
```

```
## [1] 0.0003368469
```

Example xii

```
# What about posterior probability that pi
# is between 0.08 and 0.12?
denominator <- weight(beta_vars)
# g(x) is indicator function
gvals <- as.numeric(beta_vars > 0.08 &
                      beta_vars < 0.12)
numerator <- weight(beta_vars)*gvals
mean(numerator)/mean(denominator)
```

```
## [1] 0.673715
```

–>