Importance Sampling

The methods we've introduced so far generate arbitrary points from a distribution to approximate integrals– in some cases many of these points correspond to points where the function value is very close to 0, and therefore contributes very little to the approximation. In many cases the integral "comes with" a given density, such as integrals involving calculating an expectation. However, there will be cases where another distribution gives a better fit to integral you want to approximate, and results in a more accurate estimate; importance sampling is useful here. In other cases, such as when you want to evaluate $E(X)$ where you can't even generate from the distribution of X, importance sampling is necessary. The final, and most crucial, situation where importance sampling is useful is when you want to generate from a density you only know up to a multiplicative constant.

The logic underlying importance sampling lies in a simple rearrangement of terms in the target integral and multiplying by 1:

$$\int h(x)p(x)dx = \int h(x)\frac{p(x)}{g(x)}g(x)dx = \int h(x)w(x)g(x)dx$$

here $g(x)$ is another density function whose support is the same as that of $p(x)$. That is, the sample space corresponding to $p(x)$ is the same as the sample space corresponding to $g(x)$ (at least over the range of integration). $w(x)$ is called the importance function; a good importance function will be large when the integrand is large and small otherwise.

1 Importance sampling to improve integral approximation

As a first example we will look at a case where importance sampling provides a reduction in the variance of an integral approximation. Consider the function $h(x) = 10\exp(-2|x-5|)$. Suppose that we want to calculate $E(h(X))$, where $X \sim \text{Uniform}(0,1)$. That is, we want to calculate the integral

$$\int_0^{10} \exp(-2|x-5|)\,dx$$

The true value for this integral is about 1. The simple way to do this is to use the approach from lab notes 6 and generate X_i from the uniform(0,10) density and look at the sample mean of $10 \cdot h(X_i)$ (notice this is equivalent to importance sampling with importance function $w(x) = p(x)$):

```
X <- runif(100000,0,10)
Y <- 10*exp(-2*abs(X-5))
c( mean(Y), var(Y) )
[1] 0.9919611 3.9529963
```

The function h in this case is peaked at 5, and decays quickly elsewhere, therefore, under the uniform distribution, many of the points are contributing very little to this expectation. Something more like a gaussian function (ce^{-x^2}) with a peak at 5 and small variance, say, 1, would provide greater precision. We can re-write the integral as

$$\int_0^{10} 10\exp(-2|x-5|)\,\frac{1/10}{\frac{1}{\sqrt{2\pi}}e^{-(x-5)^2/2}}\frac{1}{\sqrt{2\pi}}e^{-(x-5)^2/2}dx$$

That is, $E(h(X)w(X))$, where $X \sim N(5,1)$. So in this case $p(x) = 1/10$, $g(x)$ is the $N(5,1)$ density, and $w(x) = \frac{\sqrt{2\pi}e^{(x-5)^2/2}}{10}$ is the importance function in this case. This integral can be more compactly written as

$$\int_0^{10} \exp\left(-2|x-5|\right)\sqrt{2\pi}e^{(x-5)^2/2} \cdot \frac{1}{\sqrt{2\pi}}e^{-(x-5)^2/2}dx$$

where the part on the left is the quantity whose expectation is being calculated, and the part on the right is the density being integrated against ($N(5,1)$). We implement this second approach in R by

```
w <- function(x) dunif(x, 0, 10)/dnorm(x, mean=5, sd=1)
f <- function(x) 10*exp(-2*abs(x-5))
X=rnorm(1e5,mean=5,sd=1)
Y=w(X)*f(X)
c( mean(Y), var(Y) )
[1] 0.9999271 0.3577506
```

Notice that the integral calculation is still correct, but with a variance this is approximately 1/10 of the simple monte carlo integral approximation. This is one case where importance sampling provided a substantial increase in precision. A plot of the integrand from solution 1:

$$10\exp\left(-2|x-5|\right),$$

along with the density it is being integrated against:

$$p(x) = 1/10,$$

and a second plot of the integrand from solution 2:

$$\exp\left(-2|x-5|\right)\sqrt{2\pi}e^{(x-5)^2/2},$$

along with the density it is being integrated against:

$$p(x) = \frac{1}{\sqrt{2\pi}}e^{-(x-5)^2/2},$$

gives some intuition about why solution 2 was so much more efficient. Notice the integrands are on much larger scale than the densities (since densities must integrate to 1), so the integrands are normalized to make the plots comparable.

2 Importance sampling when you cannot generate from the density p

As another example we will consider estimating the moments of a distribution we are unable to sample from. Let

$$p(x) = \frac{1}{2}e^{-|x|}$$

which is called the double exponential density. The CDF is

$$F(x) = \frac{1}{2}e^x\mathcal{I}(x \leq 0) + (1 - e^{-x}/2)\mathcal{I}(x > 0)$$

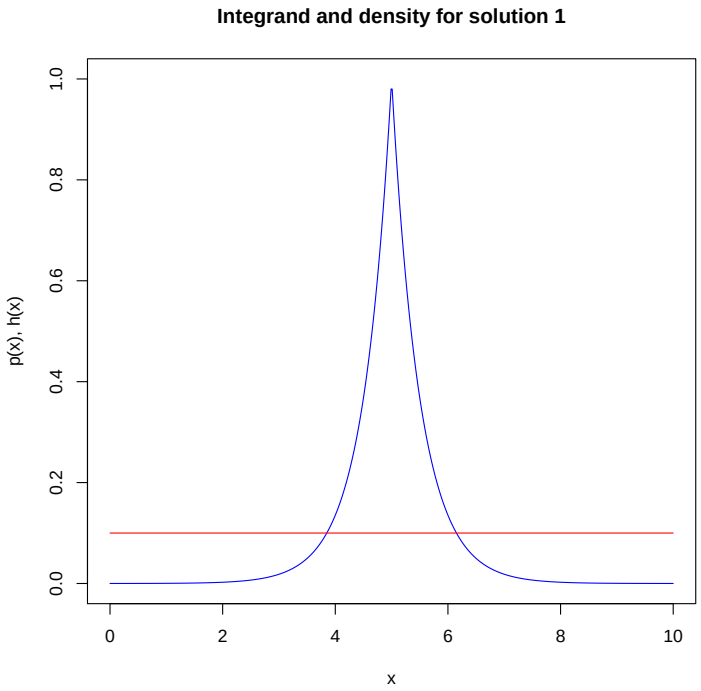

Figure 1: Normalized integrand (blue) and the density being integrated against (red) for approach 1

which is a piecewise function and difficult to invert (it is possible to generate from this distribution but lets pretend it is not). Suppose you want to estimate $E(X^2)$ for this distribution, which is support on $\mathcal{R}$. That is, we want to calculate the integral

$$\int_{-\infty}^{\infty} x^2 \frac{1}{2} e^{-|x|} dx$$

We cannot estimate this empirically without generating from p. However, we can re-write this as

$$\int_{-\infty}^{\infty} x^2 \frac{\frac{1}{2}e^{-|x|}}{\frac{1}{\sqrt{8\pi}}e^{-x^2/8}} \frac{1}{\sqrt{8\pi}} e^{-x^2/8} dx$$

Notice that $\frac{1}{\sqrt{8\pi}}e^{-x^2/8}$ is the $N(0,4)$ density. Now this amounts to generating $X_1, X_2, ..., X_N$ from $N(0,4)$ and estimating

$$E\left(X^2 \frac{\frac{1}{2}e^{-|X|}}{\frac{1}{\sqrt{8\pi}}e^{-X^2/8}} \right)$$

by the sample mean of this quantity. The following R code does this:

```
X <- rnorm(1e5, sd=2)
Y <- (X^2) * .5 * exp(-abs(X))/dnorm(X, sd=2)
mean(Y)
[1] 1.998898
```

The true value for this integral is 2, so importance sampling has done the job here. Other

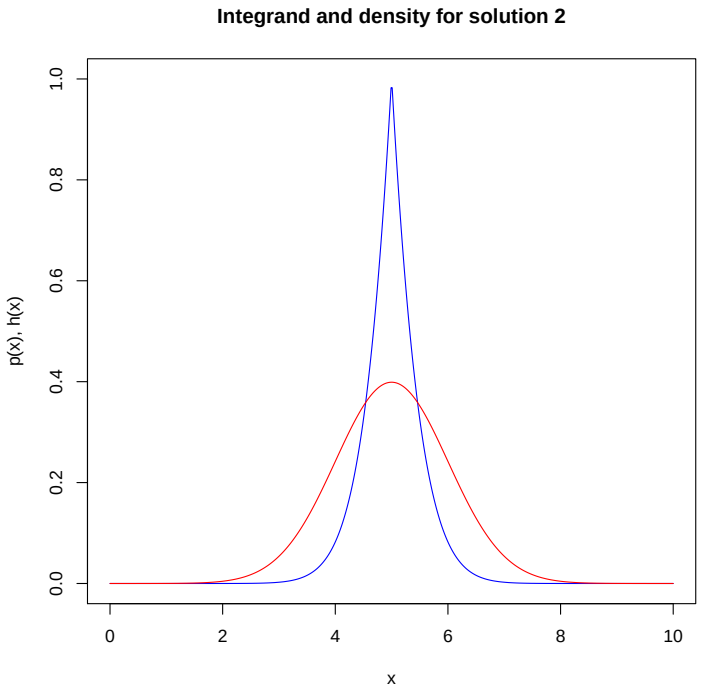

Figure 2: Normalized integrand (blue) and the density being integrated against (red) for approach 2

moments, $E(X), E(X^3), E(X^4), ...$ for $X \sim p$ can be estimated analogously, although it should be clear that all odd moments are 0.

Exercise: Can you find a better g than the $N(0, 4)$ density in this problem? A better choice of g will correspond to lower estimated variance in the estimated integral.

3 Importance Sampling when the target density is unnormalized

A function is a probability density on the interval $\mathcal{I}$ if the function is non-negative and integrates to 1 over $\mathcal{I}$. Therefore for any non-negative function f such that $\int_{\mathcal{I}} f(x)dx = C$, the function $p(x) = f(x)/C$ is a density on $\mathcal{I}$; f is referred to as the *unnormalized* density and C the *normalizing constant*. In many inference problems you will want to determine properties, such as the mean, of a distribution where you only have knowledge of the unnormalized density.

If you want to calculate $E(h(X))$ where the unnormalized density of X is f, you can still use importance sampling. This can we re-written as

$$E(h(X)) = \int h(x)f(x)/Cdx = \frac{1}{C}\int h(x)\frac{f(x)}{g(x)}g(x)dx = \frac{1}{C}\int h(x)\tilde{w}(x)g(x)dx$$

where $\tilde{w}(x) = \frac{f(x)}{g(x)} = C\frac{p(x)}{g(x)} = Cw(x)$, where p is the normalized density. By the law of large numbers, if $X_1, X_2, ..., X_N$ are iid draws from g, then

$$\frac{1}{N}\sum_{i=1}^{N} h(X_i)\tilde{w}(X_i) \to C \cdot E(h(X))$$

as $N \to \infty$. Also by LLN,

$$\frac{1}{N}\sum_{i=1}^{N} \tilde{w}(X_i) \to C$$

Therefore a sensible estimator of $E(h(X))$ is

$$\overline{E(h(X))}_{IS} = \frac{\frac{1}{N}\sum_{i=1}^{N} h(X_i)\tilde{w}(X_i)}{\sum_{i=1}^{N} \tilde{w}(X_i)}$$

This method is only considered reliable when the weights are not too variable. As a rule of thumb, when

$$ESS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\tilde{w}(X_i)}{\overline{w}} - 1\right)^2}$$

is less than 5, this method is reasonable. Here $\overline{w}$ is the sample mean of the $\tilde{w}(X_i)$'s. **You will know you've chosen a bad g is ESS is large.** When $ESS < 5$, the variance of $\overline{E(h(X))}_{IS}$ can be estimated as

$$\sigma^2_{IS} = \frac{1}{N}\sum_{i=1}^{N}(Z_i - \overline{E(h(X))}_{IS})^2$$

where

$$Z_i = \frac{\tilde{w}(X_i)h(X_i)}{\overline{w}}.$$

3.1 Bayesian Inference

The primary examples where you want to determine properties of a distribution given only the unnormalized density come from Bayesian inference. Suppose you observe data $x_1, ..., x_n$ with density p_θ indexed by a parameter θ. A typical approach is to view θ as a fixed number you want to determine and and estimate it by

$$\max_\theta p_\theta(x_1, ..., x_n)$$

which is called the maximum likelihood estimator. Instead, a bayesian approach views the parameter as having a distribution when conditioned on the data, and look to make inference about the *posterior density*, $p(\theta|x_1, ..., x_n)$. Noting that $p_\theta(x) = p(x|\theta)$ and applying bayes rule twice:

$$\begin{aligned} p(\theta|x_1, ..., x_n) &= \frac{p(\theta, x_1, ..., x_n)}{p(x_1, ..., x_n)} \\ &= \frac{p(x_1, ..., x_n|\theta)p(\theta)}{p(x_1, ..., x_n)} \end{aligned}$$

where $p(\theta)$ is called the *prior density* and is the marginal distribution of θ. The prior can be used to influence the estimator based on prior knowledge. As the sample size increases, the prior becomes less relevant, since the likelihood scales with n, but the prior does not. In the denominator is the marginal distribution of the data

$$p(x_1, ..., x_n) = \int p(\theta, x_1, ..., x_n) d\theta$$

which is the normalizing constant for the posterior distribution and is often an intractable integral. We can apply importance sampling to the unnormalized posterior density

$$\tilde{p}(\theta|x_1, ..., x_n) = p(x_1, ..., x_n|\theta)p(\theta)$$

3.2 Handling numerical problems

When calculating the likelihood for a large sample you, the likelihood will often take on astronomically small values, so numerical considerations must be taken. For example, R would conclude the quantity

$$\frac{e^{-1000}}{e^{-1000} + e^{-1001}}$$

is NaN, because both the numerator and denominator are both 0, as far as R is concerned. However, we know this quantity is equal to $1/(1 + e^{-1}) = .731$. The importance function has a similar issue, since both it and its mean appear in the importance sampling estimate. It becomes easier to work with $\log(w(x)) = \log(p(x)) - \log(g(x))$, and is absolutely necessary when the sample size is relatively large. Letting $M = \max_i \log(w(X_i))$, we can write the following derivation

$$\begin{aligned}
\overline{E(h(X))}_{IS} &= \frac{\sum_{i=1}^N h(X_i)\tilde{w}(X_i)}{\sum_{i=1}^N \tilde{w}(X_i)} \\
&= \frac{\sum_{i=1}^N h(X_i)\exp\left(\log \tilde{w}(X_i)\right)}{\sum_{i=1}^N \exp(\log \tilde{w}(X_i))} \\
&= \frac{e^M \sum_{i=1}^N h(X_i)\exp\left(\log \tilde{w}(X_i) - M\right)}{e^M \sum_{i=1}^N \exp(\log \tilde{w}(X_i) - M)} \\
&= \frac{\sum_{i=1}^N h(X_i)\exp\left(\log \tilde{w}(X_i) - M\right)}{\sum_{i=1}^N \exp(\log \tilde{w}(X_i) - M)}
\end{aligned}$$

The final line is a more numerically stable version of the importance sampling estimator. The main trick that makes this work is that the same exponential terms appear in both the numerator and the denominator.

Examples

Example 1: Suppose $X_1, ..., X_n \sim \text{Binomial}(10, \theta)$ where $\theta \in (0, 1)$ has a Beta$(5, 3)$ prior density: $p(\theta) = \frac{\Gamma(8)}{\Gamma(5)\Gamma(3)}\theta^4(1-\theta)^2$. We want to estimate the mean of the posterior distribution: $\int_0^1 \theta p(\theta|x_1, ..., x_n)d\theta$. Take g to be the Beta(α, β) density, where

$$\begin{aligned}\alpha &= c\overline{X} \\ \beta &= c(10 - \overline{X})\end{aligned}$$

where $\overline{X}$ is the sample mean, and c is a positive constant. This will ensure that g is peaked near $\overline{X}/10$, which is where the posterior distribution should have a lot of mass. The larger c is, the more sharply peaked around $\overline{X}/10$ will be. Tune c to minimize the variance of the estimate. The first step is to calculate the joint distribution of the data, given θ:

$$\begin{aligned}p(x_1, ..., x_n|\theta) &= \prod_{i=1}^n p(x_i|\theta) \\ &= \prod_{i=1}^n \binom{10}{x_i}\theta^{x_i}(1-\theta)^{10-x_i} \\ &\propto \theta^{\sum_{i=1}^n x_i}(1-\theta)^{10n-\sum_{i=1}^n x_i} \\ &= \theta^{n\overline{X}}(1-\theta)^{n(10-\overline{X})}\end{aligned}$$

Next we derive an expression for something proportional to the posterior density:

$$\begin{aligned}p(\theta|x_1, ..., x_n) &\propto p(x_1, ..., x_n|\theta)p(\theta) \\ &\propto \theta^{n\overline{X}}(1-\theta)^{n(10-\overline{X})}p(\theta) \\ &= \theta^{n\overline{X}}(1-\theta)^{n(10-\overline{X})}\frac{\Gamma(8)}{\Gamma(5)\Gamma(3)}\theta^4(1-\theta)^2 \\ &\propto \theta^{n\overline{X}}(1-\theta)^{n(10-\overline{X})}\theta^4(1-\theta)^2 \\ &= \theta^{n\overline{X}+4}(1-\theta)^{n(10-\overline{X}))+2}\end{aligned}$$

The log of this quantity is

$$\big(n\overline{X} + 4\big)\log(\theta) + \big(n(10 - \overline{X})) + 2\big)\log(1-\theta)$$

Using the above as the log of the unnormalized target density, the following R code executes the importance sampling:

```
# X is the data, res is the number of monte carlo samples,
# C is the tuning parameter for the g distribution
IS <- function(X, C, res)
```

```
{

 # sample size
 n <- length(X)

 # log posterior derived above
 log.posterior <- function(t) (sum(X)+4)*log(t) + (n*(10-mean(X))-2)*log(1-t)

 # log trial density, g
 log.g <- function(t) dbeta(t,C*mean(X),C*(10-mean(X)), log=TRUE)

 # log importance function
 log.w <- function(t) log.posterior(t) - log.g(t)

 # generate from the trial distribution
 U <- rbeta(res, C*mean(X), C*(10-mean(X)))

 # calculate the list of log.w values
 LP <-  log.w(U)

    # factor out the largest value to prevent numerical underflow
    w <- max(LP)
    LP <- LP - w

 # importance sampling estimate
 I <- mean( exp(LP)*U )/mean(exp(LP))

 # calculate ESS
 ESS <- mean( ( (exp(LP)/mean(exp(LP))) - 1)^2 )

 # calculate s.e. of the IS estimate
 Z = exp(LP)/mean(exp(LP))
 sig.sq <- (1/res)*sum( (Z-I)^2 )
 se <- sqrt( sig.sq/res )

 # return the 95 percent confidence interval for the estimate
 return(c(I - 1.96*se, I + 1.96*se, ESS))
}

# Generate 100 binomial(10,.4) random variables
X = rbinom(100, 10, .4)

## calculate ESS and var(E(theta|X))
## for a grid of values for alpha
const <- seq(.1, 200, length=2000)
A <- matrix(0, 2000, 2)
for(j in 1:2000)
{
```

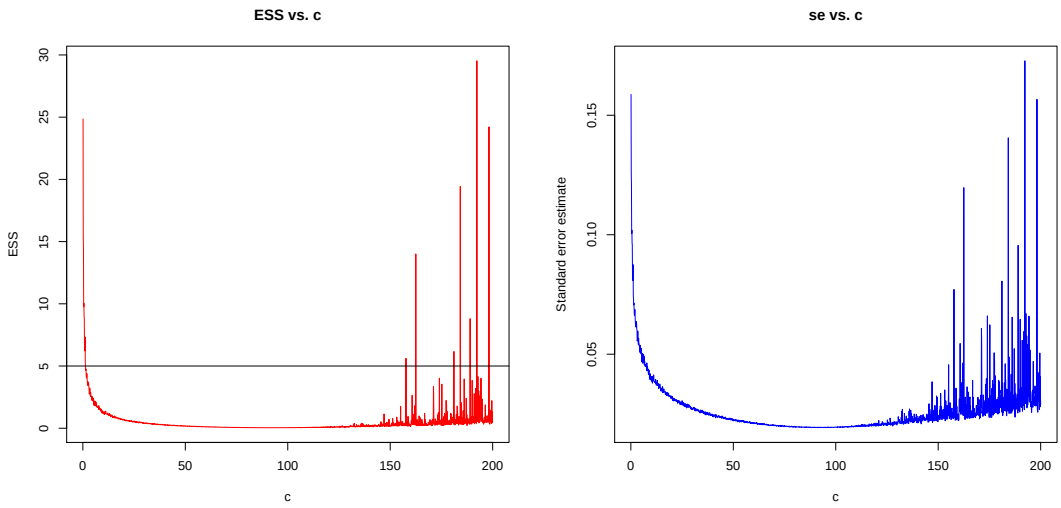

Figure 3: Values for ESS and the standard error of the integral as a function of c in example 1.

```
 OUT <- IS(X, const[j], 1000)
 ESS <- OUT[3]
 V <- (OUT[2]-OUT[1])/3.92
 A[j,] <- c(ESS, V)
}

# see where ESS is low enough
plot(const, A[,1], ylab="ESS", xlab="c", main="ESS vs. c", col=2, type="l")
abline(h=5)

# standard error estimates
plot(const, A[,2], xlab="c", ylab="Standard error estimate", main="se vs. c", col=4, t

# The final confidence interval
IS(X, const[which.min(A[,2])], 1000)[1:2]
[1] 0.3469836 0.4276585
```

It appears that for about $c > 120$ the estimates become very unstable; this is because the g density becomes practically a point mass at $\overline{X}/10$, so the ratio $p(\theta|x)/g(\theta)$ becomes a very high variance random variable. The optimal c appears to be around 100.

Example 2: Suppose $X_1, ..., X_n \sim N(0, \theta)$ and we specify a Gamma$(3, .5)$ distribution for the prior of θ. We will use a trial density g which is Gamma distributed with $\alpha = cs^2$, and $\beta = c$, where c is a positive constant, and s^2 is the sample variance. So the mean of the trial distribution will be s^2. Choose c to optimize estimation precision.

The joint distribution of the data, given θ, is

$$
\begin{aligned}
p(x_1, ..., y_n|\theta) &= \prod_{i=1}^{n} p(x_i|\theta) \\
&= \prod_{i=1}^{n} \sqrt{\frac{1}{2\pi\theta}} e^{-x_i^2/2\theta} \\
&= (\frac{1}{2\pi\theta})^{n/2} \exp\left(-\frac{1}{2\theta}\sum_{i=1}^{n} x_i^2\right) \\
&\propto \theta^{-n/2} \exp\left(-\frac{1}{2\theta}\sum_{i=1}^{n} x_i^2\right)
\end{aligned}
$$

Therefore the posterior distribution is proportional to

$$
\begin{aligned}
p(\theta|x_1, ..., x_n) &= p(x_1, ..., y_n|\theta)p(\theta) \\
&= \left(\theta^{-n/2} \exp\left(-\frac{1}{2\theta}\sum_{i=1}^{n} x_i^2\right)\right) \cdot 2^{-3}\Gamma(3)^{-1}\theta^2 e^{-\theta/2} \\
&\propto \left(\theta^{-n/2} \exp\left(\frac{1}{2\theta}\sum_{i=1}^{n} x_i^2\right)\right) \cdot \theta^2 e^{-\theta/2} \\
&= \theta^{(4-n)/2} \exp\left(-\frac{1}{2\theta}\left(\theta^2 + \sum_{i=1}^{n} x_i^2\right)\right)
\end{aligned}
$$

so the log posterior is a constant plus

$$
((4-n)/2)\log(\theta) - \frac{1}{2\theta}\left(\theta^2 + \sum_{i=1}^{n} x_i^2\right)
$$

The following R code executes importance sampling to estimate the mean of the posterior distribution and determines an optimal c for the trial density:

```
# X is the data, res is the number of monte carlo samples,
# C is the tuning parameter for the g distribution
IS <- function(X, C, res)
{

 # sample size
 n <- length(X)

 # posterior derived above
 log.posterior <- function(t) ( (4-n)/2 )*log(t) - (1/(2*t)) * (t^2 + sum(X^2))

 # parameters for the trial distribution
```

```
 a = C*var(X); b = C;

 # log trial density, g
 log.g <- function(t) dgamma(t,a,b,log=TRUE)

 # log importance function
 log.w <- function(t) log.posterior(t) - log.g(t)

 # generate from the trial distribution
 U <- rgamma(res, a, b)

 # calculate the list of log.w values
 LP <-  log.w(U)

    # factor out the largest value to prevent numerical underflow
    w <- max(LP)
    LP <- LP - w

 # importance sampling estimate
 I <- mean( exp(LP)*U )/mean(exp(LP))

 # calculate ESS
 ESS <- mean( ( (exp(LP)/mean(exp(LP))) - 1)^2 )

 # calculate s.e. of the IS estimate
 Z = exp(LP)/mean(exp(LP))
 sig.sq <- (1/res)*sum( (Z-I)^2 )
 se <- sqrt( sig.sq/res )

 # return the 95 percent confidence interval for the estimate
 return(c(I - 1.96*se, I + 1.96*se, ESS))
}

# Generate 100 N(0,4) random variables
X = rnorm(100,mean=0,sd=2)

## calculate ESS, E(theta|X), and var(E(theta|X))
## for a grid of values for c
const <- seq(.05, 20, length=500)
A <- matrix(0, 500, 2)
for(j in 1:500)
{

 OUT <- IS(X, const[j], 1000)
 ESS <- OUT[3]
 V <- (OUT[2]-OUT[1])/3.92
 A[j,] <- c(ESS, V)
}
```

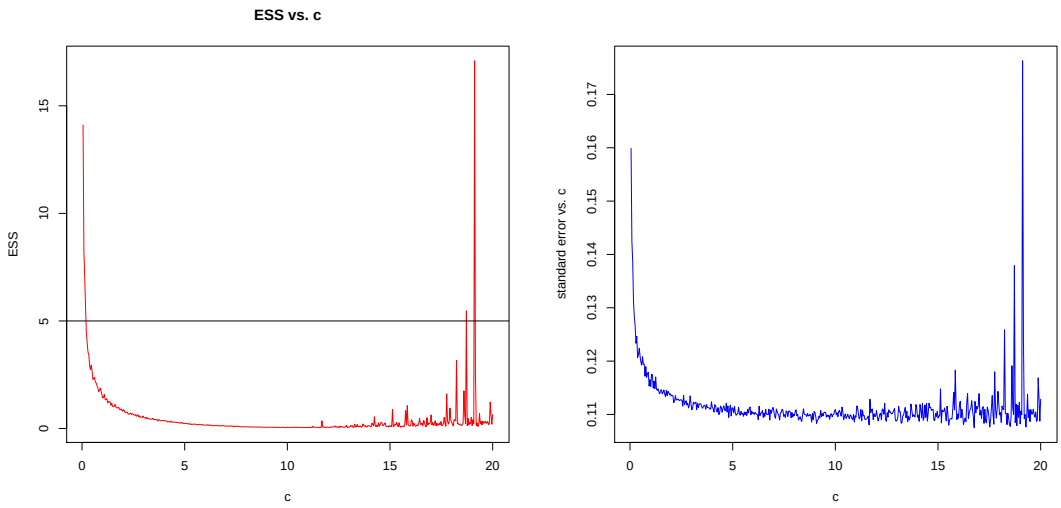

Figure 4: Values for ESS and the standard error of the integral as a function of c in Example 2

```
# see where ESS is low enough
plot(const, A[,1], ylab="ESS", xlab="c", main="ESS vs. c", col=2, type="l")
abline(h=5)

# variance estimates
plot(const, A[,2], xlab="c", ylab="standard error vs. c", col=4, type="l")

# final confidence interval
IS(X, const[which.min(A[,2])], 1000)[1:2]
[1] 3.867017 4.249979
```

Apparently when c is bigger than about 1, the corresponding estimate is stable, since the ESS is small. It appears that the larger c is, the better the corresponding integral estimate becomes. However, when c is too large, the estimates become unstable again, since the g distribution becomes to peaked at the sample variance, and the importance function becomes too high variance. It appears that $c \approx 10$ is optimal.

In both examples it is clear that the choice of g is crucial. When g is too flat ($c \downarrow 0$), the estimates are highly variable, as well as when g is too peaked around one location ($c \uparrow$).