Harrison Lam

Importance Sampling and Fast Simulation

The objective of this project is to evaluate the performance of importance sampling when estimating the probability of a rare event or when simulating a random variable is difficult.

Notation:

Let X, Y be random variables with densities $\pi(x), \pi'(x)$ respectively and the twisted density $\pi'(x) > 0$ whenever $\pi(x) > 0$

Let $L(y) = \frac{\pi(y)}{\pi'(y)}$ be the likelihood ratio

Suppose we want to estimate $\gamma = \mathbb{P}(X \in A)$. Let $\hat{\gamma} = L(Y)1[Y \in A]$. Then

$\mathbb{E}\hat{\gamma} = \mathbb{E}[L(Y)1(Y \in A)]$
$= \int_{y \in A} L(y)\pi'(y)dy$ since Y has density π'

$$= \int_{y \in A} \pi(y)dy = \mathbb{P}(X \in A) = \gamma$$

Hence $\hat{\gamma}$ is an unbiased estimate of γ.

To estimate γ:

$\frac{1}{n}\sum_{i=1}^{n} \hat{\gamma}_i \to \mathbb{E}\hat{\gamma} = \gamma$ by the law of large numbers

Suppose π, π' known, Y easy to simulate, then

1. Let n be a large integer
2. Simulate Y n times to get $Y_1 = y_1, \dots, Y_n = y_n$
3. Calculate $\hat{\gamma}_i = L(y_i)1[y_i \in A] = \frac{\pi(y_i)}{\pi'(y_i)}1[y_i \in A]$
4. Estimate $\gamma \approx \frac{1}{n}\sum_{i=1}^{n} \hat{\gamma}_i$

Now, let $X \sim Exp\left(\frac{1}{3}\right), B = \{X > 30\}$

$$\mathbb{P}(B) = \mathbb{P}(X > 30)$$
$$= e^{-\frac{1}{3}(30)} = e^{-10} \approx 4.54 \times 10^{-5}$$

Using the twisted distribution $Exp(\lambda)$, the likelihood function is

$$L(y) = \frac{\frac{1}{3}e^{-\frac{y}{3}}}{\lambda e^{-\lambda y}} = \frac{1}{3\lambda}e^{\left(\lambda - \frac{1}{3}\right)y}$$

The program to estimate γ is shown below in figure 1:

```
n = 10000;
lambda = 1/5;
%Generate n Unif[0,1] samples
u = rand(1,n);
%Transform into n Exp(lambda) samples
explambda = -(1/lambda)*log(u);

gammahat = zeros(1,n);
for i = 1:n
    if explambda(i) > 30
        gammahat(i) = (1/(3*lambda))*exp((lambda-1/3)*explambda(i));
    end
end
estimate = mean(gammahat);
```

Figure 1: estimating γ

Fixing n at 10000 and changing λ: (3 repeats for each λ)

$\lambda = \frac{1}{200}$	$\lambda = \frac{1}{50}$	$\lambda = \frac{1}{10}$	$\lambda = \frac{1}{5}$	$\lambda \geq \frac{1}{3}$
>> main estimate = 3.4786e-05 >> main estimate = 4.2843e-05 >> main estimate = 5.5905e-05	>> main estimate = 4.1977e-05 >> main estimate = 3.3464e-05 >> main estimate = 4.7394e-05	>> main estimate = 4.7214e-05 >> main estimate = 4.9495e-05 >> main estimate = 5.6331e-05	>> main estimate = 2.2545e-05 >> main estimate = 1.1666e-04 >> main estimate = 3.0083e-05	>> main estimate = 0 >> main estimate = 0 >> main estimate = 0

As seen from the table above, it seems that the program does not produce accurate estimates for larger values of λ for n = 10000. In particular, for $\lambda \geq \frac{1}{3}$, the program only produces 0 as an estimate.

To estimate the length of simulation needed to obtain a good estimate, we first repeat the estimation process for all n from 1 to N, $\lambda = \frac{1}{50}$, with the code below (figure 2)

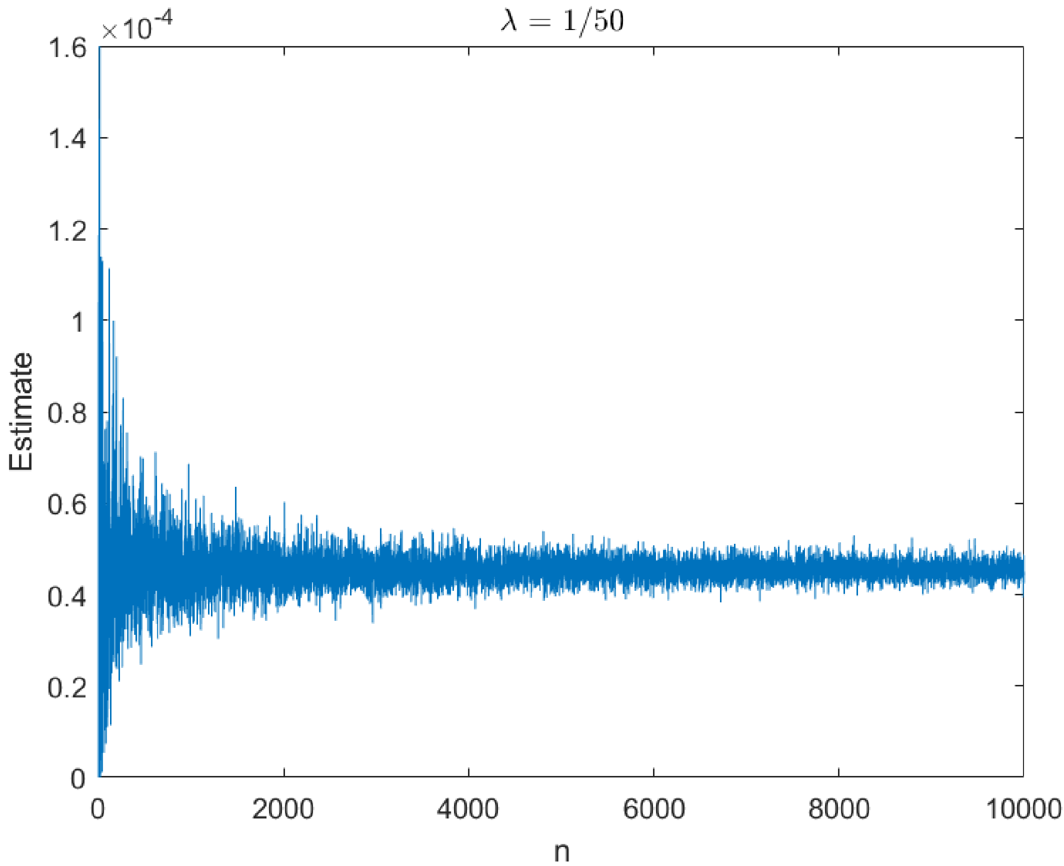
```
N = 10000;
lambda = 1/50;
result = zeros(1,N);

for n = 1:N
    u = rand(1,n);
    explambda = -(1/lambda)*log(u);

    gammahat = zeros(1,n);
    for i = 1:n
        if explambda(i) > 30
            gammahat(i) = (1/(3*lambda))*exp((lambda-1/3)*explambda(i));
        end
    end
    result(i) = mean(gammahat);
end
plot(result)
title(append('$\lambda$ = 1/', num2str(1/lambda)), interpreter = 'latex');
xlabel('n');
ylabel('Estimate')
```

Figure 2: estimation process for n from 1 to N, fixed λ

This gives the following result, plotting the estimate against the number of trials:



As seen from this plot, this procedure seems to produce a fairly accurate result past N = 5000. To proceed, first define a 'good' estimate as an estimate within 10% of the true value γ. Then, fix N and repeat the estimation process many times (rep = 10000 in this case) to estimate this quantity:

$$\mathbb{P}(estimate\ is\ within\ 10\%\ of\ true\ \gamma) = \mathbb{P}(estimate \in [4.09 \times 10^{-5}, 5 \times 10^{-5}])$$

This is achieved with the code below in figure 3

```
%Fix N
N = 5200;
rep = 10000;
resultmatrix = zeros(1,rep);
for j = 1:rep
    u = rand(1,N);
    explambda = -(1/lambda)*log(u);

    gammahat = zeros(1,N);
    for i = 1:N
        if explambda(i) > 30
            gammahat(i) = (1/(3*lambda))*exp((lambda-1/3)*explambda(i));
        end
    end
    resultmatrix(j) = mean(gammahat);
end
%Estimate the probability of the result being 'good' for fixed N
frac = sum(resultmatrix<=5e-5 & resultmatrix>=4.09e-5)/rep
```

Figure 3: estimating probability of a 'good' result

Using this code and trying different values of N, it could be seen that the program produces 'good' estimates for γ approximately 95% of the time at N = 6000 for $\lambda = \frac{1}{50}$.

Repeating this for different values of λ:

λ	N
1/200	15000
1/50	6000
1/40	5300
1/30	5000
1/20	6200
1/10	15000
1/5	200000

It seems that $\lambda = 1/30$ is the best, as it only requires N=5000 for the program to consistently (95%) produce 'good' estimates.

Given that it is difficult to compute estimates when the event is very rare (or very common), if Y is such that $\{Y \in B\}$ is very rare or common, then using importance sampling is just as ineffective as simulating X itself. When $\frac{1}{40} < \lambda < \frac{1}{30}$, the event $\{Y \in B\} = \{Y > 30\}$ occurs with probability ≈ 0.4 and is not too rare or common, which makes the simulation effective.

$$
\begin{aligned}
Var(\hat{\gamma}) &= \mathbb{E}[\hat{\gamma}^2] - \mathbb{E}[\hat{\gamma}]^2 \\
&= \int_B \left(\frac{\pi(y)}{\pi'(y)}\right)^2 \pi'(y)dy - \gamma^2 \\
&= \int_{30}^{\infty} \frac{1}{9\lambda} e^{\left(\lambda-\frac{2}{3}\right)y} dy - \gamma^2
\end{aligned}
$$

For $\lambda \geq \frac{2}{3}, Var(\hat{\gamma}) = \infty$

Hence $Var\left(\frac{1}{n}\sum_{i=1}^{n} \hat{\gamma}_i\right) = \frac{Var(\hat{\gamma})}{n} = \infty$ for any fixed, finite n

$\Rightarrow$ Simulation is useless for $\lambda \geq \frac{2}{3}$

$$Var(\hat{\gamma}) = \int_{30}^{\infty} \frac{1}{9\lambda} e^{\left(\lambda - \frac{2}{3}\right)y} dy - \gamma^2$$
$$= \frac{1}{9e^{20}} \left(\frac{1}{\lambda\left(\frac{2}{3} - \lambda\right)} e^{30\lambda} \right)$$

Set $\frac{d}{d\lambda} Var(\hat{\gamma}) = 0$

$$\Rightarrow -30\lambda^2 + 22\lambda - \frac{2}{3} = 0$$
$$\Rightarrow \lambda = \frac{-22 \pm \sqrt{404}}{-60} = 0.03167 \; or \; 0.7017 > \frac{2}{3} (rej.)$$

So $\lambda = 0.03167$ optimal ($\approx \frac{1}{32}$, consistent with table above)

We can apply importance sampling to Markov chains with jump probabilities P_{ij}, by considering the path it takes as a random variable, in the space of sample paths. Consider another Markov chain with twisted jump probabilities P'_{ij} such that $P'_{ij} > 0$ whenever $P_{ij} > 0$.

If we observe a path $x_0 \to x_1 \to \cdots \to x_n$, the likelihood ratio of the path is

$L = \frac{P_{x_0,x_1} \dots P_{x_{n-1},x_n}}{P'_{x_0,x_1} \dots P'_{x_{n-1},x_n}}$, since the likelihood in the original chain is $P_{x_0,x_1} \dots P_{x_{n-1},x_n}$

Now consider a simple random walk (X_n) on the non-negative integers with up probability $p < \frac{1}{2}$. We wish to compute the probability that the chain hits a value C before it returns to 0 on a single excursion. Let $C = 30, p = \frac{1}{4}$.

$\mathbb{P}(T_c < T_0 | X_0 = 0) = \mathbb{P}(T_c < T_0 | X_0 = 1) = q_1$ where

$q_i \coloneqq \mathbb{P}(MC \text{ reaching 30 before } 0 | X_0 = i), q_0 = 0, q_{30} = 1$

Then $q_i = \frac{1}{4} q_{i+1} + \frac{3}{4} q_{i-1} \; for \; i = 1, \dots, 29$

Auxiliary equation $\frac{1}{4}x^2 - x + \frac{3}{4} = 0$

$\Rightarrow x = 1 \; or \; 3$

$\Rightarrow q_i = A + B(3^i)$

Using $q_0 = 0, q_{30} = 1$:

$$\begin{cases} A + B = 0 \\ A + B(3^{30}) = 1 \end{cases}$$
$$\Rightarrow B = \frac{1}{3^{30} - 1}$$
$$\Rightarrow \mathbb{P}(T_c < T_0 | X_0 = 0) = q_1 = \frac{2}{3^{30} - 1}$$

Since this probability is too small to simulate directly, we attempt to use fast simulation (importance sampling for Markov chains) to compute this value, using another Markov chain with up probability $p' > \frac{1}{2}$.

For $p' > \frac{1}{2}, C$:

$q_i' := \mathbb{P}(\text{twisted MC reaching } C \text{ before } 0 | X_0 = i), q_0' = 0, q_c' = 1$

$$q_i' = p' q_{i+1}' + (1 - p') q_{i+1}'$$
$$\Rightarrow p' x^2 - x + (1 - p') = 0$$
$$\Rightarrow x = 1, \frac{1 - p'}{p'}$$
$$\Rightarrow q_i' = A + B\left(\frac{1 - p'}{p'}\right)^i = \frac{1 - \left(\frac{1 - p'}{p'}\right)^i}{1 - \left(\frac{1 - p'}{p'}\right)^C} \text{ using } q_0' = 0, q_c' = 1$$
$$\Rightarrow q_1' = \frac{2 - \frac{1}{p'}}{1 - \left(\frac{1 - p'}{p'}\right)^C}$$

As $C \to \infty$:

$$q_1 = \frac{\frac{1}{p} - 2}{\left(\frac{1-p}{p}\right)^C - 1} \to 0, q_1' \to 2 - \frac{1}{p'}$$

Although the event for the original random walk is rare and hard to simulate for large C, it occurs with probability $\approx 2 - \frac{1}{p'}$ for the twisted random walk. In particular, the fast simulation is expected to do well if $2 - \frac{1}{p'}$ is not close to 0 or 1, similar to before.

Let $\hat{\gamma} = \frac{P_{Y_0, Y_1} \dots P_{Y_{n-1}, Y_n}}{P'_{Y_0, Y_1} \dots P'_{Y_{n-1}, Y_n}} 1[T_{30}' < T_0' | Y_0 = 0]$ be an estimator for

$\gamma = \mathbb{P}(T_C < T_0 | X_0 = 0)$ where $Y =$ twisted random walk starting at $0, n = T'_{\{0,30\}}$

$$\begin{aligned}
\mathbb{E}[\hat{\gamma}] &= \int_y \frac{P_{0,y_1} \dots P_{y_{n-1},y_n}}{P'_{0,y_1} \dots P'_{y_{n-1},y_n}} 1[T_{30}' < T_0' | Y_0 = 0] P'_{0,y_1} \dots P'_{y_{n-1},y_n} dy_1 \dots dy_{n-1} \\
&= \int_{y: y_n = 30, n = T'_{\{0,30\}}} \frac{P_{0,y_1} \dots P_{y_{n-1},y_n}}{P'_{0,y_1} \dots P'_{y_{n-1},y_n}} P'_{0,y_1} \dots P'_{y_{n-1},y_n} dy_1 \dots dy_{n-1} \\
&= \int_{y: y_n = 30, n = T'_{\{0,30\}}} P_{0,y_1} \dots P_{y_{n-1},30}\, dy_1 \dots dy_{n-1} \\
&= \gamma
\end{aligned}$$

$\Rightarrow \hat{\gamma}$ unbiased

Using the same method as before, fix N and repeat the estimation process rep = 10000 times to estimate

$$\mathbb{P}(estimate\ is\ within\ 10\%\ of\ true\ \gamma) = \mathbb{P}(estimate \in [8.742 \times 10^{-15}, 1.069 \times 10^{-14}])$$

The code is shown below in figure 4:

```
p1 = 1/4;
p2 = 0.75;
C = 30;
N = 10000;
rep = 10000;
resultmatrix = zeros(1,rep);

for j = 1:rep
    total = 0;
    for i = 1:N
        x = 1;
        Lratio = 1;
        while x ~= 0 && x ~= C
            if rand < p2
                x = x+1;
                Lratio = Lratio*p1/p2;
            else
                x = x-1;
                Lratio = Lratio*(1-p1)/(1-p2);
            end
        end

        if x == C
            total = total + Lratio;
        end
    end
    resultmatrix(j) = total/N;
end

frac = sum(resultmatrix<=1.02e-14 & resultmatrix>=9.228e-15)/rep;
```

Figure 4: estimating probability of a 'good' result for fast simulation

Hence, this method produces 'good' estimates for γ approximately 95% of the time at these values of N for different p':

p'	N
0.501	>2000000
0.55	300000
0.6	30000
0.65	4000
0.7	750
0.75	180
0.8	1100
0.85	60000

When p' is close to 0.5 or 1, the probability $\approx 2 - \frac{1}{p'}$ that the event occurs for the twisted random walk is close to 0 or 1, so the method is not useful. This is consistent with the previous result, as this method is clearly a lot more effective when the event occurs with probability that is not too high or too low in the twisted system, as shown in the table above. When $p' \approx 0.75$, have $2 - \frac{1}{p'} \approx 0.65$ and the program performs very well, as it only requires 180 trials for it to produce a 'good' estimate over 95% of the time. If the number of trials is increased to 750 for $p' = 0.75$, the program could produce estimates with less than 5% error ($\in [9.228 \times 10^{-15}, 1.020 \times 10^{-14}]$) from the true value 95% of the time. This shows that fast simulation is very effective in this case, while not being computationally intensive at all.

$$\hat{\gamma} = \frac{P_{Y_0,Y_1} \dots P_{Y_{n-1},Y_n}}{P'_{Y_0,Y_1} \dots P'_{Y_{n-1},Y_n}} \, 1[T'_C < T'_0 | Y_0 = 0]$$
$$= \prod_{i=0}^{n-1} \frac{P_{Y_i,Y_{i+1}}}{P'_{Y_i,Y_{i+1}}} 1[T'_C < T'_0 | Y_0 = 0]$$
$$= \begin{cases} 0 \; if \; Y \; hits \; 0 \; before \; C \\ \prod_{i=0}^{n-1} \frac{P_{Y_i,Y_{i+1}}}{P'_{Y_i,Y_{i+1}}} \; if \; Y \; hits \; C \; before \; 0 \end{cases}$$

In the case $p' = 1 - p$:

If Y hits C before 0:
Each $\frac{P_{Y_i,Y_{i+1}}}{P'_{Y_i,Y_{i+1}}}$ is either $\frac{p}{p'} = \frac{p}{1-p}$ or $\frac{1-p}{1-p'} = \frac{1-p}{p}$

Y moves up $C - 1$ more times than it moves down (starting at 1 since $\frac{P_{0,1}}{P'_{0,1}} = 1$)

$$\Rightarrow \hat{\gamma} = \left(\frac{p}{1-p}\right)^{C-1}$$

$$\mathbb{P}(Y \text{ reaching } C \text{ before } 0 | X_0 = 1) = \frac{2 - \frac{1}{p'}}{1 - \left(\frac{1-p'}{p'}\right)^C}$$
$$= \frac{2 - \frac{1}{1-p}}{1 - \left(\frac{p}{1-p}\right)^C} = \frac{2(1-p)^C - (1-p)^{C-1}}{(1-p)^C - p^C}$$

$$\Rightarrow \hat{\gamma} \sim \left(\frac{p}{1-p}\right)^{C-1} Bernoulli\left(\frac{2(1-p)^C - (1-p)^{C-1}}{(1-p)^C - p^C}\right)$$

$\Rightarrow \hat{\gamma} \sim \left(\frac{p}{1-p}\right)^{C-1} \frac{1}{10000} Bin\left(10000, \frac{2(1-p)^C - (1-p)^{C-1}}{(1-p)^C - p^C}\right)$ after 10000 trials

$$\mathbb{E}[\hat{\gamma}] = \left(\frac{p}{1-p}\right)^{C-1} \frac{2(1-p)^C - (1-p)^{C-1}}{(1-p)^C - p^C}$$
$$= \frac{p^{C-1} - 2p^C}{(1-p)^C - p^C}$$
$$= \frac{\frac{1}{p} - 2}{\left(\frac{1-p}{p}\right)^C - 1}$$
$$= \gamma$$

Hence
$$\mathbb{E}[(\hat{\gamma} - \gamma)^2] = \mathbb{E}[(\hat{\gamma} - \mathbb{E}[\hat{\gamma}])^2]$$
$$= Var(\hat{\gamma})$$
$$= \left(\frac{p}{1-p}\right)^{2(C-1)} \left(\frac{1}{10000}\right)^2 \times$$
$$(10000) \frac{2(1-p)^C - (1-p)^{C-1}}{(1-p)^C - p^C} \left(1 - \frac{2(1-p)^C - (1-p)^{C-1}}{(1-p)^C - p^C}\right)$$
$$= \frac{1}{10000} \gamma \left(\left(\frac{p}{1-p}\right)^{C-1} - \gamma\right)$$

To verify this result, the code on page 7 is used, with $p' = 0.75 = 1 - \frac{1}{4}, C = 30, N = 10000$. The process is then repeated 10000 times, and we estimate $\mathbb{E}[(\hat{\gamma} - \gamma)^2] \approx \frac{1}{10000} \sum_{i=1}^{10000} (\hat{\gamma}_i - \gamma)^2$ where each $\hat{\gamma}_i$ is a 10000-trial estimator. The result should be close to the theoretical result

$$\mathbb{E}[(\hat{\gamma} - \gamma)^2] = \frac{1}{10000} \gamma \left(\left(\frac{p}{1-p} \right)^{C-1} - \gamma \right) \approx 4.718 \times 10^{-33} \ (\star)$$

```
p1 = 1/4;
p2 = 0.75;
C = 30;
N = 10000;
rep = 10000;
resultmatrix = zeros(1,rep);

for j = 1:rep
    total = 0;
    for i = 1:N
        x = 1;
        Lratio = 1;
        while x ~= 0 && x ~= C
            if rand < p2
                x = x+1;
                Lratio = Lratio*p1/p2;
            else
                x = x-1;
                Lratio = Lratio*(1-p1)/(1-p2);
            end
        end

        if x == C
            total = total + Lratio;
        end
    end
    resultmatrix(j) = total/N;
end

frac = sum(resultmatrix<=1.02e-14 & resultmatrix>=9.228e-15)/rep;

truevalue = 9.714e-15;
varestimate = mean((resultmatrix-truevalue).^2)
```

This gives the following estimate for $\mathbb{E}[(\hat{\gamma} - \gamma)^2]$, which is very close to the expected theoretical value $(\star)$ and verifies the calculations.

```
>> main

varestimate =

   4.7452e-33
```

Reference:

https://www.maths.cam.ac.uk/undergrad/catam/II/20pt2.pdf