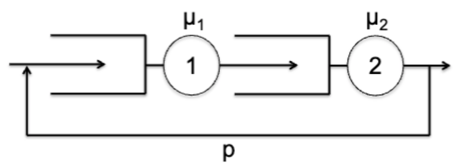
IEOR 4404 Simulation **Fall 2024**

Homework 9

Instructor: Henry Lam

Problem 1 Consider a queueing system with two servers in series, so that customers have to be first served at server 1 and then server 2. Both servers serve customers according to the first-come-first-serve policy. Suppose now that for each customer that finishes service at server 2, independently with probability $p > 0$, he/she is unhappy with the service, and joins the end of the queue at server 1 (and leaves the system with probability $q = 1 - p > 0$). Note that a customer can be served by servers 1 and 2 multiple times, until he/she is happy with the services, at which time he/she leaves the system.



Suppose that customers arrive according to a Poisson process with rate λ, their service requirements at servers 1 and 2 are independent exponential random variables with parameters μ_1 and μ_2, respectively, and independent from everything else.

(a) Let $X_1(t)$ be the number of customers in queue and (possibly) in service at server 1, at time t, and $X_2(t)$ the number of customers in queue and (possibly) in service at server 2, at time t. Show that the process $(X_1(t), X_2(t))_{t\geq 0}$ is a continuous-time discrete-state Markov chain. Specify the holding time distributions for different states, and the transition probabilities.

(b) Describe a procedure to simulate the Markov chain $(X_1(t), X_2(t))_{t\geq 0}$ up to time T. Implement it in a computer by running 100 replications, i.e., trajectories of the Markov chain, and plot the distributions of $X_1(T)$ and $X_2(T)$. Use $\lambda = 1$, $\mu_1 = 2$, $\mu_2 = 3$, $p = 0.2$, and $T = 10$.

Problem 2 Consider a $G/G/1$ queue. More specifically, suppose the interarrival times are i.i.d. $Exp(1/2)$ random variables, service times are i.i.d. $Gamma(3, 2)$ random variables with density function given by

$$f(x) = 4x^2 e^{-2x}, \quad x \geq 0,$$

and the interarrival and service times are all independent. The server serves customers in a first-come-first-serve order. Write the pseudo-code for simulating the long-run average waiting time of the customers. Implement the code in the computer by simulating 1000 customers, assuming the system is empty initially, and using a burn-in of 100 customers. You may use the fact that $Gamma(3, 2)$ can be represented as the sum of 3 i.i.d $Exp(2)$ random variables.

Problem 3 Consider simulating $P(X > 10)$ where $X \sim Gamma(2, 1)$. You are given that $Gamma(\alpha, \beta)$ has a density

$$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad x > 0$$

where Γ is the gamma function. The mean of $Gamma(\alpha, \beta)$ is α/β and the moment generating function $Ee^{\theta X}$ is $(1 - \theta/\beta)^{-\alpha}$ defined for $\theta < \beta$.

(a) Derive the output of one replication obtained from importance sampling by tuning the α parameter so that the mean of the gamma distribution matches exactly 10.

(b) Derive the output of one replication obtained from importance sampling by an exponential tilting of $Gamma(2, 1)$ so that its mean matches exactly 10.

(c) Consider simply using a shifted gamma distribution to speed up the simulation, i.e., use the random variable $X+10$, where $X \sim Gamma(2, 1)$, as the importance sampling variable. Is this a legitimate scheme? Briefly explain your answer.

Problem 4 Consider the problem of estimating

$$\alpha = \mathbb{P}\left(\min\{X_1 + X_2, X_3 + X_4, X_1 + X_4, X_2 + X_3\} \geq 4\right),$$

where X_1, X_2, X_3 and X_4 are i.i.d standard normal random variables.

(a) Consider the joint pdf of (X_1, X_2, X_3, X_4) given by

$$f(x_1, x_2, x_3, x_4) = \frac{1}{(2\pi)^2} \exp\left(-\frac{x_1^2 + x_2^2 + x_3^2 + x_4^2}{2}\right), \quad -\infty < x_1, x_2, x_3, x_4 < \infty.$$

Consider the region

$$A = \{(x_1, x_2, x_3, x_4) : \min\{x_1 + x_2, x_3 + x_4, x_1 + x_4, x_2 + x_3\} \geq 4\}.$$

Compute the maximizer $(x_1^*, x_2^*, x_3^*, x_4^*)$ of the pdf $f(x_1, x_2, x_3, x_4)$ over the region A.

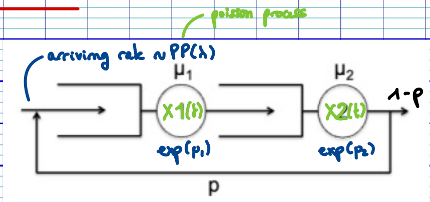
(b) By considering simulating X_1, X_2, X_3, X_4 that are independent normal random variables with means x_1^*, x_2^*, x_3^* and x_4^* respectively, and variances all being equal to 1, develop an importance sampling scheme to estimate α. Implement it using 100 replications to obtain a point estimate for α.

(c) Explain briefly the reason for computing $(x_1^*, x_2^*, x_3^*, x_4^*)$, the maximizer of pdf f over A, and for using X_1, X_2, X_3, X_4 with the changed distributions in the importance sampling method.

Alexandre
Duhamel
afd2153
MSOR

HW 9: Simulation

Problem 1:

a) 

The state space S of the Markov chain is the set of all possible pairs (i,j) where:

$i \geq 0$: # of customers at server 1
$j \geq 0$: # of customers at server 2

} queue or served

$(i,j) \longrightarrow (i+1,j)$: new customer arrives at rate λ and joins queue 1.
$(i,j) \longrightarrow (i-1,j+1)$: customer completes service 1 and moves to two at rate μ_1.
$(i,j) \longrightarrow (i,j-1)$: customer completes service 2 and leaves for good at rate μ_2 with probability q.
$(i,j) \longrightarrow (i+1,j-1)$: customer completes service 2 and joins back the first queue at rate μ_2 wp p.

For any state (i,j), the holding time until the next event is exponentially distributed where:

$$r(i,j) = \lambda + \mu_1 \mathbb{1}\{i>0\} + \mu_2 \mathbb{1}\{j>0\}$$

Hence:
$$\begin{cases} P((i,j) \longrightarrow (i+1,j)) = \dfrac{\lambda}{r(i,j)} \\ P((i,j) \longrightarrow (i-1,j+1)) = \dfrac{\mu_1 \mathbb{1}\{i>0\}}{r(i,j)} \\ P((i,j) \longrightarrow (i,j-1)) = \dfrac{\mu_2 q \mathbb{1}\{j>0\}}{r(i,j)} \\ P((i,j) \longrightarrow (i+1,j-1)) = \dfrac{\mu_2 p \mathbb{1}\{j>0\}}{r(i,j)} \end{cases}$$

Since the inter arrival times are exponentially distributed and the transition probabilities only depend on the current state, $(X_1(t), X_2(t))$ is a continuous-time discrete-state Markov chain.

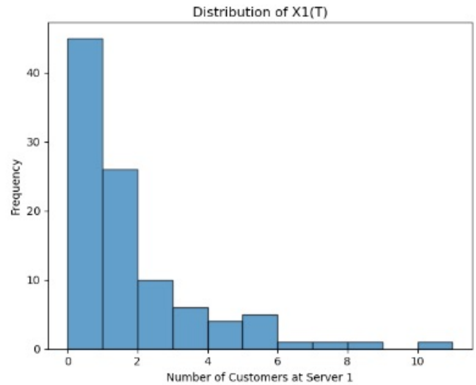
c) Initialization : $t = 0$ and $X_1 = X_2 = 0$

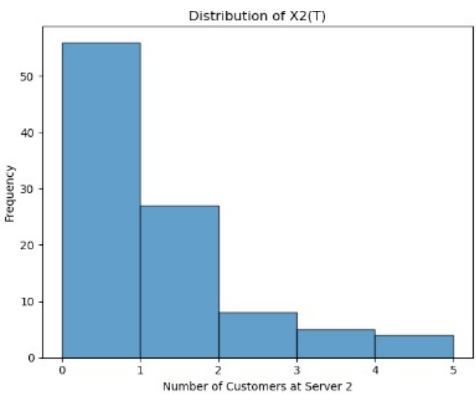
While $t < T$:

- Compute $r(i,j)$
- Update $t = t + \Delta t$ where $\Delta t \sim \exp(r(i,j))$
- Generate $U \sim \text{Uniform}(0,1)$
- Define :
  $$P_{arrival} = \frac{\lambda}{r(i,j)}$$
  $$P_{service1} = \frac{p_1 \mathbb{1}\{X_1>0\}}{r(i,j)}$$
  $$P_{service2} = \frac{p_2 \mathbb{1}\{X_2>0\}}{r(i,j)}$$
- If $U \leq P_{arrival}$: $X_1 = X_1 + 1$
- Elif $U \leq P_{arrival} + P_{service1}$: $X_1 = X_1 - 1$ and $X_2 = X_2 + 1$
- Else :
  - Generate $V \sim \text{Uniform}(0,1)$
  - If $V \leq q$: $X_2 = X_2 - 1$
  - Else : $X_2 = X_2 - 1$ and $X_1 = X_1 + 1$

Return $X_1(T)$ and $X_2(T)$

```python
X1_T_values = []
X2_T_values = []

for _ in range(replications):
    t = 0
    X1 = 0
    X2 = 0

    while t < T:
        r = lambda_arrival
        if X1 > 0:
            r += mu1
        if X2 > 0:
            r += mu2

        delta_t = random.expovariate(r)
        t += delta_t

        if t >= T:
            break

        P_arrival = lambda_arrival / r
        P_service1 = mu1 / r if X1 > 0 else 0
        P_service2 = mu2 / r if X2 > 0 else 0

        u = random.uniform(0, 1)

        if u <= P_arrival:
            X1 += 1
        elif u <= P_arrival + P_service1:
            X1 -= 1
            X2 += 1
        else:
            v = random.uniform(0, 1)
            if v <= q_leave:
                X2 -= 1
            else:
                X2 -= 1
                X1 += 1

    X1_T_values.append(X1)
    X2_T_values.append(X2)

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.hist(X1_T_values, bins=range(max(X1_T_values)+1), edgecolor='black', alpha=0.7)
plt.title('Distribution of X1(T)')
plt.xlabel('Number of Customers at Server 1')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
plt.hist(X2_T_values, bins=range(max(X2_T_values)+1), edgecolor='black', alpha=0.7)
plt.title('Distribution of X2(T)')
plt.xlabel('Number of Customers at Server 2')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```





Problem 2:

Set $N = 1100$, $burn\text{-}in = 100$, $total\ wait = 0$, $D\_prev = 0$

For n in $1 \dots N$:
- Generate $T_n \sim Exp(1/2)$ interarrival time
- Compute $A_n = A_{n-1} + T_n$ where $A_0 = 0$ arrival time
- Set $S_n = 0$
- For k in $1 \dots 3$: S_n += $Exp(2)$ service time with Gamma dist
- Compute $B_n = \max(A_n, D\_prev)$ service start time
- Compute $W_n = B_n - A_n$ waiting time
- Compute $D_n = B_n + S_n$ departure time
- Update $D_{prev} = D_n$
- If $n > burn\text{-}in$: $total\ wait$ += W_n

Return $total\ wait / (N - burn\text{-}in)$

```python
N = 1100
burn_in = 100
lambda_arrival = 0.5
beta_service = 2
alpha_service = 3
num_customers = N - burn_in

arrival_times = []
service_times = []
begin_service_times = []
waiting_times = []
departure_times = []

D_prev = 0

for n in range(N):
    T_n = random.expovariate(lambda_arrival)

    if n == 0:
        A_n = T_n
    else:
        A_n = arrival_times[-1] + T_n
    arrival_times.append(A_n)

    S_n = sum(random.expovariate(2) for _ in range(3))
    service_times.append(S_n)

    B_n = max(A_n, D_prev)
    begin_service_times.append(B_n)

    W_n = B_n - A_n
    waiting_times.append(W_n)

    D_n = B_n + S_n
    departure_times.append(D_n)

    D_prev = D_n

total_waiting_time = sum(waiting_times[burn_in:])
average_waiting_time = total_waiting_time / num_customers

print(f"Average waiting time over last {num_customers} customers: {average_waiting_time:.4f}")
```

```
Average waiting time over last 1000 customers: 2.2690
```

Problem 3:

a) We change α so that : $E[X] = 10$

We keep $\beta = 1$, originally: $E[X] = \alpha/\beta = 2$

Now we want $E[X] = 10 = \alpha'/\beta = \alpha'/1$ so $\alpha' = 10$

Set new dist to $Y \sim \text{Gamma}(10, 1)$

With density: - old: $f_X(x) = \dfrac{1^2}{\Gamma(2)} y^{2-1} e^{-1 \cdot y} = y e^{-y}/\Gamma(2)$

- new: $f_Y(y) = \dfrac{1^{10}}{\Gamma(10)} y^{10-1} e^{-1 \cdot y} = y^9 e^{-y}/\Gamma(10)$

likelihood: $L(Y) = \dfrac{f_X(Y)}{f_Y(Y)} = \dfrac{\Gamma(10) Y^1 e^{-Y}}{\Gamma(2) Y^9 e^{-Y}} = \dfrac{\Gamma(10)}{\Gamma(2)} Y^{-8}$

So output of one replication: $\begin{cases} \text{Generate } Y \sim \text{Gamma}(10,1) \\ \text{If } Y > 10: \text{return } L(Y) = \dfrac{\Gamma(10)}{\Gamma(2)} Y^{-8} \\ \text{Else : return } 0 \end{cases}$ since we must estimate $\mathbb{P}(X>10) = E[\underbrace{1\{X>10\}}_{g(X)}]$

$\sum_{i=1}^{m} g(Y_i) L(Y_i)$ for $m = 1$

b) We must find θ so that $E[X] = 10$ where $E[X] = \dfrac{\alpha}{\beta - \theta} \rightarrow \theta = 0.8$ and so $\beta' = \beta - \theta = 0.2$

Tilting is defined as: $f_Y(x) \propto f_X(x) e^{\theta x}$

So $f_Y(Y) = \dfrac{f_X(Y) e^{\theta Y}}{M_Y(\theta)}$ where $M_Y(\theta) = (1-\theta)^{-2}$

normalizing so that $\int_{\mathbb{R}} f_Y(y)\,dy = 1$

likelihood: $L(Y) = \dfrac{f_X(Y)}{f_Y(Y)} = 25 e^{-0.8Y}$

So output of one replication: $\begin{cases} \text{Generate } Y \sim \text{Gamma}(10,1) \\ \text{If } Y > 10: \text{ return } L(Y) = 25 e^{-0.8Y} \\ \text{Else : return } 0 \end{cases}$

c) The approach would involve using $Y = X + 10$ with $X \sim \text{Gamma}(2, 1)$

This is not legitimate importance sampling scheme because the likelihood cannot be computed because $f_Y(y) = f_X(y - 10)$ isn't defined for $y < 10$.

Also, $g(x) f_X(x)$ must have the support of $f_Y(x)$, which is not the case due to the shifting of 10.

$f_Y(x) = 0 \not\Rightarrow g(x) f_X(x) = 0$

Problem 4:

a) Maximizing f over A $\longleftrightarrow$ minimizing $x_1^2 + x_2^2 + x_3^2 + x_4^2$ over A

Let's assume $x_i = c$ for $i = 1, \dots, 4$ by symmetry

So $\min\{2c, 2c, 2c, 2c\} \geq 4 \iff c \geq 2$

So we take $c = 2$ and get $x_1^* = x_2^* = x_3^* = x_4^* = 2$

b) The new sampling distribution for each X_i is $\mathcal{N}(\underbrace{2}_{\text{optimal } x_i^*}, \underbrace{1}_{\text{as asked}})$: $f_Y(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-2)^2}{2}}$

Initially, we had : $\mathcal{N}(0,1)$: $f_X(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$

We compute the likelihood : $L(x) = \prod_{i=1}^{4} \frac{e^{-\frac{x_i^2}{2}}}{e^{-\frac{(x_i-2)^2}{2}}} = e^{-2\sum_{i=1}^{4} x_i + 8}$

For k in $1 \dots 10$: {
- Generate $x_i^{(k)} \sim \mathcal{N}(2,1)$, $\forall i$
- Compute $L^{(k)} = e^{-2\sum_{i=1}^{4} x_i^{(k)} + 8}$
- Compute : $\begin{cases} s_1 = x_1^{(k)} + x_2^{(k)} \\ s_2 = x_3^{(k)} + x_4^{(k)} \\ s_3 = x_1^{(k)} + x_4^{(k)} \\ s_4 = x_2^{(k)} + x_3^{(k)} \end{cases}$
- Compute $\min = \min\{s_1, s_2, s_3, s_4\}$
- If $\min \geq 4$: Accept $L^{(k)}$ same reason as problem 3
- Else : Reject

Compute $\alpha = \frac{1}{N} \sum_{i=1}^{N} L^{(k)}$

```python
N = 100
estimates = []

for _ in range(N):
    x = np.random.normal(loc=2, scale=1, size=4)

    sum_x = np.sum(x)
    L = np.exp(-2 * sum_x + 8)

    s1 = x[0] + x[1]
    s2 = x[2] + x[3]
    s3 = x[0] + x[3]
    s4 = x[1] + x[2]

    min_s = min(s1, s2, s3, s4)

    if min_s >= 4:
        estimate = L
    else:
        estimate = 0

    estimates.append(estimate)

alpha_estimate = np.mean(estimates)
print(f"Estimate of α: {alpha_estimate:.6f}")
```

```
Estimate of α: 0.000002
```

c) The maximizer $(x_1^*, x_2^*, x_3^*, x_4^*)$ identifies the point in region A where the joint PDF $f(x)$ is largest, indicating where the probability mass for α is concentrated. Shifting the means of X_1, X_2, X_3, X_4 to this maximizer focuses sampling on the region of interest, reducing variance.

The likelihood ratio ensures unbiased estimation despite the altered sampling distribution.