HW 4 M362M, Fall 2024

Wei Xuan Lee (wl22963)

2024-10-07

Problem 1

(1) It is unclear what "hits the level $l = 1/2$" means, it can mean $Y_n \geq 1/2$ or $Y_n = 1/2$. Fortunately, either way, T **is a stopping time.** Here we assume it is $Y_n \geq 1/2$. Let G_n be the decision function.

$$G^n(y_1, \cdots, y_n) = \begin{cases} 1, & y_n \geq 1/2 \text{ and } y_m < 1/2 \text{ for all } m < n \\ 0, & \text{else.} \end{cases}$$

Note that $y_n = \frac{1}{n} \sum_{k=1}^{n} x_k$, and x_k is the value of X_k on a trajectory. If the definition of "hits the level $l = 1/2$" is $Y_n = 1/2$, then the decision function is

$$G^n(y_1, \cdots, y_n) = \begin{cases} 1, & y_n = 1/2 \text{ and } y_m \neq 1/2 \text{ for all } m < n \\ 0, & \text{else.} \end{cases}$$

(2) T **is a stopping time.**

$$G^n(x_1, \cdots, x_n) = \begin{cases} 1, & \exists\, i, j : 0 \leq i < j < n \text{ s.t. } x_i = x_j = x_n = l \\ & \quad \text{and } x_k \neq l \text{ for all } k \notin \{i, j, n\} \\ 0, & \text{else.} \end{cases}$$

(3) T **is a stopping time.**

$$G^n(x_1, \cdots, x_n) = \begin{cases} 1, & n \geq 2 \text{ and } x_n - x_{n-2} \geq 2x_{n-1} \\ & \quad \text{and } x_k - x_{k-2} < 2x_{k-1} \text{ for all } k < n \\ 0, & \text{else.} \end{cases}$$

(4) T **is NOT a stopping time.** Suppose for the sake of contradiction that T is a stopping time and we have a decision function $G^n = \mathbf{1}_{\{T=n\}}$. Now, for trajectory $(x_0, x_1, x_2, x_3) = (0, 1, 2, 3)$, note that $\mathbf{1}_{\{T=2\}} = 0$ because $T = x_3 + 3 = 3 + 3 = 6 \neq 2$. On the other hand, for $(x'_0, x'_1, x'_2, x'_3) = (0, 1, 0, -1)$, we have $\mathbf{1}_{\{T=2\}} = 1$ because $T = x_3 + 3 = -1 + 3 = 2$. However, $(x_0, x_1) = (x'_0, x'_1)$, and we have $G^n(x_0, x_1) = 0, G^n(x'_0, x'_1) = 1$, a contradiction. Therefore T is not a stopping time.

(5) T **is a stopping time.**

$$G^n(x_1, \cdots, x_n) = \begin{cases} 1, & n \geq 3 \text{ and } n = x_3 + 6 \\ 0, & \text{else.} \end{cases}$$

1.1

```r
n = 10000
T = 100

sym_path = function(horizon){
  steps = sample(c(-1,1), horizon, replace=TRUE, prob=c(.5, .5))
  return (c(0,cumsum(steps)))
}

# (1) [Here we assume Y_n >= 1/2]
stop_position_time1 = function(horizon){
  path = sym_path(horizon)
  sum = 0
  for (k in 1:horizon){
    sum = sum + path[k+1]
    if (sum/k >= .5){
      return (c(path[k+1],k))
    }
  }
  return (c(path[horizon+1],horizon))
}

sim1= t(replicate(n, stop_position_time1(T)))
# Expected Stopping Position (Simulation)
exp_pos1 = mean(sim1[,1])
# Expected Stopping Time (Simulation)
exp_time1 = mean(sim1[,2])

# Wald's Theorem does not apply because expected stopping time is infinite
cat("(1)", "\n","Expected Stopping Position (Simulation) :", round(exp_pos1, 4), "\n",
      "Expected Stopping Time (Simulation)  :", round(exp_time1, 4),"\n", "\n")
## (1)
##  Expected Stopping Position (Simulation) : 0.0069
##  Expected Stopping Time (Simulation)  : 29.2403
##
# (2)
stop_position_time2 = function(horizon, l){
  path = sym_path(horizon)
  count = 0
  for (k in 1:horizon){
    if (path[k+1] == l){
      count = count + 1
    }
    if (count == 3){
      return (c(path[k+1], k))
    }
  }
  return (c(path[horizon+1], horizon))
}


sim2= t(replicate(n, stop_position_time2(T,1)))
```

```r
# Expected Stopping Position (Simulation)
exp_pos2 = mean(sim2[,1])
# Expected Stopping Time (Simulation)
exp_time2 = mean(sim2[,2])

# Wald's Theorem does not apply because expected stopping time is infinite
cat("(2)", "\n","Expected Stopping Position (Simulation) :", round(exp_pos2, 4), "\n",
      "Expected Stopping Time (Simulation)  :", round(exp_time2, 4), "\n", "\n")
## (2)
##  Expected Stopping Position (Simulation) : 0.0183
##  Expected Stopping Time (Simulation)  : 40.8761
##
# (3)
stop_position_time3 = function(horizon){
  path = sym_path(horizon)
  for (k in 2:horizon){
    if (path[k+1] - path[k-1] >= 2*path[k]){
      return (c(path[k+1], k))
    }
  }
  return (c(path[horizon + 1], horizon))
}

sim3= t(replicate(n, stop_position_time3(T)))
# Expected Stopping Position (Simulation)
exp_pos3 = mean(sim3[,1])
# Expected Stopping Time (Simulation)
exp_time3 = mean(sim3[,2])

# Wald's Theorem works as we can see the expected stopping position is close to 0
cat("(3)","\n", "Expected Stopping Position (Simulation) :", round(exp_pos3, 4), "\n",
      "Expected Stopping Time (Simulation)  :", round(exp_time3, 4), "\n", "\n")
## (3)
##  Expected Stopping Position (Simulation) : -0.0118
##  Expected Stopping Time (Simulation)  : 2.366
##
# (4) Not a stopping time

# (5)
stop_position_time5 = function(horizon){
  path = sym_path(horizon)
  return (c(path[path[4]+6+1],path[4]+6))
}

sim5= t(replicate(n, stop_position_time5(T)))
# Expected Stopping Position (Simulation)
exp_pos5 = mean(sim5[,1])
# Expected Stopping Time (Simulation)
exp_time5 = mean(sim5[,2])

# Wald's Theorem works as we can see the expected stopping position is close to 0
cat("(5)", "\n","Expected Stopping Position (Simulation) :", round(exp_pos5, 4), "\n",
      "Expected Stopping Time (Simulation)  :", round(exp_time5, 4),"\n", "\n")
```

```
## (5)
##  Expected Stopping Position (Simulation) : -0.0182
##  Expected Stopping Time (Simulation)  : 5.9864
##
```

Problem 2

(1) $p_a(0) = 0, p_a(a) = 1$, because if we are already at 0 at X_0, then surely we cannot hit a before hitting 0, therefore $p_a(0) = 0$. Similarly, we know that $p_a(a) = 1$.

(2)

$$
\begin{aligned}
p_a(x) &= P(W_n \text{ hits } a \text{ first} \mid X_1 = -1)P(X_1 = -1) \\
&\quad + P(W_n \text{ hits } a \text{ first} \mid X_1 = 1)P(X_1 = 1) \\
&= qP(W_n = (x-1) + X_n, n \geq 2 \text{ hits } a \text{ first}) \\
&\quad + pP(W_n = (x+1) + X_n, n \geq 2 \text{ hits } a \text{ first}) \\
&= qP(W_n' \text{ hits } a \text{ first}) + pP(W_n'' \text{ hits } a \text{ first}) \\
&(\text{Note: } W_n' = (x-1) + X_n, W_n'' = (x+1) + X_n) \\
&= qp_a(x-1) + pp_a(x+1)
\end{aligned}
$$

(3) Suppose $p_a(x) = A^x$ solves the equation $p_a(x) = qp_a(x-1) + pp_a(x+1)$, for $x = 1, \cdots, a-1$. Note that $p + q = 1$. When $x = 1$, we get $A = q + pA^2$. Solving this equation gives us $A = \frac{q}{p}$ or $A = 1$. We now verify that it solves the equation too when $x = 2, \cdots, a-1$. This is because $A \neq 0$, then by multiplying A^{x-1} on both sides of $A = q + pA^2$, we get$A^x = qA^{x-1} + pA^{x+1}$. Therefore we let $A_1 = \frac{q}{p}$ and $A_2 = 1$.

(4) It goes down to solving the system

$$
\begin{cases}
1 = & C_1(q/p)^a + C_2(1)^a \\
0 = & C_1(q/p)^0 + C_2(1)^0
\end{cases}
$$

which comes from $p_a(a) = 1$ and $p_a(0) = 0$. We get

$$
C_1 = \frac{-1}{1 - \left(\frac{q}{p}\right)^a}\ , \quad C_2 = \frac{1}{1 - \left(\frac{q}{p}\right)^a}.
$$

Therefore

$$
p_a(x) = \frac{1 - \left(\frac{q}{p}\right)^x}{1 - \left(\frac{q}{p}\right)^a}, \text{ for } x = 0, \cdots, a.
$$

2.1

```
n = 1000
T = 100
x=5
a=10
p=2/3
```

```r
biased_path = function(x,p,horizon){
  steps = sample(c(-1,1), horizon, replace=TRUE, prob=c(1-p, p))
  return (cumsum(c(x,steps)))
}

win = function(a, x, p, horizon) {
  path = biased_path(x, p, horizon)
  for (i in 1:horizon+1){
    if (path[i] == a){
      return (1)
    }
    if (path[i] == 0){
      return (0)
    }
  }
  # If we don't know who wins, we assume the player lost.
  return (0)
}

# Calculate probability (Simulation)
sim_prob = mean(replicate(n, win(a,x,p,T)))

# Calculate probability (Theoretical)
the_prob = 1/(1 + .5^5)

cat(" We calculate the probability of winning", "\n", "\n",
      "Probability (Theoretical) :", round(the_prob, 4), "\n",
      "Probability (Simulation)  :", round(sim_prob, 4))
##  We calculate the probability of winning
##
##  Probability (Theoretical) : 0.9697
##  Probability (Simulation)  : 0.974
```

Problem 3

(1) $f(0) = f(a) = 0$, because we are already at 0 or a at X_0.

(2)

$$\begin{aligned} f(x) &= \mathbb{E}[T_{0,a}(x) \mid X_1 = -1]P(X_1 = 1) \\ &\quad + \mathbb{E}[T_{0,a}(x) \mid X_1 = 1]P(X_1 = -1) \\ &= \frac{1}{2}\mathbb{E}[T_{0,a}(x+1)+1] + \frac{1}{2}\mathbb{E}[T_{0,a}(x-1)+1] \\ &\text{(Note: } +1 \text{ because unit time has passed for us to see } X_1, \\ &\text{and now we are at } x+1 \text{ or at } x-1) \\ &= \frac{1}{2}f(x+1) + \frac{1}{2}f(x-1) + 1 && \text{(Linearity of } \mathbb{E}[\cdot]) \end{aligned}$$

Therefore $\square_1 = \square_2 = \frac{1}{2}, \square_3 = 1$.

(3) Plug $f(x) = \alpha x^2 + \beta x + \gamma$ into the equation above,

$$\alpha\left[\frac{(x+1)^2+(x-1)^2}{2} - x^2\right] + \beta\left[\frac{(x+1)+(x-1)}{2} - x\right] + \gamma\left[\frac{1+1}{2} - 1\right] = -1.$$

Therefore $\alpha = -1$, whereas there are no restrictions on β, γ.

(4) Now we shall try to solve β, γ. From $f(0) = 0$, we have $\gamma = 0$, and now from $f(a) = 0$, we have $\beta = a$, therefore

$$f(x) = -x^2 + ax, \text{ for } x = 0, \cdots, a.$$

3.1

```
n = 1000
T = 100
x=5
a=10

sym_path = function(x,horizon){
  steps = sample(c(-1,1), horizon, replace=TRUE, prob=c(.5, .5))
  return (cumsum(c(x,steps)))
}

time = function(a, x, horizon) {
  path = sym_path(x, horizon)
  for (i in 1:horizon+1){
    if (path[i] == a | path[i] == 0){
      return (i-1)
    }
  }
  return (horizon)
}

# Calculate expectation (Simulation)
sim_exp = mean(replicate(n, time(a,x,T)))

# Calculate expectation (Theoretical)
the_exp = -x^2 + a*x

cat(" We calculate the probability of winning", "\n", "\n",
      "Expectation (Theoretical) :", round(the_exp, 4), "\n",
      "Expectation (Simulation)  :", round(sim_exp, 4))
##  We calculate the probability of winning
##
##  Expectation (Theoretical) : 25
##  Expectation (Simulation)  : 25.585
```

End of Homework