STAT529
Applied Bayesian Decision Theory
Homework 4

Fangda Li
li1208@purdue.edu

April 28, 2017

1 Interim Analysis

Letting $\omega = \theta_T - \theta_S$, we aim to find the predictive probability of $P(\omega \geq 30) \geq 0.9$ at stage q in a total of n stages ($r = n - q$ denotes the rest of stages). According to the notes, the condition holds true when the following is satisfied:

$$\Delta_n = \frac{q}{n}\Delta_q + \frac{r}{n}\Delta_r \geq 30 + 1.3\sigma\sqrt{\frac{2}{n}},$$
$$\Delta_r \geq \frac{n}{r}(30 + 1.3\sigma\sqrt{\frac{2}{n}} - \frac{q}{n}\Delta_q).$$

Taking $q = r = 100$, $n = 200$ and $\sigma = 20$, the above condition is equivalent to:

$$\Delta_r \geq 65.2 - \Delta_q.$$

Also recall that the posterior distribution of ω can be shown to be:

$$\pi(\omega|\Delta_q, \beta, \tau^2) \sim N(m, v^2),$$

where $m = \frac{\frac{2\sigma^2}{q}(0) + 2\tau^2\Delta_q}{\frac{2\sigma^2}{q} + 2\tau^2}$ and $v^2 = \frac{\frac{2\sigma^2}{q}2\tau^2}{\frac{2\sigma^2}{q} + 2\tau^2}$. Then the posterior distribution of Δ_r with β and ω being integrated out becomes:

$$\begin{aligned}
f(\Delta_r|\Delta_q) &= \int\int\int_{\tau^2,\beta,\omega} \frac{f(\Delta_r, \Delta_q|\omega, \beta, \tau^2)\pi(\omega|\beta, \tau^2)h_1(\beta)h_2(\tau^2)}{f(\Delta_q)} \\
&= \int\int\int_{\tau^2,\beta,\omega} f(\Delta_r, \omega)\pi(\omega|\Delta_q, \beta, \tau^2)f(\Delta_q|\beta, \tau^2)\frac{h_1(\beta)h_2(\tau^2)}{f(\Delta_q)} \\
&= \int_{\tau^2} N(m, \frac{2\sigma^2}{r} + v^2)\frac{f(\Delta_q|\tau^2)h_2(\tau^2)}{f(\Delta_q)}.
\end{aligned}$$

With the posterior distribution, we can compute the posterior probability of interest:

$$P(\Delta_r \geq 65.2 - \Delta_q|\Delta_q) = \int_{\tau^2}(1 - \Phi(\frac{(65.2 - \Delta_q) - \Delta_q\frac{\tau^2}{4+\tau^2}}{\sqrt{8 + \frac{8\tau^2}{4+\tau^2}}}))\frac{f(\Delta_q|\tau^2)h_2(\tau^2)}{f(\Delta_q)}.$$

In practice, MCMC is performed in order to generate posterior samples of τ^2 and the predictive probability is computed by averaging the $(1 - \Phi(*))$ values. The probability r used in MCMC for keeping τ_t^2 is:

$$r = min\{\frac{f(\Delta_q|\tau_t^2)}{f(\Delta_q|\tau_{t-1}^2)}, 1\}.$$

According to my simulation, the computed predictive probabilities $P(\Delta_r \geq 65.2 - \Delta_q)$ for different Δ can be found in Table 1:

Table 1: Predictive probabilities $P(\Delta_r \geq 65.2 - \Delta_q)$

Δ_q	30	32	34	36	38	40
P	0.008	0.078	0.325	0.699	0.932	0.993

2 Sequential Analysis

The problem is setup as follows: observations $X_1, X_2, ..., X_n \sim uniform(0, 50)$, "payoff" function $Y_n = X_n - nc$, and the number of possible trials $N = 15$.

a) **Backward Induction**: We start by looking at the second last stage, $n = 14$. First, the expected payoff at stage $n = 15$ is:

$$E[Y_{15}] = E[X_{15} - 15c] = 25 - 15c.$$

We will stop at stage 14 if the payoff at stage 14, y_{14}, is greater than $E[Y_{15}]$, which is $x_{14} \geq 25 - c$. Now we can define the optimal payoff at stage 14 as the larger value between the observed payoff y_{14} and the expected payoff at the next stage $E[Y_{15}]$: $\beta_{14}^{15} = max(y_{14}, E[Y_{15}])$.

The optimal rule says we should stop at stage $n = 13$ if the observed payoff at stage 13, y_{13}, is greater than the expected optimal payoff at stage 14, which is:

$$\begin{aligned} y_{13} &\geq E[\beta_{14}^{15}] \\ x_{13} - 13c &\geq \int_0^{50} max(y_{14}, E[Y_{15}]) f(x_{14}) dx_{14} \\ x_{13} - 13c &\geq \int_0^{50} max(x_{14} - 14c, 25 - 15c) \frac{1}{50} dx_{14} \\ x_{13} &\geq \int_0^{50} max(x_{14} - 14c, 25 - 15c) \frac{1}{50} dx_{14} + 13c \end{aligned}$$

Continuing on to stage $12, 11, .., 1$ using backward induction, we can therefore obtain the optimal stopping rules at each stage. For example, at stage 12, the optimal rule says we should continue as long as:

$$y_{12} \geq E[\beta_{13}^{15}] = E[max(Y_{13}, E[\beta_{14}^{15}])].$$

In my simulation, I have chosen $\mathbf{c = 5}$ for which it is worthwhile to take observations. According to the optimal stopping rule, we will stop at stage n if x_n is greater than the threshold t_n, where t_n for $n = 0, 1, ..., 14$. The obtained t_n values for $c = 5$ are plotted in Figure 1, which also includes results for other choices of c for the sake of comparison. Note that $c = 25$, for example, provides us a scenario where no observations are taken. At stage 1, the rule will tell us to stop if any positive payoff (which is always the case).

b) The expected payoff for the optimal stopping rule can be found in Figure 1. For $c = 5$, the expected payoff is $\mathbf{E[\beta_1^{15}] - c = 22.633}$.

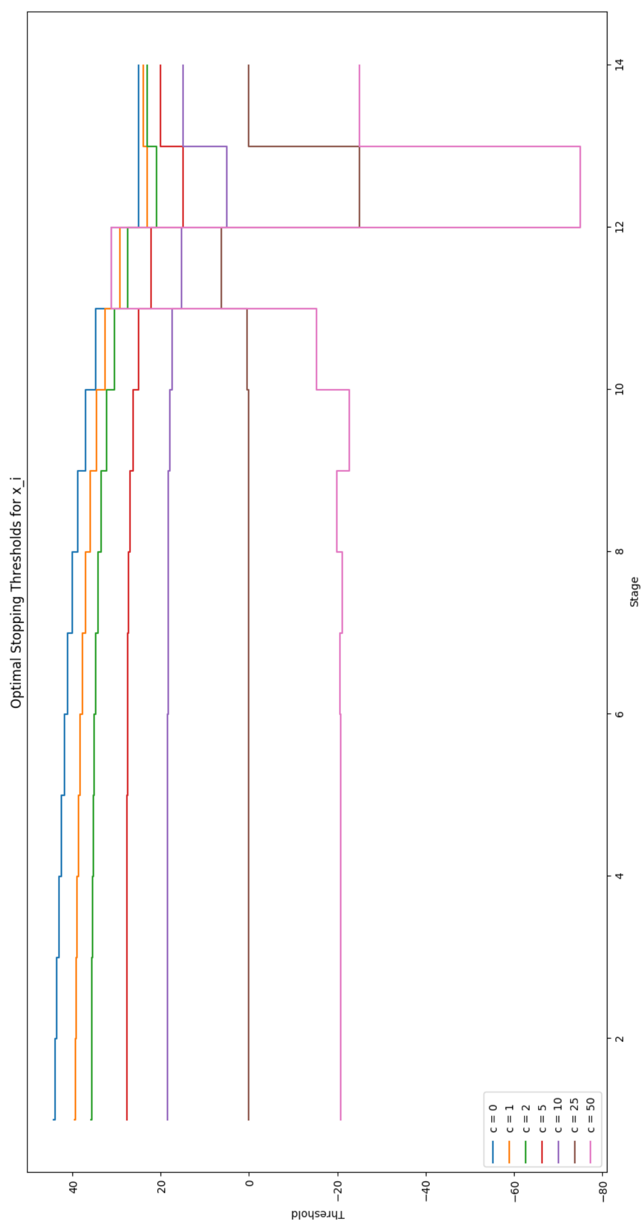

Figure 1: Optimal stopping thresholds for various choices of c.

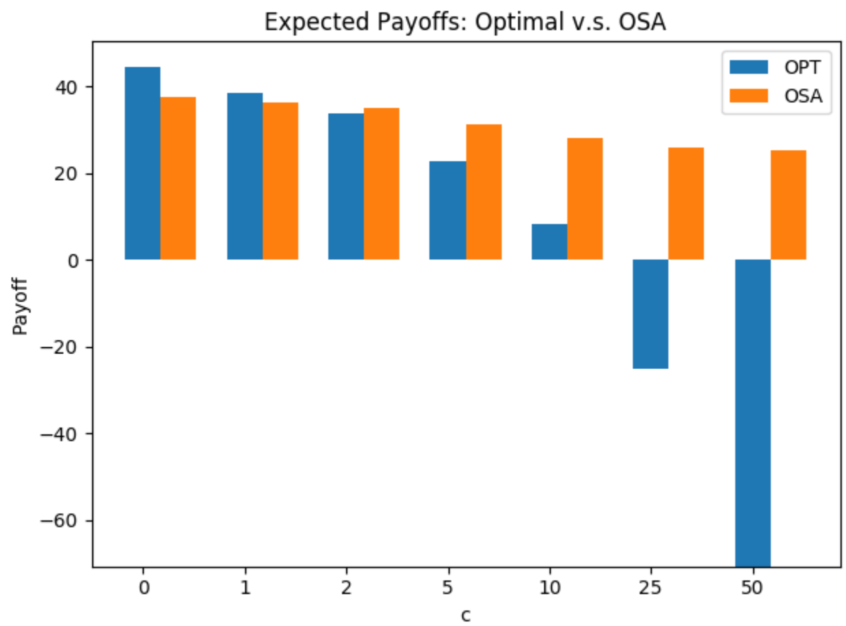

Figure 2: Expected payoffs for optimal stopping rule and OSA.

c) **One Step Ahead (OSA)**: we stop at stage n where the following condition is met:

$$y_n \geq E[Y_{n+1}],$$

which expands to

$$\begin{aligned} x_n - nc &\geq E[X_{n+1}] - (n+1)c \\ x_n - nc &\geq 25 - (n+1)c \\ x_n &\geq 25 - c. \end{aligned}$$

Simply put, we stop as soon as the observation x_n is greater than $25 - c$. The expected payoff of the OSA stopping rule is obtained by "playing the game" many times and averaging the outcomes. The resulting OSA expected payoffs of different c can be also found in Figure 2. For $c = 5$, the expected payoff is $\mathbf{31.327}$.

3 Secretary Problem

First of all, the following constants are provided: $\sigma^2 = 50$, $\tau^2 = 25$ and $\beta = 150$. In this problem, the observed performance of candidate n is $x_n \sim N(\theta_n|\sigma^2)$, where $\theta \sim N(\beta, \tau^2)$. Our goal is to hire the interviewee with the highest posterior mean:

$$\begin{aligned} E[\pi(\theta_n|x_n)] &= E[N(\frac{\sigma^2\beta + \tau^2 x_n}{\sigma^2 + \tau^2}, \frac{\sigma^2\tau^2}{\sigma^2 + \tau^2})] \\ &= \frac{\sigma^2\beta + \tau^2 x_n}{\sigma^2 + \tau^2} \\ &= 100 + \frac{x_n}{3} \\ &= w_n \sim N(150, 8.3). \end{aligned}$$

In other words, our payoff function is $Y_n = max(w_1, w_2, ..., w_n) - a_n$. Since the cost a_n only increments by a constant value c, our model is a Monotone Case, where the OSA stopping rule is optimal. Recall that OSA says we should stop when:

$$y_n \geq E[Y_{n+1}].$$

Letting m_n denote $max(w_1, w_2, ..., w_n)$, we can expand the condition to:

$$\begin{aligned}
m_n - a_n &\geq E[m_{n+1} - a_{n+1}] \\
m_n &\geq E[m_{n+1}] - c \\
m_n &\geq m_n + E[(w_{n+1} - m_n)^+] - c \\
E[(w_{n+1} - m_n)^+] &\geq c.
\end{aligned}$$

Now define γ such that $E[(w_{n+1} - \gamma)^+] = c$. Then the OSA stopping rule becomes: stop the first time $m_n \geq \gamma$, and we can solve for γ using the following integral equation:

$$\int_\gamma^\infty (w - \gamma) f(w) dw = c$$
$$\int_\gamma^\infty (w - \gamma) \frac{1}{2.88\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(w-150)^2}{8.3}} dw = c.$$

Using simulation, we arrive at $\gamma \approx$ **145** when $c = 5$.

4 Code

4.1 Q1

```
from pylab import *
from scipy.stats import norm as normal

def get_tau2_posterior_samples_MCMC(deltaq):
  '''
    Get the posterior samples for tau2.
  '''
  N = 20000
  # Posteriors before selection
  tau2 = zeros(N)
  # Initial guesses
  tau2[0] = rand() * 20 + 10
  # Generating loop
  i = 1
  while i < N:
    # Generate new sample
    tau2_i = rand() * 20 + 10
    # Determine whether to keep
```

```
    numer = exp(-deltaq**2 / 2 / (8 + 2 * tau2_i)) / sqrt(8 + 2 * tau2_i)
    denom = exp(-deltaq**2 / 2 / (8 + 2 * tau2[i-1])) / sqrt(8 + 2 *
     ↪  tau2[i-1])
    r = min(numer / denom, 1)
    # Keep with probability r
    if rand() > r: continue
    tau2[i] = tau2_i
    i += 1
  # Discard the first 500 to minimize the influence of the initial guess
  tau2 = tau2[500:]
  return tau2

def main():
  set_printoptions(precision=3)
  P = zeros(6)
  for i, deltaq in enumerate([30, 32, 34, 36, 38, 40]):
    tau2 = get_tau2_posterior_samples_MCMC(deltaq)
    rv = normal()
    numer = 65.2 - deltaq - deltaq * tau2 / (4 + tau2)
    denom = sqrt(8 + 8 * tau2 / (4 + tau2))
    Ps = 1 - rv.cdf(numer / denom)
    P[i] = mean(Ps)
  print P

if __name__ == '__main__':
  main()
```

4.2 Q2

```
from pylab import *

def get_optimal_rules_backward_induction(c):
  '''
    Obtain the optimal stopping thresholds for each stage.
    Thresholds are designated for x.
  '''
  # Optimal thresholds from stage 0 to 14
  # Stage 15 doesn't have a rule
  thresh = zeros(15)
  thresh[14] = 25 - c # Stage 14
  E_beta = zeros(15)
  E_beta[14] = 25 - 15 * c # beta_14^15
  for n in range(13, 0, -1):
    thresh[n] = E_beta[n + 1] + n * c
    E_beta[n] = (E_beta[n + 1] + n * c)**2 / 100. + 25 - n * c
```

```
  return thresh[1:], E_beta[1] - c

def get_rules_osa(c):
  '''
    Get OSA x thresholds, but no optimality guaranteed.
  '''
  return ones(15) * (25 - c)

def get_expected_payoff_osa(c, thresh):
  '''
    Repeat the game and return the average payoff.
  '''
  N = 1000
  Y = zeros(N)
  for i in range(N):
    j = 0
    x = rand() * 50
    while j < 15 and x < thresh[j]:
      j += 1
      x = rand() * 50
    Y[i] = x - j * c
  return mean(Y)

def main():
  set_printoptions(precision=3, suppress=True)
  C = [0, 1, 2, 5, 10, 25, 50]
  Y_opt = zeros(len(C))
  Y_osa = zeros((len(C)))
  figure()
  for i, c in enumerate(C):
    t_opt, Y_opt[i] = get_optimal_rules_backward_induction(c)
    t_osa = get_rules_osa(c)
    Y_osa[i] = get_expected_payoff_osa(c, t_osa)
    print "c =", c
    print t_opt
    print "Optimal expected payoff =", Y_opt[i]
    print "OSA expected payoff =", Y_osa[i]
    step(arange(14) + 1, t_opt, label="c = %d" % c)
  title('Optimal Stopping Thresholds for x_i')
  xlabel('Stage')
  ylabel('Threshold')
  legend()
  # Plot the expected payoffs
  figure()
  bar_width = 0.35
```

```
    index = arange(len(C))
    bar(index, Y_opt, bar_width, label='OPT')
    bar(index + bar_width, Y_osa, bar_width, label='OSA')
    title('Expected Payoffs: Optimal v.s. OSA')
    xlabel('c')
    ylabel('Payoff')
    xticks(index, [str(c) for c in C])
    legend()
    tight_layout()
    show()

if __name__ == '__main__':
    main()
```