Decision Theory Assignment 2

Mikael Montén

2024-12-12

Question 1

One decision-making criterion for decision-making under ignorance (non-probabilistic criterion) involves the consideration of a weighted average of the highest and lowest payoffs for each action. The weights, which must sum to 1, can be thought of as an optimism-pessimism index. The action with the highest weighted average of the highest and lowest payoffs is the action chosen by this criterion. Comment on this decision-making criterion and use it (i.e. find the optimal action) for payoff table (i) below with the highest payoff in each row receiving a weight of 0.4 and the lowest payoff receiving a weight of 0.6.

Table 1: Payoff table (i)

	State A	State B	State C	State D	State E
Action 1	-50	80	20	100	0
Action 2	30	40	70	20	30
Action 3	10	30	-30	10	40
Action 4	-10	-50	-70	-20	200

```r
payoff_wa <- matrix(0,4,1)
for(i in 1:4){
  # weighted average per action
  payoff_wa[i] <- 0.6*min(payoff[i,]) + 0.4*max(payoff[i,])
}
```

Table 2: Weighted average payoff table

	Weighted Average Payoff
Action 1	10
Action 2	40
Action 3	-2
Action 4	38

Action 2 has highest payoff. The weighted average of the highest and lowest payoffs for each action essentially means you denote a parameter θ which balances the optimistic vs pessimitic outlook in the actions to be taken according to the formula $\theta(a_{max}) + (1-\theta)(a_{min})$. For this case, say $\theta = 0.4$ (and as a result $\bar{\theta} = 0.6$) results in a decision with more pessimistic weight. $\theta = 1$ would mean the weighted average reduces to a completely optimistic maximax and $\theta = 0$ completely pessimistic maximin. The obvious advantage is the ability to incorporate the decision makers personal best and worst case scenarios rather than using any of the extremes.

Question 2

a)

Find the optimal action for payoff table (ii) below using the decision-making criterion described in task 1, with the highest payoff in each row receiving a weight of 0.8 and the lowest payoff receiving a weight of 0.2.

Table 3: Payoff table (ii)

	I	II
Action 1	10	4
Action 2	7	9

```
payoff_wa <- matrix(0,2,1)
for(i in 1:2){
  payoff_wa[i] <- 0.2*min(payoff[i,]) + 0.8*max(payoff[i,])
}
```

Table 4: Weighted average payoff table

	Weighted Average Payoff
Action 1	8.8
Action 2	8.6

b)

For payoff table (ii) the ER criterion (maximising the expected payoff) would also involve a weighted average of the two payoffs in each row. Assign the probability 0.8 to state I in payoff table (ii) and find the optimal action with the ER-criterion.

Expected Return is calculated using,

$$ER = \sum R(a, \theta) \cdot P(\theta)$$

where $R(a, \theta)$ is the payoff for action a in state θ, and $P(\theta)$ is the probability of state θ.

Payoff table (ii) has values $\{10, 7\}$ for State I and $\{4, 9\}$ for State II which are multiplied by their respective state probabilities $\{0.8, 0.2\}$ for their actions.

Table 5: Payoff table (ii) with explicit state probability

	I	II
Action 1	8.0	0.8
Action 2	5.6	1.8

```
payoff_er_wa <- matrix(0,2,1)
for(i in 1:2){
  payoff_er_wa[i] <- 0.8*payoff[i,1] + 0.2*payoff[i,2]
}
```

Table 6: Weighted average ER payoff table

	Weighted Average ER payoff
Action 1	8.8
Action 2	7.4

c)

Compare the outcomes with the two criteria and comment.

The Expected Return (ER) maximizes expected payoff by using the probability-weighted average of all possible payoffs for each action. This is a decision approach under risk (i.e. probabilistic criterion) as compared to under ignorance from question 1. It returns a lower expected payoff for both actions, but the lower probability for State II has reduced the absolute value of Action 2 more as that action has a higher reward for State II as opposed to Action 1 which has a higher payoff for State I.

For both cases, Action 1 is calculated to generate more payoff.

Question 3

Consider payoff table (i) in task 1. Assume the utility function of a person is $U(R) = log(R + 71)$*, where* R *is the payoff (and log is the natural logarithm with base e). Assign the probabilities* $p = (0.10, 0.20, 0.25, 0.10, 0.35)$ *to the states vector* (A, B, C, D, E) *and find the optimal action according to the EU-criterion (maximising expected utility).*

The optimal action according to the EU-criterion is the action that maximizes the expected utility under the probabiliy distribution that rules the state of the world. In this case we have discrete data, which gives us the Expected Utility formula

$$EU(a) = \sum U(a, \theta) \cdot P(\theta)$$

where,

- θ are states of the world
- a are available actions
- $P(\theta)$ are probabilities for each state
- $U(a, \theta)$ are utilities for each action-state combination

```
# utility function created
util <- function(R){
  return(log(R+71))
}
```

```
# payoff table i
payoff <- rbind(c(-50,80,20,100,0),
                c(30,40,70,20,30),
                c(10,30,-30,10,40),
                c(-10,-50,-70,-20,200))
```

First applying the utility to the states and then the probabilities. This is to calculate $E(U(R))$ and not $U(E(R))$ as I did before.

```
util_payoff_eu <- payoff
probs <- c(0.10,0.20,0.25,0.10,0.35)

for(i in 1:4){
  for(j in 1:5){
    util_payoff_eu[i,j] <- util(util_payoff_eu[i,j]) * probs[j]
  }
}
```

Table 7: Payoff table (i) with action-state utilities

	State A	State B	State C	State D	State E
Action 1	0.3044522	1.0034560	1.127715	0.5141664	1.491938
Action 2	0.4615121	0.9419060	1.237190	0.4510860	1.615292
Action 3	0.4394449	0.9230241	0.928393	0.4394449	1.648336
Action 4	0.4110874	0.6089045	0.000000	0.3931826	1.960742

```
util_sums <- rowSums(util_payoff_eu)
```

	Utility
Action 1	4.441727
Action 2	4.706986
Action 3	4.378642
Action 4	3.373916

Action 2 is the action with the most utility, which is the same as it was in question 1.

Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
payoff <- rbind(c(-50,80,20,100,0),
                c(30,40,70,20,30),
                c(10,30,-30,10,40),
                c(-10,-50,-70,-20,200))

colnames(payoff) <- paste0("State ",LETTERS[1:5])
rownames(payoff) <- paste0("Action ",1:4)
knitr::kable(payoff, caption = "Payoff table (i)")
payoff_wa <- matrix(0,4,1)
for(i in 1:4){
  # weighted average per action
  payoff_wa[i] <- 0.6*min(payoff[i,]) + 0.4*max(payoff[i,])
}
colnames(payoff_wa) <- c("Weighted Average Payoff")
rownames(payoff_wa) <- paste0("Action ",1:4)
knitr::kable(payoff_wa, caption = "Weighted average payoff table")
payoff <- rbind(c(10,4),
                c(7,9))
colnames(payoff) <- c("I","II")
rownames(payoff) <- paste0("Action ",1:2)
knitr::kable(payoff, caption = "Payoff table (ii)")
payoff_wa <- matrix(0,2,1)
for(i in 1:2){
  payoff_wa[i] <- 0.2*min(payoff[i,]) + 0.8*max(payoff[i,])
}
colnames(payoff_wa) <- c("Weighted Average Payoff")
rownames(payoff_wa) <- paste0("Action ",1:2)
knitr::kable(payoff_wa, caption = "Weighted average payoff table")
payoff_er <- rbind(c(10,4)*c(0.8,0.2),
                   c(7,9)*c(0.8,0.2))
colnames(payoff_er) <- c("I","II")
rownames(payoff_er) <- paste0("Action ",1:2)
knitr::kable(payoff_er, caption = "Payoff table (ii) with explicit state probability")
payoff_er_wa <- matrix(0,2,1)
for(i in 1:2){
  payoff_er_wa[i] <- 0.8*payoff[i,1] + 0.2*payoff[i,2]
}
colnames(payoff_er_wa) <- c("Weighted Average ER payoff")
rownames(payoff_er_wa) <- paste0("Action ",1:2)
knitr::kable(payoff_er_wa, caption = "Weighted average ER payoff table")
# utility function created
util <- function(R){
  return(log(R+71))
}
# payoff table i
payoff <- rbind(c(-50,80,20,100,0),
                c(30,40,70,20,30),
                c(10,30,-30,10,40),
                c(-10,-50,-70,-20,200))
util_payoff_eu <- payoff
```

```r
probs <- c(0.10,0.20,0.25,0.10,0.35)

for(i in 1:4){
  for(j in 1:5){
    util_payoff_eu[i,j] <- util(util_payoff_eu[i,j]) * probs[j]
  }
}
colnames(util_payoff_eu) <- paste0("State ",LETTERS[1:5])
rownames(util_payoff_eu) <- paste0("Action ",1:4)
knitr::kable(util_payoff_eu, caption = "Payoff table (i) with action-state utilities")
util_sums <- rowSums(util_payoff_eu)
util_sums <- as.matrix(util_sums)
colnames(util_sums) <- c("Utility")
knitr::kable(util_sums)
```