Assignment 2

Thijs Quast (thiqu264)

06/01/2019

Contents

Contributions

This report was written individually. However, during a meeting I have discussed solutions and collaborated with Julius Kittler, Faton Rekathati, Gustav Lundberg, Alexander Karlsson and Brian Masinde. During this meeting everyone contributed with ideas, remarks and thoughts on how to solve the assignments.

1

a

This decision making criteria considers the highest and lowest payoffs only, regardless of how probable they are. In a way this is not a very profound criteria as it does not consider payoffs which are not the highest nor the lowest. Maximima:

Row1 = 100 Row2 = 70 Row3 = 40 Row4 = 200

Minima:

Row1 = -50 Row2 = 20 Row3 = -30 Row4 = -70

Optimism-pessimism index:

- Row1 = 0.4 * 100 + 0.6 * -50 = 10
- Row2 = 0.4 * 70 + 0.6 * 20 = 40
- Row3 = 0.4 * 40 + 0.6 * -30 = -2
- Row4 = 0.4 * 200 + 0.6 * -70 = 38

b

Maxima: Row1 = 10 Row2 = 9

Minima: Row1 = 4 Row2 = 7

Optimism-pessimism index:

- Row1 = 0.8 * 10 + 0.2 * 4 = 8.8
- Row2 = 0.8 * 9 + 0.2 * 7 = 8.6

ER criterion: With ER, the probabilities would belong to the states of the world rather than to the values. Therefore this would result in a different conclusion.

c

$$U(R) = log(R + 71)$$

```r
utility<- function(R){
  utility = log(R+71)
  return(utility)
}
```

```r
prior_a = 0.10
prior_b = 0.20
prior_c = 0.25
prior_d = 0.10
prior_e = 0.35

priors = c(prior_a, prior_b, prior_c, prior_d, prior_e)
```

```r
payoff_i = matrix(data = c(-50, 80, 20, 100, 0,
                            30, 40, 70, 20, 50,
                            10, 30, -30, 10, 40,
                            -10, -50, -70, -20, 200), nrow = 4, ncol = 5, byrow = TRUE
                  )

colnames(payoff_i) = c("A", "B", "C", "D", "E")
rownames(payoff_i) = c(1,2,3,4)

payoff_i = as.data.frame(payoff_i)
```

```r
# Computing utility
utility_i = apply(payoff_i, c(1,2), utility)
```

We can now obtain the payoff for every action by multiplying the prior probability vectors with the transpose of the utility matrix.

```r
# Computing expected utilty
expected_payoffs = priors %*% t(utility_i)
expected_payoffs
```

```
##             1        2        3        4
## [1,] 4.441727 4.770221 4.378643 3.373916
```

The action which yields the highest expected utility is action 2 with an expected utility of 4.770221

2

a

Because in this problem we are facing a linear utility function we can immediately look at the payoff. I.e. there is no difference in marginal utility from payoff.

First compute posterior probabilities from given priors and likelihoods based on this one draw.

Samples: - 70%B, 30%R - 30%R, 70%B

Sample	State	Prior	Likelihood	Posterior
B	70% B, 30% R	0.6	0.7	0.42/(0.42+0.12) = 0.78
B	30% B, 70% R	0.4	0.3	0.12/(0.42+0.12) = 0.22
R	70% B, 30% R	0.6	0.3	0.18/(0.18+0.28) = 0.39
R	30% B, 70% R	0.4	0.7	0.28/(0.18+0.28) =0.61

Actions: - option1 : Guessing 70% is red - option2: Guessing 70% is blue

- ER(option1) = 0.6 x (-3) + 0.4 x (5) = 0.2
- ER(option2) = 0.6 x (5) + 0.4 x (-3) = 1.8

So the optimal prior choice is to go for option2, guessing 70% is blue.

- ER(option1 | B) = 0.77 x (5) + 0.22 x (-3) = 3.24
- ER(option2 | B) = 0.77 x (-3) + 0.22 x (5) = -1.24
- ER(option1 | R) = 0.39 x (5) + 0.61 x (-3) = 0.12
- ER(option2 | R) = 0.39 x (-3) + 0.61 x (5) = 1.88
- VSI(B) = E(option1 | B) - E(option1 | B) = 3.24 - 3.24 = 0
- VSI(R) = E(option2 | R) - E(option1 | R) = 1.88-0.12 = 1.76
- p(B) = 0.42 + 0.12 = 0.54
- p(R) = 0.18 + 0.28 = 0.46
- EVSI = VSI(B) x p(B) + VSI(R) x p(R) = 0 + 1.76*0.46 = 0.81

b

$$ENGS(n) = EVSI(n) - CS(n)$$
$$Given : n = 10, CS = 0.25$$

In this case, the scenario is as follows:

Decision	70% R, 30% B	30% R, 70% B
R	5	-3
B	-3	5

Out of a sample of 10. The number of reds can range from 0-10. This will give 11 posterior distributions.

```
# possible number of red samples varies from 0-10.
EVSI <- c()
```

```
for (nred in c(0:10)){
  # Probabilities of obtaining red or blue
  posterior_red <- dbinom(x = nred, size = 10, prob = 0.7) * 0.4
  posterior_blue <- dbinom(x= nred, size = 10, prob = 0.3) * 0.6
  posterior_nred <- posterior_red + posterior_blue
  posterior_nblue <- 1-posterior_nred

  # For red
  p_option1_nred <- (posterior_red)/posterior_nred
  p_option2_nred <- 1-p_option1_nred


  ER_option1_nred <- p_option1_nred*5 + p_option2_nred*-3
  ER_option2_nred <- p_option1_nred*-3 + p_option2_nred*5

  ER_options_nred <- c(ER_option1_nred, ER_option2_nred)
```

```r
  ER_prior <- 2
  VSI_nred <- max(ER_options_nred) - ER_options_nred[2]

  # For blue
  p_option1_nblue <- (posterior_blue)/posterior_nblue
  p_option2_nblue <- 1-p_option1_nblue

  ER_option1_nblue <- p_option1_nblue*5 + p_option2_nblue*-3
  ER_option2_nblue <- p_option1_nblue*-3 + p_option2_nblue*5

  ER_options_nblue <- c(ER_option1_nblue, ER_option2_nblue)


  VSI_nblue <- max(ER_options_nblue) - ER_options_nblue[2]


  EVSI[nred+1] = (VSI_nred * posterior_nred) + (VSI_nblue * posterior_nblue)
}
```

```r
EVSI
```

```
##  [1] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [7] 0.46395387 0.81064126 0.74017405 0.38673328 0.09036374
```

```r
ENGS <- sum(EVSI)-10*0.25
ENGS
```

```
## [1] -0.008133804
```

c

For simplicity I assume a 1 sample draw.

State	probabilities
70% red	0.4
70% blue	0.6

Sample	state: 70% red	state: 70% blue
red	0.61	0.39
blue	0.22	0.78

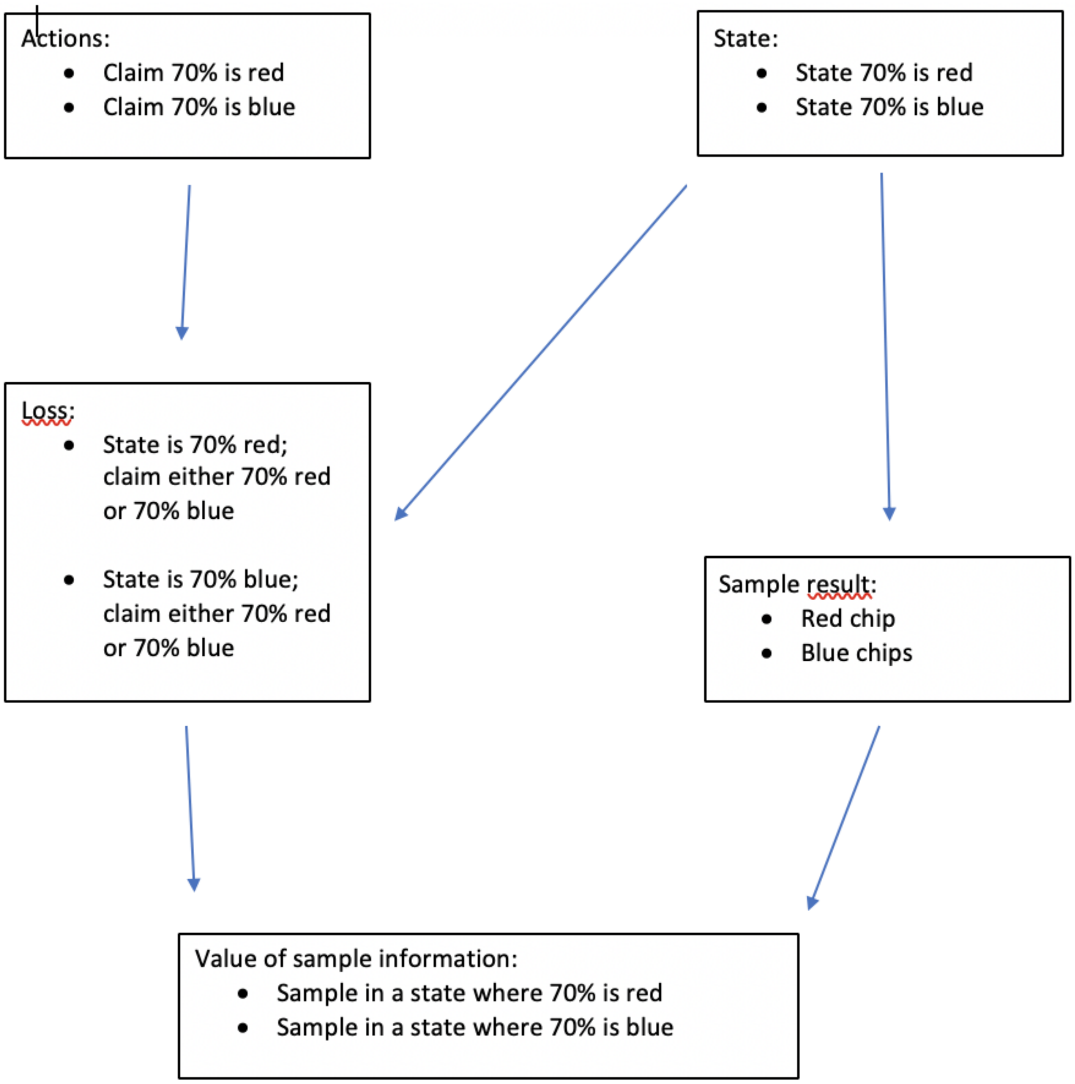
Action	Decision
a1	70% = red

Action	Decision
a2	70% = blue

Loss	state: 70% red	state: 70% blue
Actions	a1 , a2	a1 , a2
	-5 , 3	3 , -5

VSI	state: 70% red	state: 70% blue
Sample	1.76	0

Diagram:



3

a

We want to see at which point the cost functions intersect.

$$15 = 14 + 10p$$
$$p = 0.1$$

$$p(\tilde{p} < 0.1)$$

Using:

```r
pbeta(q = 0.1, shape1 = 2, shape2 = 48)
```

```
## [1] 0.9630964
```

$$p(\tilde{p} < 0.1) = 0.96$$

Therefore, choose option B.

b

Cost functions are indifferent when p = 0.1.

Loss functions under different scenarios:

LossA = 7500 - 500(14 + 10p), when p<0.1 LossA = 0, when p>0.1

LossB = 500(14+10p) - 7500, when p>0.1 LossB = 0, when p<0.1

Because we now look for the exact number, we look for the EVPI (meeting 11, slide 19). Which is defined as:

$$EVPI = \int VPI(\theta) * f(\theta)d\theta$$

When p > 0.1, option B is no longer profitable, therefore we have to look at the cost of this part of the integral.

$$Loss(B|state : p > 0.1) = 500(14 + 10p) - 7500$$
$$Loss(B|state : p > 0.1) = 7000 + 5000p - 7500 = 5000p - 500 = 500(10p - 1)$$

$$EVPI = 500 \int_{0.1}^{1} (10p - 1) * f(p)$$
$$EVPI = 500(10 \int_{0.1}^{1} p * f(p)d(p) - \int_{0.1}^{1} f(p)d(p)$$
$$EVPI = 500(10 \int_{0.1}^{1} \frac{p^2(1-p)^{47}}{B(2,48)} - \int_{0.1}^{1} \frac{p(1-p)^{47}}{B(2,48)}$$

Now a creative way of multiplying by one:

$$EVPI = 500(10 \int_{0.1}^{1} \frac{p^2(1-p)^{47}}{B(2,48)} * \frac{B(3,48)}{B(3,48)} - \int_{0.1}^{1} \frac{p(1-p)^{47}}{B(2,48)})$$
$$EVPI = 5000 \int_{0.1}^{1} dbeta(3,48) * \frac{B(3,48)}{B(2,48)} - 500 \int_{0.1}^{1} dbeta(2,48)$$

```r
evpi <- 5000*(1-pbeta(0.1, 3, 48)) * (beta(3,48)/beta(2,48)) - 500*(1-pbeta(0.1, 2, 48))
evpi
```

```
## [1] 3.893963
```

Thus the EVPI is:

3.89

c

The trials are a form of Bernouilli sampling. The beta is a conjugate prior, leading to a beta posterior, in which the parameters are:

- alpha = alpha + nr_successes
- beta = beta + nr_failures

The optimal prior choise is B, under the mean of p prior, which is 0.04, this will cost 7200.

```
prob <- c()
```

```
for (each in c(0:10)){
  prob[each+1] <- (2+each)/(48+10-each)
}
```

```
costs <- as.data.frame(matrix(data = NA, nrow = 11, ncol = 2))
colnames(costs) <- c("prior", "posterior")
costs[,1] <- 7200
costs[,2] <- 500*prob*10 + 7000
```

```
costs
```

```
##    prior posterior
## 1   7200  7172.414
## 2   7200  7263.158
## 3   7200  7357.143
## 4   7200  7454.545
## 5   7200  7555.556
## 6   7200  7660.377
## 7   7200  7769.231
## 8   7200  7882.353
## 9   7200  8000.000
## 10  7200  8122.449
## 11  7200  8250.000
```

```
evsi <- c()
```

```
evsi <- costs[,2] - 7500
# For options 1 till 4 it remains best to stick with option B. So:
evsi[1:4] <- 0
```

```
# Multiply posterior probabilities by probability of such a sample outcome to occur
p_theta <- c()
for (each in 0:10){
  p_theta[each+1] <- dbinom(each, size = 10, prob = 0.04)
}
```

```
evsi <- sum(p_theta * evsi)
evsi
```

```
## [1] 0.02695627
```

d

```
nr_defects <- 1
p <- (2+1)/(48+10-1)
```

If one decides to go for option A, the costs will be: 500x15 = 7500

If one decides to go for option B, the costs will be:

```
(500*14) + 10*p*500
```

```
## [1] 7263.158
```

In this case it is wiser to go for option B.

4

a

$$\bar{x} = \frac{1}{6}\sum_{i=1}^{6} x_i = 121000$$

$$X \sim N(\theta, \sigma = 12000)$$

where

$$\theta \sim N(115000, \sigma = 9000)$$

This exercise can be solved with simulation, e.g. with n=100 000

```
n <- 1000000
theta <- rnorm(n = n, mean = 115000, sd = 9000)
x <- c()

for (i in 1:n){
  x[i] <- rnorm(n = 1, mean = theta[i], sd = 12000)
}
```

```
x_greater <- length(x[x>120000])
x_smaller <- length(x[x<=120000])

prior_odds <- x_greater/x_smaller
prior_odds
```

```
## [1] 0.5859145
```

b

In order to find the Bayes factor, one needs the posterior distribution.In the course Bayesian Learning (732A91), we covered how one can update to a posterior distribution based on obtained data.

$$Prior : \theta \sim N(\mu_0, \tau_0^2)$$

$$Posterior : p(\theta|x_1, ..., x_n) \propto p(x_1, .., x_n|\theta, \sigma^2)p(\theta)$$
$$\propto N(\theta|\mu_n, \tau_n^2)$$

Where:

$$\frac{1}{\tau_n^2} = \frac{n}{\sigma^2} + \frac{1}{\tau_0^2}$$
$$\mu_n = w\bar{x} + (1 - w)\mu_0$$
$$w = \frac{\frac{n}{\sigma^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}}$$

Thus:

$$w = \frac{\frac{6}{12000^2}}{\frac{6}{12000^2} + \frac{1}{9000^2}} = 0.7714$$

$$\tau_n = \sqrt{\frac{1}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}}} = \sqrt{\frac{1}{\frac{6}{12000^2} + \frac{1}{9000^2}}} = 4302.82$$

$$\mu_n = 0.7714 \times 121000 + (1 - 0.7714) \times 115000 = 119628.4$$

```
n <- 1000000
theta <- rnorm(n = n, mean = 119628, sd = 4302)
x <- c()

for (i in 1:n){
  x[i] <- rnorm(n = 1, mean = theta[i], sd = 12000)
}
```

```
x_greater_posterior <- length(x[x>120000])
x_smaller_posterior <- length(x[x<=120000])
posterior_odds <- x_greater_posterior/x_smaller_posterior
posterior_odds
```

```
## [1] 0.9525949
```

$$Bayesfactor = \frac{posteriorodds}{priorodds} = \frac{0.9542472}{0.5858466} = 1.628834$$

```
bayes_factor <- posterior_odds/prior_odds
```

c

In meeting 15, slide 25 we discussed how to move from posterior odds to posterior probability.

$$P(H_0|Data) = \frac{B}{B + Priorodds}$$

```
posterior_probability = bayes_factor/(bayes_factor + prior_odds)
posterior_probability
```

```
## [1] 0.7350889
```

The probabliity that the costs will exceed 120 000 is approximately (depending per sample) 0.735, whereas the probability that the costs are lower or equal to 120 000 is approximately 1-0.73 = 0.264.

```
# Expected costs assuming that costs are lower than 120 000, but they turn out to be higher is:
posterior_probability * 4000
```

```
## [1] 2940.356
```

```
# Expected costs assuming that costs are higher than 120 000, but they turn out to be lower is:
(1-posterior_probability) * 6000
```

```
## [1] 1589.467
```

When minimizing expected loss, it is optimal to assume costs will exceed 120 000 SEK.