Decision Theory Assignment 4

Mikael Montén

2024-12-20

Question 1

Assume some budget calculations depend on whether a certain cost will be at least SEK 120 000 or lower than this amount. A reasonable model for this cost is a normal distribution with standard deviation SEK 12 000 (independent of the mean) and a mean that can be modelled as normally distributed with mean 115 000 (SEK) and standard deviation 9 000 (SEK). No trend is anticipated for this cost and for the 6 previous periods the average cost was SEK 121 000.

Note that the hypotheses are about the actual cost, not the expected cost.

a. Show that the prior odds for the hypothesis that the cost will exceed SEK 120 000 (against the alternative that it will not) is about 0.59. [hint: write the observed variable $\bar{x}$ as a sum of two independent random variables $\bar{x} = \bar{\mu} + \bar{\epsilon}$]

H_0 : The cost will be 120 000 SEK or higher

H_1 : The cost will be lower than 120 000 SEK

```
set.seed(123456)

mu_mean <- 115000
mu_sd <- 9000

prior_sd <- 12000

# calculating mu
prior_mu <- rnorm(1e05, mean = mu_mean, sd = mu_sd)
# calculating prior cost
prior <- rnorm(1e05, mean = prior_mu, sd = prior_sd)

# prior probabilities for each hypothesis
pr_p0 <- mean(prior < 120000)
pr_p1 <- 1-pr_p0

# prior odds of cost exceeding 120 000
pr_p1_odds <- pr_p1/pr_p0
pr_p0_odds <- pr_p0/pr_p1
print(pr_p1_odds)
```

```
## [1] 0.5878309
```

The result is 0.5878 so the calculations seem correct.

b. Show that the Bayes factor (considering the average cost for the 6 previous periods) for the hypothesis that the cost will exceed SEK 120 000 (against the alternative that it will not) is about 1.63 [Not about 1.60 or about 1.70].

The Bayes factor is defined as the ratio of the posterior odds to the prior odds of a hypothesis.

$$B = \frac{Odds(H_0|Data, I)}{Odds(H_0|I)} \implies Odds(H_0|Data, I) = B \cdot Odds(H_0|I)$$

If we know the Bayes factor, we can calculate the posterior odds. According to the information given, the standard deviation for the model is 12 000 SEK independent of mean which is used to model the posterior.

```
set.seed(123456)

n <- 6
obs_mean <- 121000

# posterior calculation from Bayesian Learning course slides
# posterior std dev with sigma2_0 and tau2_0 being given model sd

# prior sd for posterior predictive
prior_sd_pred <- sqrt(prior_sd^2 + mu_sd^2)

post_sd <- sqrt(1 / ((n/prior_sd_pred^2) + (1/prior_sd_pred^2)))

#post_sd <- 15000
# posterior mean
w <- (n/prior_sd_pred^2) / ((n/prior_sd_pred^2) + (1/prior_sd_pred^2))
post_mean <- w*obs_mean + (1-w)*prior_mu

posterior <- rnorm(1e05, mean = post_mean, sd = post_sd)

# posterior probabilities
po_p0 <- mean(posterior < 120000)
po_p1 <- 1-po_p0

# posteror odds
po_p1_odds <- po_p1/po_p0
po_p0_odds <- po_p0/po_p1

# bayes factor for h1
po_p0_odds / pr_p1_odds
```

```
## [1] 1.631974
```

My calculated Bayes Factor is 1.632.

c. *If the loss of accepting the hypothesis that the cost will be lower than SEK 120 000 while the opposite will be true is SEK 4 000, and the loss of accepting the hypothesis that the cost will be at least SEK 120 000 while the opposite will be true is SEK 6 000, which decision should be made for the budget (according to the rule of minimizing the expected loss)?*

The loss matrix looks as follows

```
dec_mat <- rbind(c(0,6000),
                 c(4000,0))
```

	Cost < 120000	Cost >= 120000
Accept H0	0	6000
Accept H1	4000	0

Incorporate the posterior probabilities:

```
dec_mat[1,2] <- dec_mat[1,2] * po_p1
dec_mat[2,1] <- dec_mat[2,1] * po_p0
```

	Cost < 120000	Cost >= 120000
Accept H0	0.00	3062.28
Accept H1	1958.48	0.00

To minimize the expected loss, the decision maker should accept H1.

Question 2

Consider a big box filled with an enormous amount of poker chips. You know that either 70% of the chips are red and the remainder blue, or 70% are blue and the remainder red. You must guess whether the big box has 70% red / 30% blue or 70% blue / 30% red. If you guess correctly, you win US $5. If you guess incorrectly, you lose US $3. Your prior probability that the big box contains 70% red / 30% blue is 0.40, and you are risk neutral in your decision making (i.e. your utility is linear in money).

The decision matrix looks as following:

```
dec_mat <- rbind(c(5,-3),c(-3,5))
```

	70% red / 30% blue	70% blue / 30% red
Guess 70% red / 30% blue	5	-3
Guess 70% blue / 30% red	-3	5

Into this we incorporate the probabilities of each state

```
majority_red <- 0.4
dec_mat[,1] <- dec_mat[,1]*majority_red
dec_mat[,2] <- dec_mat[,2]*(1-majority_red)
```

	70% red / 30% blue	70% blue / 30% red
Guess 70% red / 30% blue	2.0	-1.8
Guess 70% blue / 30% red	-1.2	3.0

a. If you could purchase sample information in the form of one draw of a chip from the big box, how much should you be willing to pay for it?

This is found by calculating the Expected Value of Sample Information $EVSI$. This requires calculating the Value of Sample Information $VSI(y) = E(R(a''|y)|y) - E(R(a'|y)|y)$ and calculating the $EVSI = \sum_y VSI(y) \cdot P(y)$ where $y = [red, blue]$.

The guesses will be formulated as $70R/30B$ and $30R/70B$ from now on.

Expected posterior utilities for each action $E(R(a'|y)|y)$ are

$E(R(\text{"70B/30R"}|y)|y) = (-3) \cdot 0.4 + 5 \cdot 0.6 = 1.8$

$E(R(\text{"70R/30B"}|y)|y) = (-3) \cdot 0.6 + 5 \cdot 0.4 = 0.2$

```
# expected value for the majority count of each color
e_red <- 0.2
e_blue <- 1.8

# prior beliefs
prior_red <- 0.4
prior_blue <- 0.6

# chips distributions
majority <- 0.7
minority <- 0.3

# probabilities given draws
prob_red <- prior_red * majority + prior_blue * minority
prob_blue <- prior_red * minority + prior_blue * majority
```

	Red	Blue
Probability given draws	0.46	0.54

```
# posterior probabilities given red
post_red_gr <- (prior_red * majority) / prob_red # majority red given red
post_blue_gr <- (prior_blue * minority) / prob_red # majority blue given red

# posterior probabilities given blue
post_red_gb <- (prior_red * minority) / prob_blue # majority red given blue
post_blue_gb <- (prior_blue * majority) / prob_blue # majority blue given blue
```

	Given red	Given blue
Posteriors guess red	0.6086957	0.3913043
Posteriors guess blue	0.2222222	0.7777778

```
# multiply posterior probs by losses
right <- 5
wrong <- (-3)

# draw red
d_red_g_red <- post_red_gr * right + post_blue_gr * wrong # guess red
d_red_g_blue <- post_red_gr * wrong + post_blue_gr * right # guess blue
```

```r
# draw blue
d_blue_g_red <- post_red_gb * right + post_blue_gb * wrong # guess red
d_blue_g_blue <- post_red_gb * wrong + post_blue_gb * right # guess blue
```

	Draw blue	Draw blue
Guess red	1.869565	0.1304348
Guess blue	-1.222222	3.2222222

From this, we can calculate the EVSI. Our prior belief tells us to accept H1 to maximize utility. This means we don't gain any "new" information from drawing a blue chips; thus only drawing a red chip will affect the Value of Sample Information. We calculate this with difference in expected values when drawing a red.

```r
vsi_red <- d_red_g_red - d_red_g_blue
evsi <- vsi_red * prob_red + 0
```

	Sample information values
VSI(red)	1.73913
EVSI	0.80000

Assume now that the cost of sampling is US$0.25 (i.e. 25 US cents) per draw.

b. *What is the ENGS for a sample of 10 chips using a single-stage sampling plan.*

The Expected Net Gain of Sampling (ENGS) is calculated by subtracting the Cost of Sampling (CS) from the Expected Value of Sample Information (EVSI)

$$ENGS = EVSI - CS$$

The calculations are modeled for drawing red chips. That is, y represents the number of red chips drawn and probabilities are calculated according to that as well.

```r
set.seed(123456)
# draws
n <- 10
# costs
cost_ps <- 0.25
cost_tot <- cost_ps * n

# marginal probabilities calculated by using probs from binomial distribution
marginal_prob <- function(y,n){
  dist_min <- dbinom(y, n, minority)
  dist_maj <- dbinom(y, n, majority)
  marg <- dist_maj*prior_red + dist_min*prior_blue
  return(marg)
}

# posterior probability for red
posterior_prob <- function(y,n){
  dist_min <- dbinom(y, n, minority)
  dist_maj <- dbinom(y, n, majority)

  num <- dist_maj * prior_red
```

```
  den <- dist_maj * prior_red+dist_min * prior_blue
  post <- num/den
  return(post)
}

# initiailize evsi
evsi <- c()

for(y in 0:n){
  # marginal probability of observing y red chips
  p_y <- marginal_prob(y, n)

  # posterior probs based on draw
  post_red <- posterior_prob(y,n)
  post_blue <- 1-post_red

  # expected value for each action
  ev_red <- 5 * post_red - 3*post_blue
  ev_blue <- 5 * post_blue -3*post_red

  # best action
  best_action <- max(ev_red, ev_blue)

  evsi[y+1] <- best_action * p_y

}

# expected value for the majority count being blue
e_blue <- 1.8

evsi_final <- sum(evsi)-e_blue

engs <- evsi_final - cost_tot
```

	Results from sampling
Total cost	2.5000000
EVSI	2.4918662
ENGS	-0.0081338

The ENGS is slightly negative which shows that sampling results in a net negative expected profit.

Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
set.seed(123456)

mu_mean <- 115000
mu_sd <- 9000

prior_sd <- 12000

# calculating mu
prior_mu <- rnorm(1e05, mean = mu_mean, sd = mu_sd)
# calculating prior cost
prior <- rnorm(1e05, mean = prior_mu, sd = prior_sd)

# prior probabilities for each hypothesis
pr_p0 <- mean(prior < 120000)
pr_p1 <- 1-pr_p0

# prior odds of cost exceeding 120 000
pr_p1_odds <- pr_p1/pr_p0
pr_p0_odds <- pr_p0/pr_p1
print(pr_p1_odds)
set.seed(123456)

n <- 6
obs_mean <- 121000

# posterior calculation from Bayesian Learning course slides
# posterior std dev with sigma2_0 and tau2_0 being given model sd

# prior sd for posterior predictive
prior_sd_pred <- sqrt(prior_sd^2 + mu_sd^2)

post_sd <- sqrt(1 / ((n/prior_sd_pred^2) + (1/prior_sd_pred^2)))

#post_sd <- 15000
# posterior mean
w <- (n/prior_sd_pred^2) / ((n/prior_sd_pred^2) + (1/prior_sd_pred^2))
post_mean <- w*obs_mean + (1-w)*prior_mu

posterior <- rnorm(1e05, mean = post_mean, sd = post_sd)

# posterior probabilities
po_p0 <- mean(posterior < 120000)
po_p1 <- 1-po_p0

# posteror odds
po_p1_odds <- po_p1/po_p0
po_p0_odds <- po_p0/po_p1

# bayes factor for h1
po_p0_odds / pr_p1_odds
```

```r
dec_mat <- rbind(c(0,6000),
                 c(4000,0))
colnames(dec_mat) <- c("Cost < 120000", "Cost >= 120000")
rownames(dec_mat) <- c("Accept H0", "Accept H1")
knitr::kable(dec_mat)
dec_mat[1,2] <- dec_mat[1,2] * po_p1
dec_mat[2,1] <- dec_mat[2,1] * po_p0
knitr::kable(dec_mat)
dec_mat <- rbind(c(5,-3),c(-3,5))
colnames(dec_mat) <- c("70% red / 30% blue", "70% blue / 30% red")
rownames(dec_mat) <- c("Guess 70% red / 30% blue", "Guess 70% blue / 30% red")
knitr::kable(dec_mat, align = 'c')
majority_red <- 0.4
dec_mat[,1] <- dec_mat[,1]*majority_red
dec_mat[,2] <- dec_mat[,2]*(1-majority_red)
knitr::kable(dec_mat, align = 'c')
# expected value for the majority count of each color
e_red <- 0.2
e_blue <- 1.8

# prior beliefs
prior_red <- 0.4
prior_blue <- 0.6

# chips distributions
majority <- 0.7
minority <- 0.3

# probabilities given draws
prob_red <- prior_red * majority + prior_blue * minority
prob_blue <- prior_red * minority + prior_blue * majority
probs <- rbind(c(prob_red, prob_blue))
rownames(probs) <- c("Probability given draws")
colnames(probs) <- c("Red", "Blue")

knitr::kable(probs)
# posterior probabilities given red
post_red_gr <- (prior_red * majority) / prob_red # majority red given red
post_blue_gr <- (prior_blue * minority) / prob_red # majority blue given red

# posterior probabilities given blue
post_red_gb <- (prior_red * minority) / prob_blue # majority red given blue
post_blue_gb <- (prior_blue * majority) / prob_blue # majority blue given blue
post_probs <- rbind(c(post_red_gr, post_blue_gr),
                    c(post_red_gb, post_blue_gb))
rownames(post_probs) <- c("Posteriors guess red",
                          "Posteriors guess blue")
colnames(post_probs) <- c("Given red", "Given blue")

knitr::kable(post_probs)
# multiply posterior probs by losses
right <- 5
wrong <- (-3)
```

```r
# draw red
d_red_g_red <- post_red_gr * right + post_blue_gr * wrong # guess red
d_red_g_blue <- post_red_gr * wrong + post_blue_gr * right # guess blue

# draw blue
d_blue_g_red <- post_red_gb * right + post_blue_gb * wrong # guess red
d_blue_g_blue <- post_red_gb * wrong + post_blue_gb * right # guess blue
payoffs <- rbind(c(d_red_g_red, d_red_g_blue),
                 c(d_blue_g_red, d_blue_g_blue))
rownames(payoffs) <- c("Guess red", "Guess blue")
colnames(payoffs) <- c("Draw blue", "Draw blue")


knitr::kable(payoffs)
vsi_red <- d_red_g_red - d_red_g_blue
evsi <- vsi_red * prob_red + 0
si <- rbind(vsi_red, evsi)
colnames(si) <- "Sample information values"
rownames(si) <- c("VSI(red)", "EVSI")
knitr::kable(si)
set.seed(123456)
# draws
n <- 10
# costs
cost_ps <- 0.25
cost_tot <- cost_ps * n

# marginal probabilities calculated by using probs from binomial distribution
marginal_prob <- function(y,n){
  dist_min <- dbinom(y, n, minority)
  dist_maj <- dbinom(y, n, majority)
  marg <- dist_maj*prior_red + dist_min*prior_blue
  return(marg)
}

# posterior probability for red
posterior_prob <- function(y,n){
  dist_min <- dbinom(y, n, minority)
  dist_maj <- dbinom(y, n, majority)

  num <- dist_maj * prior_red
  den <- dist_maj * prior_red+dist_min * prior_blue
  post <- num/den
  return(post)
}

# initiailize evsi
evsi <- c()

for(y in 0:n){
  # marginal probability of observing y red chips
  p_y <- marginal_prob(y, n)
```

```r
  # posterior probs based on draw
  post_red <- posterior_prob(y,n)
  post_blue <- 1-post_red

  # expected value for each action
  ev_red <- 5 * post_red - 3*post_blue
  ev_blue <- 5 * post_blue -3*post_red

  # best action
  best_action <- max(ev_red, ev_blue)

  evsi[y+1] <- best_action * p_y

}

# expected value for the majority count being blue
e_blue <- 1.8

evsi_final <- sum(evsi)-e_blue

engs <- evsi_final - cost_tot
results <- rbind(cost_tot, evsi_final, engs)
colnames(results) <- "Results from sampling"
rownames(results) <- c("Total cost", "EVSI", "ENGS")
knitr::kable(results)
```