

IS606 Bayesian (Homework 9)

Daniel Dittenhafer

December 6, 2015

2.1 (p31)

Model A The following code defines a function for the Model A probabilities.

```
xSeq <- seq(1, 4)
modelA <- function(x)
{
  return (1 / 4)
}

dfModelA <- data.frame(x=xSeq, Px=modelA(xSeq))
kable(dfModelA, caption="Model A Probabilities")
```

Table 1: Model A Probabilities

x	Px
1	0.25
2	0.25
3	0.25
4	0.25

We don't see any bias in the Model A probabilities. They are all equal.

Model B The following code defines a function for the Model B probabilities.

```
modelB <- function(x)
{
  return (x / 10)
}

dfModelB <- data.frame(x=xSeq, Px=modelB(xSeq))
kable(dfModelB, caption="Model B Probabilities")
```

Table 2: Model B Probabilities

x	Px
1	0.1
2	0.2
3	0.3
4	0.4

Model B is biased toward the higher numbers.

Model C The following code defines a function for the Model C probabilities.

```
modelC <- function(x)
{
  return (12 / (25 * x))
}

dfModelC <- data.frame(x=xSeq, Px=modelC(xSeq))
kable(dfModelC, caption="Model C Probabilities")
```

Table 3: Model C Probabilities

x	Px
1	0.48
2	0.24
3	0.16
4	0.12

Model C is biased toward the lower numbers, slightly more so than Model B was toward the higher numbers.

5.1 (p118)

The following code works through the test for a disease and retest scenario for an individual.

```
# Define marginal probabilities
marginalProbDiseasePresent <- 0.001
marginalProbDiseaseAbsent <- 1 - marginalProbDiseasePresent
# Define composite probabilities
probTestPosGivenDisPresent <- 0.99
probTestPosGivenDisAbsent <- 0.05
# Func of Bayes rule
bayesRule <- function(likelihood, prior, evidence)
{
  return ((likelihood * prior) / sum(evidence))
}
```

First we compute the probability of having the disease given the test is positive.

```
# Denominator in component form
evidence <- c(probTestPosGivenDisPresent * marginalProbDiseasePresent,
              probTestPosGivenDisAbsent * marginalProbDiseaseAbsent)
# Call function to compute probability of disease present given a positive test.
probDisPresentGivenTestPost <- bayesRule(probTestPosGivenDisPresent,
                                          marginalProbDiseasePresent,
                                          evidence)
probDisPresentGivenTestPost
```

```
## [1] 0.01943463
```

Now, after a re-test and a negative result, what is the probability of having the disease?

```

pPrior <- probDisPresentGivenTestPost
probTestNegGivenDisPresent <- (1 - probTestPosGivenDisPresent)
evidence <- c(probTestNegGivenDisPresent * pPrior,
              (1 - probTestPosGivenDisAbsent) * (1 - pPrior))
# Call function to compute probability of disease present given a negative test.
probDisPresentGivenTestNeg <- bayesRule(probTestNegGivenDisPresent,
                                         pPrior,
                                         evidence)
format(probDisPresentGivenTestNeg, scientific = FALSE)

## [1] "0.0002085862"

```

5.2 (p118)

```

n <- 100000
probTestNegGivenDisAbsent <- 1 - probTestPosGivenDisAbsent
dfTable <- data.frame(ThetaSad=c(99, 1, 100),
                      ThetaHappy=c(probTestPosGivenDisAbsent * marginalProbDiseaseAbsent * n,
                                   probTestNegGivenDisAbsent * marginalProbDiseaseAbsent * n,
                                   99900),
                      Col3=c(NA, NA, 100000))
dfTable$Col3 <- dfTable$ThetaSad + dfTable$ThetaHappy
dfTable

```

a) Table

```

##   ThetaSad ThetaHappy   Col3
## 1      99      4995   5094
## 2       1     94905  94906
## 3     100     99900 100000

```

b) Proportion of people who have the disease given their test result is positive It looks like about 2%. Computing:

```

b <- 99 / 5094
b

```

```

## [1] 0.01943463

```

c) The following code computes the “Markov” natural frequencies and displays the table.

```

nDisPresent <- 10000
nDisAbsent <- 9990000
dfNaturalFreq <- data.frame(ThetaSad=c(nDisPresent,
                                       nDisPresent * (0.99),
                                       (nDisPresent * (0.99)) * ((1-0.99))),
                           ThetaHappy=c(nDisAbsent,
                                       nDisAbsent * (0.05),
                                       (nDisAbsent * (0.05)) * (1-0.05)))
kable(dfNaturalFreq)

```

ThetaSad	ThetaHappy
10000	9990000
9900	499500
99	474525

d) What proportion of people who test positive at first and then negative on retest, actually have the disease?

```

numerator <- ((nDisPresent * (0.99)) * ((1-0.99)))
denom <- ((nDisPresent * (0.99)) * (1-0.99)) + (nDisAbsent * (0.05) * (1-0.05))
nActuallyHaveDisease <- numerator / denom

nActuallyHaveDisease

## [1] 0.0002085862

```