

Bayesian Statistics- HW1

魏上傑

2023-04-14

1.

It is estimated that roughly 8% of incoming email is spam. A spam filter has an accuracy rate of 92% for spam emails, and it incorrectly categorizes 3% as non-spam emails as spam.

(a) What percent of all email is marked as spam?

I will use S as spam, NS as non-spam, and MS as marked-spam for short.

$$\begin{aligned} P(MS) &= P(S, MS) + P(NS, MS) \\ &= P(MS|S) * P(S) + P(MS|NS) * P(NS) \\ &= 0.92 * 0.08 + 0.03 * 0.92 \\ &= 0.102 \end{aligned}$$

```
0.93*0.08+0.03*0.92
```

```
## [1] 0.102
```

(b) If an email is marked as spam, what is the probability that it is indeed a spam email?

$$P(S|MS) = \frac{P(S, MS)}{P(MS)} = \frac{0.92 * 0.08}{0.102} = 0.72$$

```
0.92*0.08/0.102
```

```
## [1] 0.7215686
```

2.

Marginal and conditional probability: The social mobility data from Section 2.5 gives a joint probability distribution on $(Y_1, Y_2) = (\text{father's occupation}, \text{son's occupation})$. Using this joint distribution, calculate the following distributions:

(a) the marginal probability distribution of a father's occupation;

$$P(\text{farm}) = 0.018 + 0.035 + 0.031 + 0.008 + 0.018 = 0.11$$

$$P(\text{operatives}) = 0.002 + 0.112 + 0.064 + 0.032 + 0.069 = 0.279$$

$$P(\text{craftsmen}) = 0.001 + 0.066 + 0.094 + 0.032 + 0.084 = 0.277$$

$$P(\text{sales}) = 0.001 + 0.018 + 0.019 + 0.010 + 0.051 = 0.099$$

$$P(\text{professional}) = 0.001 + 0.029 + 0.032 + 0.043 + 0.130 = 0.235$$

```
0.018+0.035+0.031+0.008+0.018
```

```
## [1] 0.11
```

```
0.002+0.112+0.064+0.032+0.069
```

```
## [1] 0.279
```

```
0.001+0.066+0.094+0.032+0.084
```

```
## [1] 0.277
```

```
0.001+0.018+0.019+0.010+0.051
```

```
## [1] 0.099
```

```
0.001+0.029+0.032+0.043+0.130
```

```
## [1] 0.235
```

(b) the marginal probability distribution of a son's occupation;

$$P(\text{farm}) = 0.018 + 0.002 + 0.001 + 0.001 + 0.001 = 0.023$$

$$P(\text{operatives}) = 0.035 + 0.112 + 0.066 + 0.018 + 0.029 = 0.26$$

$$P(\text{craftsmen}) = 0.031 + 0.064 + 0.094 + 0.019 + 0.032 = 0.24$$

$$P(\text{sales}) = 0.008 + 0.032 + 0.032 + 0.010 + 0.043 = 0.125$$

$$P(\text{professional}) = 0.018 + 0.069 + 0.084 + 0.051 + 0.130 = 0.352$$

```
0.018+0.002+0.001+0.001+0.001
```

```
## [1] 0.023
```

```
0.035+0.112+0.066+0.018+0.029
```

```
## [1] 0.26
```

```
0.031+0.064+0.094+0.019+0.032
```

```
## [1] 0.24
```

```
0.008+0.032+0.032+0.010+0.043
```

```
## [1] 0.125
```

```
0.018+0.069+0.084+0.051+0.130
```

```
## [1] 0.352
```

(c) the conditional distribution of a son's occupation, given that the father is a farmer;

$$P(sfarm|ffarm) = 0.018/0.11 = 0.164$$

$$P(sOper|ffarm) = 0.035/0.11 = 0.318$$

$$P(sCraft|ffarm) = 0.031/0.11 = 0.282$$

$$P(sSales|ffarm) = 0.008/0.11 = 0.073$$

$$P(sPro|ffarm) = 0.018/0.11 = 0.164$$

```
0.018/0.11
```

```
## [1] 0.1636364
```

```
0.035/0.11
```

```
## [1] 0.3181818
```

```
0.031/0.11
```

```
## [1] 0.2818182
```

```
0.008/0.11
```

```
## [1] 0.07272727
```

```
0.018/0.11
```

```
## [1] 0.1636364
```

(d) the conditional distribution of a father's occupation, given that the son is a farmer.

$$P(ffarm|sfarm) = 0.018/0.023 = 0.783$$

$$P(fOper|sfarm) = 0.002/0.023 = 0.087$$

$$P(fCraft|sfarm) = 0.001/0.023 = 0.043$$

$$P(fSales|sfarm) = 0.001/0.023 = 0.043$$

$$P(fPro|sfarm) = 0.001/0.023 = 0.043$$

```
0.018/0.023
```

```
## [1] 0.7826087
```

```
0.002/0.023
```

```
## [1] 0.08695652
```

```
0.001/0.023
```

```
## [1] 0.04347826
```

```
0.001/0.023
```

```
## [1] 0.04347826
```

0.001/0.023

[1] 0.04347826

3.

Let $X \sim N(\mu, \sigma^2)$, what distribution does $Y = (X - \mu)/\sigma$ have? (Hint: review the change of variable material.)

$$\begin{aligned}
 Y = g(X) = X - \mu/\sigma &\implies X = g^{-1}(Y) = \sigma Y + \mu \\
 f_Y(y) &= f_X(g^{-1}(y)) \left| \frac{dg^{-1}(y)}{dy} \right| \\
 &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \times \sigma \\
 &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(\sigma y)^2}{2\sigma^2}\right) \\
 &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) \implies Y \sim N(0, 1)
 \end{aligned}$$

4.

Let $X_1, \dots, X_n \sim \text{Normal}(\mu_X, \sigma_X)$, and $Y_1, \dots, Y_m \sim \text{Normal}(\mu_Y, \sigma_Y)$, and assume all X's and Y's are independent. Write out the joint density:

$$f(X_1 = x_1, \dots, X_n = x_n, Y_1 = y_1, \dots, Y_m = y_m | \mu_X, \mu_Y, \sigma_X, \sigma_Y).$$

$$\begin{aligned}
 &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-\frac{(x_i - \mu_x)^2}{2\sigma_x^2}\right) \prod_{j=1}^m \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left(-\frac{(y_j - \mu_y)^2}{2\sigma_y^2}\right) \\
 &= (2\pi\sigma_x^2)^{n/2} (2\pi\sigma_y^2)^{m/2} \exp\left(-\frac{\sum_i (x_i - \mu_x)^2}{2\sigma_x^2}\right) \exp\left(-\frac{\sum_j (y_j - \mu_y)^2}{2\sigma_y^2}\right)
 \end{aligned}$$

5.

We say a random variable X has a logistic distribution, if its cdf is $F(x) = 1 + \frac{e^x}{1+e^x}, -\infty < x < \infty$.

To compare this distribution with the standard normal distribution, use R to plot their pdf's (in the same graph). Suppose you draw 1000 random samples from each distribution, then which distribution will you expect to see more samples with extreme values (Heavy tail, i.e., very large in absolute value, either positive or negative)?

(Hint: in R, `dnorm()` is the function to evaluate the pdf of a normal distribution, and `dlogis()` is the function to evaluate the pdf of a logistic distribution.)

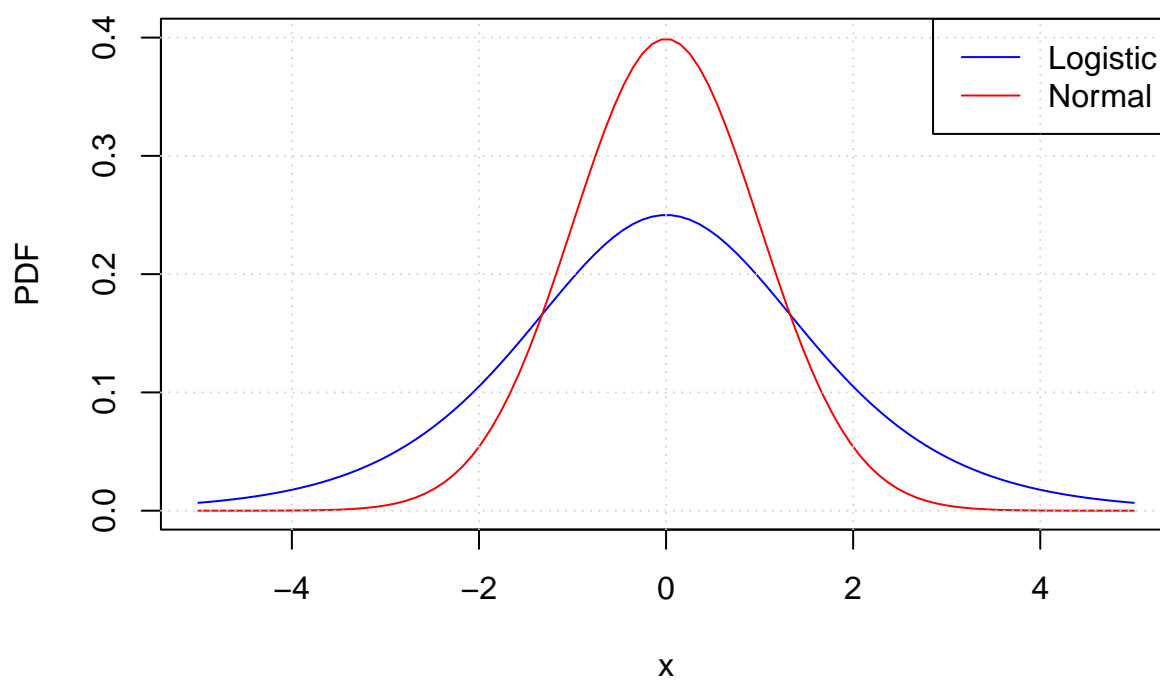
```

# Generate x-values
x <- seq(-5, 5, length.out = 100)

# Compute PDF values
pdf_logis <- dlogis(x)
pdf_norm <- dnorm(x)

# Plot PDFs
plot(x, pdf_logis, type = "l", col = "blue", ylim = c(0, 0.4),
     ylab = "PDF", xlab = "x")
lines(x, pdf_norm, col = "red")
legend("topright", legend = c("Logistic", "Normal"), lty = 1,
     col = c("blue", "red"))
grid()

```



```

# Generate random samples
n <- 1000
samples_logis <- rlogis(n)
samples_norm <- rnorm(n)

# Count extreme values
num_extreme_logis <- sum(abs(samples_logis) > 3)

```

```

num_extreme_norm <- sum(abs(samples_norm) > 3)

# Print results
cat("Number of extreme values in logistic distribution:", num_extreme_logis, "\n")

## Number of extreme values in logistic distribution: 98

cat("Number of extreme values in normal distribution:", num_extreme_norm, "\n")

## Number of extreme values in normal distribution: 2

```

6.

In the all-nighters example in class, we took a sample of $n = 10$ Vassar students and obtained $y = 3$ out of 10 who had pulled an all-nighter in last academic year. Recall that we came up with a prior distribution for the percentage p of all Vassar students who had pulled an all-nighter in last academic year as (in R):

```

priorvalues <- c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)
priorprob <- c(1/23, 1/23, 7/23, 7/23, 3/23, 3/23, 1/23, 0/23, 0/23, 0/23, 0/23)

```

We used the pre-written RMarkdown file to calculate the posterior probabilities of 11 values of p and plot the prior and posterior on the same graph.

Now consider another sample of $n_2 = 10$ with $y_2 = 5$ (use new notation for the previous sample $n_1 = 10, y_1 = 3$). Use the R script and make changes to perform the following:

- (a) Treat two samples as one sample, that is, $n = n_1 + n_2 = 10 + 10 = 20$ and $y = y_1 + y_2 = 3 + 5 = 8$. Calculate the posterior probabilities of p given this new combined sample (n, y) .

```

priorvalues <- c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)
priorprob <- c(1/23, 1/23, 7/23, 7/23, 3/23, 3/23, 1/23, 0/23, 0/23, 0/23, 0/23)

n <- 20
y <- 8

```

```

set.seed(1111)
#vector for storing results
jointprob <- numeric(length = length(priorvalues))

for(i in 1:length(priorvalues))
{

```

```

#compute Binomial probability given value of p - likelihood
binomprob <- dbinom(y, n, p = priorvalues[i])

#compute joint probability - posterior
jointprob[i] <- binomprob * priorprob[i]

}

#compute marginal probability of y
pofy <- sum(jointprob)

#compute posterior probabilities
posteriorprob <- jointprob/pofy

#put posterior probabilities in one matrix object for easy viewing
allnighterposterior <- as.data.frame(cbind(priorvalues, priorprob, posteriorprob))
names(allnighterposterior) <- c("p", "prior", "posterior")

#list the final posterior distribution, based on our prior derived in class
allnighterposterior

##      p      prior  posterior
## 1 0.0 0.04347826 0.000000000
## 2 0.1 0.04347826 0.0001881144
## 3 0.2 0.30434783 0.0820219042
## 4 0.3 0.30434783 0.4234055597
## 5 0.4 0.13043478 0.2850546884
## 6 0.5 0.13043478 0.1905607002
## 7 0.6 0.04347826 0.0187690330
## 8 0.7 0.00000000 0.000000000
## 9 0.8 0.00000000 0.000000000
## 10 0.9 0.00000000 0.000000000
## 11 1.0 0.00000000 0.000000000

#plot the prior and posterior probabilities
require(ggplot2)

```

```
## Loading required package: ggplot2
```

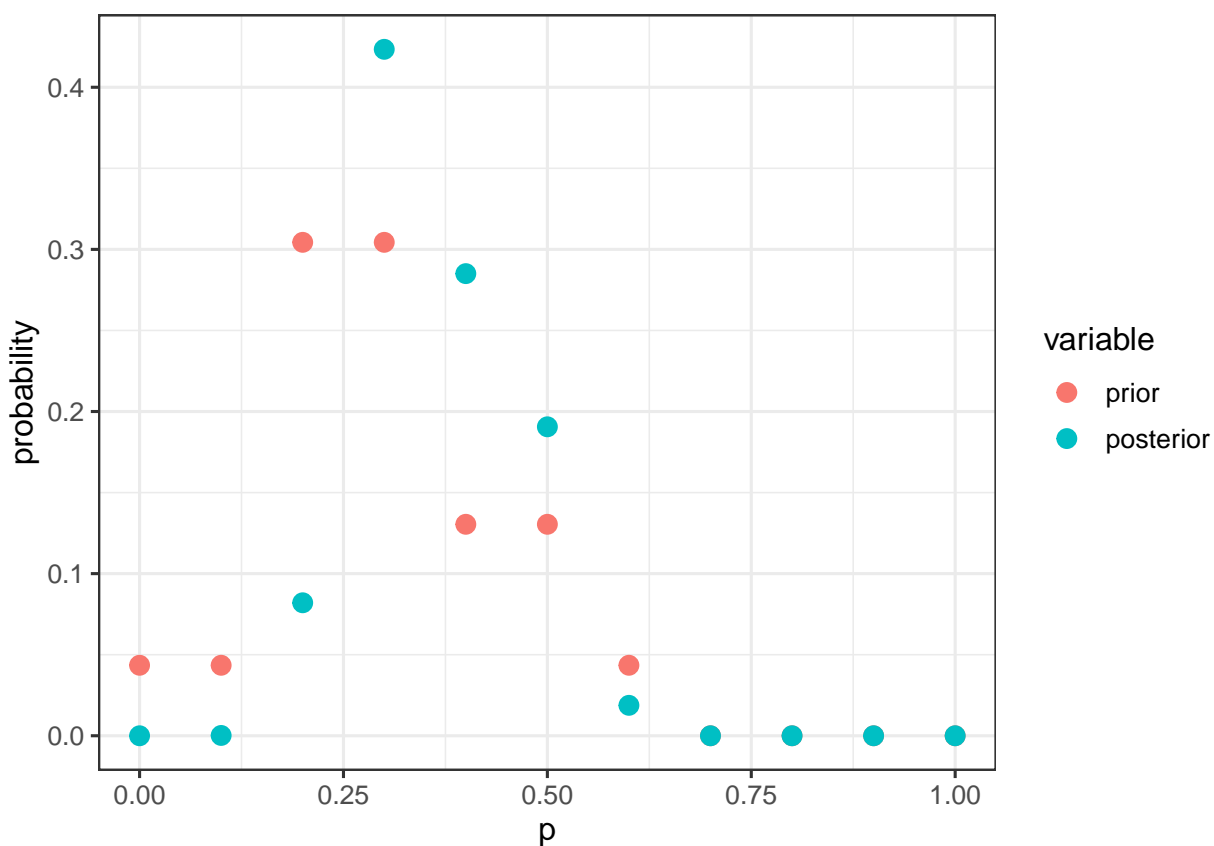


```
require(reshape2)
```

```
## Loading required package: reshape2
```

```
allnighterposterior_all <- melt(allnighterposterior, id = "p")

ggplot(allnighterposterior_all, aes(x = p, y = value, colour = variable)) +
  geom_point(size = 3) +
  xlab("p") + ylab("probability") +
  theme_bw(base_size = 12, base_family = "")
```



- (b) Recall that a sequential update is to use the previous posterior from the first sample (n_1, y_1) as the prior for the second sample (n_2, y_2). Do a sequential update and check whether the posterior probabilities match with what you calculated in (a).

We need to run sample 1 first.

```
set.seed(1111)
priorvalues <- c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)
priorprob <- c(1/23, 1/23, 7/23, 7/23, 3/23, 3/23, 1/23, 0/23, 0/23, 0/23, 0/23)

n <- 10
```

```

y <- 3

#vector for storing results
jointprob <- numeric(length = length(priorvalues))

for(i in 1:length(priorvalues))
{

  #compute Binomial probability given value of p - likelihood
  binomprob <- dbinom(y, n, p = priorvalues[i])

  #compute joint probability - posterior
  jointprob[i] <- binomprob * priorprob[i]

}

#compute marginal probability of y
pofy <- sum(jointprob)

#compute posterior probabilities
posteriorprob <- jointprob/pofy

#put posterior probabilities in one matrix object for easy viewing
allnighterposterior <- as.data.frame(cbind(priorvalues, priorprob, posteriorprob))
names(allnighterposterior) <- c("p", "prior", "posterior")

allnighterposterior_sample1 <- allnighterposterior$posterior

```

Now, we use the posteriors obtained from sample 1 as our new prior for sample 2.

```

set.seed(1111)
priorvalues <- c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)
priorprob <- allnighterposterior_sample1

n <- 10
y <- 5

```

```

#vector for storing results
jointprob <- numeric(length = length(priorvalues))

for(i in 1:length(priorvalues))
{

  #compute Binomial probability given value of p - likelihood
  binomprob <- dbinom(y, n, p = priorvalues[i])

  #compute joint probability - posterior
  jointprob[i] <- binomprob * priorprob[i]

}

#compute marginal probability of y
pofy <- sum(jointprob)

#compute posterior probabilities
posteriorprob <- jointprob/pofy

#put posterior probabilities in one matrix object for easy viewing
allnighterposterior <- as.data.frame(cbind(priorvalues, priorprob, posteriorprob))
names(allnighterposterior) <- c("p", "prior", "posterior")

#list the final posterior distribution, based on our prior derived in class
allnighterposterior

```

```

##      p      prior    posterior
## 1  0.0 0.000000000 0.000000000
## 2  0.1 0.013123561 0.0001881144
## 3  0.2 0.322234525 0.0820219042
## 4  0.3 0.427073101 0.4234055597
## 5  0.4 0.147473543 0.2850546884
## 6  0.5 0.080385077 0.1905607002
## 7  0.6 0.009710192 0.0187690330
## 8  0.7 0.000000000 0.0000000000
## 9  0.8 0.000000000 0.0000000000
## 10 0.9 0.000000000 0.0000000000

```

```
## 11 1.0 0.000000000 0.000000000
```

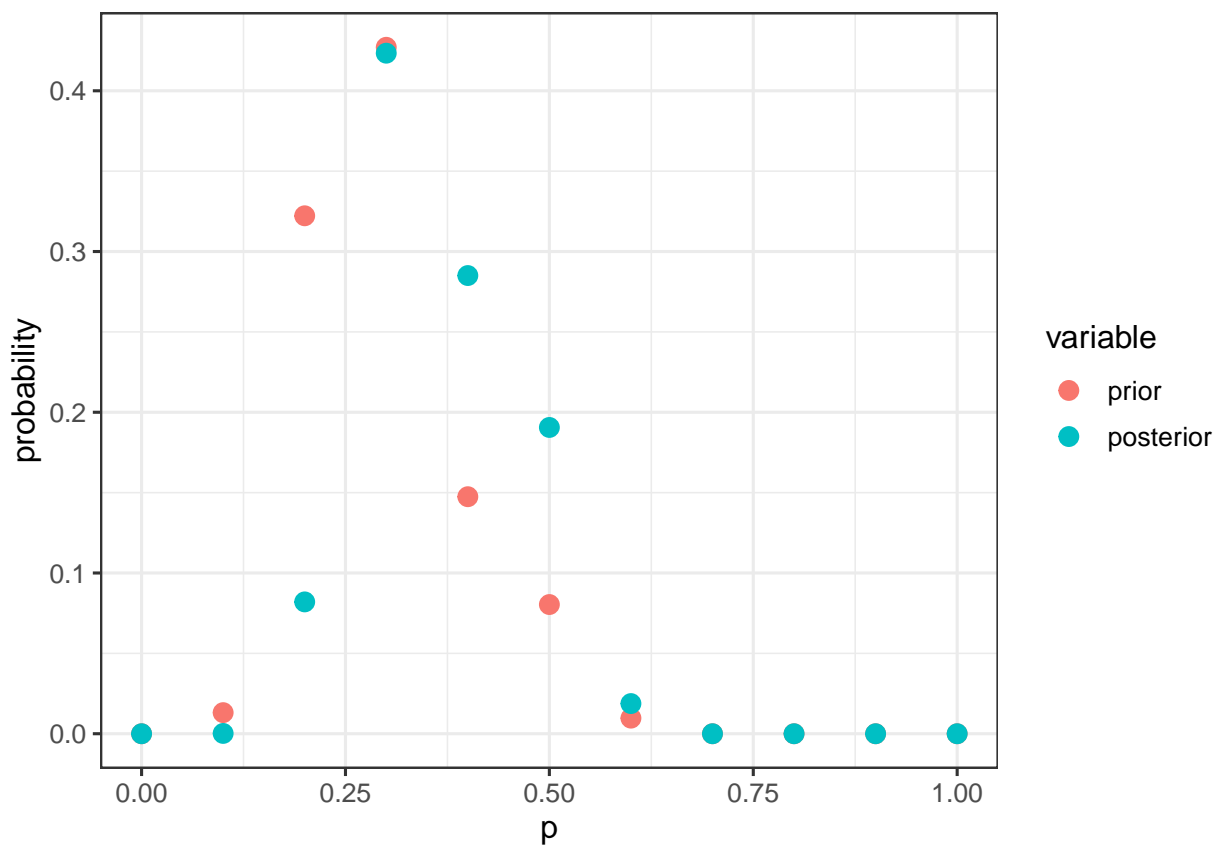
```
#plot the prior and posterior probabilities
```

```
require(ggplot2)
```

```
require(reshape2)
```

```
allnighterposterior_all <- melt(allnighterposterior, id = "p")
```

```
ggplot(allnighterposterior_all, aes(x = p, y = value, colour = variable)) +  
  geom_point(size = 3) +  
  xlab("p") + ylab("probability") +  
  theme_bw(base_size = 12, base_family = "")
```



The posterior probabilities match with what I calculated in (a).

(c) Would sequential update make sense in this case? Why or why not?

Yes, this is because these two samples share the same prior belief, which in Bayesian viewpoint means that these two samples are similar. It wouldn't make sense if these two samples are not alike.