Solutions to Homework 1

Shashank Sule

9/10/2019

Problem 1

a. Let S be the event that an email is spam and M be the event that an email is marked as spam. Then

$$P(M) = P(M|S)P(S) + P(M|S^c)P(S^c) = (0.92)(0.08) + (0.03)(0.92) = 0.1012$$

```
(0.92)*(0.08) + (0.03)*(0.92)
```

```
## [1] 0.1012
```

b. We can calculate $P(S|M)$ via Bayes' theorem:

$$P(S|M) = \frac{P(M|S)P(S)}{P(M|S)P(S) + P(M|S^c)P(S^c)} = 0.73$$

```
(0.92 * 0.08)/((0.92)*(0.08) + (0.03)*(0.92))
```

```
## [1] 0.7272727
```

Problem 2

```
# Making the data matrix

vals <- c(0.018,0.035,0.031,0.008,0.018,0.002,0.112,0.064,0.032,0.069,0.001,0.066,0.094,

joint <- matrix(vals, nrow = 5, byrow=TRUE, dimnames = list(c("farm", "operatives","craf
joint
```

```
##                farm operatives craftsmen sales professional
## farm          0.018      0.035     0.031 0.008        0.018
## operatives    0.002      0.112     0.064 0.032        0.069
## craftsmen     0.001      0.066     0.094 0.032        0.084
## sales         0.001      0.018     0.019 0.010        0.051
## professional  0.001      0.029     0.032 0.043        0.130
```

a) Let F be the father's occupation and S the son's. Then the marginal probability distribution of the father's occupation is given by $P(F = f_i) = \sum_{s_i \in S} P(F = f_i, S = s_i)$. The marginal probability distribution of the father's occupation:

```
mfather <- numeric(length = 5)
rvarb <- c("farm", "operatives","craftsmen","sales","professional")
for(i in 1:5)
{
  mfather[i] = sum(joint[i,])
}
names(mfather) <- rvarb
mfather
```

```
##         farm   operatives    craftsmen        sales professional
##        0.110        0.279        0.277        0.099        0.235
```

b) The marginal probability distribution of the son's occupation:

```
mson <- numeric(length = 5)
rvarb <- c("farm", "operatives","craftsmen","sales","professional")
for(i in 1:5)
{
  mson[i] = sum(joint[,i])
}
names(mson) <- rvarb
mson
```

```
##         farm   operatives    craftsmen        sales professional
##        0.023        0.260        0.240        0.125        0.352
```

c) The conditional distribution $P(S = s_i \,|\, F = f_i)$ is given as

$$P(S = s_i \,|\, F = f_i) = \frac{P(S = s_i, F = f_i)}{P(F = f_i)}$$

Hence the conditional distribution of the son's occupation given that the father is a farmer is as follows:

```
cson <- joint["farm",]/mfather["farm"]
cson
```

```
##         farm   operatives    craftsmen        sales professional
##   0.16363636   0.31818182   0.28181818   0.07272727   0.16363636
```

d) Similarly, the father's conditional distribution given that the son is a farmer is as follows:

```
cfather <- joint[,"farm"]/mson["farm"]
cfather
```

```
##         farm   operatives    craftsmen        sales professional
##   0.78260870   0.08695652   0.04347826   0.04347826   0.04347826
```

Problem 3

Set Y $N(\mu, \sigma)$ and $Z = \frac{X-\mu}{\sigma}$. Then we consider the cumulative distribution function $F_Z(z)$, which is given as follows:

$$F_Z(z) = P(Z \leq z) = P(\frac{X-\mu}{\sigma} \leq z) = P(X \leq \sigma z + \mu) = \int_{-\infty}^{\sigma z+\mu} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx$$

Then we know that the density function $f_Z(z) = F_z'(z)$ so we have that

$$f_Z(Z) = \frac{d}{dz}\int_{-\infty}^{\sigma z+\mu} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx = \frac{\sigma}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\sigma z+\mu-\mu}{\sigma}\right)^2} dx = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}z^2} dx$$

Thus Z has a Gaussian density function with $\mu = 0$ and $\sigma = 1$ and so Z $N(0, 1)$

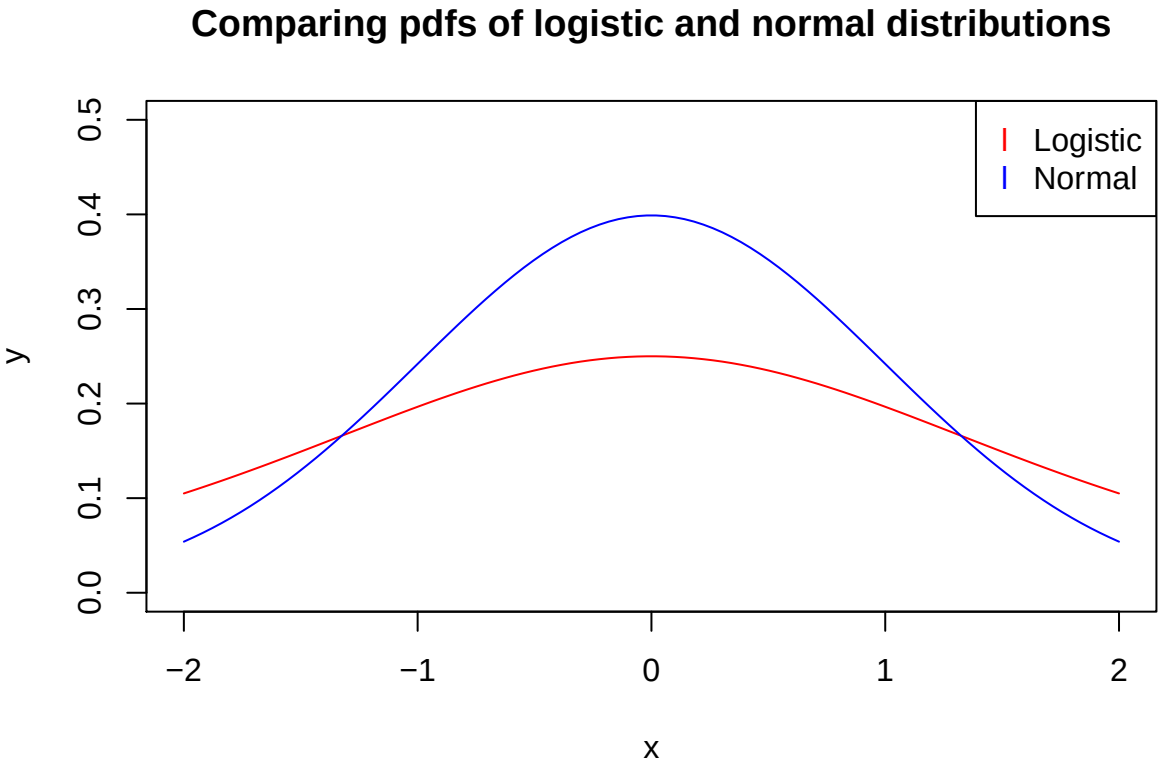
Problem 4

From the conditional independence formula we have that

$$\begin{aligned}
&P(X_1 = x_1, \ldots, Y_m = y_m | \mu_X, \sigma_X, \mu_Y, \sigma_Y) \\
&= \prod_{i=1}^{n} P(X_i = x_i | \mu_X, \sigma_X, \mu_Y, \sigma_Y) \prod_{i=1}^{m} P(Y_i = y_i | \mu_X, \sigma_X, \mu_Y, \sigma_Y) \\
&= \prod_{i=1}^{n} P(X_i = x_i | \mu_X, \sigma_X) \prod_{i=1}^{m} P(Y_i = y_i | \mu_Y, \sigma_Y) \\
&= \frac{1}{\sigma_X^n (2\pi)^{n/2}} e^{-\frac{1}{2}\frac{\sum_{i=1}^{n}(x_i-\mu_X)^2}{\sigma_X^2}} \frac{1}{\sigma_Y^n (2\pi)^{m/2}} e^{-\frac{1}{2}\frac{\sum_{i=1}^{m}(y_i-\mu_Y)^2}{\sigma_Y^2}} \\
&= \frac{1}{\sigma_X^n \sigma_Y^m (2\pi)^{(n+m)/2}} e^{-\frac{1}{2}\left(\frac{\sum_{i=1}^{n}(x_i-\mu_X)^2}{\sigma_X^2} + \frac{\sum_{i=1}^{m}(y_i-\mu_Y)^2}{\sigma_Y^2}\right)}
\end{aligned}$$

Problem 5

Let $F(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$. Then $f(x) = F'(x) = \frac{e^{-x}}{(1+e^{-x})^2}$



The logistic distribution is more likely to sample extreme values as the probabilities at the extremes are higher than those given by the normal distribution.

Problem 6

Here we calculate the posterior distribution given $n = 20$ and $y = 8$:

```
require(knitr)
```

```
## Loading required package: knitr
```

```
#Entering data and the prior probabilities
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

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

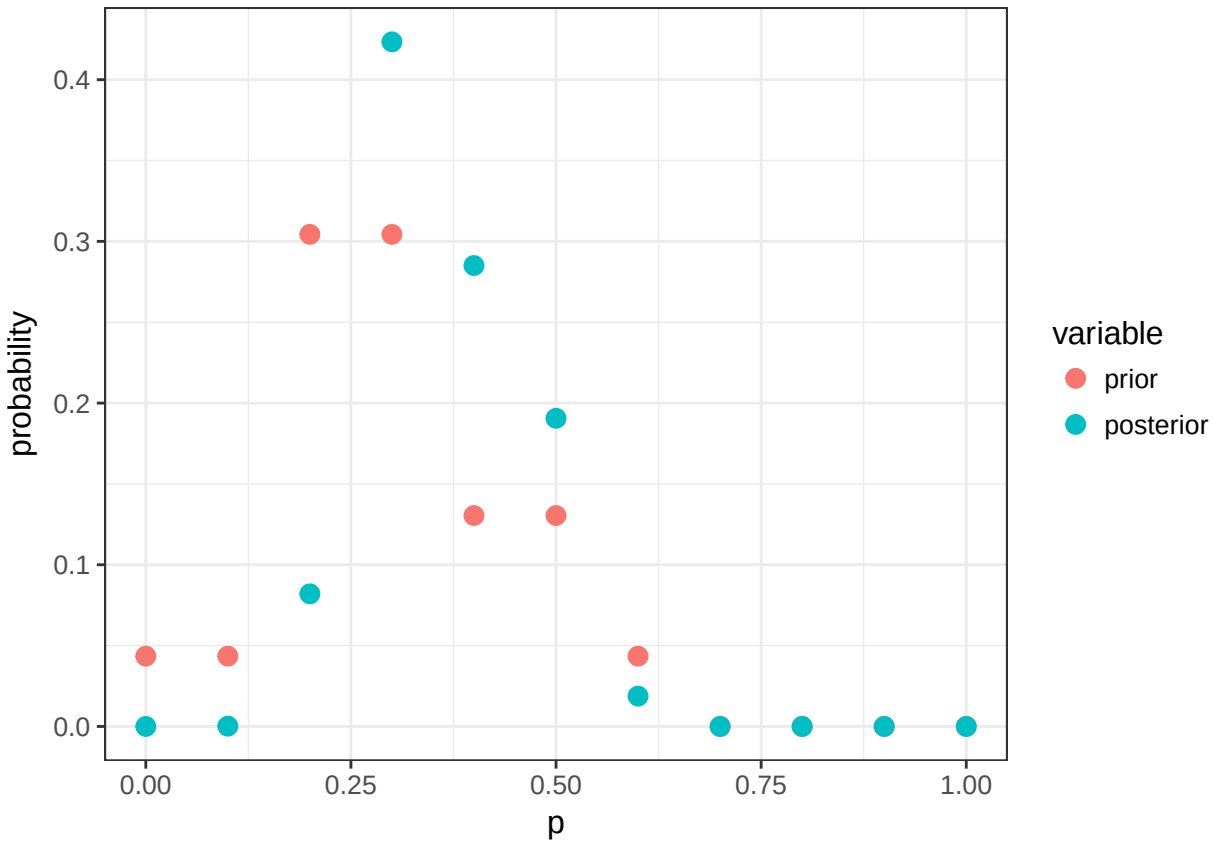
```r
for(i in 1:length(priorvalues))
{

  #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

#visualizing the posterior

#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

#plot the prior and posterior probabilities
require(ggplot2)
```

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

```r
require(reshape2)
```

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

```r
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 = "")
```




```
kable(allnighterposterior, caption="Posterior using one-step updating")
```

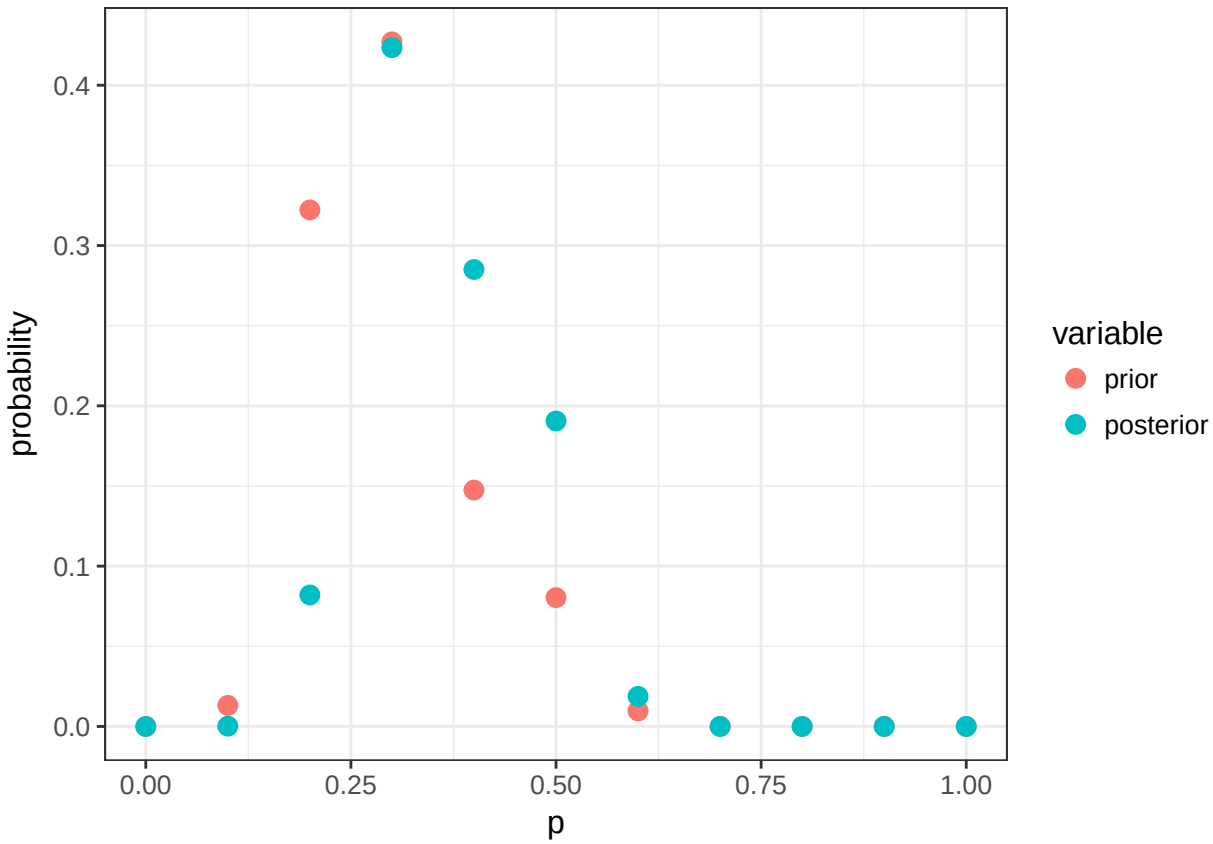
Table 1: Posterior using one-step updating

p	prior	posterior
0.0	0.0434783	0.0000000
0.1	0.0434783	0.0001881
0.2	0.3043478	0.0820219
0.3	0.3043478	0.4234056
0.4	0.1304348	0.2850547
0.5	0.1304348	0.1905607
0.6	0.0434783	0.0187690
0.7	0.0000000	0.0000000
0.8	0.0000000	0.0000000
0.9	0.0000000	0.0000000
1.0	0.0000000	0.0000000

In the following code, we assign the posterior calculated in class as prior and compute the posterior for $n = 10$ and $y = 5$.

```r
#Calculating posterior (n=10, y=3) from the prior distribution given in class
require(knitr)
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


#Now we will do a sequential update by settting the prior probabilities to the compute

priorprob <- posteriorprob
n <- 10
y <- 5


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])
```

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

}

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

#compute posterior probabilities
posteriorprob <- jointprob/pofy
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

#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 = "")
```




```
kable(allnighterposterior, caption="Posterior using sequential updating")
```

Table 2: Posterior using sequential updating

p	prior	posterior
0.0	0.0000000	0.0000000
0.1	0.0131236	0.0001881
0.2	0.3222345	0.0820219
0.3	0.4270731	0.4234056
0.4	0.1474735	0.2850547
0.5	0.0803851	0.1905607
0.6	0.0097102	0.0187690
0.7	0.0000000	0.0000000
0.8	0.0000000	0.0000000
0.9	0.0000000	0.0000000
1.0	0.0000000	0.0000000

Thus we see that the posterior with the bigger data set calculated in one update is the same as the posterior calculated by sequential updates of 10 at a time. This is because we assume that the trials are independent so the fact that out of 10 people 3 stayed up last year does not affect the fact that out of 10 other people, 5 stayed up last year.