Lab 6

PSTAT 115, Fall 2020

November 14, 2020

This lab will focus on the following topics:

- Bayes estimators
- Single-parameter normal-normal model
- Grid approximation to the posterior distribution

Bayes estimators

Loss Function:

$$L(\theta, \hat{\theta}) \geq 0$$

squared loss:

$$L(\theta, \hat{\theta}) = (\theta - \hat{\theta})^2$$

absolute error loss:

$$L(\theta, \hat{\theta}) = |\theta - \hat{\theta}|$$

Risk function of an estimator $\hat{\theta}(X)$ is:

$$R(\theta, \hat{\theta}) = E_\theta[L(\theta, \hat{\theta})] = \int L(\theta, \hat{\theta}) f(X|\theta) dX$$

Bayes Risk:

$$R = \int R(\theta, \hat{\theta}) \pi(\theta) d\theta$$

We want to minimize it

$$\begin{aligned}
R &= \int R(\theta, \hat{\theta}(X)) \pi(\theta) d\theta \\
&= \int (\int L(\theta, \hat{\theta}(X)) f(X|\theta) dX) \pi(\theta) d\theta \\
&= \int \int L(\theta, \hat{\theta}(X)) f(X|\theta) \pi(\theta) dX d\theta \\
&= \int \int L(\theta, \hat{\theta}(X)) f(X) \pi(\theta|X) d\theta dX \\
&= \int (\int L(\theta, \hat{\theta}(X)) \pi(\theta|X) d\theta) f(X) dX \\
&= \int r(\hat{\theta}) f(X) dX
\end{aligned}$$

If we use squared loss, then the Bayes estimator is the posterior mean.

Review of single-parameter normal-normal model

Unknown μ, known σ^2

Prior:

$$\mu \sim N\left(\mu_0, \frac{\sigma^2}{\kappa_0}\right)$$

Likelihood:

$$Y_i \sim N\left(\mu, \sigma^2\right)$$

Posterior:

$$\mu | y_1, \dots y_n \sim N\left(\mu_n, \frac{\sigma^2}{\kappa_n}\right)$$

$$\text{where } \kappa_n = \kappa_0 + n \text{ and } \mu_n = \frac{\left(\kappa_0/\sigma^2\right)\mu_0 + \left(n/\sigma^2\right)\overline{y}}{\kappa_0/\sigma^2 + n/\sigma^2} = \frac{\kappa_0\mu_0 + n\overline{y}}{\kappa_n}$$

Discussion:

What is the meaning of κ_0 (consider pseudo-counts)?

How does the posterior parameters κ_n and μ_n relate to the prior parameters κ_0 and μ_0?

How would you interpret κ_n and μ_n?

Bias and variance of posterior mean:

$E\hat{\mu} = E\mu_n = \frac{\kappa_0\mu_0 + nE\overline{y}}{\kappa_n} = \frac{\kappa_0\mu_0 + n\mu}{\kappa_0 + n}$. Bias is $E\hat{\mu} - \mu = \frac{\kappa_0\mu_0 - \kappa_0\mu}{\kappa_0 + n}$

$var(\hat{\mu}) = \frac{n^2 var(\overline{y})}{\kappa_n^2} = \frac{n\sigma^2}{(\kappa_0 + n)^2}$

Grid approximation to the posterior distribution

First of all we set up the parameters based on the above formulas.

```
## prior mean for mu and prior counts for mu
mu0 <- 1.9
k0 <- 1

## sufficient statistics are sample mean and sample variance
y <- c(1.64, 1.7, 1.72, 1.74, 1.82, 1.82, 1.82, 1.9, 2.08)
n <- length(y)
ybar <- mean(y)

## posterior parameters, see the formula above
kn <- k0 + n
mun <- (k0 * mu0 + n * ybar) / kn
```

Now we need to learn several basic functions to facilitate our plots.

(1) The `expand.grid()` function creates a grid based on two vectors by justaposing a pair of values. The value from the first vector changes first, then the value from the second vector will change.

```
## example for expand.grid
grid <- as_tibble(expand.grid(seq(1, 3, by = 1), seq(0, 2, by=1)))
colnames(grid) <- c('x', 'y')
grid
```

```
## # A tibble: 9 x 2
##       x     y
##   <dbl> <dbl>
## 1     1     0
## 2     2     0
## 3     3     0
## 4     1     1
## 5     2     1
## 6     3     1
## 7     1     2
## 8     2     2
## 9     3     2
```

(2) The `Vectorize()` function is a wrapper for an arbitrary function. The resulting function can be applied to each row for a tibble object.

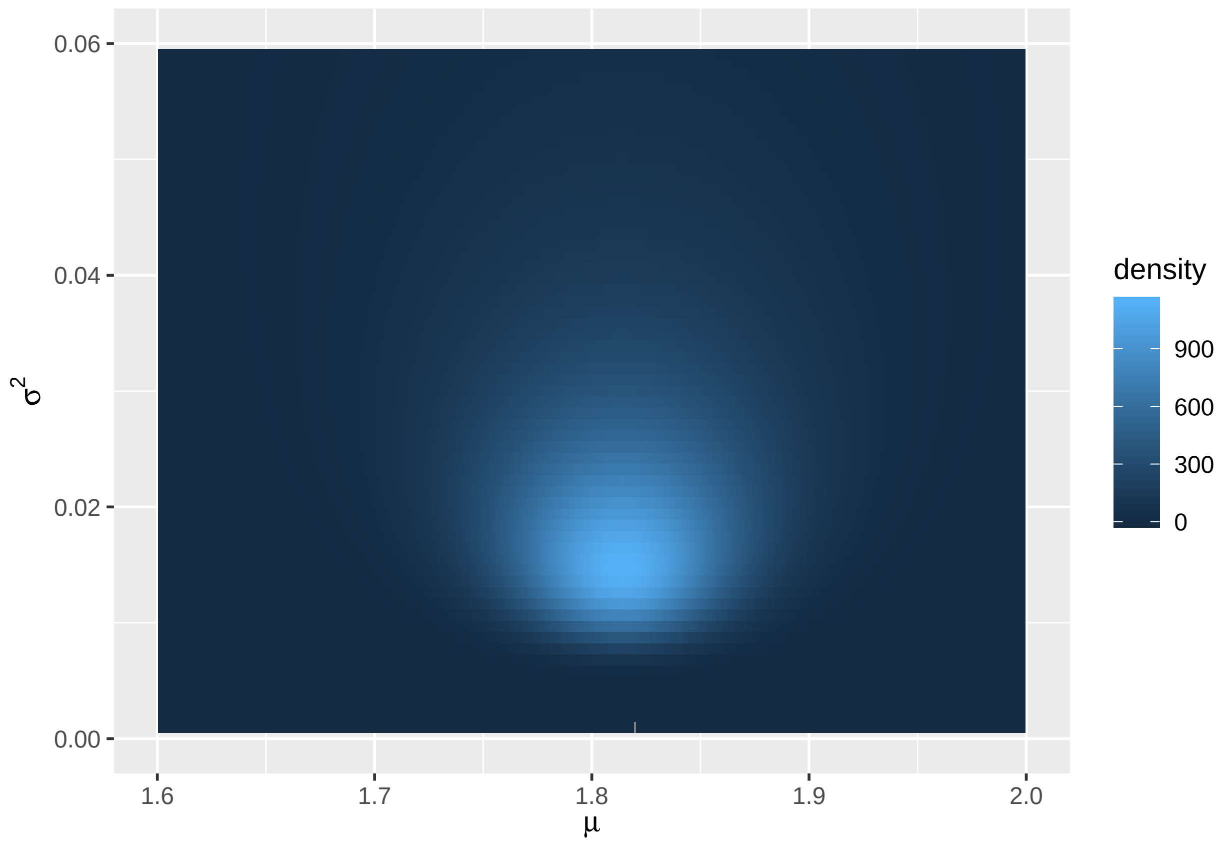
```
## example of Vectorize()
sum <- Vectorize(function (a, b) {
  a + b
})

grid %>% mutate("Sum" = sum(x, y))
```

```
## # A tibble: 9 x 3
##       x     y   Sum
##   <dbl> <dbl> <dbl>
## 1     1     0     1
## 2     2     0     2
## 3     3     0     3
## 4     1     1     2
## 5     2     1     3
## 6     3     1     4
## 7     1     2     3
## 8     2     2     4
## 9     3     2     5
```

Armed with the above new functions, let us go about plotting the posterior joint density.

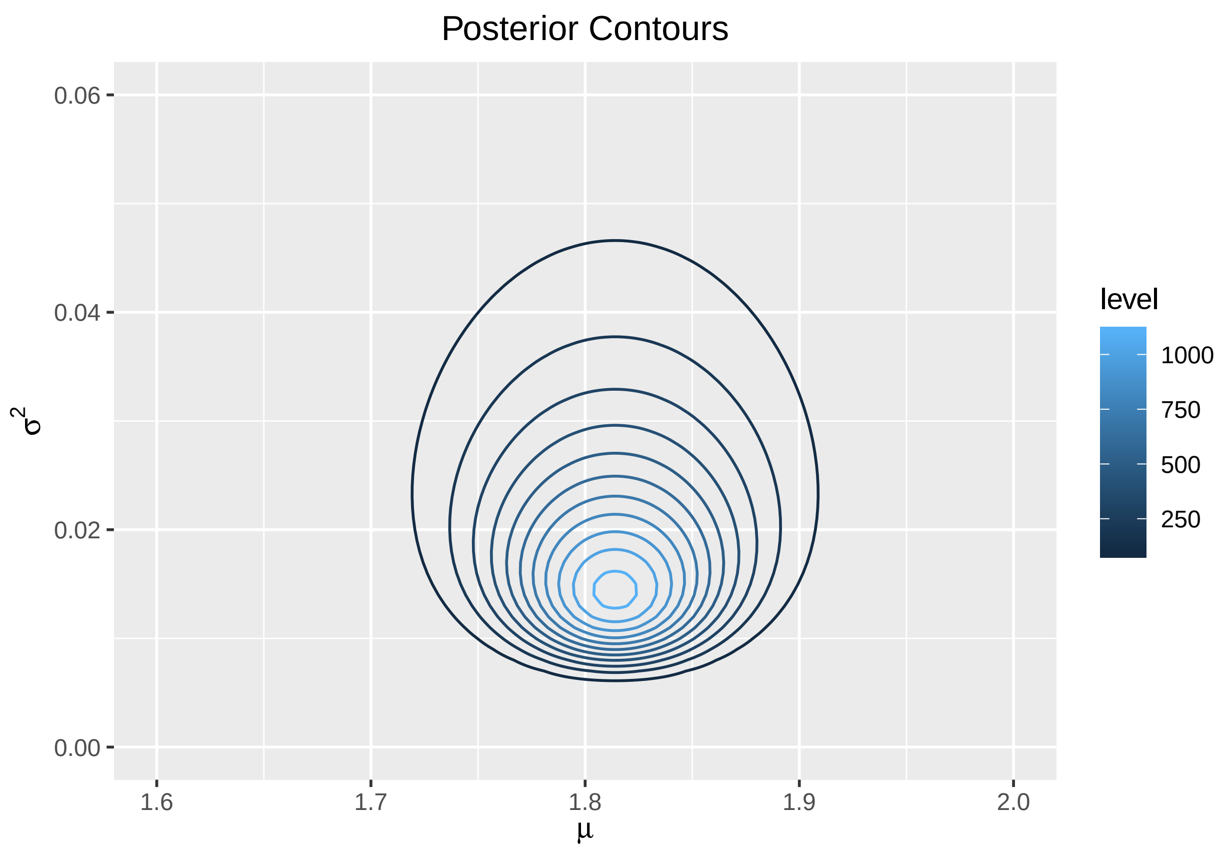
```
## create the grid on which the posterior joint distribution we are interested in
grid <- as_tibble(expand.grid(seq(1.6, 2.0, by=0.001), seq(0, 0.06, by=0.001)))
colnames(grid) <- c("mu", "s2")

## create the wrapped function to be applied to each row in the tibble
normal_posterior <- Vectorize(function(mu, sigma2) {

  ## likelihood times prior
```

```
  prod(dnorm(y, mu, sqrt(sigma2))) *
    dnorm(mu, mu0, sqrt(sigma2/k0))
})

## applied the wrapped function to each row of grid and
## plot the density using the geom_raster function,
## which fill each location based on the corresponding density value

grid %>% mutate(density = normal_posterior(mu, s2)) %>% ggplot() +
  geom_raster(aes(mu, s2, fill=density)) +
  xlim(c(1.6, 2)) + ylim(c(0, 0.06)) +
  xlab(expression(mu)) + ylab(expression(sigma^2))
```



Contour Plot

```
grid %>% mutate(density = normal_posterior(mu, s2)) %>%
  ggplot() + geom_contour(aes(mu, s2, z=density, colour=stat(level))) +
  xlim(c(1.6, 2)) + ylim(c(0, 0.06)) +
  xlab(expression(mu)) + ylab(expression(sigma^2)) +
  ggtitle("Posterior Contours") +
  theme(plot.title = element_text(hjust = 0.5))
```



Posterior Contours
0.06
0.04
0.02
0.00
σ²
1.6
1.7
1.8
1.9
2.0
μ
level
1000
750
500
250