BIOST 527 Homework 5

Benjamin Stan

June 7, 2021

Question 1

(a)

For B random variables with pairwise correlation $\rho \geq 0$, the variance of the mean of the variables can be expressed

$$\begin{aligned}
Var\left(\frac{1}{B}\sum_{i=1}^{B} x_i\right) &= \frac{1}{B^2} Var\left(\sum_{i=1}^{B} x_i\right) \\
&= \frac{1}{B^2}\left[\sum_{i=1}^{B} Var(x_i) + \sum_{i=1}^{B}\sum_{j\neq i}^{B} Cov(x_i, x_j)\right] \\
&= \frac{1}{B^2}\left[B\sigma^2 + (B^2 - B)\rho\sigma^2\right] \\
&= \frac{1}{B^2}\left[B\sigma^2 + B(B-1)\rho\sigma^2\right] \\
&= \frac{\sigma^2}{B} + \frac{B-1}{B}\rho\sigma^2 \\
&= \frac{\sigma^2}{B} + \frac{B\rho\sigma^2}{B} - \frac{\rho\sigma^2}{B} \\
&= \frac{\sigma^2}{B} + \rho\sigma^2 - \frac{\rho\sigma^2}{B} \\
&= \rho\sigma^2 + \sigma^2\frac{(1-\rho)}{B}
\end{aligned}$$

Question 2

(a)

When considering the variance of the first realization of the bootstrap mean:

$$Var(\bar{x_1^*}) = E[Var(\bar{x_1}^*|x)] + Var[E(\bar{x_1}^*|x)]$$

Consider the second term:

$$Var[E(\bar{x_1}^*|x)] = Var(\bar{x}) = \frac{\sigma^2}{N}$$

Consider the first term:

$$\begin{aligned}
E[Var(\bar{x_1}^*|x)] &= E\left[Var\left(\frac{1}{N}\sum_{i=1}^{N} x_{1i}^*|x\right)\right] \\
&= E\left[\frac{1}{N^2}\sum_{i=1}^{N} Var(x_{1i}^*|x)\right] \\
&= E\left[\frac{1}{N}Var(x_{1i}^*|x)\right] \\
&= E\left[\frac{1}{N}E\left((x_{1i}^* - E(x_{1i}^*))^2|x\right)\right] \\
&= E\left[\frac{1}{N}E\left((x_{1i}^* - \bar{x})^2|x\right)\right] \\
&= E\left[\frac{1}{N}E\left((x_{1i}^{*2} - 2x_{1i}^*\bar{x} - \bar{x}^2)|x\right)\right] \\
&= E\left[\frac{1}{N}\left(E(x_{1i}^{*2}|x) - E(2x_{1i}^*\bar{x}|x) - E(\bar{x}^2|x)\right)\right] \\
&= E\left[\frac{1}{N}\left(E(x_{1i}^{*2}|x) - 2\bar{x}^2 + \bar{x}^2\right)\right] \\
&= E\left[\frac{1}{N}\left(E(x_{1i}^{*2}|x) - \bar{x}^2\right)\right] \\
&= E\left[\frac{1}{N}\left(\frac{1}{N}\sum_{i=1}^{N} x_i^2 - \bar{x}^2\right)\right] \\
&= E\left[\frac{1}{N^2}\sum_{i=1}^{N}(x_i - \bar{x})^2\right] \\
&= \frac{N-1}{N^2}\sigma^2
\end{aligned}$$

Combining the terms:

$$\begin{aligned}
Var(\bar{x_1}^*) &= E[Var(\bar{x_1}^*|x)] + Var[E(\bar{x_1}^*|x)] \\
&= \frac{N-1}{N^2}\sigma^2 + \frac{\sigma^2}{N} \\
&= \frac{\sigma^2}{N}\left(\frac{N-1}{N} + 1\right) \\
&= \frac{\sigma^2}{N}\left(\frac{2N-1}{N}\right)
\end{aligned}$$

(b)

When considering the covariance of the means of two bootstrap realizations:

$$Cov(\bar{x_1}^*, \bar{x_2}^*) = E[Cov(\bar{x_1}^*, \bar{x_2}^*|x)] + Cov[E(\bar{x_1}^*|x), E(\bar{x_2}^*|x)]$$

Consider the second term:

$$Cov[E(\bar{x_1}^*|x), E(\bar{x_2}^*|x)] = Cov(\bar{x}, \bar{x}) = Var(\bar{x}) = \frac{\sigma^2}{N}$$

Consider the first term:

$$\begin{aligned} E[Cov(\bar{x_1}^*, \bar{x_2}^*|x)] &= E(\bar{x_1}^*\bar{x_2}^*|x) - E(\bar{x_1}^*|x)E(\bar{x_2}^*|x) \\ &= E(\bar{x_1}^*\bar{x_2}^*|x) - \bar{x}^2 \\ &= E(\bar{x_1}^*|x)E(\bar{x_2}^*|x) - \bar{x}^2 \\ &= \bar{x}^2 - \bar{x}^2 \\ &= 0 \end{aligned}$$

Combining the terms:

$$Cov(\bar{x_1}^*, \bar{x_2}^*) = \frac{\sigma^2}{N}$$

(c)

$$\begin{aligned} Cor(\bar{x_1}^*, \bar{x_2}^*) &= \frac{Cov(\bar{x_1}^*, \bar{x_2}^*)}{\sqrt{Var(\bar{x_1}^*)Var(\bar{x_2}^*)}} \\ &= \frac{\frac{\sigma^2}{N}}{\frac{\sigma^2}{N}\left(\frac{2N-1}{N}\right)} \\ &= \frac{N}{2N-1} \end{aligned}$$

This value $\approx 1/2$ when N is large.

(d)

When considering the means of the means of the bootstrap realizations:

$$\begin{aligned} Var(\bar{x}_{bag}) &= Var\left[\frac{1}{B}\sum_{b=1}^{B}\bar{x}_b^*\right] \\ &= Cor(\bar{x_1}^*, \bar{x_2}^*)Var(\bar{x_1}^*) - \frac{1 - Cor(\bar{x_1}^*, \bar{x_2}^*)}{B}Var(\bar{x_1}^*) \\ &= \left(\frac{N}{2N-1}\right)\frac{\sigma^2}{N}\left(\frac{2N-1}{N}\right) + \left(\frac{1 - \frac{N}{2N-1}}{B}\right)\frac{\sigma^2}{N}\left(\frac{2N-1}{N}\right) \\ &= \frac{\sigma^2}{N} + \left(\frac{\sigma^2}{NB}\right)\left(\frac{2N-1-N}{2N-1}\right)\left(\frac{2N-1}{N}\right) \\ &= \frac{\sigma^2}{N} + \frac{\sigma^2}{N}\left(\frac{N-1}{NB}\right) \\ &= \frac{\sigma^2}{N}\left[1 + \frac{N-1}{NB}\right] \end{aligned}$$

(e)

$$\begin{aligned} Var(\bar{x}) &= Var\left(\frac{1}{N}\sum_{i=1}^{N} x_i\right) \\ &= \frac{1}{N^2} Var\left(\sum_{i=1}^{N} x_i\right) \\ &= \frac{1}{N} Var(x_i) \\ &= \frac{\sigma^2}{N} \end{aligned}$$

(f)

The answers to parts (d) and (e) can be used to explain how bagging produces no reduction in variance in estimating the mean. The result from part (e) computed the variance in the sample mean, $\bar{x}$, which was $\frac{\sigma^2}{N}$. Compare this to the variance of the bagged mean estimate, $\bar{x}_{bag}$, which was $\frac{\sigma^2}{N}\left[1 + \frac{N-1}{NB}\right]$, and it can be concluded that the sample mean variance is necessarily smaller due to N and B being positive values.

Question 3

To verify the results from parts (a) through (e) of Question 2, we will use a simulation performed 1000 times. In each simulation, an X dataset will be generated with 1000 samples (N) from a N(1,2) distribution. 100 bootstrap means will then be computed for each simulation by taking the mean of N samples from X with replacement.

```
#####
# Q3
#####
## Define parameters
set.seed(124)
niter = 1000
N = 1000
B = 100
bs_means = matrix(data = NA,
                  nrow = B,
                  ncol = niter)
X_bar = NULL
## Perform simulation
for(i in 1:niter) {
  X = rnorm(N, mean = 1, sd = sqrt(2))
  X_bar = c(X_bar,mean(X))
  bs_means_i = unlist(map(1:B,function(z) mean(sample(X,N,replace = T))))
  bs_means[,i] = bs_means_i
}

## Verify answer to 2(a)
answer_a = 2/N*(2*N-1)/N
sim_answer_a = var(bs_means[1,])
```

```
## Verify answer to 2(b)
answer_b = 2/N
sim_answer_b = cov(bs_means[1,], bs_means[2,])

## Verify answer to 2(c)
answer_c = N/(2*N-1)
sim_answer_c = cor(bs_means[1,], bs_means[2,])

## Verify answer to 2(d)
answer_d = (2/N)*(1+(N-1)/(N*B))
sim_answer_d = var(apply(bs_means, 2, mean))

## Verify answer to 2(e)
answer_e = 2/N
sim_answer_e = var(X_bar)
```

The quantities computed using the formulas from part (a) through (e) of Question 2 and those computed using the simulation are compared in the table below:

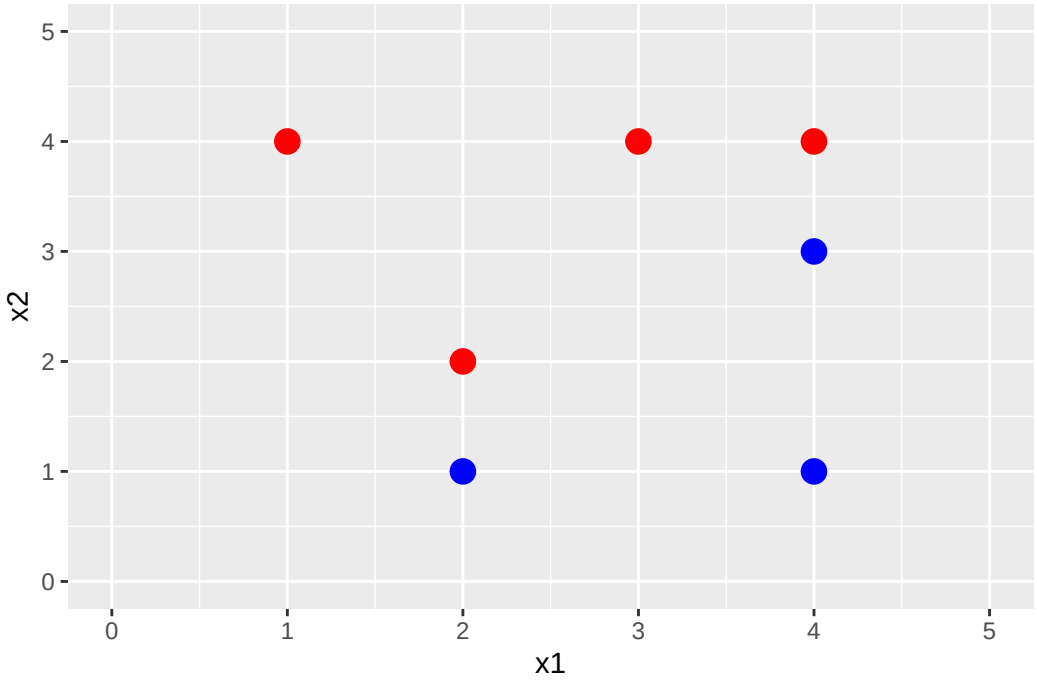
	Formula	Simulation
a	0.00400	0.00399
b	0.00200	0.00204
c	0.500	0.518
d	0.00202	0.00206
e	0.00200	0.00203

As can be seen by the table, the results from the formulas generated in Question 2 and the simulation match to approximately the third significant figure. Thus, the results from Question 2 can be considered verified.

Question 4

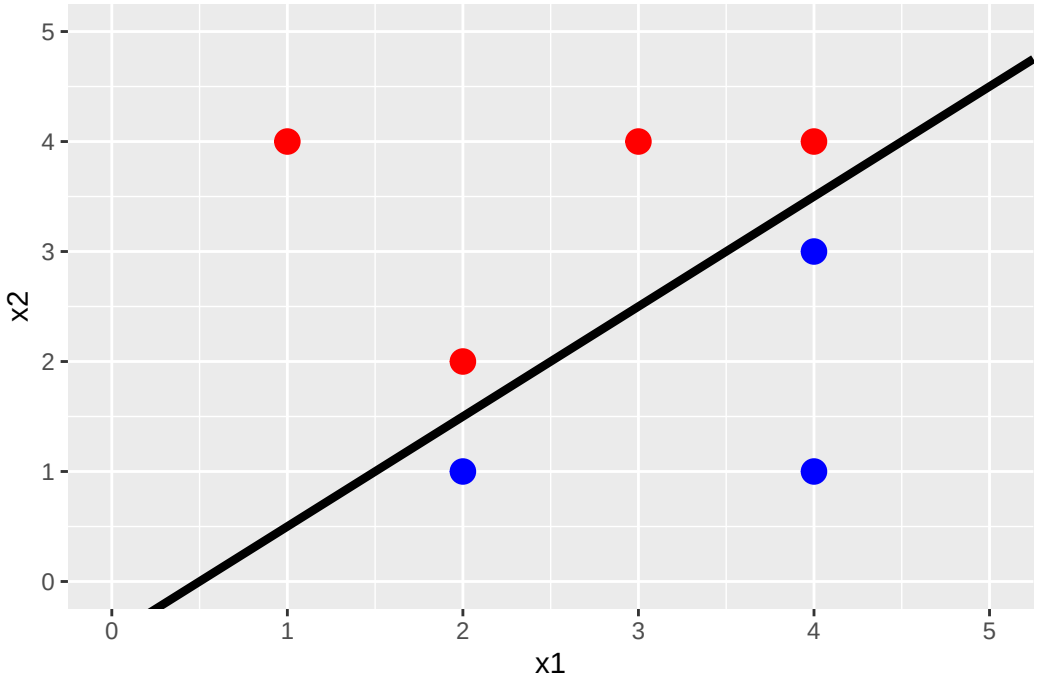
(a)

Below are the labeled observations from the dataset.



(b)

The optimal separating hyperplane is shown below; the equation for the hyperplane is $X_1 - X_2 - 0.5 = 0$.

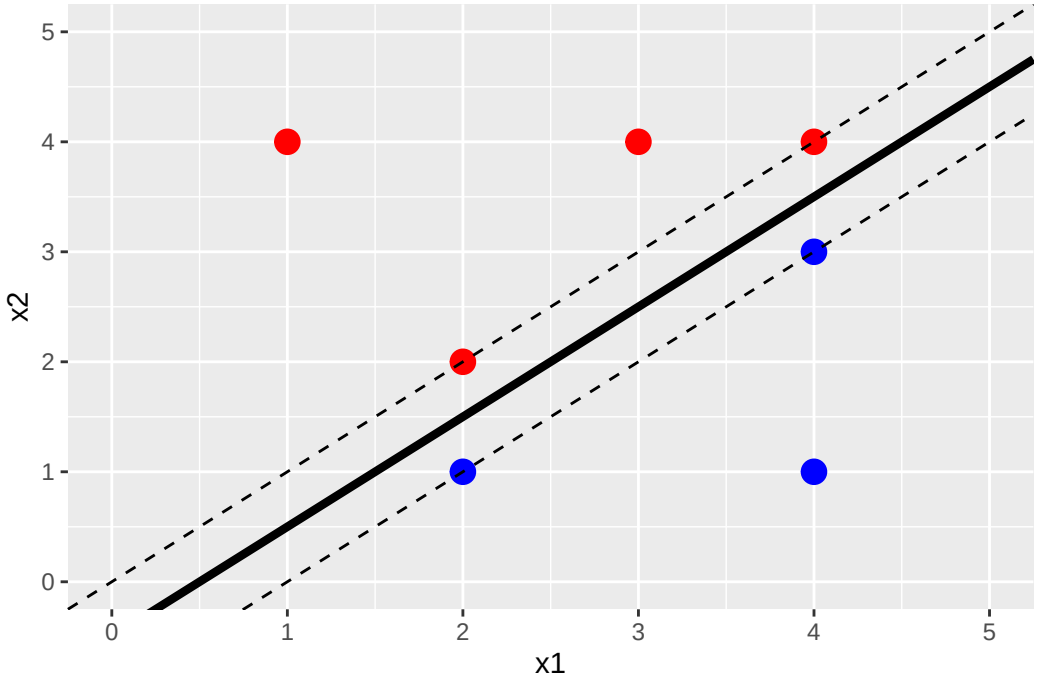


(c)

The classification rule for the maximal margin classifier is classify to Red if $X_1 - X_2 - 0.5 < 0$, and classify to Blue otherwise.

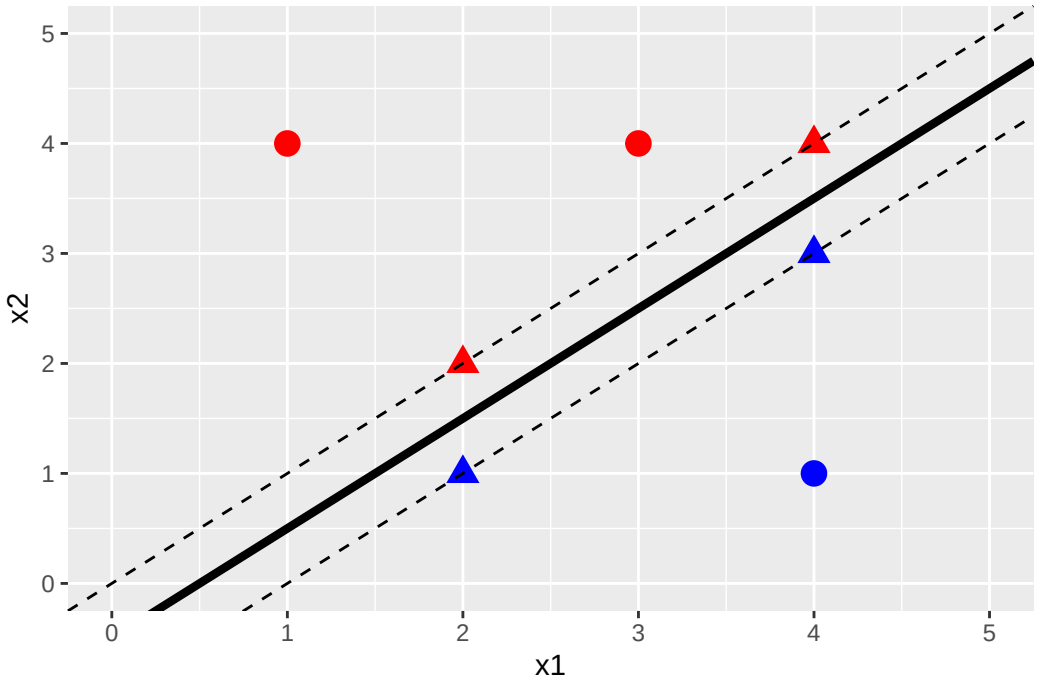
(d)

The margin, M, is the perpendicular distance from either of the dashed lines to the maximal separating hyperplane in the graph below.



(e)

The support vectors are indicated by the triangles in the graph below. They are points (4,3), (4,4), (2,1), and (2,2).

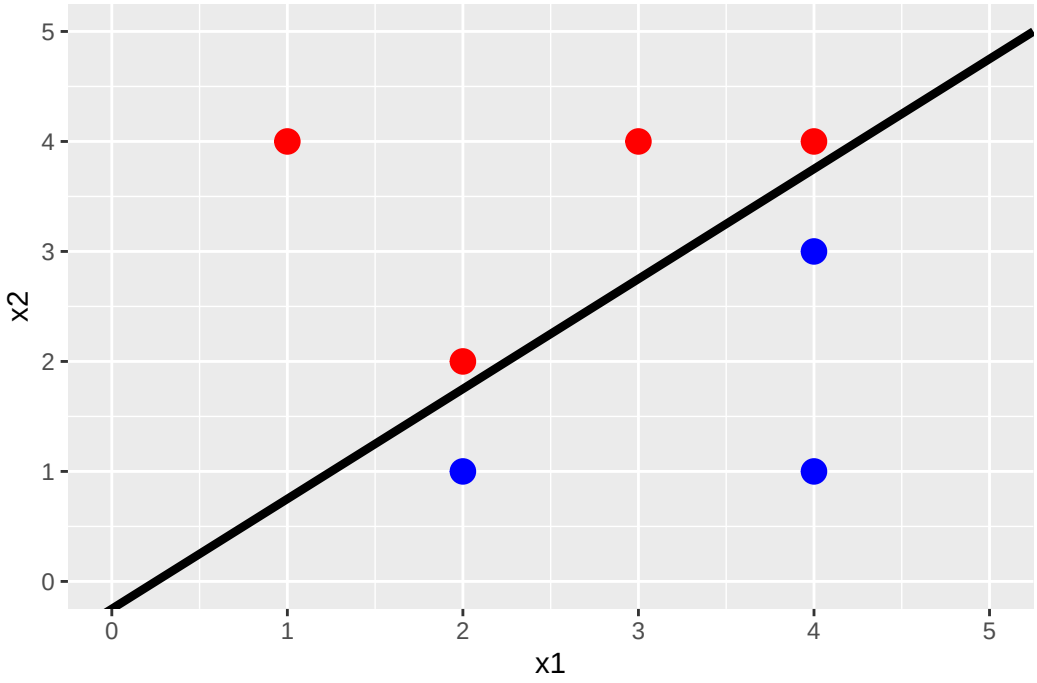


(f)

A slight movement of the seventh point (4,1) would not affect the maximal margin hyperplane because this is not one of the support vectors, and a slight movement would not cause it to become a support vector.

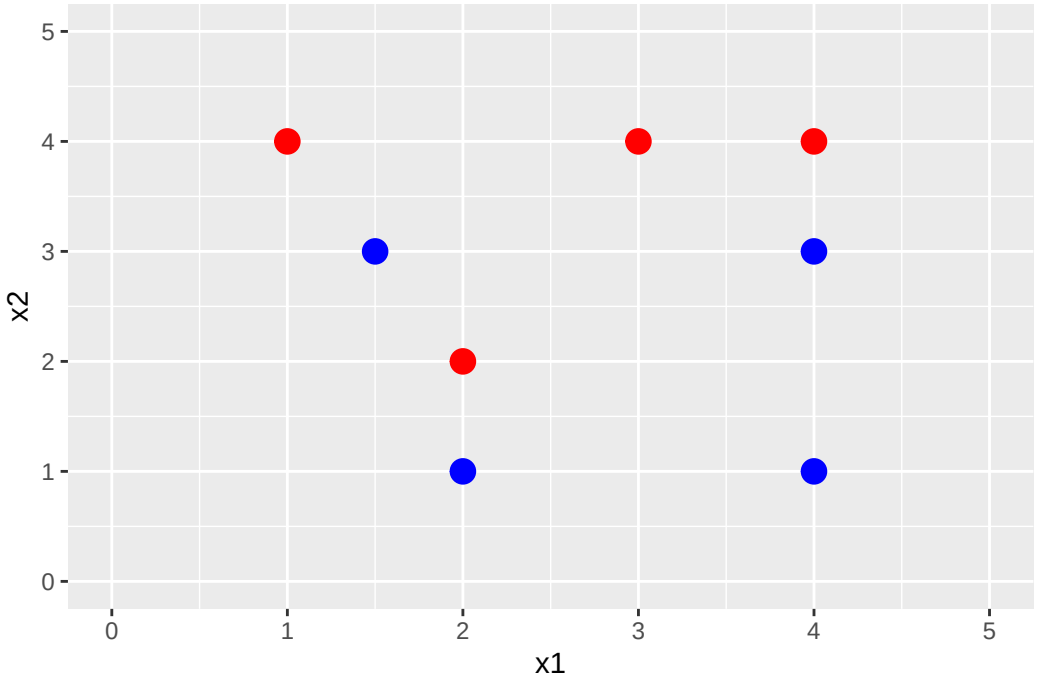
(g)

A separating hyperplane that is not the optimal separating hyperplane is shown below. The equation for the hyperplane is $X_1 - X_2 - 0.25 = 0$.



(h)

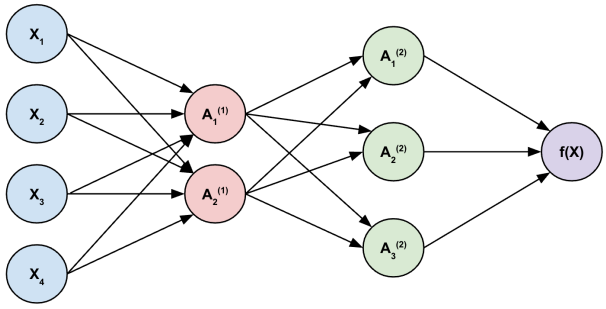
As shown below, the addition of blue point (1.5,3) makes it so the two classes are no longer separable by a hyperplane.



Question 5

(a)

Below is a diagram of the neural network:



(b)

For this neural network, we will consider the regression setting. The function $f(X)$ takes the form

$$f(X) = \beta_0 + \sum_{l=1}^{3} \beta_l A_l^{(2)}$$

with

$$A_l^{(2)} = \left(w_{l0}^{(2)} + \sum_{k=1}^{2} w_{lk}^{(2)} A_k^{(1)} \right)_+$$
$$A_k^{(1)} = \left(w_{k0}^{(1)} + \sum_{j=1}^{4} w_{kj}^{(1)} X_j \right)_+$$

Combining these terms:

$$f(X) = \beta_0 + \sum_{l=1}^{3} \beta_l \left(w_{l0}^{(2)} + \sum_{k=1}^{2} w_{lk}^{(2)} \left(w_{k0}^{(1)} + \sum_{j=1}^{4} w_{kj}^{(1)} X_j \right)_+ \right)_+$$

(c)

We will use the following coefficient matrices:

$$\beta = \begin{bmatrix} 1 \\ 1.25 \\ -1 \\ 1.5 \end{bmatrix}$$
$$w_1 = \begin{bmatrix} -1 & 1.5 & -1 & 0.5 & 2.5 \\ 2 & -0.5 & 2 & 1 & 1.5 \end{bmatrix}$$
$$w_2 = \begin{bmatrix} 1 & -1 & 1.5 \\ 2 & 0.5 & -1 \\ 0.5 & 1 & -2 \end{bmatrix}$$

We can then compute the following terms:

$$\begin{aligned}
A_1^{(1)} &= (-1 + 1.5X_1 - X_2 + 0.5X_3 + 2.5X_4)_+ \\
A_2^{(1)} &= (2 - 0.5X_1 + 2X_2 + X_3 + 1.5X_4)_+ \\
A_1^{(2)} &= (1 - A_1^{(1)} + 1.5A_2^{(1)})_+ \\
&= \big(1 - (-1 + 1.5X_1 - X_2 + 0.5X_3 + 2.5X_4)_+ + 1.5(2 - 0.5X_1 + 2X_2 + X_3 + 1.5X_4)_+\big)_+ \\
A_2^{(2)} &= (2 + 0.5A_1^{(1)} - A_2^{(1)})_+ \\
&= \big(2 + 0.5(-1 + 1.5X_1 - X_2 + 0.5X_3 + 2.5X_4)_+ - (2 - 0.5X_1 + 2X_2 + X_3 + 1.5X_4)_+\big)_+ \\
A_3^{(2)} &= (0.5 + A_1^{(1)} - 2A_2^{(1)})_+ \\
&= \big(0.5 + (-1 + 1.5X_1 - X_2 + 0.5X_3 + 2.5X_4)_+ - 2(2 - 0.5X_1 + 2X_2 + X_3 + 1.5X_4)_+\big)_+
\end{aligned}$$

The value of $f(X)$ is

$$\begin{aligned}
f(X) &= 1 + 1.25A_1^{(2)} - A_2^{(2)} + 1.5A_3^{(2)} \\
&= 1 + 1.25\big(1 - (-1 + 1.5X_1 - X_2 + 0.5X_3 + 2.5X_4)_+ + 1.5(2 - 0.5X_1 + 2X_2 + X_3 + 1.5X_4)_+\big)_+ \\
&\quad - \big(2 + 0.5(-1 + 1.5X_1 - X_2 + 0.5X_3 + 2.5X_4)_+ - (2 - 0.5X_1 + 2X_2 + X_3 + 1.5X_4)_+\big)_+ \\
&\quad + 1.5\big(0.5 + (-1 + 1.5X_1 - X_2 + 0.5X_3 + 2.5X_4)_+ - 2(2 - 0.5X_1 + 2X_2 + X_3 + 1.5X_4)_+\big)_+
\end{aligned}$$

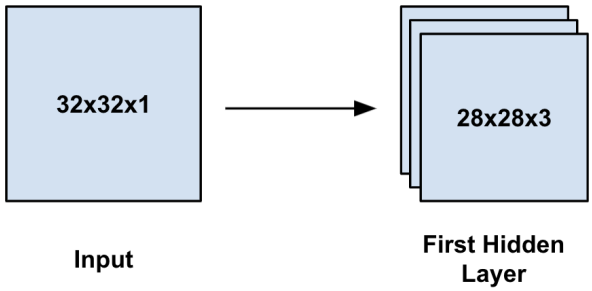
(d)

The total number of parameters is 23, resulting from the sum of parameters from β (4), w_1 (10), and w_2 (9).

Question 6

(a)

The diagram of the input and first hidden layer are shown below.



(b)

There are 5*5*3 = 75 parameters in this model, resulting from the dimensions of the filters and assuming no intercept term.

(c)

This model can be thought of as an ordinary feed-forward neural network by treating each of the the pixels of the input image as a feature and putting constraints on the weights of the hidden units (the convolutional filters). The hidden units in this case are each of the pixels of the convolved images, of which there are 28*28*3 = 2352. There are two constraints on these weights; the first is that the filter only operates on pieces of the original image, so each input pixel does not have a weight on each pixel of the hidden layer (or some of those weights can be thought of as zero). The second constraint is that the non-zero weights are used multiple times, as these weights correspond to the convolutional filter that is applied. As such, it is constrained relative to a typical feed-forward network, in which each pair of input and hidden layer units would have a unique weight.

(d)

If there were no constraints, the ordinary feed-forward neural network described in part (c) would have (32*32)(28*28*3) = 2,408,448 weights resulting from the dimensions of the input and hidden layers.

Appendix

```
## Load libraries and set working directory
setwd("/Users/bstan/Documents/UW/Courses/BIOST 527")
rm(list = ls())
library(tidyverse)
library(tidyr)
library(tinytex)
library(class)
library(MASS)
library(ISLR)
library(leaps)

#####
# Q3
#####
```

```r
## Define parameters
set.seed(124)
niter = 1000
N = 1000
B = 100
bs_means = matrix(data = NA,
                  nrow = B,
                  ncol = niter)
X_bar = NULL
## Perform simulation
for(i in 1:niter) {
  X = rnorm(N, mean = 1, sd = sqrt(2))
  X_bar = c(X_bar,mean(X))
  bs_means_i = unlist(map(1:B,function(z) mean(sample(X,N,replace = T))))
  bs_means[,i] = bs_means_i
}

## Verify answer to 2(a)
answer_a = 2/N*(2*N-1)/N
sim_answer_a = var(bs_means[1,])

## Verify answer to 2(b)
answer_b = 2/N
sim_answer_b = cov(bs_means[1,], bs_means[2,])

## Verify answer to 2(c)
answer_c = N/(2*N-1)
sim_answer_c = cor(bs_means[1,], bs_means[2,])

## Verify answer to 2(d)
answer_d = (2/N)*(1+(N-1)/(N*B))
sim_answer_d = var(apply(bs_means, 2, mean))

## Verify answer to 2(e)
answer_e = 2/N
sim_answer_e = var(X_bar)

#####
# Q4a
#####
toy_data = data.frame(
  x1 = c(3, 2, 4, 1, 2, 4, 4),
  x2 = c(4, 2, 4, 4, 1, 3, 1),
  y = c("red", "red", "red", "red", "blue", "blue", "blue")
)
g1 = ggplot(data = toy_data) +
      scale_color_manual(values=c("blue", "red")) +
      geom_point(aes(x=x1,y=x2,col=y),size=4) +
      xlim(0,5) +
      ylim(0,5) +
      theme(legend.position = "none")
g1
```

```
#####
# Q4b
#####
g1+geom_abline(slope = 1,intercept = -0.5, size=1.5)

#####
# Q4d
#####
g1 +
  geom_abline(slope = 1,intercept = -0.5, size=1.5) +
  geom_abline(slope = 1,intercept = 0, size=0.5, linetype='dashed') +
  geom_abline(slope = 1,intercept = -1, size=0.5, linetype='dashed')

#####
# Q4e
#####
toy_data = data.frame(
  x1 = c(3, 1, 4, 2, 4, 2, 4),
  x2 = c(4, 4, 1, 1, 3, 2, 4),
  y = c("red", "red", "blue", "blue", "blue", "red", "red")
)
g2 = ggplot(data = ) +
      scale_color_manual(values=c("blue", "red")) +
      geom_point(aes(x=toy_data$x1[1:3],y=toy_data$x2[1:3],col=toy_data$y[1:3]),size=4) +
      geom_point(aes(x=toy_data$x1[4:7],y=toy_data$x2[4:7],col=toy_data$y[4:7]),
                 shape=17,size=4) +
      xlab("x1") +
      ylab("x2") +
      xlim(0,5) +
      ylim(0,5)
g2 +
  geom_abline(slope = 1,intercept = -0.5, size=1.5) +
  geom_abline(slope = 1,intercept = 0, size=0.5, linetype='dashed') +
  geom_abline(slope = 1,intercept = -1, size=0.5, linetype='dashed') +
  theme(legend.position = "none")

#####
# Q4g
#####
g1+geom_abline(slope = 1,intercept = -0.25, size=1.5)

#####
# Q4h
#####
toy_data = data.frame(
  x1 = c(3, 2, 4, 1, 2, 4, 4, 1.5),
  x2 = c(4, 2, 4, 4, 1, 3, 1, 3),
  y = c("red", "red", "red", "red", "blue", "blue", "blue", "blue")
)
g3 = ggplot(data = toy_data) +
      scale_color_manual(values=c("blue", "red")) +
      geom_point(aes(x=x1,y=x2,col=y),size=4) +
      xlim(0,5) +
```

```
      ylim(0,5) +
      theme(legend.position = "none")
g3
```