Problem Solutions

e-Chapter 9

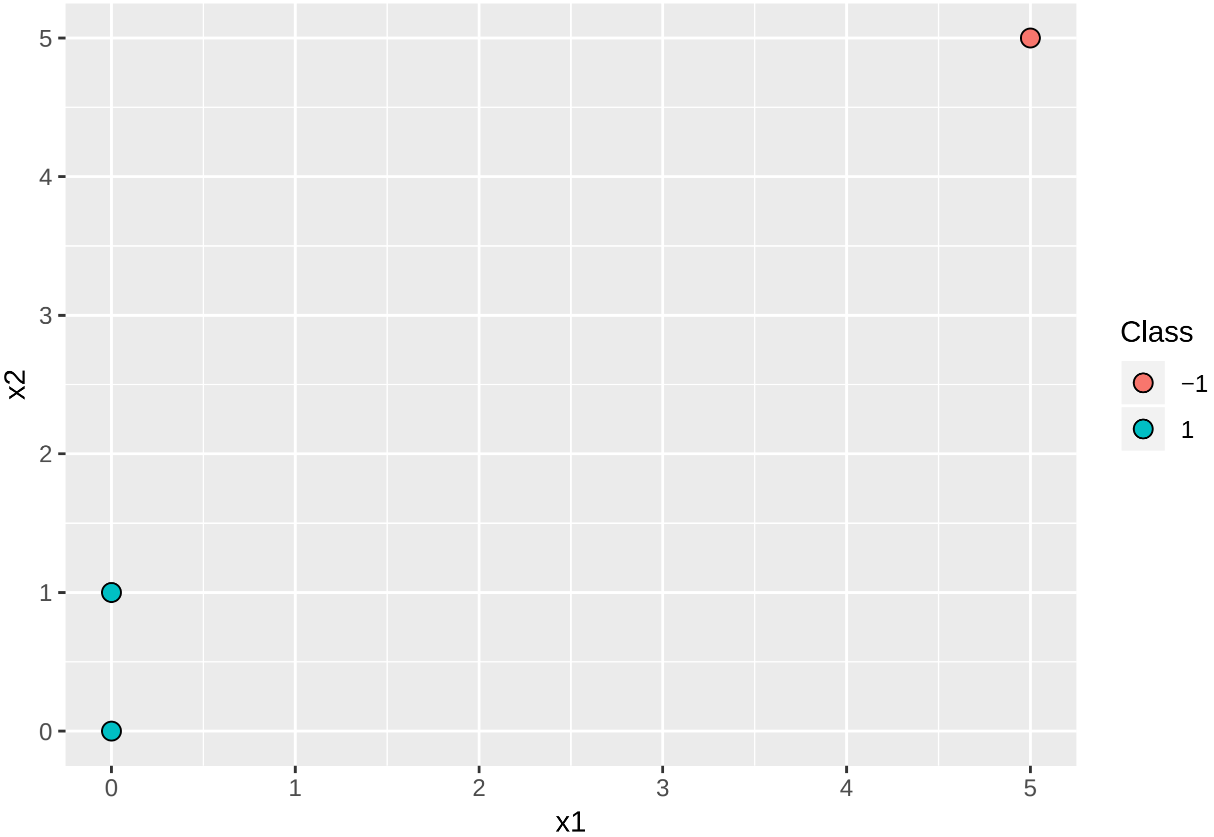
Pierre Paquay

Problem 9.1

(*a*) We begin by implementing the nearest neighbor method on the raw data.

```r
data <- data.frame(x1 = c(0, 0, 5), x2 = c(0, 1, 5))
class <- as.factor(c(1, 1, -1))

ggplot(data, aes(x = x1, y = x2, fill = class)) + geom_point(size = 3, shape = 21) +
  guides(fill = guide_legend(title = "Class"))
```

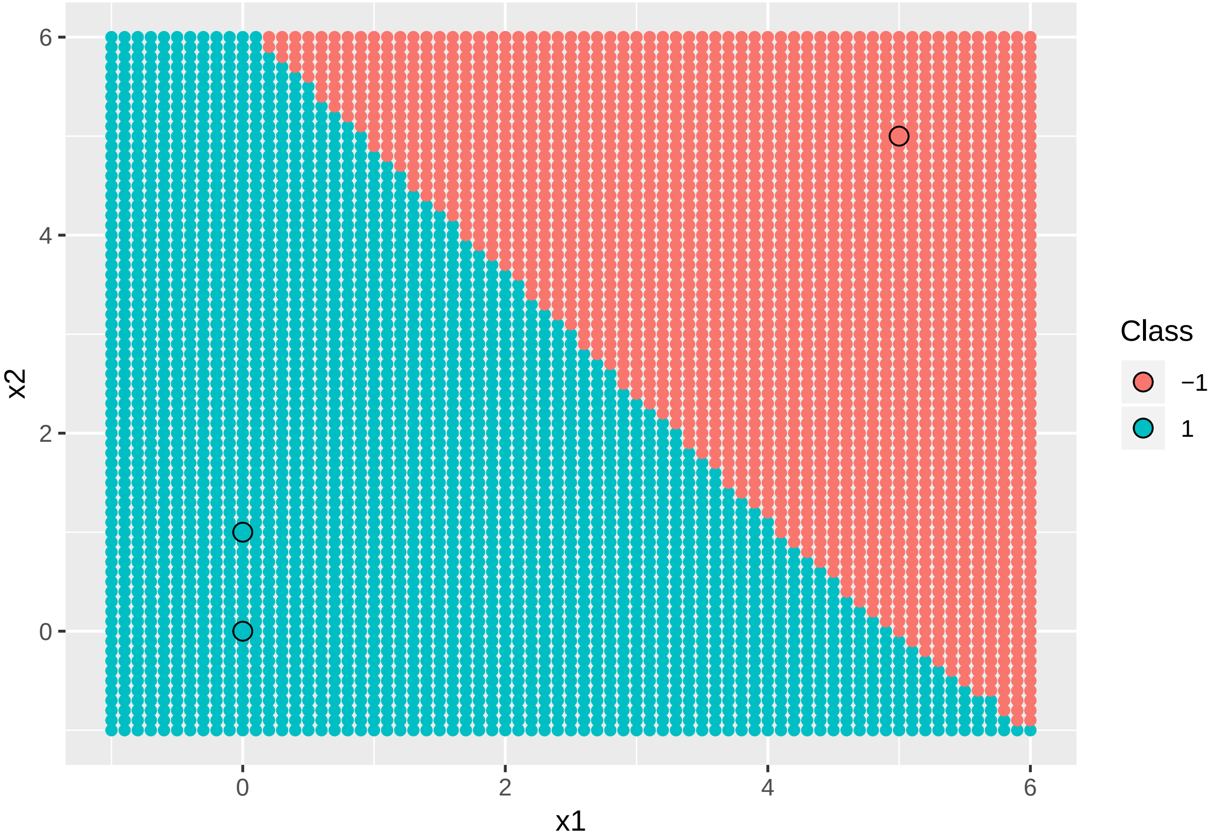


```r
grid <- expand.grid(x1 = seq(min(data[, 1] - 1), max(data[, 1] + 1), by = 0.1),
                    x2 = seq(min(data[, 2] - 1), max(data[, 2] + 1), by = 0.1))

knn_mod <- knn(data, grid, class, k = 1, prob = TRUE)
```

Below, we show the decision regions of the final hypothesis.

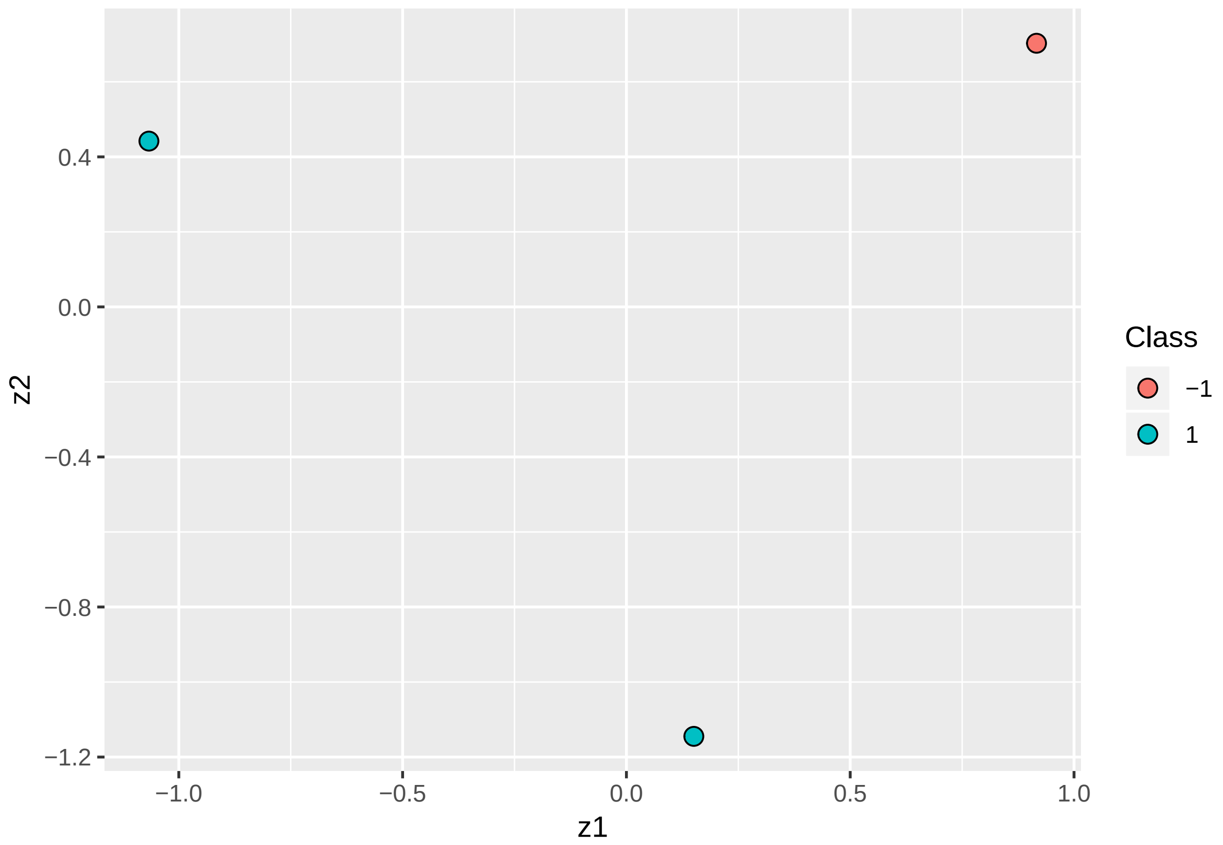
```r
ggplot() + geom_point(data = grid, aes(x = x1, y = x2, col = knn_mod)) +
  geom_point(data = data, aes(x = x1, y = x2, fill = class), size = 3, shape = 21) +
  guides(fill = guide_legend(title = "Class")) +
  guides(col = guide_legend(title = "Class"))
```



(*b*) Here, we transform to whitened coordinates and we run the nearest neighbor rule.

```
data_centered <- apply(data, 2, function(y) y - mean(y))
sigma <- t(data_centered) %*% as.matrix(data_centered) / 2
sigma_sqr <- sqrtm(sigma)
sigma_sqr_inv <- solve(sigma_sqr)
data_whitened <- as.matrix(data_centered) %*% sigma_sqr_inv
data_whitened <- as.data.frame(data_whitened)
colnames(data_whitened) <- c("z1", "z2")

ggplot(data_whitened, aes(x = z1, y = z2, fill = class)) + geom_point(size = 3, shape = 21) +
  guides(fill = guide_legend(title = "Class"))
```

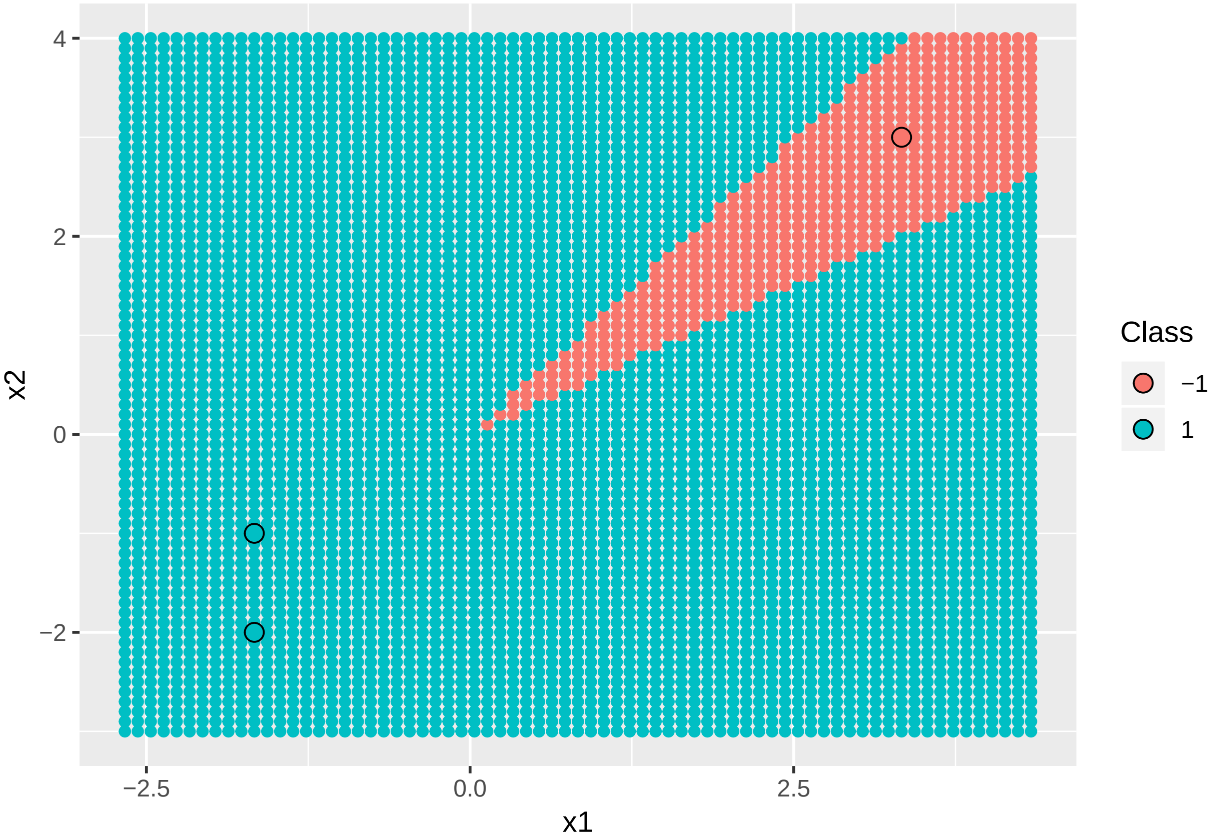


```
grid_centered <- expand.grid(x1 = seq(min(data_centered[, 1] - 1),
                                      max(data_centered[, 1] + 1), by = 0.1),
                             x2 = seq(min(data_centered[, 2] - 1),
                                      max(data_centered[, 2] + 1), by = 0.1))
grid_whitened <- as.matrix(grid_centered) %*% sigma_sqr_inv

knn_mod_whitened <- knn(data_whitened, grid_whitened, class, k = 1, prob = TRUE)
```

We show the decision region of the final hypothesis in the original space as well.

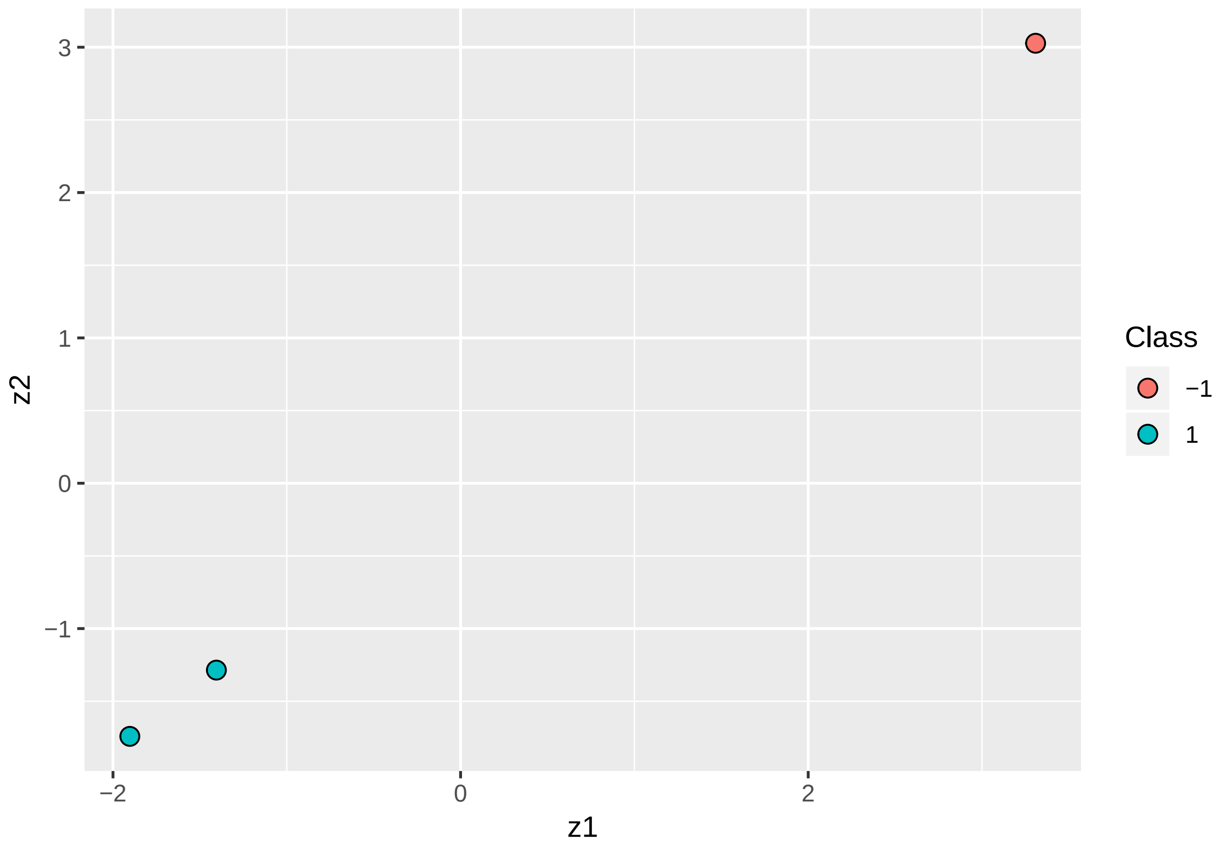
```
ggplot() + geom_point(data = grid_centered, aes(x = x1, y = x2, col = knn_mod_whitened)) +
  geom_point(data = as.data.frame(data_centered), aes(x = x1, y = x2, fill = class),
             size = 3, shape = 21) +
  guides(fill = guide_legend(title = "Class")) +
  guides(col = guide_legend(title = "Class"))
```



(c) Finally, we use principal component analysis to reduce the data to 1 dimension for our nearest neighbor classifier.

```
SVD_decomp <- svd(data_centered)
V1 <- SVD_decomp$v[, 1]
Z <- data_centered %*% V1
data_pca <- Z %*% t(V1)
data_pca <- as.data.frame(data_pca)
colnames(data_pca) <- c("z1", "z2")

ggplot(data_pca, aes(x = z1, y = z2, fill = class)) + geom_point(size = 3, shape = 21) +
  guides(fill = guide_legend(title = "Class"))
```

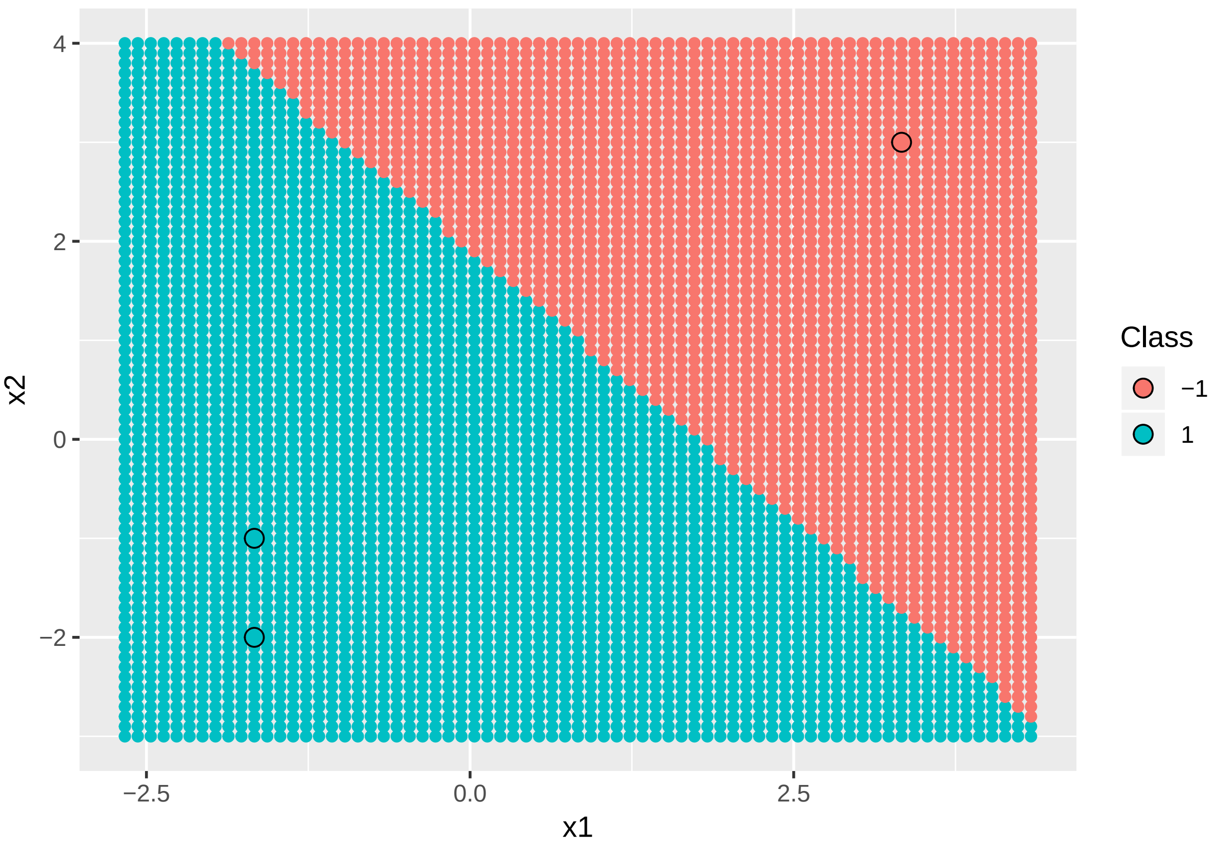


```
grid_pca <- (as.matrix(grid_centered) %*% V1) %*% t(V1)

knn_mod_pca <- knn(data_pca, grid_pca, class, k = 1, prob = TRUE)
```

Once again, we show the decision regions of the final hypothesis in the original space.

```
ggplot() + geom_point(data = grid_centered, aes(x = x1, y = x2, col = knn_mod_pca)) +
  geom_point(data = as.data.frame(data_centered), aes(x = x1, y = x2, fill = class),
             size = 3, shape = 21) +
  guides(fill = guide_legend(title = "Class")) +
  guides(col = guide_legend(title = "Class"))
```


4
2
0
−2
x2
−2.5
0.0
2.5
x1
Class
−1
1