Problem Solutions

e-Chapter 6

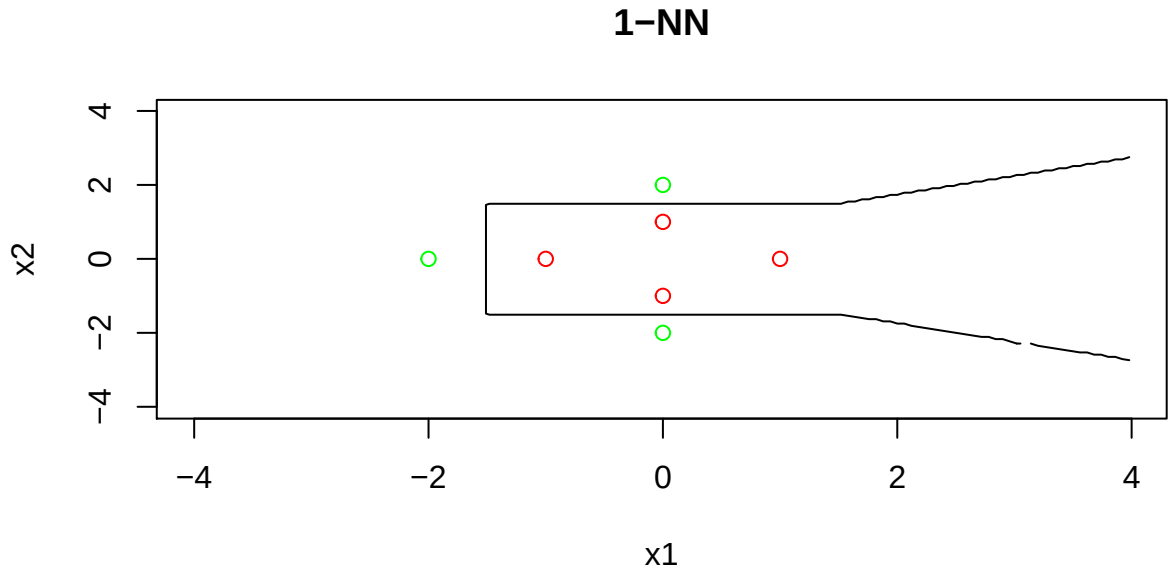
Pierre Paquay

Problem 6.1

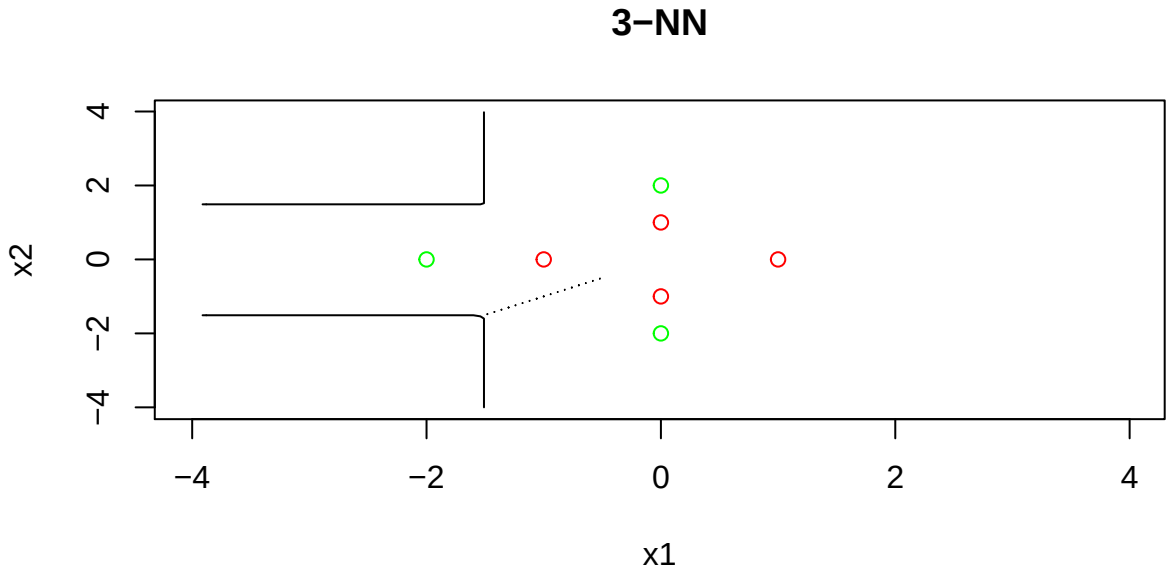
(a) Below we plot the decision regions for the 1-NN and 3-NN rules.

```
set.seed(10)
X <- matrix(c(1, 0, 0, -1, 0, 0, -2, 0, 1, -1, 0, 2, -2, 0), nrow = 7)
seq <- seq(-4, 4, by = 0.06)
Xnew <- expand.grid(seq, seq)
labels <- c(-1, -1, -1, -1, 1, 1, 1)

NN_1 <- knn(X, Xnew, labels, k = 1, prob = TRUE)
prob <- attr(NN_1, "prob")
prob <- ifelse(NN_1 == "1", prob, 1-prob)
prob1 <- matrix(prob, length(seq), length(seq))
contour(seq, seq, prob1, levels = 0.5, labels = "", xlab = "x1", ylab = "x2",
        main = "1-NN")
points(X, col = ifelse(labels == -1, "red", "green"))
```



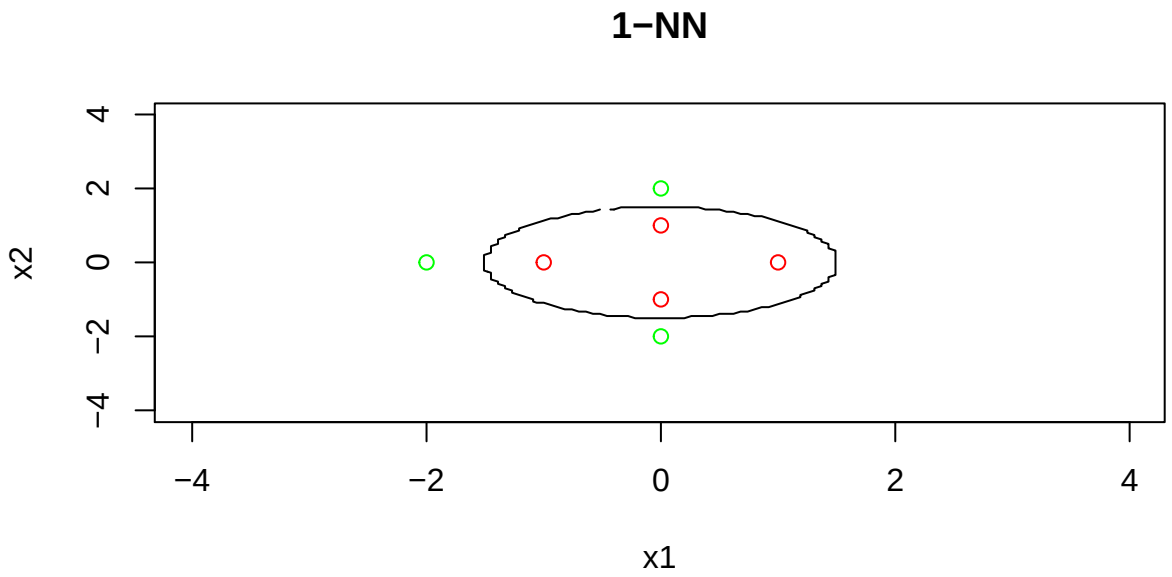

```
NN_3 <- knn(X, Xnew, labels, k = 3, prob = TRUE)
prob <- attr(NN_3, "prob")
prob <- ifelse(NN_3 == "1", prob, 1-prob)
prob1 <- matrix(prob, length(seq), length(seq))
contour(seq, seq, prob1, levels = 0.5, labels = "", xlab = "x1", ylab = "x2",
        main = "3-NN")
points(X, col = ifelse(labels == -1, "red", "green"))
```



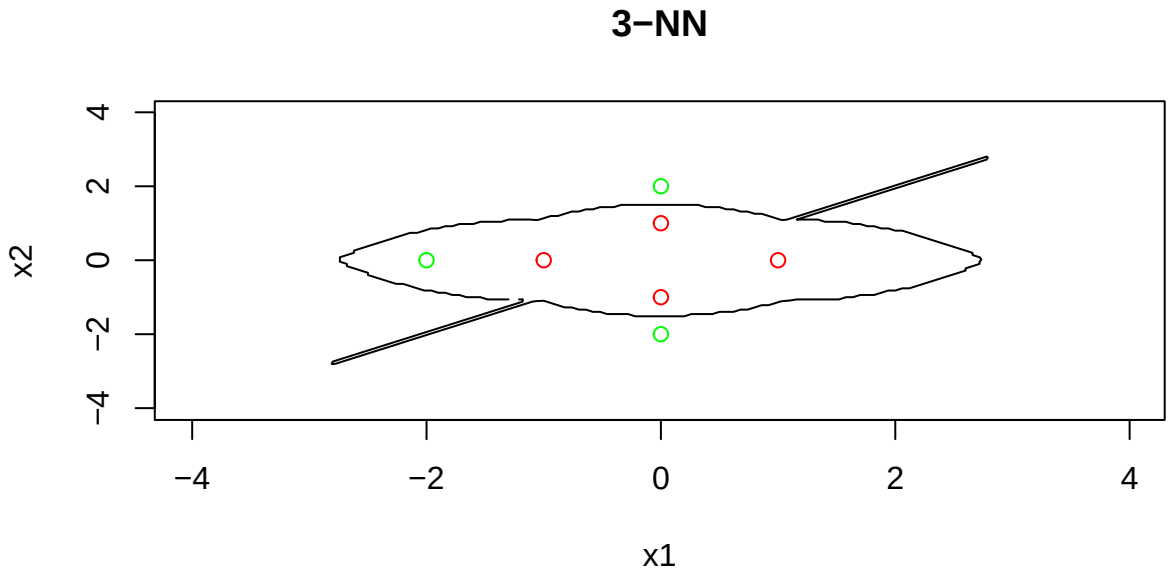
(*b*) Below we plot the classification regions in the x-space for the 1-NN and 3-NN rules implemented on the data in the z-space.

```
Z1 <- apply(X, 1, function(x) sqrt(x[1]^2 + x[2]^2))
Z2 <- apply(X, 1, function(x) atan(x[2]/ x[1]))
Z <- matrix(c(Z1, Z2), byrow = FALSE, ncol = 2)
Znew1 <- apply(Xnew, 1, function(x) sqrt(x[1]^2 + x[2]^2))
Znew2 <- apply(Xnew, 1, function(x) atan(x[2]/ x[1]))
Znew <- matrix(c(Znew1, Znew2), byrow = FALSE, ncol = 2)

NN_1 <- knn(Z, Znew, labels, k = 1, prob = TRUE)
prob <- attr(NN_1, "prob")
prob <- ifelse(NN_1 == "1", prob, 1-prob)
prob1 <- matrix(prob, length(seq), length(seq))
contour(seq, seq, prob1, levels = 0.5, labels = "", xlab = "x1", ylab = "x2",
        main = "1-NN")
points(X, col = ifelse(labels == -1, "red", "green"))
```



```
NN_3 <- knn(Z, Znew, labels, k = 3, prob = TRUE)
prob <- attr(NN_3, "prob")
prob <- ifelse(NN_3 == "1", prob, 1-prob)
prob1 <- matrix(prob, length(seq), length(seq))
contour(seq, seq, prob1, levels = 0.5, labels = "", xlab = "x1", ylab = "x2",
        main = "3-NN")
points(X, col = ifelse(labels == -1, "red", "green"))
```

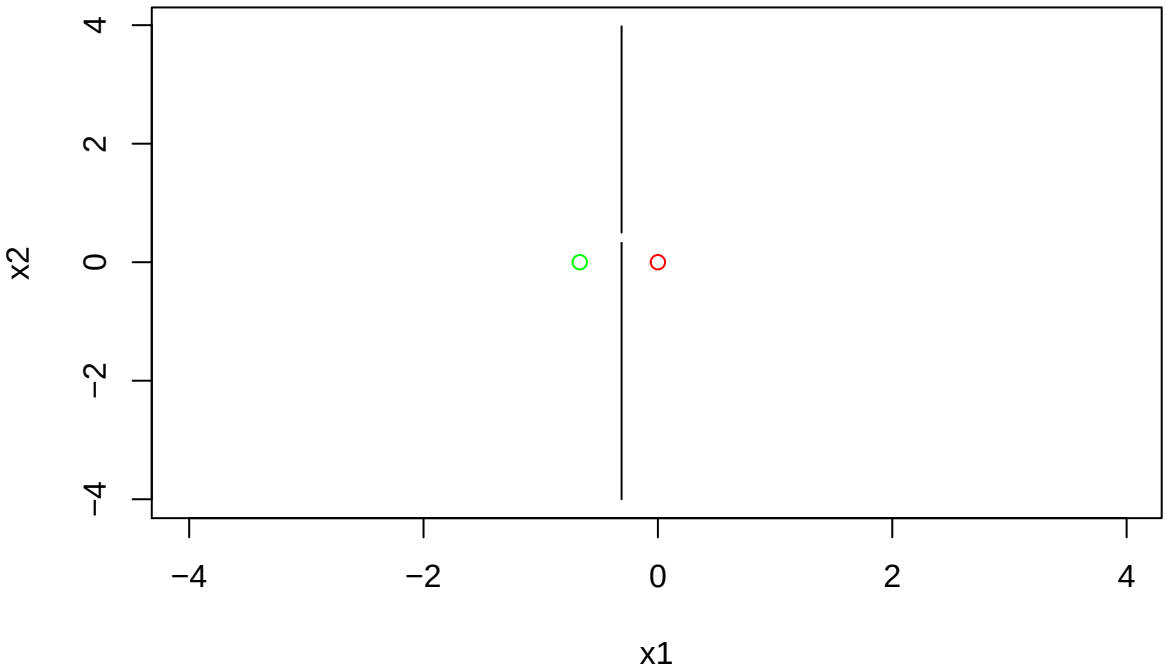


Problem 6.2

(a) Below we plot the classification regions for the 1-NN rule using the condensed data.

```
mu_1 <- apply(X[labels == 1, ], 2, mean)
mu_min1 <- apply(X[labels == -1, ], 2, mean)
Xcond <- rbind(mu_1, mu_min1)
labels2 <- c(1, -1)
knn_pred <- knn(Xcond, X, labels2, k = 1, prob = TRUE)

NN_1 <- knn(Xcond, Xnew, labels2, k = 1, prob = TRUE)
prob <- attr(NN_1, "prob")
prob <- ifelse(NN_1 == "1", prob, 1-prob)
prob1 <- matrix(prob, length(seq), length(seq))
contour(seq, seq, prob1, levels = 0.5, labels = "", xlab = "x1", ylab = "x2",
        main = "1-NN for condensed data")
points(Xcond, col = ifelse(labels2 == -1, "red", "green"))
```

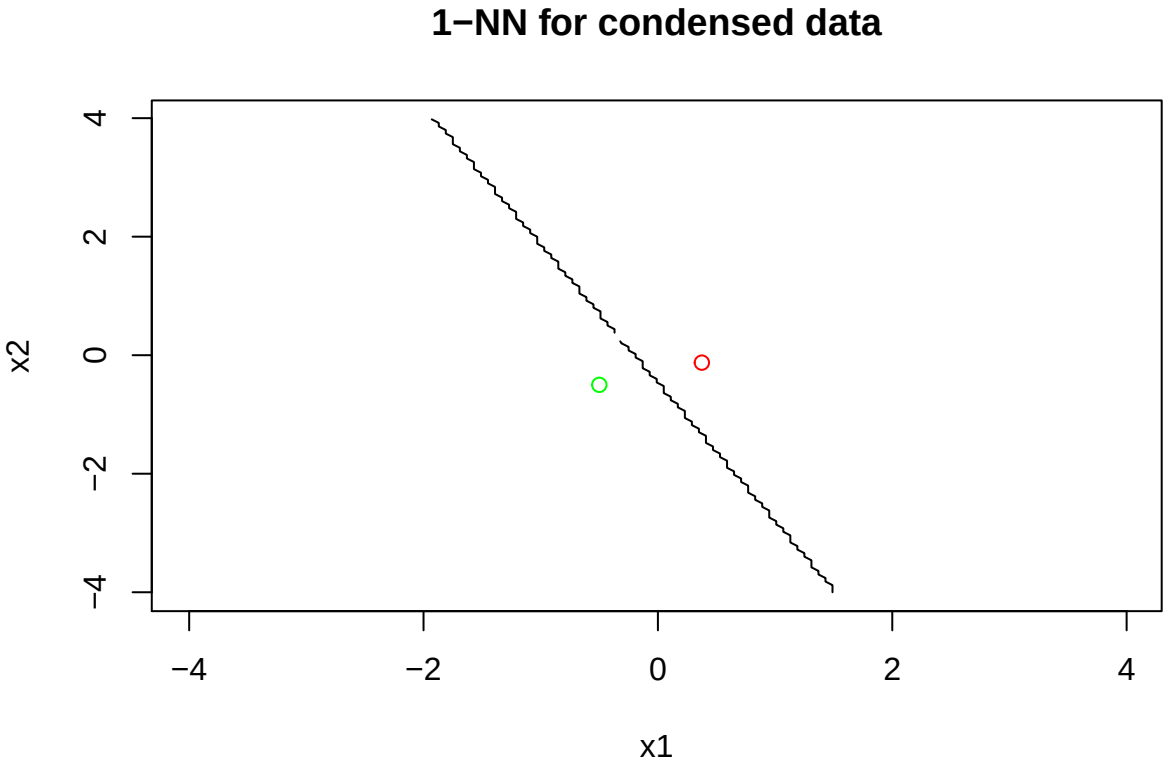


Here, we have an in-sample error equal to 0.4285714.

(b) It is easy to see that the method of condensing gives us two points of coordinates $(-1/2, -1/2)$ and $(3/8, -1/8)$ with labels of $+1$ and -1 respectively.

```
cond_1 <- c(-1/2, -1/2)
cond_min1 <- c(3/8, -1/8)
Xcond <- rbind(cond_1, cond_min1)
labels2 <- c(1, -1)
knn_pred <- knn(Xcond, X, labels2, k = 1, prob = TRUE)

NN_1 <- knn(Xcond, Xnew, labels2, k = 1, prob = TRUE)
prob <- attr(NN_1, "prob")
prob <- ifelse(NN_1 == "1", prob, 1-prob)
prob1 <- matrix(prob, length(seq), length(seq))
contour(seq, seq, prob1, levels = 0.5, labels = "", xlab = "x1", ylab = "x2",
        main = "1-NN for condensed data")
points(Xcond, col = ifelse(labels2 == -1, "red", "green"))
```



In this case, we have an in-sample error equal to 0.4285714 which is exactly equal to the in-sample error of point (a).