module\_6\_assignment\_rmd

library(tidyverse); library(e1071); library(cvms); library(tibble); library(broom)  
  
### Problem 1a  
q1a <- function(x1, x2) {  
 return(1 + 3\*x1 - x2)  
}  
  
x1s <- seq(-1.5, 1.5, length.out = 1e2)  
x2s <- seq(-1.5, 1.5, length.out = 1e2)  
  
# Reference: https://selbydavid.com/2018/01/09/neural-network/  
decision\_grid <- expand.grid(x1 = x1s, x2 = x2s)  
decision\_grid\_mat <- data.matrix(decision\_grid[,c('x1','x2')])  
q1a\_grid\_output <- mapply(q1a, decision\_grid\_mat[,1], decision\_grid\_mat[,2])  
decision\_grid$class\_label <- factor(x = q1a\_grid\_output > 0, labels = c('Blue', 'Red'))  
  
# Reference: https://stackoverflow.com/questions/40675778/center-plot-title-in-ggplot2  
theme\_update(plot.title = element\_text(hjust = 0.5))  
  
# Reference: https://stackoverflow.com/questions/43129280/color-points-with-the-color-as-a-column-in-ggplot2  
p1 <- ggplot(data = decision\_grid, aes(x = x1, y = x2)) +  
 geom\_point(colour = decision\_grid$class\_label, size = 0.1) +  
 # Reference: http://www.sthda.com/english/wiki/ggplot2-add-straight-lines-to-a-plot-horizontal-vertical-and-regression-lines  
 geom\_abline(intercept = 1, slope = 3) +  
 geom\_hline(yintercept = 0) + geom\_vline(xintercept = 0) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 1a') +  
 theme\_minimal() + theme(aspect.ratio = 1, plot.title = element\_text(hjust = 0.5))  
  
### Problem 1b  
x1s <- seq(-1.5, 1.5, length.out = 1e2)  
x2s <- seq(-1.5, 1.5, length.out = 1e2)  
q1b <- function(x1, x2) {  
 return(-2 + x1 + 2\*x2)  
}  
decision\_gridb <- expand.grid(x1 = x1s, x2 = x2s)  
decision\_grid\_matb <- data.matrix(decision\_gridb[,c('x1','x2')])  
q1b\_grid\_output <- mapply(q1b, decision\_grid\_matb[,1], decision\_grid\_matb[,2])  
decision\_gridb$class\_label <- factor(x = q1b\_grid\_output > 0, labels = c('Green', 'Purple'))  
  
# Shrink the # of points in the geom\_point  
x1s <- seq(-1.5, 1.5, length.out = 20)  
x2s <- seq(-1.5, 1.5, length.out = 20)  
decision\_grid <- expand.grid(x1 = x1s, x2 = x2s)  
decision\_grid\_mat <- data.matrix(decision\_grid[,c('x1','x2')])  
q1a\_grid\_output <- mapply(q1a, decision\_grid\_mat[,1], decision\_grid\_mat[,2])  
decision\_grid$class\_label <- factor(x = q1a\_grid\_output > 0, labels = c('Blue', 'Red'))  
  
p2 <- ggplot(data = decision\_grid, aes(x = x1, y = x2)) +  
 # Reference: http://www.math-evry.cnrs.fr/\_media/members/cambroise/teaching/tp\_r\_part1\_corrected.pdf  
 geom\_tile(data = decision\_gridb, aes(fill = class\_label)) +  
 # Reference: http://www.cookbook-r.com/Graphs/Colors\_(ggplot2)/  
 # Reference: https://stackoverflow.com/questions/25195869/how-to-change-color-palette-of-geom-tile-in-r-ggplot2  
 # Reference: https://stackoverflow.com/questions/25176399/scale-fill-discrete-and-scale-fill-manual-legend-options-confusion  
 scale\_fill\_manual(values=c("#99FF33", "#3300FF"), name = 'Class Label') +  
 # Reference: https://stackoverflow.com/questions/47023781/how-to-add-a-legend-for-two-geom-layers-in-one-ggplot2-plot  
 geom\_point(aes(colour = class\_label), size = 0.5) +  
 scale\_color\_manual(values=c("#0099FF", "#CC0033"), name = 'Class Label') +  
 geom\_abline(intercept = 1, slope = 3) +  
 geom\_abline(intercept = 1, slope = -0.5, color = 'red') +  
 geom\_hline(yintercept = 0) + geom\_vline(xintercept = 0) +  
 theme\_minimal() + theme(aspect.ratio = 1, plot.title = element\_text(hjust = 0.5)) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 1b') +  
 guides(shape = FALSE,  
 colour = guide\_legend(override.aes = list(  
 fill = c("#99FF33", "#3300FF"),  
 size = c(3, 3),  
 shape = c(16, 16))))  
  
### Problem 2  
### part (a)  
# Reference: https://stackoverflow.com/questions/6862742/draw-a-circle-with-ggplot2  
plot\_circle <- function(center = c(0,0), diameter = 1, npoints = 1e3){  
 r = diameter / 2  
 tt <- seq(0,2\*pi,length.out = npoints)  
 xx <- center[1] + r \* cos(tt)  
 yy <- center[2] + r \* sin(tt)  
 return(data.frame(x = xx, y = yy))  
}  
circle\_data <- plot\_circle(center = c(-1, 2), diameter = 2\*2, npoints = 1e3)  
  
p3 <- ggplot(circle\_data, aes(x,y)) +  
 geom\_path() +  
 # Reference: https://stackoverflow.com/questions/21294196/how-do-i-make-my-facets-perfectly-square  
 theme(aspect.ratio = 1) +  
 geom\_hline(yintercept = 0) + geom\_vline(xintercept = 0) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 2a') +  
 theme\_minimal() + theme(aspect.ratio = 1, plot.title = element\_text(hjust = 0.5))  
  
### part (b)  
# (-4,5), (2,5), (-4,-1), (1,-1)  
fill\_area <- data.frame(x1 = c(-4), x2 = c(2),  
 y1 = c(-1), y2 = c(5), class\_label = c('Blue'))  
circle\_data$class\_label <- 'Red'  
  
p4 <- ggplot() +  
 # Reference: http://sape.inf.usi.ch/quick-reference/ggplot2/geom\_rect  
 # Reference: https://stackoverflow.com/questions/50343911/remove-border-from-geom-rect-using-ggplot2  
 # Reference: https://stackoverflow.com/questions/31599146/ggplot2-change-geom-rect-colour-in-a-stacked-barplot  
 geom\_rect(data = fill\_area,  
 mapping = aes(xmin = x1, xmax = x2, ymin = y1, ymax = y2), fill = '#0099FF',  
 color = NA, alpha = 0.5) +  
 geom\_polygon(data = circle\_data, mapping = aes(x, y), fill = "#CC0033", alpha = 0.5) +  
 # Reference: https://stackoverflow.com/questions/21294196/how-do-i-make-my-facets-perfectly-square  
 geom\_hline(yintercept = 0) + geom\_vline(xintercept = 0) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 2a') +  
 # Reference: https://stackoverflow.com/questions/45346885/center-plot-title-in-ggplot2-using-theme-bw  
 theme\_minimal() + theme(aspect.ratio = 1, plot.title = element\_text(hjust = 0.5)) +  
 scale\_colour\_manual(name = 'the colour',   
 values =c('#0099FF'='black','#CC0033'='red'), labels = c('c2','c1'))  
  
### Problem 3  
### part (a)  
df <- data.frame(x1 = c(3,2,4,1,2,4,4),  
 x2 = c(4,2,4,4,1,3,1),  
 y = rep(c('Red', 'Blue'), each = 4)[-8])  
  
p5 <- ggplot(data = df, mapping = aes(x = x1, y = x2, colour = y)) +  
 geom\_hline(yintercept = 0) + geom\_vline(xintercept = 0) +  
 geom\_point(aes(colour = y), size = 1.5) +  
 scale\_color\_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3a') +  
 theme\_minimal() + theme(aspect.ratio = 1, plot.title = element\_text(hjust = 0.5))  
  
### part (b)  
p6 <- ggplot(data = df, mapping = aes(x = x1, y = x2, colour = y)) +  
 geom\_hline(yintercept = 0) + geom\_vline(xintercept = 0) +  
 geom\_point(aes(colour = y), size = 1.5) +  
 geom\_abline(slope = 1, intercept = -0.5) +  
 scale\_color\_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3b') +  
 theme\_minimal() + theme(aspect.ratio = 1, plot.title = element\_text(hjust = 0.5))  
  
### part (d)  
p7 <- ggplot(data = df, mapping = aes(x = x1, y = x2, colour = y)) +  
 geom\_hline(yintercept = 0) + geom\_vline(xintercept = 0) +  
 geom\_point(aes(colour = y), size = 1.5) +  
 geom\_abline(slope = 1, intercept = -0.5) +  
 geom\_abline(slope = 1, intercept = -1, linetype = 'dashed') +  
 geom\_abline(slope = 1, intercept = 0, linetype = 'dashed') +  
 scale\_color\_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3d') +  
 theme\_minimal() + theme(aspect.ratio = 1, plot.title = element\_text(hjust = 0.5))  
  
### part (e)  
# y = x - 0.5  
# y - y1 = (x - x1)  
# (2,2)  
# y - 2 = x - 2  
# y = x  
d=data.frame(x=c(1,2,5,6,8), y=c(3,6,2,8,7), vx=c(1,1.5,0.8,0.5,1.3), vy=c(0.2,1.3,1.7,0.8,1.4))  
  
margin\_points <- data.frame(  
 x = c(2, 2, 4, 4),  
 y = c(2, 1, 4, 3),  
 vx = c(0.25, -0.25, 0.25, -0.25),  
 vy = c(-0.25, 0.25, -0.25, 0.25)  
)  
  
p8 <- ggplot(data = df, mapping = aes(x = x1, y = x2, colour = y)) +  
 geom\_hline(yintercept = 0) + geom\_vline(xintercept = 0) +  
 geom\_point(aes(colour = y), size = 1.5) +  
 geom\_abline(slope = 1, intercept = -0.5) +  
 geom\_abline(slope = 1, intercept = -1, linetype = 'dashed') +  
 geom\_abline(slope = 1, intercept = 0, linetype = 'dashed') +  
 # Reference: http://sape.inf.usi.ch/quick-reference/ggplot2/geom\_segment  
 geom\_segment(data=margin\_points, mapping=aes(x=x, y=y, xend=x+vx, yend=y+vy), size=0.5, color="green") +  
 scale\_color\_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3e') +  
 xlim(0, 5) + theme\_minimal() +  
 theme(aspect.ratio = 1, plot.title = element\_text(hjust = 0.5))  
  
### part (g)  
p9 <- ggplot(data = df, mapping = aes(x = x1, y = x2, colour = y)) +  
 geom\_hline(yintercept = 0) + geom\_vline(xintercept = 0) +  
 geom\_point(aes(colour = y), size = 1.5) +  
 geom\_abline(slope = -0.5, intercept = 3.5) +  
 scale\_color\_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3f') +  
 xlim(0, 5) + theme\_minimal() +  
 theme(aspect.ratio = 1, plot.title = element\_text(hjust = 0.5))  
  
### part (h)  
df2 <- data.frame(x1 = c(3,2,4,1,2,4,4,2),  
 x2 = c(4,2,4,4,1,3,1,3),  
 y = rep(c('Red', 'Blue'), each = 4))  
  
p10 <- ggplot(data = df2, mapping = aes(x = x1, y = x2, colour = y)) +  
 geom\_hline(yintercept = 0) + geom\_vline(xintercept = 0) +  
 geom\_point(aes(colour = y), size = 1.5) +  
 geom\_abline(slope = 1, intercept = -0.5) +  
 geom\_abline(slope = 1, intercept = -1, linetype = 'dashed') +  
 geom\_abline(slope = 1, intercept = 0, linetype = 'dashed') +  
 scale\_color\_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3f') +  
 xlim(0, 5) + theme\_minimal() +  
 theme(aspect.ratio = 1, plot.title = element\_text(hjust = 0.5))  
  
### Problem 4  
set.seed(111)  
x1\_class1 <- rnorm(50)  
x1\_class2 <- rnorm(50, mean = 0.5)  
x2\_class1 <- 1.5 \* x1\_class1^(2) + 0.5 + runif(50)  
x2\_class2 <- 1.5 \* x1\_class2^(2) + - 0.5 - runif(50)  
  
df <- data.frame(x = c(x1\_class1, x1\_class2),  
 y = c(x2\_class1, x2\_class2),  
 z = rep(c(1,2), each = 50))  
  
p11 <- ggplot(df, aes(x, y, color = factor(z))) +  
 geom\_point() +  
 scale\_color\_manual(values=c("#0099FF", "#CC0033"), name = 'Class Label') +  
 theme\_minimal() +  
 theme(aspect.ratio = 0.55, plot.title = element\_text(hjust = 0.5)) +  
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 4')  
  
set.seed(666)  
train\_idx <- sample(seq(1,1e2), 80)  
  
data.train <- data.frame(x = df[train\_idx,c('x')],  
 y = df[train\_idx,c('y')],  
 z = as.factor(df$z[train\_idx]))  
  
data.test <- data.frame(x = df[-train\_idx,'x'],  
 y = df[-train\_idx,'y'],  
 z = as.factor(df$z[-train\_idx]))  
  
svm.linear <- svm(z~., data=data.train, kernel="linear", cost=10)  
plot(svm.linear, data.train)  
table(df$z[train\_idx], predict(svm.linear, data.train))  
  
svm.poly = svm(z~., data=data.train, kernel="polynomial", cost=10)  
plot(svm.poly, data.train)  
table(df$z[train\_idx], predict(svm.poly, data.train))  
  
svm.radial = svm(z~., data=data.train, kernel="radial", gamma=1, cost=10)  
plot(svm.radial, data.train)  
table(df$z[train\_idx], predict(svm.radial, data.train))  
  
plot(svm.linear, data.test)  
plot(svm.poly, data.test)  
plot(svm.radial, data.test)  
table(df$z[-train\_idx], predict(svm.linear, data.test))  
table(df$z[-train\_idx], predict(svm.poly, data.test))  
table(df$z[-train\_idx], predict(svm.radial, data.test))  
  
linear\_cm\_train <- tidy(table(tibble("target"=df$z[train\_idx],  
 "prediction"=predict(svm.linear, data.train))))  
plot\_confusion\_matrix(linear\_cm\_train,   
 targets\_col = "target",   
 predictions\_col = "prediction",  
 counts\_col = "n")  
  
poly\_cm\_train <- tidy(table(tibble("target"=df$z[train\_idx],  
 "prediction"=predict(svm.poly, data.train))))  
plot\_confusion\_matrix(poly\_cm\_train,   
 targets\_col = "target",   
 predictions\_col = "prediction",  
 counts\_col = "n")  
  
radial\_cm\_train <- tidy(table(tibble("target"=df$z[train\_idx],  
 "prediction"=predict(svm.radial, data.train))))  
plot\_confusion\_matrix(radial\_cm\_train,   
 targets\_col = "target",   
 predictions\_col = "prediction",  
 counts\_col = "n")  
  
  
linear\_cm <- tidy(table(tibble("target"=df$z[-train\_idx],  
 "prediction"=predict(svm.linear, data.test))))  
plot\_confusion\_matrix(linear\_cm,   
 targets\_col = "target",   
 predictions\_col = "prediction",  
 counts\_col = "n")  
  
poly\_cm <- tidy(table(tibble("target"=df$z[-train\_idx],  
 "prediction"=predict(svm.poly, data.test))))  
plot\_confusion\_matrix(poly\_cm,   
 targets\_col = "target",   
 predictions\_col = "prediction",  
 counts\_col = "n")  
  
radial\_cm <- tidy(table(tibble("target"=df$z[-train\_idx],  
 "prediction"=predict(svm.radial, data.test))))  
plot\_confusion\_matrix(radial\_cm,   
 targets\_col = "target",   
 predictions\_col = "prediction",  
 counts\_col = "n")