

HW4 : Support Vector Machine

Avantika Diwadkar

11/15/2022

Load packages

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
library(data.table)
```

```
##  
## Attaching package: 'data.table'  
  
## The following objects are masked from 'package:dplyr':  
##  
##   between, first, last
```

```
library(foreach)  
library(ggplot2)  
library(quadprog)  
library(MASS)
```

```
##  
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':  
##  
##      select
```

```
library(mvtnorm)  
library(Matrix)
```

Homework 4

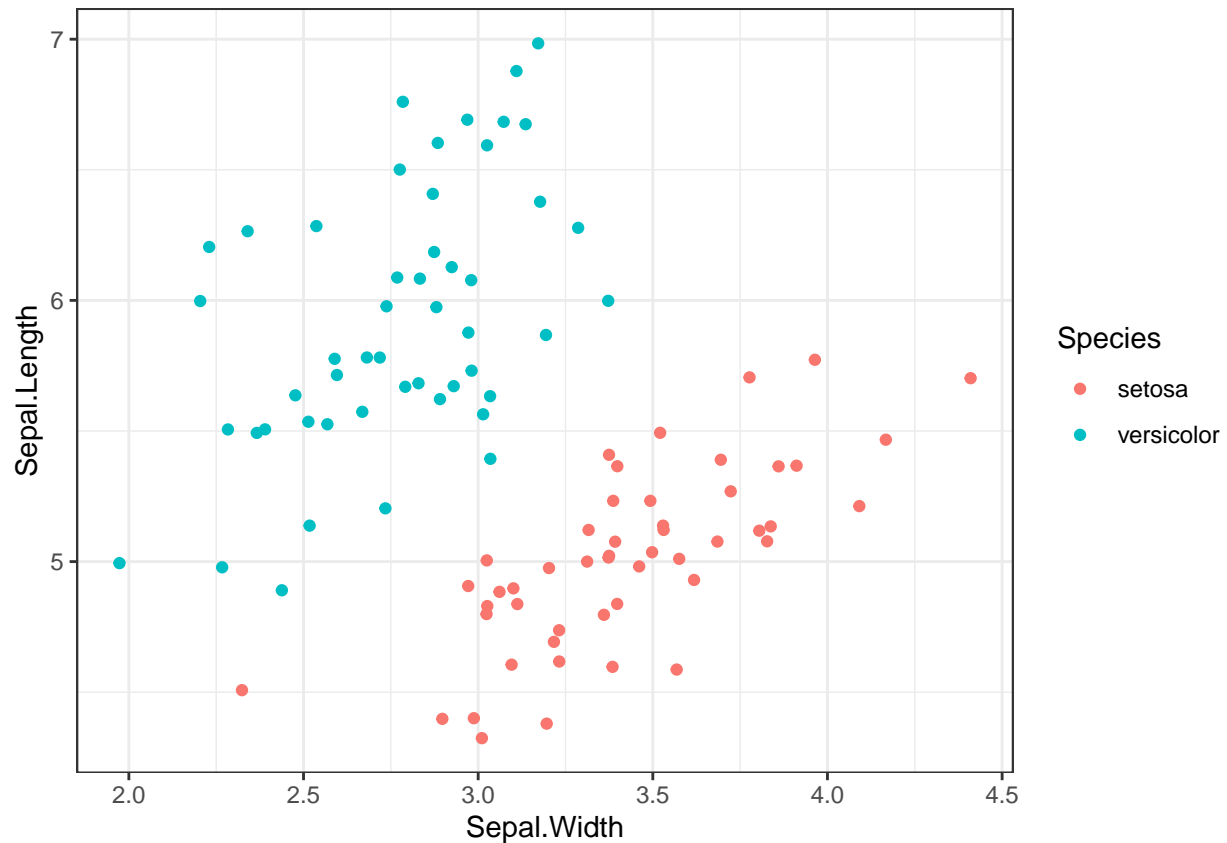
For homework, please implement the following algorithms

1. support vector machine without the slack variable.
2. support vector machine with slack variables.
3. support vector regression.

For 1 and 2, please identify some simulated examples where using slack variables improves classification and the examples where using the slack variable reduces the accuracy.

Visualize example dataset for separable case

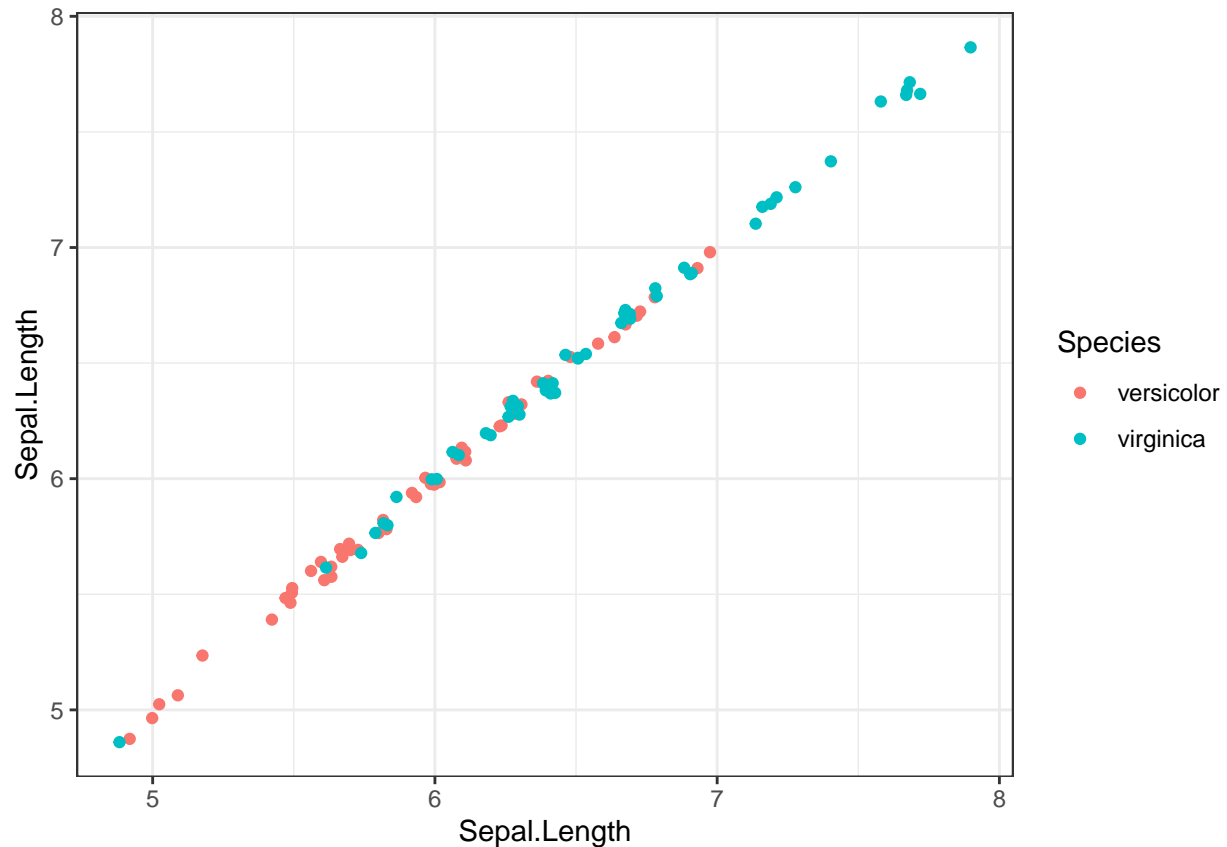
```
data(iris)  
iris2cat = iris[1:100,c("Sepal.Length", "Sepal.Width", "Species")]  
f = ggplot(iris2cat, aes(Sepal.Width, Sepal.Length, colour = Species))  
f + geom_jitter() + scale_colour_hue() + theme(legend.position = "bottom") + theme_bw()
```



```
#Change encoding to 1 and -1
iris2cat$Species <- as.integer(ifelse(iris2cat$Species == "versicolor",-1, 1))
```

Visualize example dataset for non-separable case

```
iris_ns <- iris %>% dplyr::filter(Species != "setosa")
iris2len <- iris_ns[1:100,c("Sepal.Length","Sepal.Width","Species")]
f = ggplot(iris2len, aes(Sepal.Length, Sepal.Length, colour = Species))
f + geom_jitter() + scale_colour_hue() + theme(legend.position = "bottom") + theme_bw()
```



```
#Change encoding to 1 and -1
iris2len$Species <- as.integer(ifelse(iris2len$Species == "versicolor",-1, 1))
```

Generate the SVM function

```
svm.fit = function(X, y, C=NULL) {
  no_of_samples = length(y)
  no_of_features = ncol(X)
  K = matrix(rep(0, no_of_samples*no_of_samples), nrow=no_of_samples)

  # for (i in 1:no_of_samples){
#   for (j in 1:no_of_samples){
#     K[i,j] = X[i,] %*% X[j,] }}
# Dmat = outer(y,y) * K

  Dmat <- (X %*% t(X)) * (y %*% t(y))
  Dmat = as.matrix(nearPD(Dmat)$mat)
  dvec = rep(1, no_of_samples)

  Amat = rbind(y, diag(no_of_samples), -1*diag(no_of_samples))
  bvec = c(0, rep(0, no_of_samples), rep(-C, no_of_samples))
```

```

#Amat <- rbind(y, diag(1, nrow = no_of_samples))
#bvec = c(0, rep(0, no_of_samples))

res_qp = solve.QP(Dmat,dvec,t(Amat),bvec=bvec, meq=1)
#a = res_qp$solution
#bomega = apply(a*y*X,2,sum)
return(res_qp)
}

```

Run SVM for separable case

```

X = as.matrix(iris2cat[,1:2])
center = function(z) (z-mean(z))/sd(z)
for(j in 1:2) {X[,j] = center(X[,j])}
for(j in 1:2) {X[,j] = X[,j]+3}

y = iris2cat$Species #*2-1

#Get the QP result
res_qp <- svm.fit(X,y,C=1)

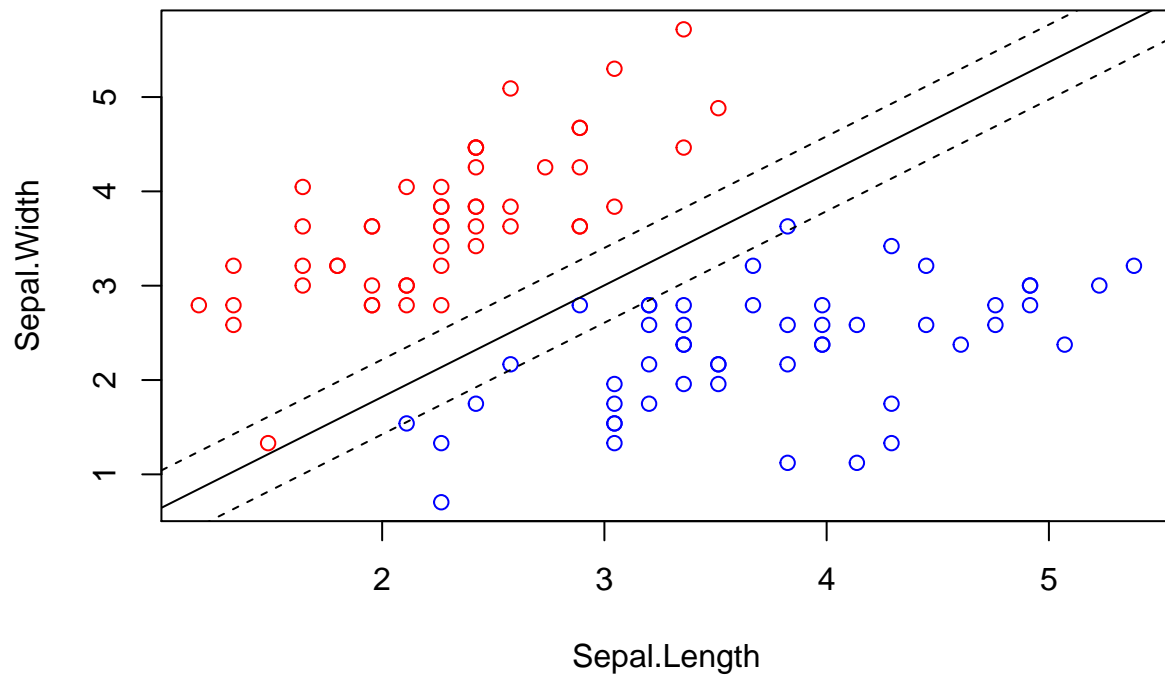
#Run the SVM model
a = res_qp$solution
bomega1 = apply(a*y*X,2,sum)

## deifning support vectors
#indx <- which(a > 0)
indx <- res_qp$iaact

#Get optimum hyperplane values w and beta0
w <- bomega1
b <- foreach(i = 1:length(indx), .combine = "c") %do% {
as.numeric(y[i] - (w %*% X[i,]))} %>% mean()

#Plot hyperplane
margin = 1 / sqrt(sum(w^2))
plot(X, col=(3-y))
abline(b/-w[2], b=w[1]/-w[2])
abline(b/-w[2]+margin, b=w[1]/-w[2], lty = 2)
abline(b/-w[2]-margin, b=w[1]/-w[2], lty = 2)

```



```
#Get prediction
#y_pred = 2*((X %*% bomega1) > 0 )-1
#table(obs=y,pred=y_pred)

# x_min = min(iris2cat$Sepal.Width)
# x_max = max(iris2cat$Sepal.Width)
# y_min = min(iris2cat$Sepal.Length)
# y_max = max(iris2cat$Sepal.Length)
# xx = seq(x_min, x_max)
# alpha = -1*w[1]/w[2]
# yy = alpha*xx - (b)/w[2]
# margin = 1 / sqrt(sum(w^2))
# abline(xx, yy)
# abline(xx+margin, yy, lty = 2)
# abline(xx-margin, yy, lty = 2)
```

Run SVM for non-separable case

```
X = as.matrix(iris2len[,1:2])
for(j in 1:2) {X[,j] = center(X[,j])}
for(j in 1:2) {X[,j] = X[,j]+3}
```

```

y = iris2len$Species #*2-1

#Get the QP result
res_qp <- svm.fit(X,y,C=1)

#Run the SVM model
a = res_qp$solution
bomega1 = apply(a*y*X,2,sum)
#Get prediction
#y_pred = 2*((X %*% bomega1) > 0 )-1
#table(obs=y,pred=y_pred)

## deifning support vectors
indx <- res_qp$iact

#Get optimum hyperplane values w and beta0
w <- bomega1
b <- foreach(i = 1:length(indx), .combine = "c") %do% {
as.numeric(y[i] - (w %*% X[i,]))} %>% mean()

#Plot hyperplane
margin = 1 / sqrt(sum(w^2))
plot(X, col=(3-y))
abline(b/-w[2], b=w[1]/-w[2])
abline(b/-w[2]+margin, b=w[1]/-w[2], lty = 2)
abline(b/-w[2]-margin, b=w[1]/-w[2], lty = 2)

```

