HW 3

Carlyle Morgan

3/25/2022

Problem 1

```
digit_5<-read.csv("train.5.txt")
digit_5$digit<-"5"
digit_6<-read.csv("train.6.txt")
digit_6$digit<-"6"
colnames(digit_5)<-NA
colnames(digit_6)<-NA
digits<-rbind(digit_5,digit_6)
colnames(digits)<-seq(1,256,1)
colnames(digits)[257]<-"digit"
test_ind<-sample(seq(1,nrow(digits)), nrow(digits)/5)
digits$digit<-as.factor(digits$digit)

digits_train<-digits[-test_ind,]
digits_test<-digits[test_ind,]

linearcrossval <- tune(svm, digit~., data = digits_train, scale = FALSE, kernel = "linear", ranges = lis

summary(linearcrossval$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = digit ~ ., data = digits_train,
##     ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)), scale = FALSE,
##     kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.01
##
## Number of Support Vectors:  131
##
##  ( 62 69 )
##
##
## Number of Classes:  2
##
## Levels:
##  5 6
```

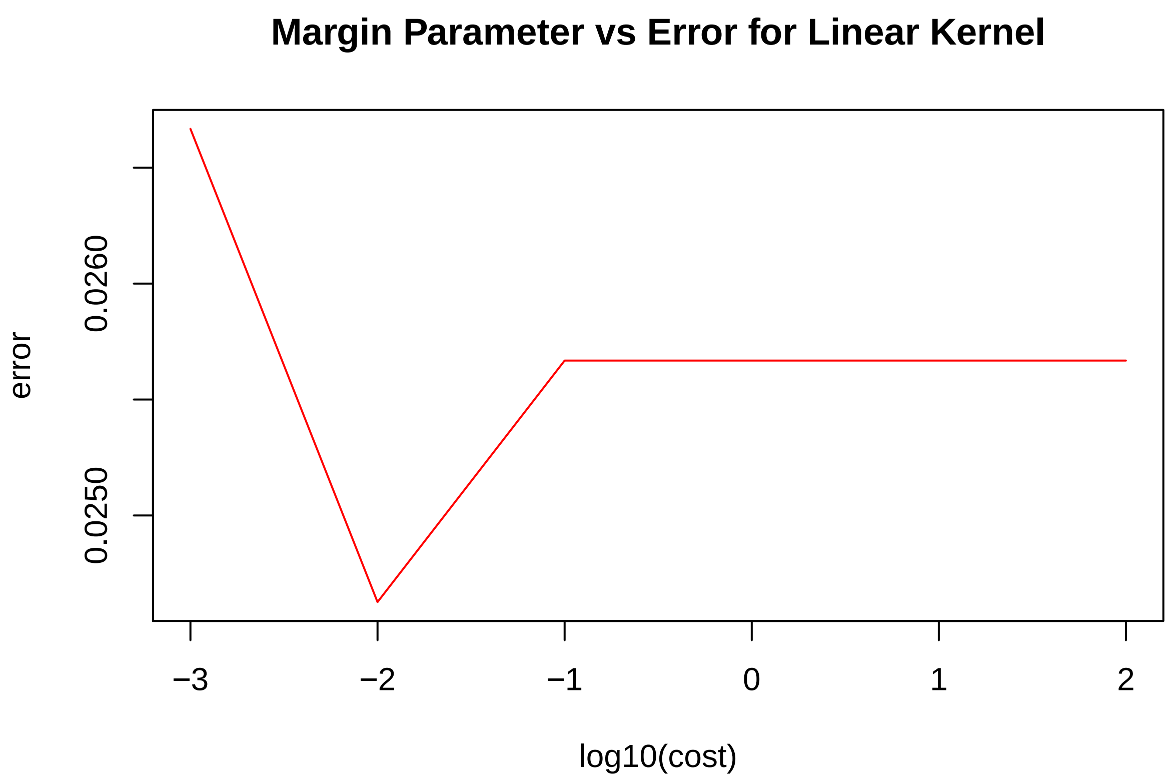
```r
### Because cross-validation with radial kernels is computationally intensive, I ran this
### process once and then exported the results to a csv

radialperformances<-read.csv("radialperformances.csv")
```

1.

a.

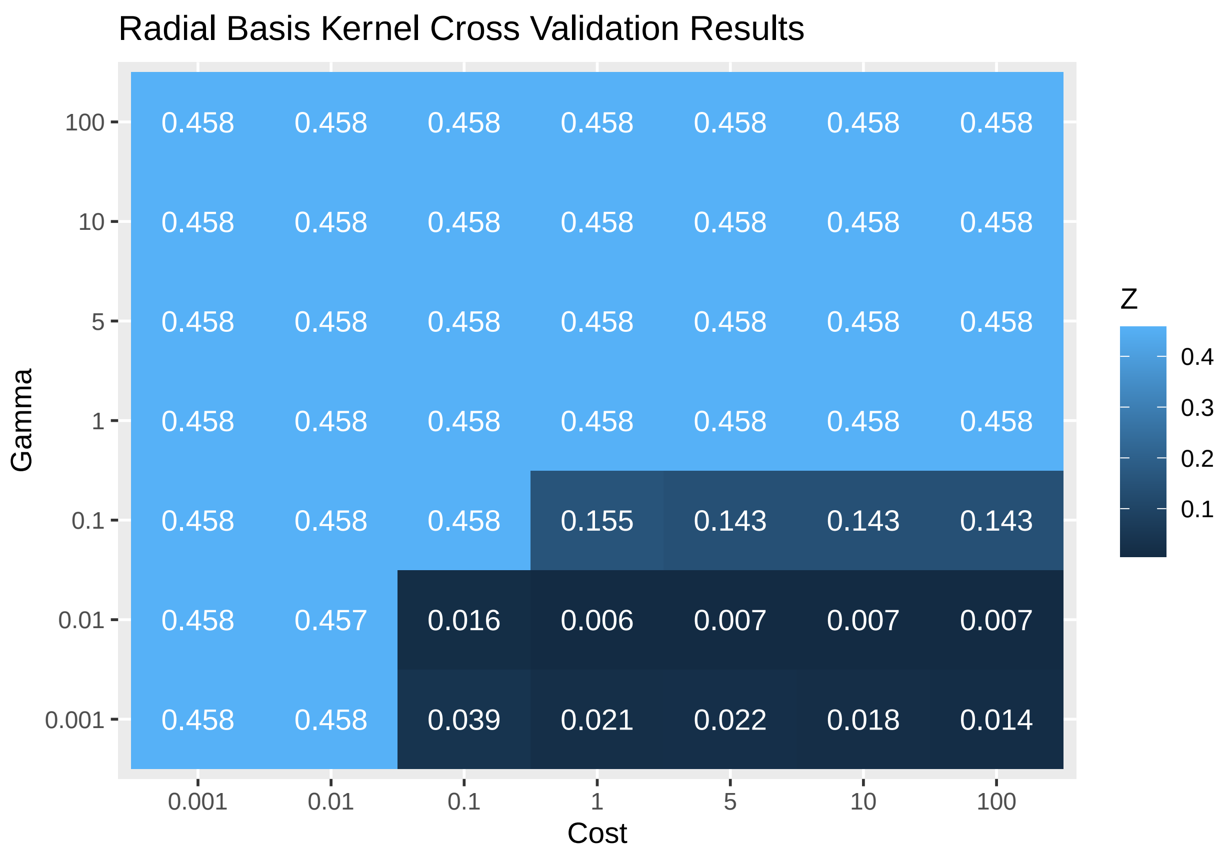
```r
plot(error~log10(cost), linearcrossval$performances, type = "l", main = "Margin Parameter vs Error for L
```



b.

```r
x<-c("0.001", "0.01", "0.1", "1", "5", "10", "100")
y<-c("0.001", "0.01", "0.1", "1", "5", "10", "100")
data <- expand.grid(X=x, Y=y)
data$Z<-round(radialperformances$error,3)

ggplot(data, aes(X, Y, fill= Z)) +
  geom_tile()+
  geom_text(aes(label = Z), color = "white", size = 4)+
  labs(title = "Radial Basis Kernel Cross Validation Results", x = "Cost", y = "Gamma")
```



2.

From our earlier results, we should run a linear kernel SVM with cost equal to 0.01 and a radial kernel with cost equal to 1 and Gamma equal to 0.01.

```
linsvm<-svm(digit~., digits_train, kernel = "linear", cost = 0.01, scale = FALSE)
radialsvm<-svm(digit~., digits_train, kernel = "radial", cost = 1, Gamma = 0.01, scale = FALSE)

sum(ifelse(predict(linsvm, digits_test)==digits_test$digit,0,1))/nrow(digits_test)
```

```
## [1] 0.02880658
```

```
sum(ifelse(predict(radialsvm, digits_test)==digits_test$digit,0,1))/nrow(digits_test)
```

```
## [1] 0.01234568
```

The radial kernel has a smaller test error under zero-one loss than the linear kernel, so a RBF kernel seems more ideal for this scenario.

Problem 2

i. Let $\phi(x) = x$

$$k^*(x, x') = \langle x, x' \rangle \qquad \forall x, x' \in \mathbb{R}^d$$
$$= \langle \phi(x), \phi(x') \rangle$$

$\Rightarrow \exists\, \phi : \mathbb{R}^d \to F$ s.t. $k^*(x, x') = \langle \phi(x), \phi(x') \rangle_F$
$\forall\, x, x' \in \mathbb{R}^d$

Definition of a kernel $\Rightarrow k^*(x, x')$ is a valid kernel

ii. $k_1^*(x, x')$ is a valid kernel $\Rightarrow \exists\, \phi : \mathbb{R}^d \to F$ s.t. $k_1^*(x, x') = \langle \phi(x), \phi(x') \rangle_F$
$\forall\, x, x' \in \mathbb{R}^d$

Let $\tilde{\phi}(x) = \sqrt{a}\, \phi(x)$

$$k^*(x, x') = a\, k_1^*(x, x') \qquad \forall x, x' \in \mathbb{R}^d$$
$$= a \langle \phi(x), \phi(x') \rangle$$
$$= \langle \sqrt{a}\, \phi(x), \sqrt{a}\, \phi(x') \rangle$$
$$= \langle \tilde{\phi}(x), \tilde{\phi}(x') \rangle$$

$\Rightarrow \exists\, \tilde{\phi} : \mathbb{R}^d \to F$ s.t. $k^*(x, x') = \langle \tilde{\phi}(x), \tilde{\phi}(x') \rangle_F$
$\forall\, x, x' \in \mathbb{R}^d$

Definition of a kernel $\Rightarrow k^*(x, x')$ is a valid kernel iii.

$k_1(x, x')$ is a valid kernel $\Rightarrow$ $k_1(x, x') = k_1(x', x)$
and therefore symmetric

$$\Rightarrow k^*(x, x') = g(x)\, k_1(x, x')\, g(x')$$
$$= g(x)\, k_1(x', x)\, g(x')$$

commutativity $\Rightarrow$ $= g(x')\, k_1(x', x)\, g(x)$

$$= k^*(x', x)$$

$\Rightarrow$ k^* is symmetric

Additionally,

$k_1(x, x)$ is a valid kernel $\Rightarrow$ $f' K_1 f \geq 0 \quad \forall f \in \mathbb{R}^n$
and therefore positive-definite

$$f' K^* f = \sum_{ij} f_i\, K^*(x_i, x_j)\, f_j$$
$$= \sum_{ij} f_i\, g(x_i)\, k_1(x_i, x_j)\, g(x_j)\, f_j$$

Let $h = f_i g(x_i)$,

$$\Rightarrow = \sum_{ij} h_i\, k_1(x_i, x_j)\, h_j$$

$$= h' K_1 h$$

K_1 positive definite $\Rightarrow$ $h' K_1 h \geq 0$
$\Rightarrow f' K^* f \geq 0 \quad \forall f \in \mathbb{R}^n$
$\Rightarrow k^*$ is positive definite

Mercer's Theorem $\Rightarrow$ $k^*(x, x')$ is a valid kernel.

By spectral decomposition,

iv. $K_1 := \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j(x')$

$K_2 := \sum_{j=1}^{\infty} \gamma_j \theta_j(x) \theta_j(x')$

$\therefore K_1(x, x') K_2(x, x') = K^*(x, x')$

$= \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j(x') \sum_{j=1}^{\infty} \gamma_j \theta_j(x) \theta_j(x')$

Let $c_j := \lambda_j \sum_{j=1}^{\infty} \gamma_j$ and $\phi_j'(x) := \phi_j(x) \sum_{j=1}^{\infty} \theta_j(x)$

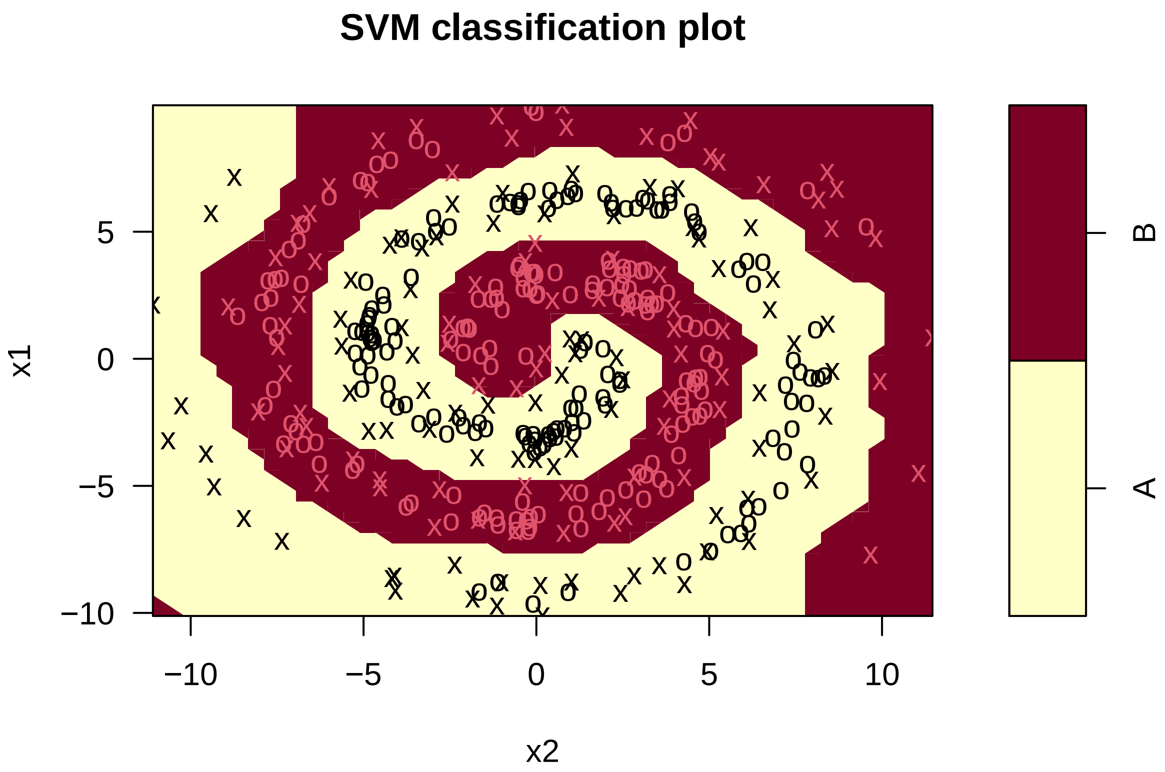
$\therefore = \sum_{j=1}^{\infty} c_j \phi_j'(x) \phi_j'(x')$

Mercer's Theorem

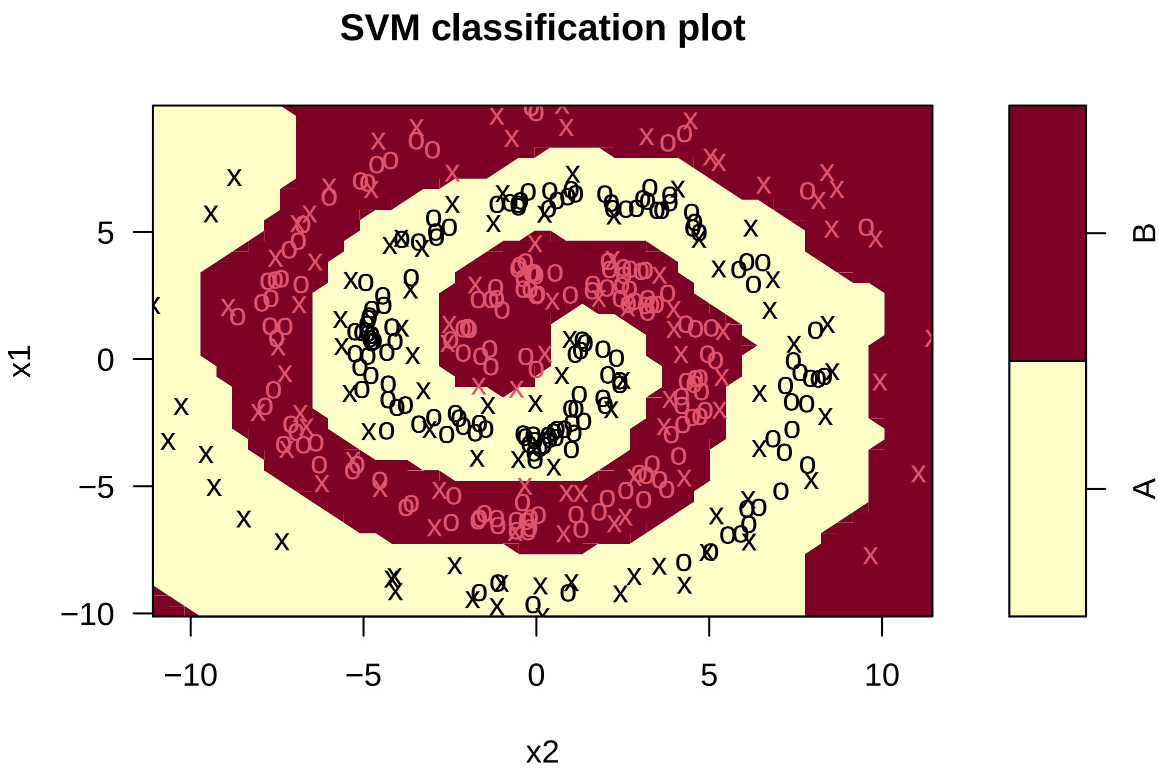
$\Rightarrow K^*(x, x')$ is a valid kernel

v.

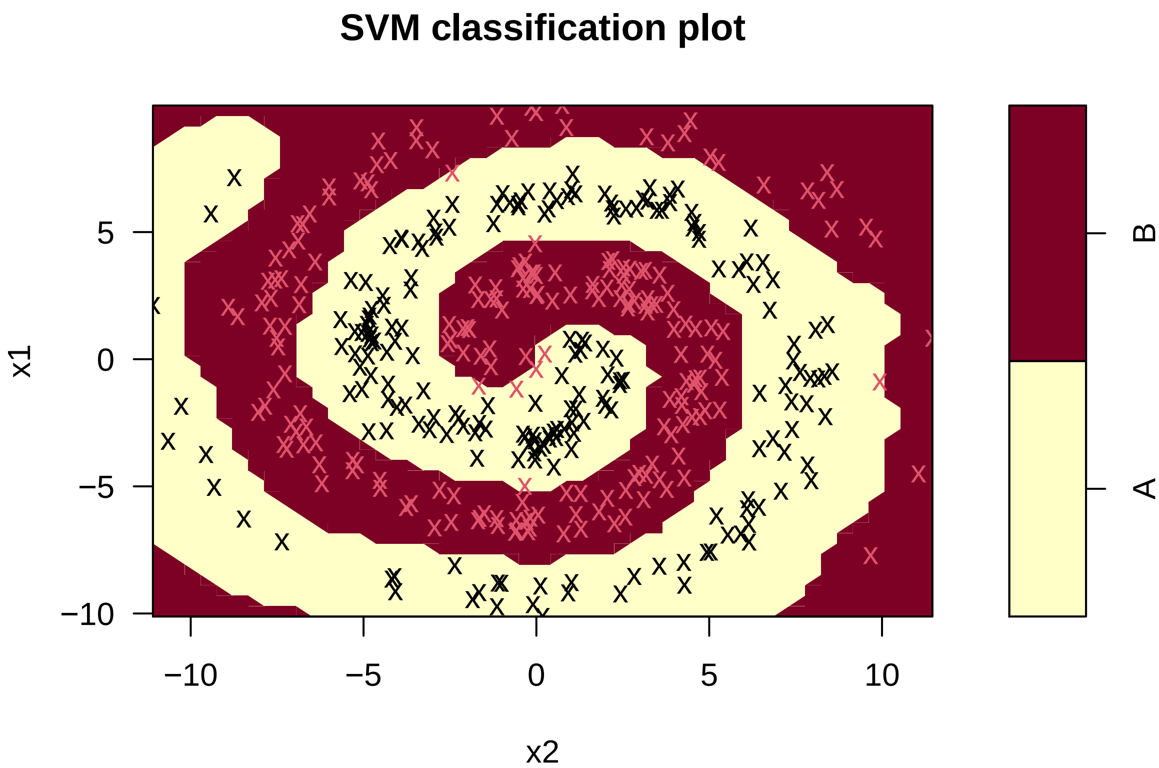
```
radialdata<-read.csv("HW3Problem2.csv")
pr2pt51<-svm(as.factor(class)~., data = radialdata, kernel = "radial", cost = 1, Gamma = 0.05, scale = F
plot(pr2pt51, radialdata)
```



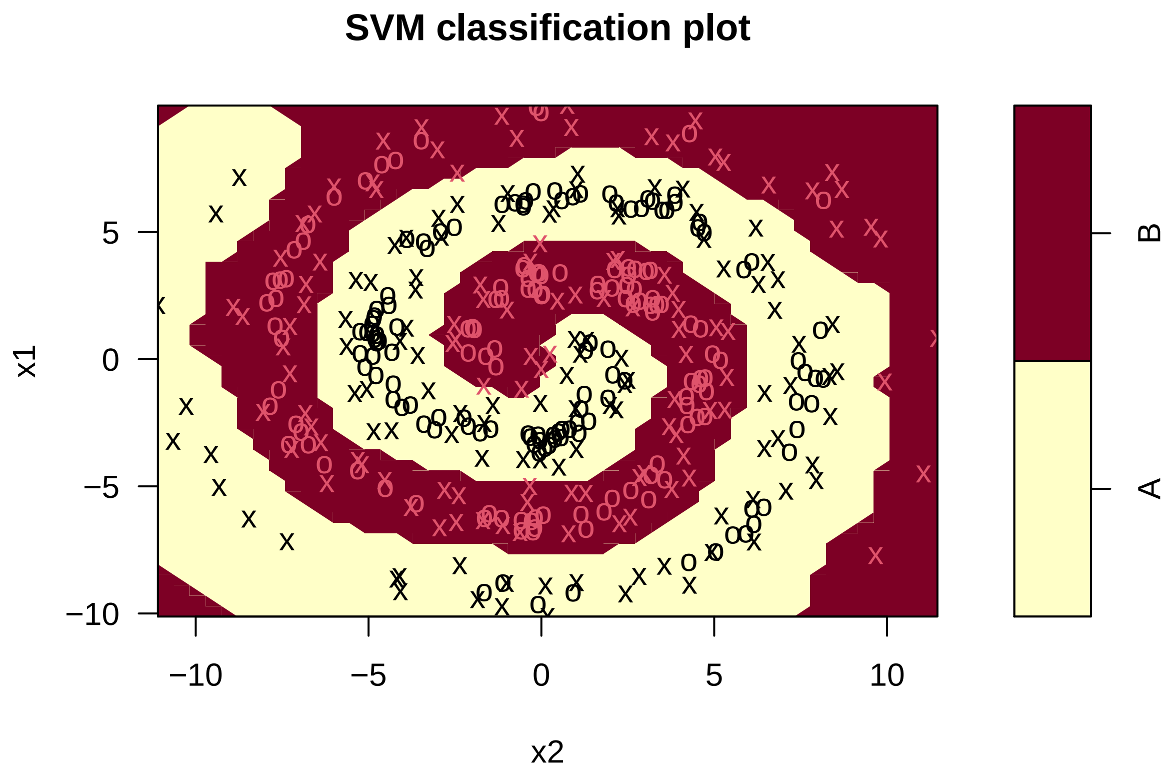

```
pr2pt52<-svm(as.factor(class)~., data = radialdata, kernel = "radial", cost = 5, Gamma = 0.05, scale = F
plot(pr2pt52, radialdata)
```




```r
pr2pt53<-svm(as.factor(class)~., data = radialdata, kernel = "radial", cost = 0.01, Gamma = 0.02, scale
plot(pr2pt53, radialdata)
```



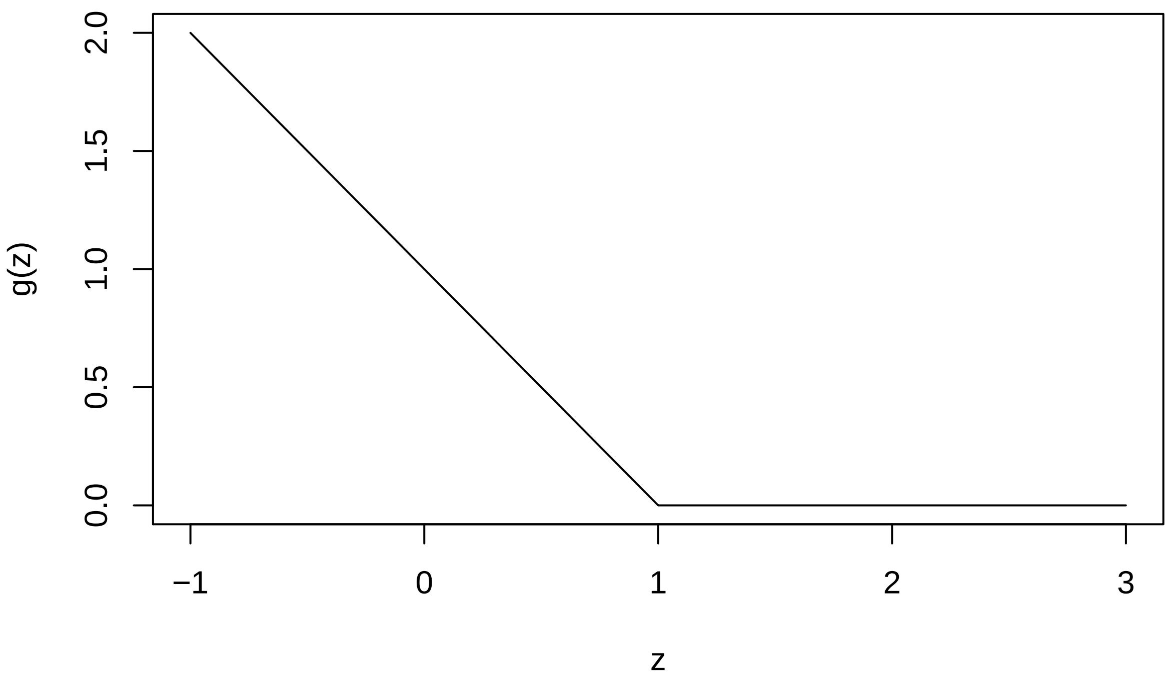

```
pr2pt54<-svm(as.factor(class)~., data = radialdata, kernel = "radial", cost = 0.5, Gamma = 0.05, scale =
plot(pr2pt54, radialdata)
```



Problem 3

i.

```
f1<-function(x){
  return(max(1-x,0))
}
xes<-seq(-1,3,.01)
yes<-sapply(xes,f1)
plot(xes,yes, type = "l", xlab = "z", ylab = "g(z)")
```



g(z)
2.0
1.5
1.0
0.5
0.0
−1
0
1
2
3
z

ii.

For $z \leq 1$

By above, $Q_i(v_H, c) =$

$$\max\{0, 1 - y_i(w_1 x_{i_1} + w_2 x_{i_2} + c)\} + \frac{\lambda}{2}(w_1^2 + w_2^2)$$

Since $z \leq 1$, $\max\{0, 1 - y_i(w_1 x_{i_1} + w_2 x_{i_2} + c\} = 1 - y_i(w_1 x_{i_1} + w_2 x_{i_2} + c)$

$\therefore \dfrac{\partial Q_i}{\partial w_1} = -y_i x_{i_1} + \lambda w_1$

$\therefore \dfrac{\partial Q_i}{\partial w_2} = -y_i x_{i_2} + \lambda w_2$

$\therefore \dfrac{\partial Q_i}{\partial c} = -y_i$

For $z > 1$

Since $z > 1$,

$\max\{0, 1 - y_i(w_1 x_{i_1} + w_2 x_{i_2} + c)\} = 0$

$\therefore \dfrac{\partial Q_i}{\partial w_1} = \lambda w_1$

$\dfrac{\partial Q_i}{\partial w_2} = \lambda w_2$

$\dfrac{\partial Q_i}{\partial c} = 0$

iii.

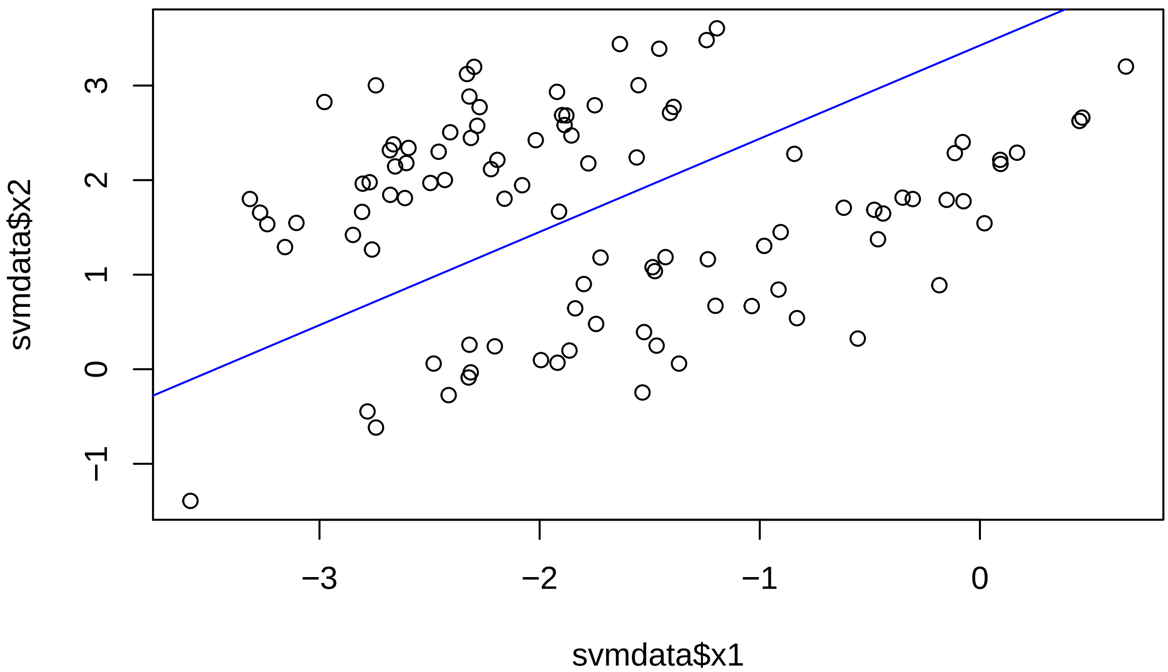
```
svmdata <- read.csv("svmdata.csv")
x1<-svmdata$x1
x2<-svmdata$x2
y<-svmdata$y
w1<-0
w2<-0
c<-0
lambda<-0.25

svmgraddesc<-function(y,x1,x2, lambda, epsilon, maxiter, w1_0,w2_0,c_0){
  w1<-w1_0
  w2<-w2_0
  c<-c_0
  convergence <- FALSE
  iterations <- 1
  while(convergence == FALSE & iterations < maxiter){
    eta<-1/(iterations*lambda)
    gradient_i<-matrix(NA, length(x1),3)
    for (i in 1:length(x1)){
      if((y[i]*(w1*x1[i]+w2*x2[i]+c))<1){
        gradient_i[i,]<- c(-y[i], -y[i]*x1[i]+lambda*w1, -y[i]*x2[i]+lambda*w2)
      }
      if((y[i]*(w1*x1[i]+w2*x2[i]+c))>1){
         gradient_i[i,]<- c(0, lambda*w1, lambda*w2)
        }
    }
    c<-c-(eta*mean(gradient_i[,1]))
    w1<-w1-(eta*mean(gradient_i[,2]))
    w2<-w2-(eta*mean(gradient_i[,3]))
    if(abs(mean(gradient_i[,1]))+abs(mean(gradient_i[,2]))+abs(mean(gradient_i[,3]))<epsilon){
      convergence<-TRUE
    }
    iterations<-iterations+1
  }
  return(c(c,w1,w2))
}

test_svm<-svmgraddesc(y,x1,x2,0.3,0.001,10000,0,0,0)
```

```
c<-test_svm[1]
w1<-test_svm[2]
w2<-test_svm[3]

plot(svmdata$x1, svmdata$x2)
abline(-c/w2, -w1/w2, col = "blue", xlab ="x1", ylab = "x2")
```




```r
## Coefficients:
c(c,w1,w2)
```

```
## [1] -2.6736027 -0.7697816  0.7808708
```