

```

---
title: "homework7"
author: "Travis Barton"
date: "11/19/2018"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(e1071)
library(LilRhino)
library(ggplot2)
```

```{r #1}
##LOAD USPS
names(test) = names(train)

PCA on USPS

pca <- prcomp(train[,-257])
which(cumsum(pca$sdev^2)/sum(pca$sdev^2) < .95)
88 needed

train95 <- as.matrix(train[,-257]) %*% pca$rotation[,1:88]
train95 <- as.data.frame(train95)
y_train <- train$label

test95 <- as.matrix(test[,-257]) %*% pca$rotation[,1:88]
y_test <- test$label

annoying = rbind(train95, test95)
annoying <- cbind(annoying, c(y_train, y_test))
colnames(annoying) = c(colnames(train95), 'label')
SVM time

#set c
Cs <- c(2^-6, 2^-5, 2^-4, 2^-3, 2^-2, 2^-1, 1, 2, 4, 8, 16)

#set loop
acc = {}
for(i in 1:length(Cs))
{

```

```

 svmmod <- svm(label~., data = annoying[1:nrow(train95),],
kernel = 'linear', type = 'C-classification', cost = Cs[i])
 preds <- predict(svmmod, annoying[-c(1:nrow(train95)),])
 acc = c(acc, Percent(y_test, preds))
}
acc = 1-acc
Graph it

m <- ggplot(data = as.data.frame(acc), aes(x = seq(1,
length(Cs)), y = acc)) + geom_line(aes(x = seq(1,
length(Cs)), y = acc), colour = "red") +
 ylab("error") + xlab("C Values (linearly scaled)") +
 scale_x_continuous(breaks=seq(1, length(Cs)), labels =
as.character(Cs), limits = c(min(Cs), max(Cs))) +
theme(axis.text.x = element_text(angle=45))

m
Construct graph as you have done

Get the breaks
bs <- ggplot_build(m)[[2]]$ranges[[1]]$x.major_source

add new labels
m + scale_x_discrete(breaks=bs, labels = as.character(Cs))

...
```{r #2}
acc = {}
for(i in 1:length(Cs))
{
  svmmod <- svm(label~., data = annoying[1:nrow(train95),],
kernel = 'polynomial', type = 'C-classification', cost =
Cs[i], degree = 3)
  preds <- predict(svmmod, annoying[-c(1:nrow(train95)),])
  acc = c(acc, Percent(y_test, preds))
}
acc = 1-acc
m <- ggplot(data = as.data.frame(acc), aes(x = seq(1,
length(Cs)), y = acc)) + geom_line(aes(x = seq(1,
length(Cs)), y = acc), colour = "red") +
  ylab("error") + xlab("C Values (linearly scaled)") +
  scale_x_continuous(breaks=seq(1, length(Cs)), labels =

```

```
as.character(Cs), limits = c(min(Cs), max(Cs))) +
theme(axis.text.x = element_text(angle=45))
```

```
m
```
```

```
```{r #3}
```

```
annoyingtemp <- annoying
class <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
results <- matrix(0, nrow = nrow(test), ncol = 10)
acc <- {}
for(j in 1:length(Cs))
{
  for(i in 1:10){
    annoyingtemp$label <-
ifelse(annoying$label==class[i],1,0)
    svmmod <- svm(label~., data =
annoyingtemp[1:nrow(train95),], kernel = 'polynomial', type
= 'C-classification', cost = Cs[j], degree = 3)
    preds <- predict(svmmod, annoyingtemp[-
c(1:nrow(train95)),])
    results[,i] = preds
  }
  finals <- {}
  for(i in 1:nrow(test))
  {
    can <- max(which(results[i,] == max(results[i,]))) -1)
    finals = c(finals, can)
  }
  acc <- c(acc, Percent(test$label, finals))
}
```

```
Codes_done("Done", "check that shit", sound = T)
```

```
##### New shot #####
```

```
library(probsvm)
svmmod <- probsvm(train95, y_train, kernel = 'polynomial',
kparam = 3)
```

```

acc = 1-acc
m <- ggplot(data = as.data.frame(acc), aes(x = seq(1,
length(Cs)), y = acc)) + geom_line(aes(x = seq(1,
length(Cs)), y = acc), colour = "red") +
  ylab("error") + xlab("C Values (linearly scaled)") +
  scale_x_continuous(breaks=seq(1, length(Cs)), labels =
as.character(Cs), limits = c(min(Cs), max(Cs))) +
theme(axis.text.x = element_text(angle=45))

```

```

m
```

```

```

```{r #4}
library(FNN)
library(kknn)
data = annoying[1:nrow(train95),]
choosing_sigma <- function(data, k)
{
  #1 sample the 100 points
  newdata <- data[sample(c(1:nrow(data)), 100),]

  #2 get kth neighbor for a given point
  goalie = 0
  for(i in 1:100)
  {
    point = newdata[i,]
    dist = get.knnx(data[which(point$label ==
data$label),-89], point[-89], k = k+1)$nn.dist[k+1]
    goalie = goalie + dist
  }
  print(goalie)
  goalie = (goalie)/100
  return(goalie)
}
sigmas <- {}
for(i in 3:8)
{
  sigmas <- c(sigmas, choosing_sigma(data, i))
}
acc <- {}
gam = 1/(2*sigmas^2)

```

```

for(i in 1:length(sigmas))
{
  svmmod2 <- svm(label~., data = annoying[1:7290,], type =
'C-classification', gamma = gam[i])
  preds <- predict(svmmod2, annoying[-c(1:7290),-89])
  acc = c(acc, Percent(test$label, preds))
}
gamm = 0.01045944

newacc <- {}
for(i in 1:length(Cs))
{
  svmmod <- svm(label~., data = annoying[1:7290,], gamma =
gamm, cost = Cs[i], type = 'C-classification')
  preds <- predict(svmmod, annoying[-c(1:7290),-89])
  newacc <- c(newacc, Percent(test$label, preds))
}
knnacc <- {}
for(i in 3:10)
{
  knnmod <- kknns(label~., as.data.frame(annoying[1:7290,]),
as.data.frame(annoying[-c(1:7290),]), k = i, kernel =
"gaussian")
  preds <- apply(knnmod$CL, 1, median)
  knnacc <- c(knnacc, Percent(test$label, round(preds)))
}
Codes_done("dome", "dome", sound = T)
newacc <- 1-newacc
knnacc <- 1-knnacc

m <- ggplot(data = as.data.frame(acc), aes(x = seq(1,
length(Cs)), y = acc)) +
  geom_line(aes(x = seq(1, length(Cs)), y = acc), colour =
"red") +
  geom_line(aes(x = seq(1, length(Cs)), y = c(knnacc, 1, 1,
1)), colour = "green") +
  ylab("error") + xlab("C Values (linearly scaled)") +
  scale_x_continuous(breaks=seq(1, length(Cs)), labels =
as.character(Cs), limits = c(min(Cs), max(Cs))) +
  theme(axis.text.x = element_text(angle=45))

```

m

```

```
```{r #5}
library(progress)
pb = txtProgressBar(min = 0, max = 12, title = "Last one
baby", style = 3)
acc <- {}
for(j in 1:11)
{
  setTxtProgressBar(pb, j)
  for(i in 1:10){
    annoyingtemp$label <-
ifelse(annoying$label==class[i],1,0)
    svmmod <- svm(label~., data =
annoyingtemp[1:nrow(train95),], type = 'C-classification',
cost = Cs[j], gamma = gamm)
    preds <- predict(svmmod, annoyingtemp[-
c(1:nrow(train95)),])
    results[,i] = preds
  }
  finals <- {}
  for(i in 1:nrow(test))
  {
    can <- max(which(results[i,] == max(results[i,]))) -1)
    finals = c(finals, can)
  }
  acc <- c(acc, Percent(test$label, finals))
}
Codes_done("done", "done", sound = T)
acc <- 1-acc
m <- ggplot(data = as.data.frame(newacc), aes(x = seq(1,
length(Cs)), y = newacc)) +
  geom_line(aes(x = seq(1, length(Cs)), y = newacc), colour
= "red") +
  geom_line(aes(x = seq(1, length(Cs)), y = c(knnacc, 1, 1,
1)), colour = "green") +
  ylab("error") + xlab("C Values (linearly scaled)") +
  scale_x_continuous(breaks=seq(1, length(Cs)), labels =
```

```
as.character(Cs), limits = c(min(Cs), max(Cs))) +  
theme(axis.text.x = element_text(angle=45))
```

```
m  
```\
```