

Homework 5

Yiying Wu (yw3996)

R packages

```
library(tidyverse)
library(caret)
library(tidymodels)
library(ISLR)
library(e1071)
```

1. auto.csv data

In this problem, we will apply support vector machines to predict whether a given car gets high or low gas mileage based on the dataset “auto.csv” (used in Homework 3; see Homework 3 for more details of the dataset). The response variable is mpg cat. The predictors are cylinders, displacement, horsepower, weight, acceleration, year, and origin. Split the dataset into two parts: training data (70%) and test data (30%).

Input dataset

```
dat<-read_csv("./data/auto.csv")%>%
  mutate(
    mpg_cat = as.factor(mpg_cat),
    origin = as.factor(origin))
dat <- dat%>%
  na.omit()
```

Response: mpg_cat

```
contrasts(dat$mpg_cat)
```

```
##      low
## high   0
## low    1
```

Split the dataset into two parts: training data (70%) and test data (30%).

```
set.seed(1)
data_split <- initial_split(dat, prop = 0.7)

# Extract the training and test data
training_data <- training(data_split)
testing_data <- testing(data_split)
```

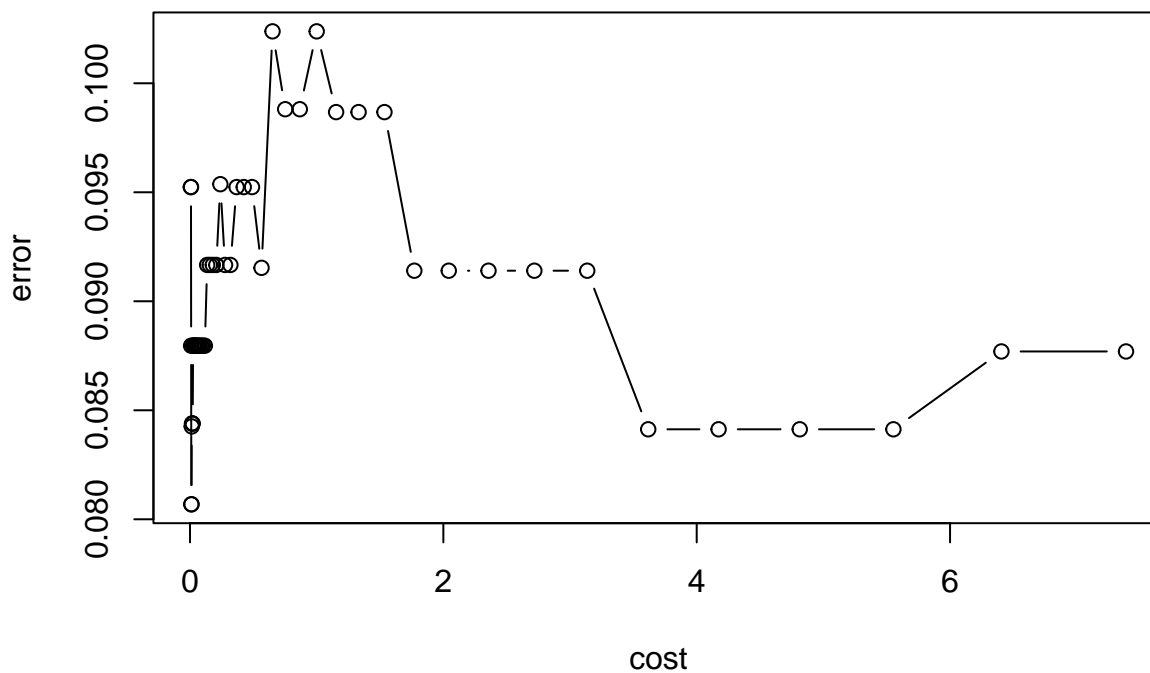
```
ctrl <- trainControl(method = "cv", number = 10,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)
```

(a) Fit a support vector classifier to the training data. What are the training and test error rates?

```
set.seed(1)
linear.tune <- tune.svm(mpg_cat ~ . ,
                       data = training_data,
                       kernel = "linear",
                       cost = exp(seq(-5,2, len = 50)),
                       scale = TRUE)

plot(linear.tune)
```

Performance of 'svm'



```
# summary(linear.tune)
linear.tune$best.parameters
```

```
##          cost
## 5 0.01193152
```

```
best.linear <- linear.tune$best.model
summary(best.linear)
```

```
##
```

```
## Call:
## best.svm(x = mpg_cat ~ ., data = training_data, cost = exp(seq(-5,
##      2, len = 50)), kernel = "linear", scale = TRUE)
##
##
## Parameters:
##   SVM-Type:  C-classification
##   SVM-Kernel: linear
##      cost:  0.01193152
##
## Number of Support Vectors:  123
##
## ( 62 61 )
##
##
## Number of Classes:  2
##
## Levels:
##   high low
```

```
pred.linear <- predict(best.linear, newdata = testing_data)

confusionMatrix(data = pred.linear,
                 reference = testing_data$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction high low
##      high    56   10
##      low     1    51
##
##              Accuracy : 0.9068
##              95% CI : (0.8393, 0.9525)
##      No Information Rate : 0.5169
##      P-Value [Acc > NIR] : < 2e-16
##
##              Kappa : 0.8143
##
## Mcnemar's Test P-Value : 0.01586
##
##              Sensitivity : 0.9825
##              Specificity : 0.8361
##              Pos Pred Value : 0.8485
##              Neg Pred Value : 0.9808
##              Prevalence : 0.4831
##              Detection Rate : 0.4746
##      Detection Prevalence : 0.5593
##              Balanced Accuracy : 0.9093
##
##              'Positive' Class : high
##
```

The **training error rate** is 0.0119.

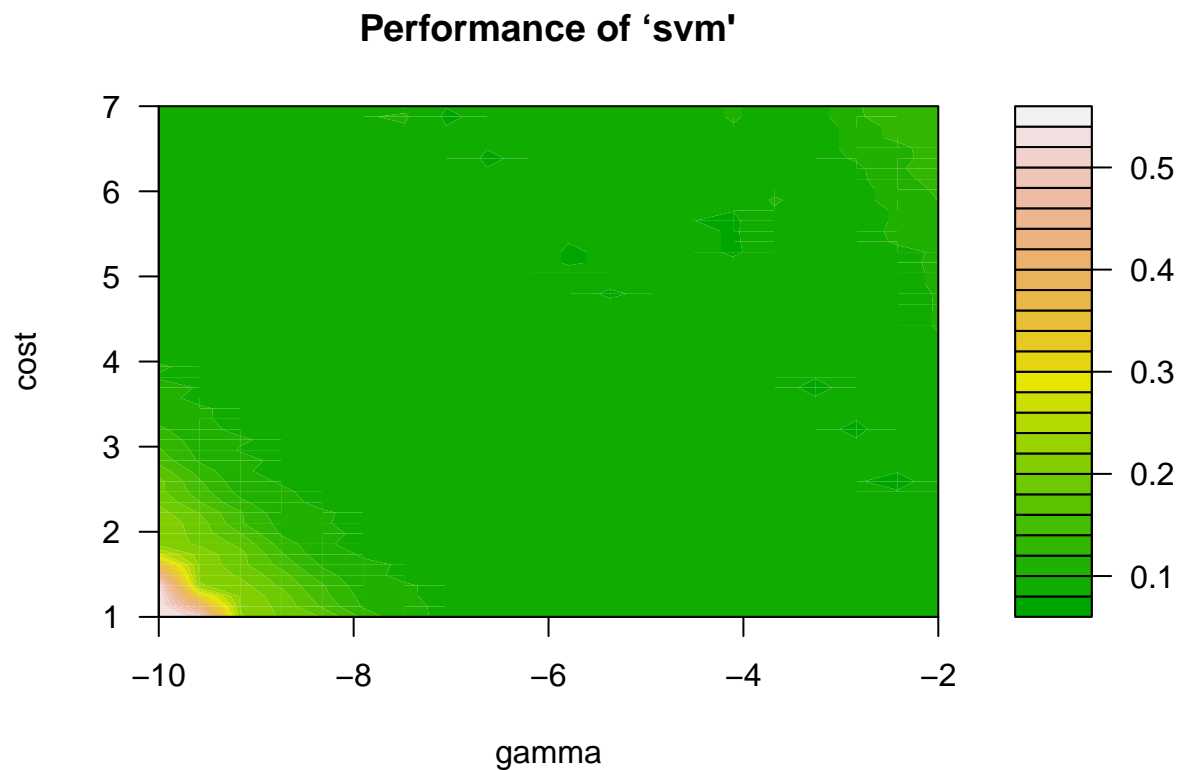
Test Error Rate = $1 - \text{Accuracy} = 1 - 0.9068 = 0.0932$

The **test error rate** for the model on the testing data is approximately 0.0932.

(b) Fit a support vector machine with a radial kernel to the training data. What are the training and test error rates?

```
set.seed(1)
radial.tune <- tune.svm(mpg_cat ~ . ,
  data = training_data,
  kernel = "radial",
  cost = exp(seq(1, 7, len = 50)),
  gamma = exp(seq(-10, -2, len = 20)))

plot(radial.tune, transform.y = log, transform.x = log,
  color.palette = terrain.colors)
```



```
# summary(radial.tune)
```

```
radial.tune$best.parameters
```

```
##           gamma      cost
## 715 0.01648568 197.4952
```

```
best.radial <- radial.tune$best.model
summary(best.radial)
```

```
##
## Call:
## best.svm(x = mpg_cat ~ ., data = training_data, gamma = exp(seq(-10,
##      -2, len = 20)), cost = exp(seq(1, 7, len = 50)), kernel = "radial")
##
##
## Parameters:
##   SVM-Type:  C-classification
##   SVM-Kernel: radial
##       cost: 197.4952
##
## Number of Support Vectors: 63
##
## ( 32 31 )
##
##
## Number of Classes: 2
##
## Levels:
##   high low
```

```
pred.radial <- predict(best.radial, newdata = testing_data)

confusionMatrix(data = pred.radial,
                 reference = testing_data$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high low
##      high  56   9
##      low   1  52
##
##              Accuracy : 0.9153
##              95% CI : (0.8497, 0.9586)
##      No Information Rate : 0.5169
##      P-Value [Acc > NIR] : < 2e-16
##
##              Kappa : 0.8311
##
## Mcnemar's Test P-Value : 0.02686
##
##              Sensitivity : 0.9825
##              Specificity : 0.8525
##              Pos Pred Value : 0.8615
##              Neg Pred Value : 0.9811
##              Prevalence : 0.4831
##              Detection Rate : 0.4746
##      Detection Prevalence : 0.5508
##              Balanced Accuracy : 0.9175
##
##              'Positive' Class : high
##
```

2. USArrests data

- (a) Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states. Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters?
- (b) Hierarchically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one.
- (c) Does scaling the variables change the clustering results? Why? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed?