Homework 5

Minghe Wang

2025-05-03

1

```
auto <- read_csv("./auto.csv")
```

```
## Rows: 392 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): mpg_cat
## dbl (7): cylinders, displacement, horsepower, weight, acceleration, year, or...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#head(auto)

str(auto)
```

```
## spc_tbl_ [392 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ cylinders   : num [1:392] 8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num [1:392] 307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num [1:392] 130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num [1:392] 3504 3693 3436 3433 3449 ...
##  $ acceleration: num [1:392] 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num [1:392] 70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num [1:392] 1 1 1 1 1 1 1 1 1 1 ...
##  $ mpg_cat     : chr [1:392] "low" "low" "low" "low" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   cylinders = col_double(),
##   ..   displacement = col_double(),
##   ..   horsepower = col_double(),
##   ..   weight = col_double(),
##   ..   acceleration = col_double(),
##   ..   year = col_double(),
##   ..   origin = col_double(),
##   ..   mpg_cat = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

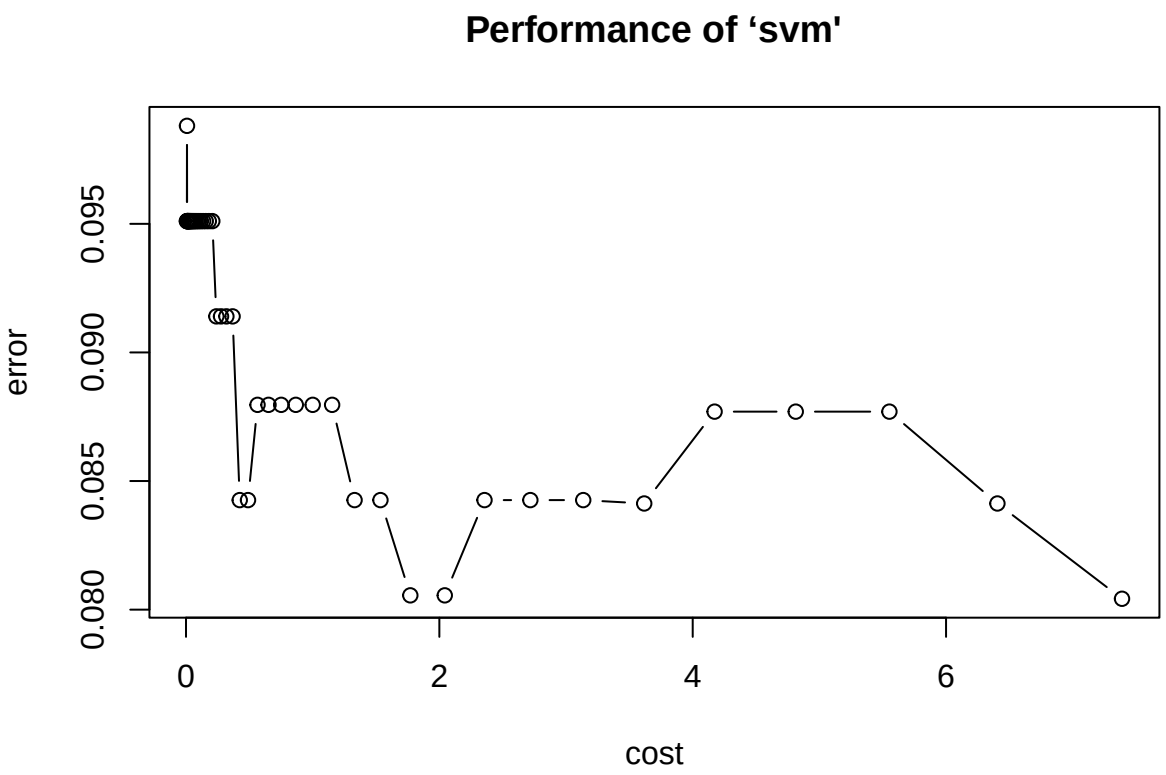
```
auto$origin <- as.factor(auto$origin)
auto$cylinders <- as.factor(auto$cylinders)
auto$mpg_cat <- as.factor(auto$mpg_cat)
contrasts(auto$mpg_cat)
```

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

```
auto <- na.omit(auto)

set.seed(1)
split_auto <- initial_split(auto, prop = 0.7)
training_data <- training(split_auto)
testing_data <- testing(split_auto)
```

a

```
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) # tuning curve
```




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

```
##        cost
## 50 7.389056
```

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

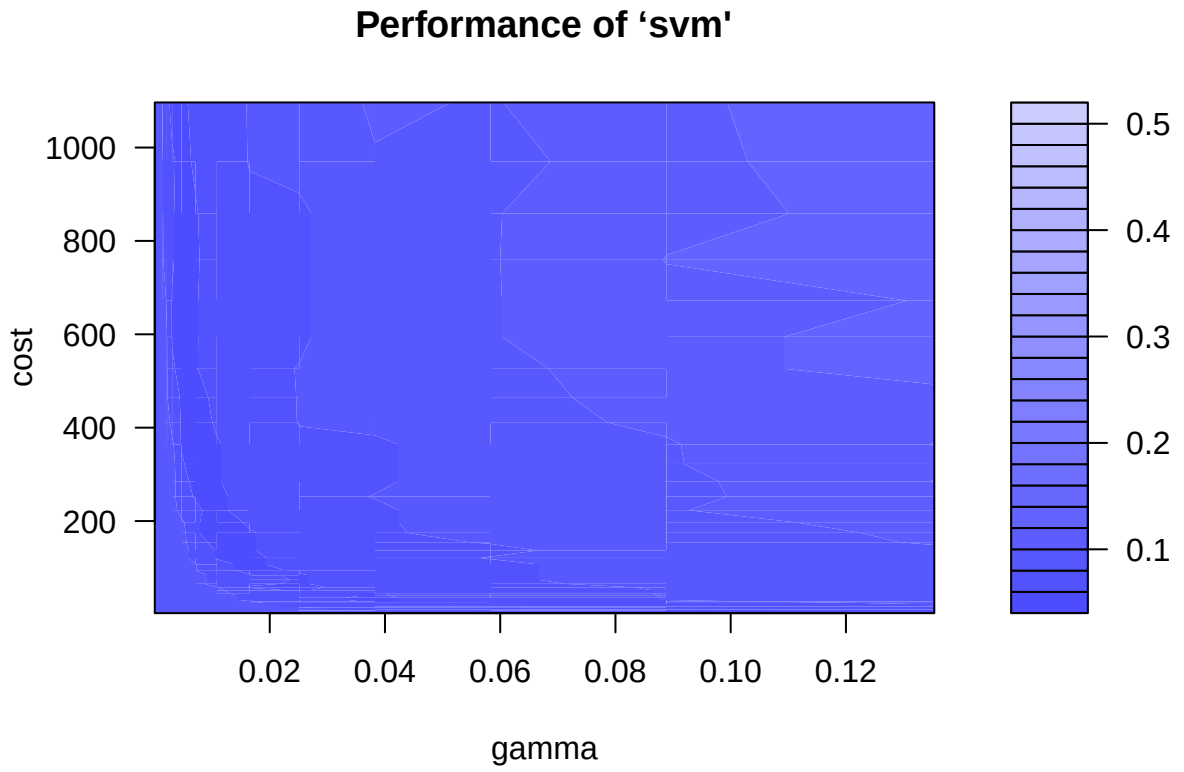
```
##
## Call:
## best.svm(x = mpg_cat ~ ., data = training_data, cost = exp(seq(-5,
##     2, len = 50)), kernel = "linear", scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  7.389056
##
## Number of Support Vectors:  50
##
##  ( 26 24 )
##
##
## Number of Classes:  2
##
## Levels:
##  high low
```

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

cm.train <- confusionMatrix(data = pred.train,
                reference = training_data$mpg_cat)
cm.test <- confusionMatrix(data = pred.test,
                reference = testing_data$mpg_cat)

train_error <- 1 - cm.train$overall['Accuracy']
test_error <- 1 - cm.test$overall['Accuracy']
```

For linear SVM, train and test error rates are: 0.0583942, 0.0932203.

b

```
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) # tuning curve
```




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

```
##          gamma     cost
## 694 0.01082047 174.7341
```

```
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:  174.7341
##
## Number of Support Vectors:  48
##
##  ( 25 23 )
##
##
## Number of Classes:  2
##
## Levels:
##  high low
```

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

```
cm.train_radial <- confusionMatrix(data = pred.train_radial,
                reference = training_data$mpg_cat)
cm.test_radial <- confusionMatrix(data = pred.test_radial,
                reference = testing_data$mpg_cat)

train_error_radial <- 1 - cm.train_radial$overall['Accuracy']
test_error_radial <- 1 - cm.test_radial$overall['Accuracy']
```

For radial kernelized SVM, train and test error rates are: 0.0474453, 0.0847458.

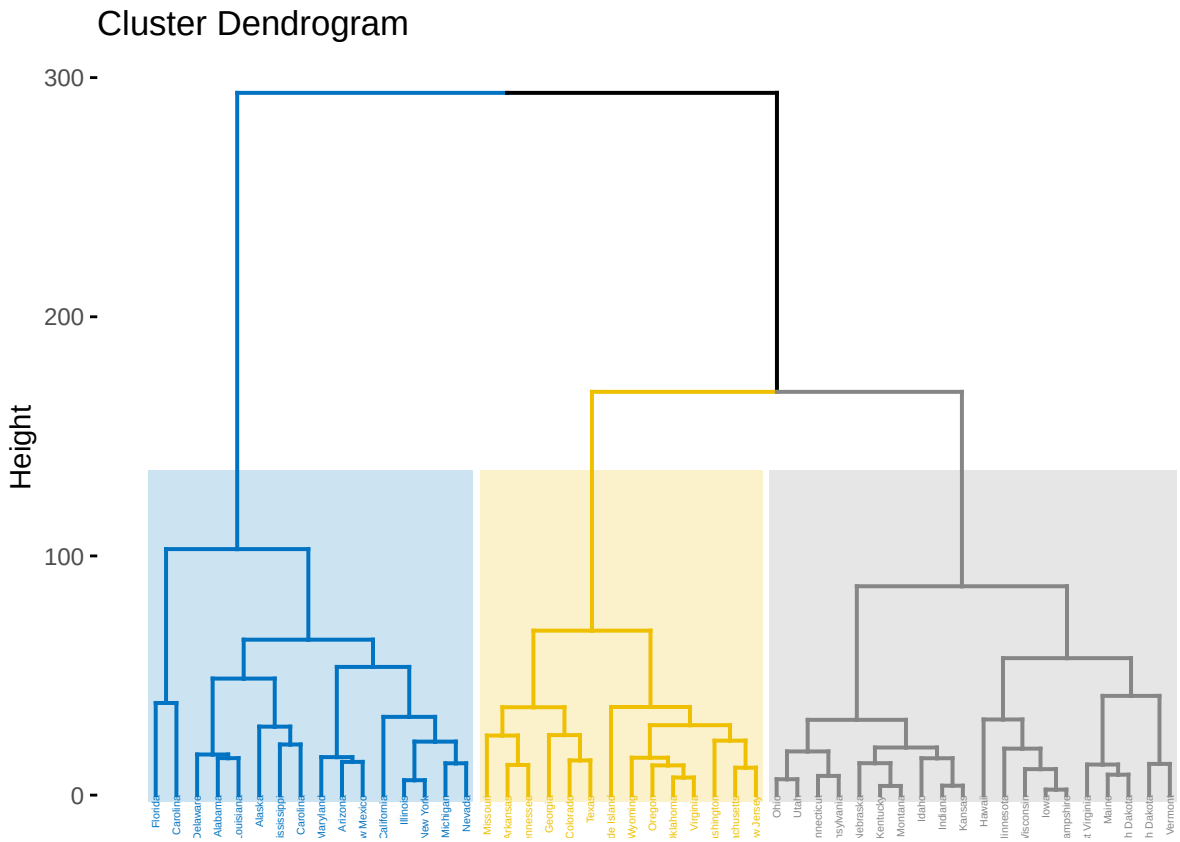
2

a

```
data("USArrests")
#head(USArrests)
```

```
hc.complete <- hclust(dist(USArrests), method = "complete")

fviz_dend(hc.complete, k = 3,
          cex = 0.3,
          palette = "jco", # color scheme; other palettes:"npg","aaas"...
          color_labels_by_k = TRUE,
          rect = TRUE, # whether to add a rectangle around groups.
          rect_fill = TRUE,
          rect_border = "jco",
          labels_track_height = 2.5)
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
ind3.complete <- cutree(hc.complete, 3)
state1 <- rownames(USArrests[ind3.complete == 1,])
state1_names <- state1[!startsWith(state1, "NA")]

state2 <- rownames(USArrests[ind3.complete == 2,])
state2_names <- state2[!startsWith(state2, "NA")]

state3 <- rownames(USArrests[ind3.complete == 3,])
state3_names <- state3[!startsWith(state3, "NA")]
```

Cluster 1 includes: Alabama, Alaska, Arizona, California, Delaware, Florida, Illinois, Louisiana, Maryland, Michigan, Mississippi, Nevada, New Mexico, New York, North Carolina, South Carolina

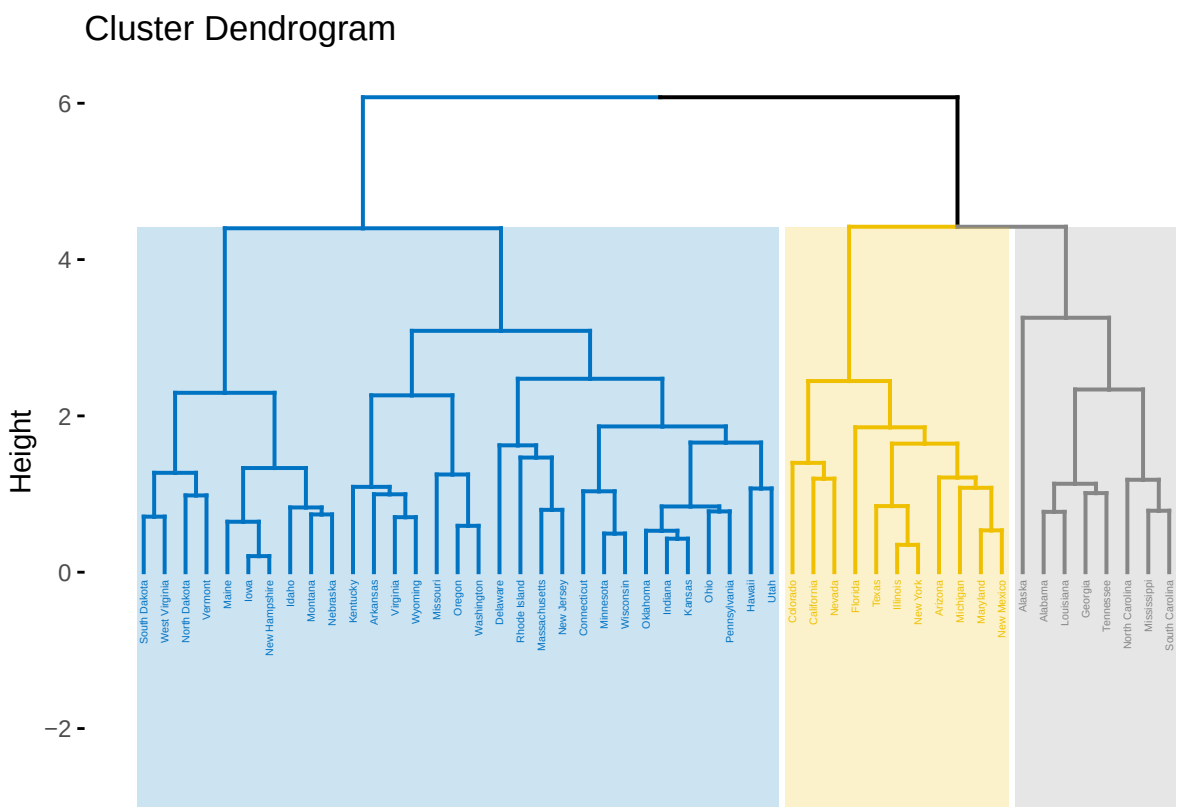
Cluster 2 includes: Arkansas, Colorado, Georgia, Massachusetts, Missouri, New Jersey, Oklahoma, Oregon, Rhode Island, Tennessee, Texas, Virginia, Washington, Wyoming

Cluster 3 includes: Connecticut, Hawaii, Idaho, Indiana, Iowa, Kansas, Kentucky, Maine, Minnesota, Montana, Nebraska, New Hampshire, North Dakota, Ohio, Pennsylvania, South Dakota, Utah, Vermont, West Virginia, Wisconsin

b

```
USArrests_scale <- scale(USArrests)

hc.complete_scaled <- hclust(dist(USArrests_scale), method = "complete")

fviz_dend(hc.complete_scaled, k = 3,
          cex = 0.3,
```

```
          palette = "jco", # color scheme; other palettes:"npg","aaas"...
          color_labels_by_k = TRUE,
          rect = TRUE, # whether to add a rectangle around groups.
          rect_fill = TRUE,
          rect_border = "jco",
          labels_track_height = 2.5)
```




```
ind3.complete_scaled <- cutree(hc.complete_scaled, 3)
state1_scale <- rownames(USArrests_scale[ind3.complete_scaled == 1,])
state1_names_scale <- state1_scale[!startsWith(state1_scale, "NA")]

state2_scale <- rownames(USArrests_scale[ind3.complete_scaled == 2,])
state2_names_scale <- state2_scale[!startsWith(state2_scale, "NA")]

state3_scale <- rownames(USArrests_scale[ind3.complete_scaled == 3,])
state3_names_scale <- state3_scale[!startsWith(state3_scale, "NA")]
```

After scaling:

Cluster 1 includes: Alabama, Alaska, Georgia, Louisiana, Mississippi, North Carolina, South Carolina, Tennessee

Cluster 2 includes: Arizona, California, Colorado, Florida, Illinois, Maryland, Michigan, Nevada, New Mexico, New York, Texas

Cluster 3 includes: Arkansas, Connecticut, Delaware, Hawaii, Idaho, Indiana, Iowa, Kansas, Kentucky, Maine, Massachusetts, Minnesota, Missouri, Montana, Nebraska, New Hampshire, New Jersey, North Dakota, Ohio, Oklahoma, Oregon, Pennsylvania, Rhode Island, South Dakota, Utah, Vermont, Virginia, Washington, West Virginia, Wisconsin, Wyoming

Scaling the variables changes the clustering results significantly because without scaling, variables with larger absolute ranges will dominate the Euclidean distance calculations. Thus, clustering would primarily reflect variation in high-magnitude variables, not all variables equally.

Yes, they should be scaled — especially when variables are measured in different units or have different variances, which is the case in `USArrests`. This ensures that all variables contribute equally to the distance calculations, and clustering reflects balanced structure across all features.