Homework 9: Support Vector Machines

Abra Brisbin

7/27/2021

Possible Solutions to Selected Questions

Important: This document contains R code solutions and example answers for problems posed in the DS740: Data Mining homework. We are intentionally sharing this document with learners *who have already completed the associated homework.* We want you to be able to review and troubleshoot your code, as well as ask questions to prepare you for later assessments. *By using this document, you are accepting responsibility* **not** *to share it with anyone else, including other students in the course or the program who might not have completed the homework yet.* By upholding this agreement, you are helping us use this tool with learners in future terms.

From Problem 1:

Question:

Which method do you expect will be better for categorizing the regions of species of oak trees: A support vector classifier or logistic regression? Explain.

Possible Answer: A support vector classifier, because there is little overlap between the groups.

Question:

Use `caret` to build a support vector classifier (a SVM with a linear kernel) to categorize the trees' regions based on their (unstandardized) logSize and logRange. Test the following values of cost: .001, .01, .1, 1, 5, 10, 100. Because of this data set's small size, use leave-one-out cross-validation. Enter your R code below.

Possible Answer:

```
data_used = oak

#ctrl = trainControl(method = "cv", number = 39) # This also works
ctrl = trainControl(method = "LOOCV")
fit_oak = train(Region ~ logSize + logRange,
               data = data_used,
               method = "svmLinear",
               tuneGrid = expand.grid(C = c(.001, .01, .1, 1, 5, 10, 100)),
               preProcess = c("center","scale"),
               trControl = ctrl)
```

Question:

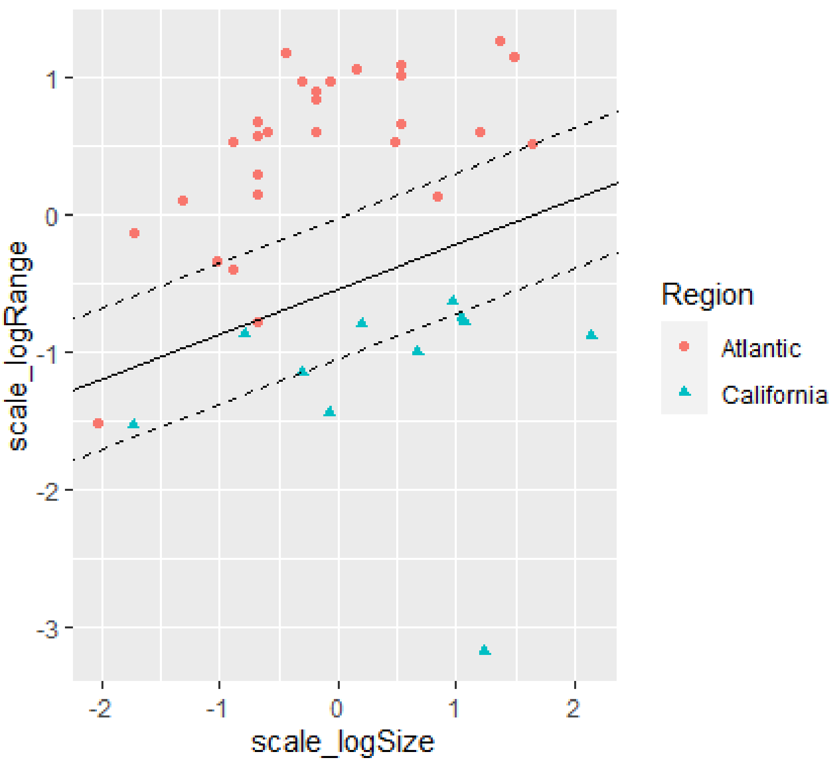
Make a graph showing the data points (similar to question 1), optimal hyperplane (line), and its margins. Include a legend.

- Use different colors and/or plotting characters to show the Regions of the points.
- You may use either the standardized or the unstandardized log(acorn size) and log(range).

Use *Insert -> Image* to upload your plot to this question on Canvas.

Possible Answer:

```
oak <- oak %>%
  tibble::rownames_to_column("Row") %>%
  mutate(is_SV = Row %in% attr(fit_oak$finalModel, "SVindex"))

b = attr(fit_oak$finalModel, "b")
coefs = attr(fit_oak$finalModel, "coef")[[1]]


oak_SV <- oak %>%
  filter(is_SV) %>%
  select(c(scale_logSize, scale_logRange)) %>%
  as.matrix()

#head(oak_SV) # Columns should be in the order x, y
              # relative to the graph

w = colSums(coefs * oak_SV) # beta_1, ... beta_p

oak %>%
  gf_point(scale_logRange ~ scale_logSize,
           color =~ Region,
           pch =~ Region) %>%
  gf_abline(intercept = b/w[2], slope = -w[1]/w[2]) %>%
  gf_abline(intercept = (b+1)/w[2], slope = -w[1]/w[2], lty = 2) %>%
  gf_abline(intercept = (b-1)/w[2], slope = -w[1]/w[2], lty = 2)
```



Question:

In question 3, we used `caret` to perform cross-validation for model selection (picking the best value of the cost parameter). In this question, we will use a *for* loop wrapper to perform a second layer of cross-validation. This will allow us to honestly assess the accuracy of our model-selection process.

- Set the random seed to 9.
- Create a vector to store the predicted regions (it should have length = the number of rows in the data set).
- Create vectors `groups` and `cv_groups` to perform 10-fold CV (like we did before learning about `caret`).
- Create a *for* loop to iterate through the folds of the outer layer of CV. Inside the *for* loop:
  - Create the variables `groupii`, `train_set`, and `test_set`, like we did for CV before learning about `caret`.
  - Use `caret` to perform an inner layer of LOOCV. `caret` should fit a support vector classifier and choose among costs of .001, .01, .1, 1, 5, 10, 100. This code can be the same as you used in question 3, **except that it should use the training set from the outer layer of CV** instead of the entire data set.
  - Use the model from `caret` to predict the regions of the data in the test set from the outer layer of CV.

Enter your R code below.

Possible Answer:

```
set.seed(9)
n = dim(oak)[1]
ngroups = 10 # 10-fold outer CV
groups = rep(1:ngroups, length = n)
cv_groups = sample(groups, n)
ctrl = trainControl(method = "LOOCV")
preds = vector(length = n)
best_cost = numeric(length = ngroups)

for(ii in 1:ngroups){
  groupii = (cv_groups == ii)
  train_set = oak[!groupii, ]
  test_set = oak[groupii, ]

  data_used = train_set

  fit = train(Region ~ logSize + logRange,
              data = data_used,
              method = "svmLinear",
              tuneGrid = expand.grid(C = c(.001, .01, .1, 1, 5, 10, 100)),
              preProcess = c("center","scale"),
              trControl = ctrl)

  best_cost[ii] = fit$bestTune[[1]]
  preds[groupii] = predict(fit, newdata = test_set)
}

best_cost

##  [1] 100.0   0.1   1.0   1.0   1.0   1.0   1.0   1.0   1.0   0.1
```

From Problem 2:

Question:

Set the random seed to 9. Use caret to perform 10-fold cross-validation to compare different values of cost and sigma for a radial support vector machine. Use the same values of cost as listed previously: .001, .01, .1, 1, 5, 10, 100. Use sigma = 0.5, 1, 2, 3, and 4.

- Model the binary gas mileage variable as a function of all the other variables that are in Auto after question 10.
- Ask caret to model the probability that each point belongs to each category. (For purposes of this homework, it's OK if the model fails to converge.)

Enter your R code below.

Possible Answer:

```
set.seed(9)
data_used = Auto

ctrl = trainControl(method = "cv", number = 10)
fit_radial = train(mpg_bin ~ .,
                   data = data_used,
                   method = "svmRadial",
                   tuneGrid = expand.grid(C = c(.001, .01, .1, 1, 5, 10, 100),
                                          sigma = c(0.5, 1, 2, 3, 4)),
                   preProcess = c("center","scale"),
                   prob.model = TRUE,
                   trControl = ctrl)
```

Question:

Which combination of parameters gave the highest cross-validation accuracy?

```
fit_radial$bestTune
```

Question:

What was the cross-validation accuracy of the best model? Enter your answer to 4 decimal places.

```
fit_radial$results %>%
  filter(sigma == 1 & C == 1)
```

Question:

Use the best model to predict the probability that the following car would have **high** gas mileage:
1977 Chrysler Sunbeam
Cylinders: 4
Engine displacement: 132.5 cubic inches
Horsepower: 155
Weight: 2,910 lbs
Acceleration: 8.3 seconds
Origin: American (1)

Enter your answer to 4 decimal places.

```
example_car = data.frame(cylinders = 4, displacement = 132.5, horsepower =
155,
                         weight = 2910, acceleration = 8.3, year = 77,
                         origin = 1)
example_car <- example_car %>%
  mutate(origin = factor(origin, levels = c(1,2,3)))

predict(fit_radial, newdata = example_car, type = "prob")
```

Question

Make a grid of example data points with

- weight = seq(min(Auto$weight), max(Auto$weight), length = 100)
- cylinders = 4
- origin = 1
- all other predictors set equal to their medians.

Predict the probability of having high gas mileage for each data point in the grid. Include your R code below.

```
xgrid = expand.grid(cylinders = 4,
                    displacement = median(Auto$displacement),
                    horsepower = median(Auto$horsepower),
                    weight = seq(min(Auto$weight), max(Auto$weight), length = 100),
                    acceleration = median(Auto$acceleration),
                    year = median(Auto$year),
                    origin = 1)

xgrid <- xgrid %>%
  mutate(origin = factor(origin, levels = c(1,2,3)))

preds = predict(fit_radial, newdata = xgrid, type = "prob")
xgrid <- xgrid %>%
  mutate(prob_high = preds[ ,1])
```