SVM Exercise

Rommel Bartolome

April 3, 2019

a.)

Fit an SVM with temperature, relative humidity, light, and CO2 as predictors. Try different kernels and choose what gives the best predictive performance for the test set.

We first load our data, then divide it to training and test dataset:

```r
load("occup.Rdata")
set.seed(1)
sample <- sample.int(n = nrow(occup),
                     size = floor(.75*nrow(occup)),
                     replace = F)
train_occup <- occup[sample, ]
test_occup  <- occup[-sample, ]
```

We try to tune the best SVM, we try the radial first:

```r
library(e1071)
set.seed(1)
tune.out_radial <- tune(svm, Occupancy~., data=train_occup[,-c(5)],
                        kernel="radial", ranges = list(cost=c(0.1, 1, 5, 10),
                                                       gamma = c(0.1, 0.5, 1, 2, 3)))
tune.out_radial$best.model
```

```
##
## Call:
## best.tune(method = svm, train.x = Occupancy ~ ., data = train_occup[,
##     -c(5)], ranges = list(cost = c(0.1, 1, 5, 10), gamma = c(0.1,
##     0.5, 1, 2, 3)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##       gamma:  3
##
## Number of Support Vectors:  230
```

Here, we see that the best cost is 10 and the best gamma is 3. We use that to predict to the test set:

```r
table(predict = predict(tune.out_radial$best.model, test_occup),
      actual=test_occup$Occupancy)
```

```
##        actual
## predict    0    1
##       0 1628    8
##       1    6  394
```

This gives a pretty good result, with only 14 misclassified. We try to use other kernels:

```
set.seed(1)
tune.out_linear <- tune(svm, Occupancy~., data=train_occup[,-c(5)],
                        kernel="linear",
                        ranges = list(list(cost=c(0.001, 0.01, 0.1, 1, 5, 10, 100))))
table(predict = predict(tune.out_linear$best.model, test_occup),
      actual=test_occup$Occupancy)
```

```
##        actual
## predict    0    1
##       0 1609    4
##       1   25  398
```

The linear one is worse, with a total of 29 misclassified. We also try a polynomial kernel:

```
set.seed(1)
tune.out_poly <- tune(svm, Occupancy~., data=train_occup[,-c(5)],
                      kernel="polynomial",
                      ranges = list(cost=c(0.001, 0.01, 0.1, 1, 5, 10, 100),
                                    degree = c(2, 3, 4)))
table(predict = predict(tune.out_poly$best.model, test_occup),
      actual=test_occup$Occupancy)
```

```
##        actual
## predict    0    1
##       0 1609    1
##       1   25  401
```

Here we see that a total of 26 was misclassified. As such, we will choose **radial** as the best one, with the following specifications:

```
tune.out_radial$best.model
```

```
##
## Call:
## best.tune(method = svm, train.x = Occupancy ~ ., data = train_occup[,
##     -c(5)], ranges = list(cost = c(0.1, 1, 5, 10), gamma = c(0.1,
##     0.5, 1, 2, 3)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##       gamma:  3
##
## Number of Support Vectors:  230
```

b.)

Now, fit an SVM with light, CO2, and humidity ratio as predictors. Try different kernels and choose what gives the best predictive performance for the test set.

We first try the radial set:

```
set.seed(1)
tune.out_radial2 <- tune(svm, Occupancy~., data=train_occup[,-c(1,2)],
                        kernel="radial", ranges = list(cost=c(0.1, 1, 5, 10),
                                                       gamma = c(0.1, 0.5, 1, 2, 3)))
table(predict = predict(tune.out_radial2$best.model, test_occup),
      actual=test_occup$Occupancy)
```

```
##        actual
## predict    0    1
##       0 1627   11
##       1    7  391
```

Here, we see that it has 20 misclassified entries. We try a linear one:

```
set.seed(1)
tune.out_linear2 <- tune(svm, Occupancy~., data=train_occup[,-c(1,2)],
                        kernel="linear", ranges = list(cost=c(0.001, 0.01, 0.1, 1, 5, 10, 100)))
table(predict = predict(tune.out_linear2$best.model, test_occup),
      actual=test_occup$Occupancy)
```

```
##        actual
## predict    0    1
##       0 1609    1
##       1   25  401
```

Similarly, there are 26 misclassified entries. We will try a polynomial one:

```
set.seed(1)
tune.out_poly2 <- tune(svm, Occupancy~., data=train_occup[,-c(1,2)],
                      kernel="polynomial",
                      ranges = list(cost=c(0.001, 0.01, 0.1, 1, 5, 10, 100),
                                    degree = c(2, 3, 4)))
table(predict = predict(tune.out_poly2$best.model, test_occup),
      actual=test_occup$Occupancy)
```

```
##        actual
## predict    0    1
##       0 1610    7
##       1   24  395
```

By far, this is the worst, with 31 misclassified entries. Here, we see that **radial** is still the best one. The parameters of such is as follows:

```
tune.out_radial2
```

```
## 
## Parameter tuning of 'svm':
## 
## - sampling method: 10-fold cross validation
## 
## - best parameters:
##  cost gamma
##    10     2
## 
## - best performance: 0.007041936
```

c.)

Compare the performance of your SVMs in (a) and (b). What was the effect of changing the way you treat temperature and relative humidity?

Here are the number of misclassified entries on each:

Model Used	**Misclassified**
No Humidity Ratio - Radial	14
No Humidity Ratio - Linear	29
No Humidity Ratio - Polynomial	26
No Temp & Humidity - Radial	20
No Temp & Humidity - Linear	26
No Temp & Humidity - Polynomial	31

I would say that since Humidity Ratio is a derived quantity from temperature and relative humidity, we can say that we should be able to have similar results, which is the case here (low error rate). However, since Temparature and Humidity are still separated in `a)`, we would expect that it would be better overall, which is also the case here.

All in all, the SVM using the radial kernel with a separate temperature and relative humidity factor has the lowest error.