HW04

Zixin Ouyang

10/6/2017

Exercise 1

```
hw04_train<-read.csv('hw04-trn-data.csv')
hw04_test<-read.csv('hw04-tst-data.csv')
```

```
library(caret)
```

```
c1 = function(data) {
  with(data, ifelse(x1 > 0, yes = "dodgerblue", no = "darkorange"))
}
```

```
c2 = function(data) {
  with(data, ifelse(x2 > x1 + 1, yes = "dodgerblue", no = "darkorange"))
}
```

```
c3 = function(data) {
  with(data, ifelse(x2 > x1 + 1,
                    yes = "dodgerblue",
                    no = ifelse(x2 < x1 - 1,
                                yes = "dodgerblue",
                                no = "darkorange")))
}
```

```
c4 = function(data) {
  with(data, ifelse(x2 > (x1 + 1) ^ 2,
                    yes = "dodgerblue",
                    no = ifelse(x2 < -(x1 - 1) ^ 2,
                                yes = "dodgerblue",
                                no = "darkorange")))
}
```

```
calc_error = function(classifier, data) {
  mean(data$y != classifier(data))
}
```

```
classifiers = list(c1, c2, c3, c4)
```

```
results = data.frame(
  c("`c1`", "`c2`", "`c3`", "`c4`"),
  sapply(classifiers, calc_error, data = hw04_train),
  sapply(classifiers, calc_error, data = hw04_test)
)
```

```
colnames(results) = c("Classifier", "Train Error Rate", "Test Error Rate")
knitr::kable(results)
```

Classifier	Train Error Rate	Test Error Rate
`c1`	0.468	0.5160
`c2`	0.216	0.2240

Classifier	Train Error Rate	Test Error Rate
c3	0.096	0.1270
c4	0.050	0.0665

Exercise 2

```
get_logistic_error = function(model, data) {
  predicted = ifelse(predict(model, data) > 0.5,
                     yes = "dodgerblue",
                     no = "darkorange")
  mean(data$y != predicted)
}
```

```
model_1 = glm(y ~ 1, data = hw04_train, family = "binomial")
model_2 = glm(y ~ x1 + x2, data = hw04_train, family = "binomial")
model_3 = glm(y ~ x1 + x2 + I(x1 ^ 2) + I(x2 ^ 2), data = hw04_train, family = "binomial")
model_4 = glm(y ~  x1 * x2 + I(x1 ^ 2) + I(x2 ^ 2), data = hw04_train, family = "binomial")
```

```
model_list = list(model_1, model_2, model_3, model_4)
train_errors = sapply(model_list, get_logistic_error, data = hw04_train)
test_errors  = sapply(model_list, get_logistic_error, data = hw04_test)
```

Models	Train Error Rate	Test Error Rate
mod_1	0.334	0.3305
mod_2	0.334	0.3305
mod_3	0.33	0.343
mod_4	0.098	0.136

Exercise 3

```
make_sim_data = function(n_obs = 25) {
  x1 = runif(n = n_obs, min = 0, max = 2)
  x2 = runif(n = n_obs, min = 0, max = 4)
  prob = exp(1 + 2 * x1 - 1 * x2) / (1 + exp(1 + 2 * x1 - 1 * x2))
  y = rbinom(n = n_obs, size = 1, prob = prob)
  data.frame(y, x1, x2)
}
```

```
set.seed(659017838)
n_sims = 1000
n_models = 3
x = data.frame(x1=1, x2=1)
predictions = matrix(0, nrow = n_sims, ncol = n_models)
```

```
for(sim in 1:n_sims) {
  sim_data = make_sim_data()
  mod_1 = glm(y ~ 1, data = sim_data, family = "binomial")
  mod_2 = glm(y ~ ., data = sim_data, family = "binomial")
  mod_3 = glm(y ~ x1*x2 + I(x1 ^ 2) + I(x2 ^ 2), data = sim_data, family = "binomial")
```

```
  predictions[sim, 1] = predict(mod_1, newdata=x , type = "response")
  predictions[sim, 2] = predict(mod_2, newdata=x, type = "response")
  predictions[sim, 3] = predict(mod_3, newdata=x, type = "response")
}
```

```
get_mse = function(truth, estimate) {
  mean((estimate - truth) ^ 2)
}
```

```
get_bias = function(estimate, truth) {
  mean(estimate) - truth
}
```

```
get_var = function(estimate) {
  mean((estimate - mean(estimate)) ^ 2)
}
```

```
p = function(x) {
with(x,
     exp(1 + 2 * x1 - 1 * x2) / (1 + exp(1 + 2 * x1 - 1 * x2))
     )
}
```

```
bias = apply(predictions, 2, get_bias, truth = p(x))
variance = apply(predictions, 2, get_var)
mse = apply(predictions, 2, get_mse, truth = p(x))
```

K	Mean Squared Error	Bias Squared	Variance
Intercept Only	0.05819	0.04921	0.00897
Additive	0.00933	0.00006	0.00927
Second Order	0.02317	0.00074	0.02243

Exercise 4

(a) The true decision boundaries are non-linear since the fourth classifier performs best.

(b) Model 4 performs best because it has smallest train and test error rates.

(c) The first three models are underfitting as they are all simpler than the "best" model.

(d) None of these models are overfitting as the "best" model is also the most complex.

(e) Both the additive and second order models are performing unbiased estimation.

(f) The additive model performs best because it has the lowest MSE.