Homework 9 - Support Vector Machines

Shiloh Bradley

6/28/2020

```
normalize <- function(x) {
  x <- na.omit(x)
  return ((x - min(x)) / (max(x) - min(x)))
}
```

Titanic

```
df <- read.csv("titanic3.csv", header = TRUE)
dim(df) ## 1309 14
```

```
## [1] 1309   14
```

```
names(df)
```

```
##  [1] "pclass"    "survived"  "name"      "sex"       "age"
##  [6] "sibsp"     "parch"     "ticket"    "fare"      "cabin"
## [11] "embarked"  "boat"      "body"      "home.dest"
```

```
head(df)
```

```
##   pclass survived                                            name sex
## 1      1        1                   Allen, Miss. Elisabeth Walton   1
## 2      1        1                  Allison, Master. Hudson Trevor   0
## 3      1        0                    Allison, Miss. Helen Loraine   1
## 4      1        0            Allison, Mr. Hudson Joshua Creighton   0
## 5      1        0 Allison, Mrs. Hudson J C (Bessie Waldo Daniels)   1
## 6      1        1                         Anderson, Mr. Harry   0
##       age sibsp parch ticket     fare   cabin embarked boat body
## 1 29.0000     0     0  24160 211.3375      B5        S    2   NA
## 2  0.9167     1     2 113781 151.5500 C22 C26        S   11   NA
## 3  2.0000     1     2 113781 151.5500 C22 C26        S        NA
## 4 30.0000     1     2 113781 151.5500 C22 C26        S       135
## 5 25.0000     1     2 113781 151.5500 C22 C26        S        NA
## 6 48.0000     0     0  19952  26.5500     E12        S    3   NA
##                         home.dest
## 1                     St Louis, MO
## 2 Montreal, PQ / Chesterville, ON
## 3 Montreal, PQ / Chesterville, ON
## 4 Montreal, PQ / Chesterville, ON
## 5 Montreal, PQ / Chesterville, ON
## 6                     New York, NY
```

```
summary(df)
```

```
##      pclass          survived                                    name
##  Min.   :1.000   Min.   :0.000   Connolly, Miss. Kate            :   2
##  1st Qu.:2.000   1st Qu.:0.000   Kelly, Mr. James                :   2
##  Median :3.000   Median :0.000   Abbing, Mr. Anthony             :   1
```

```
##  Mean   :2.295   Mean   :0.382   Abbott, Master. Eugene Joseph   :   1
##  3rd Qu.:3.000   3rd Qu.:1.000   Abbott, Mr. Rossmore Edward     :   1
##  Max.   :3.000   Max.   :1.000   Abbott, Mrs. Stanton (Rosa Hunt):   1
##                                  (Other)                         :1301
##       sex             age              sibsp            parch
##  Min.   :0.000   Min.   : 0.1667   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:21.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.000   Median :28.0000   Median :0.0000   Median :0.000
##  Mean   :0.356   Mean   :29.8811   Mean   :0.4989   Mean   :0.385
##  3rd Qu.:1.000   3rd Qu.:39.0000   3rd Qu.:1.0000   3rd Qu.:0.000
##  Max.   :1.000   Max.   :80.0000   Max.   :8.0000   Max.   :9.000
##                  NA's   :263
##       ticket          fare                    cabin      embarked
##  CA. 2343:  11   Min.   :  0.000                 :1014    :  2
##  1601    :   8   1st Qu.:  7.896   C23 C25 C27   :   6   C:270
##  CA 2144 :   8   Median : 14.454   B57 B59 B63 B66:   5   Q:123
##  3101295 :   7   Mean   : 33.295   G6            :   5   S:914
##  347077  :   7   3rd Qu.: 31.275   B96 B98       :   4
##  347082  :   7   Max.   :512.329   C22 C26       :   4
##  (Other) :1261   NA's   :1         (Other)       : 271
##       boat           body                       home.dest
##         :823   Min.   :  1.0                         :564
##  13     : 39   1st Qu.: 72.0   New York, NY          : 64
##  C      : 38   Median :155.0   London                : 14
##  15     : 37   Mean   :160.8   Montreal, PQ          : 10
##  14     : 33   3rd Qu.:256.0   Cornwall / Akron, OH:  9
##  4      : 31   Max.   :328.0   Paris, France         :  9
##  (Other):308   NA's   :1188    (Other)               :639
```

```
df <- df %>%
      select(survived, pclass, sex, age, sibsp, parch) %>%
      filter(!is.na(pclass) & !is.na(sex) & !is.na(age) & !is.na(sibsp) & !is.na(parch)) %>%
      mutate(survived = as.factor(survived))
## don't need to normalize any variables
```

```
n.NA <- colSums(is.na(df))
n.NA
```

```
## survived   pclass      sex      age    sibsp    parch
##        0        0        0        0        0        0
```

```
features0 <- setdiff(names(df), "survived")
Formula0 <- formula(paste("survived ~ ",
                          paste(features0, collapse = " + ")))
Formula0
```

```
## survived ~ pclass + sex + age + sibsp + parch
```

```
## Don't split the data before normalizing the data and creating the dummy variables
M <- trunc(.25 * nrow(df))

# to be able to replicate the results, set initial seed for random
# number generator
set.seed(1797317)
holdout <- sample(1:nrow(df), M, replace = F)
df.train <- df[-holdout, ]   # Training set 785  6
```

```
df.test <- df[holdout, ]    # Test set  of 261  6
dim(df.train)
```

```
## [1] 785   6
```

```
dim(df.test)
```

```
## [1] 261   6
```

```
svm1 <- svm(Formula0, kernel = "radial", type = "C-classification", probability = TRUE, data = df.train)
print(svm1)
```

```
##
## Call:
## svm(formula = Formula0, data = df.train, kernel = "radial", type = "C-classification",
##     probability = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  375
```

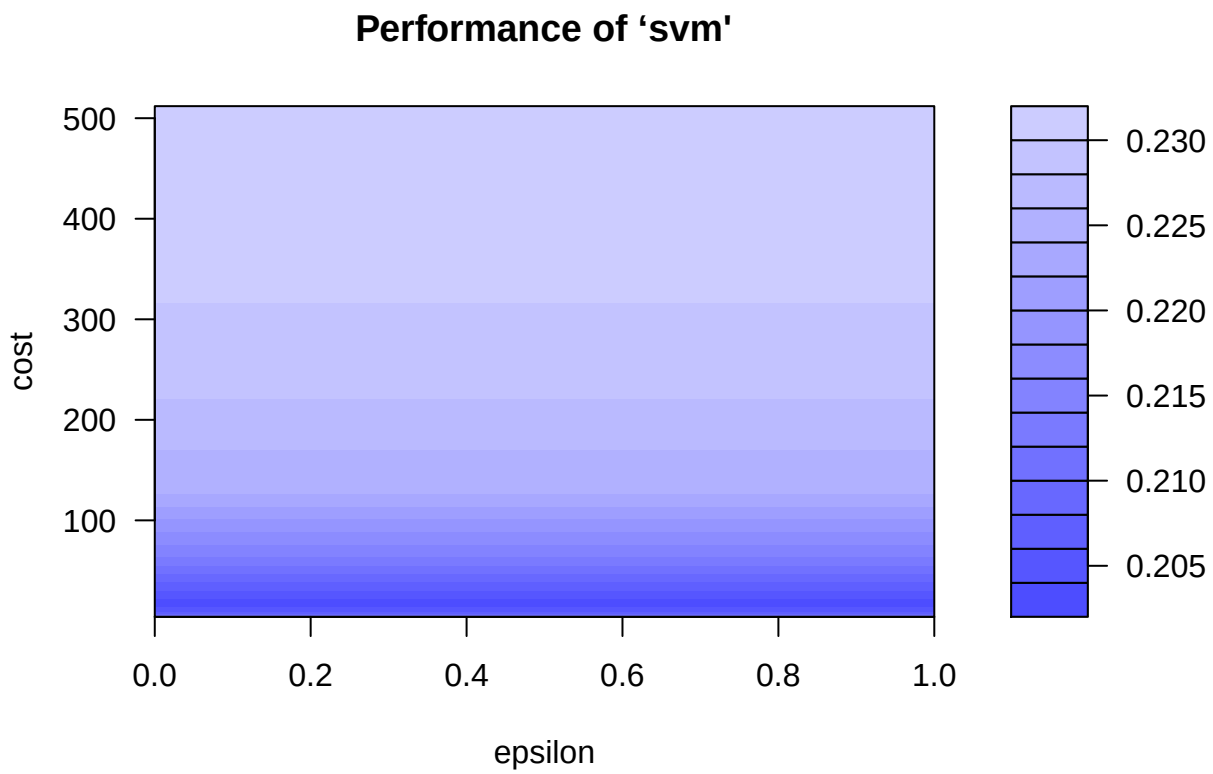
```
names(svm1)
```

```
##  [1] "call"            "type"            "kernel"
##  [4] "cost"            "degree"          "gamma"
##  [7] "coef0"           "nu"              "epsilon"
## [10] "sparse"          "scaled"          "x.scale"
## [13] "y.scale"         "nclasses"        "levels"
## [16] "tot.nSV"         "nSV"             "labels"
## [19] "SV"              "index"           "rho"
## [22] "compprob"        "probA"           "probB"
## [25] "sigma"           "coefs"           "na.action"
## [28] "fitted"          "decision.values" "terms"
```

```
# SVM performance can be improved further by tuning the SVM
# perform a grid search to tune(optimize) SVM HYPERPARAMETERS
tune.svm <- tune(svm, Formula0,
                 kernel = "radial", data = df.train,
                 type = "C-classification", probability = TRUE,
                 ranges = list(epsilon = seq(0, 1, 0.1), cost = 2^(2:9)))
print(tune.svm)
```

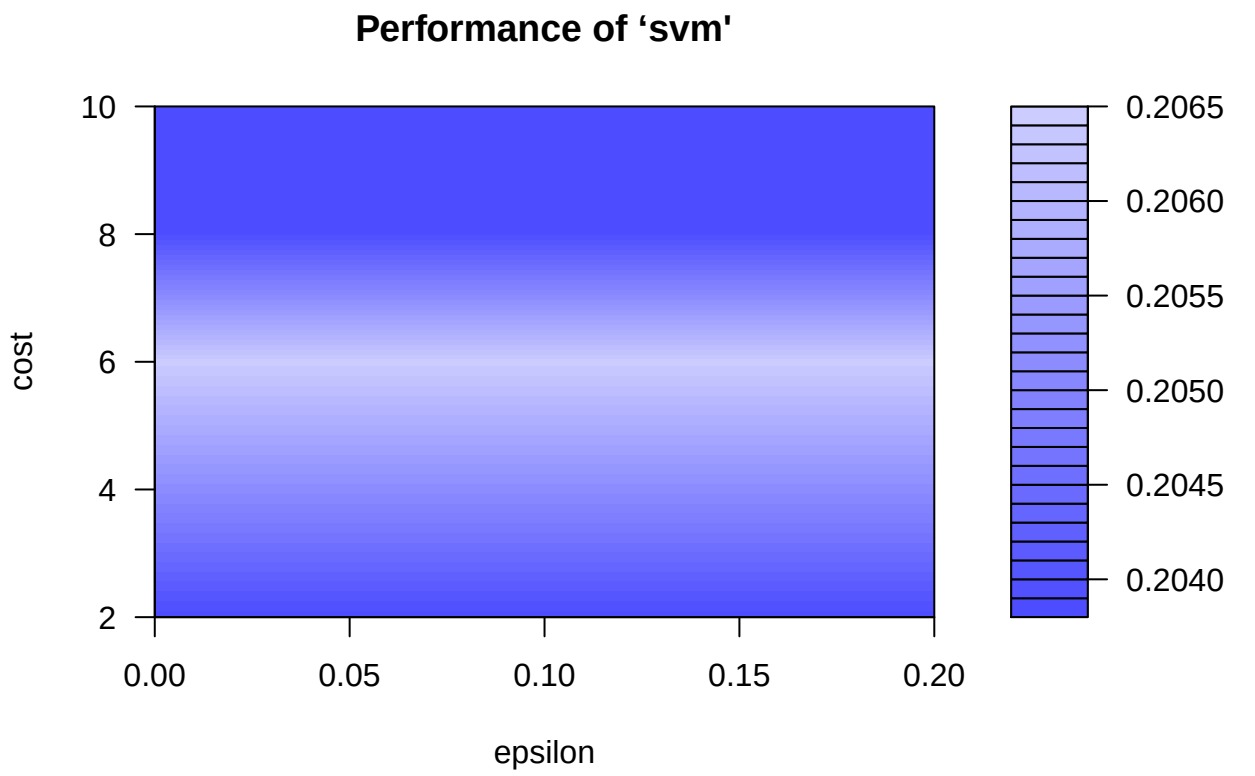
```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  epsilon cost
##        0   16
##
## - best performance: 0.2025803
```

```r
# Draw the tuning graph
plot(tune.svm)
```




```r
tune.svm2 <- tune(svm, Formula0,
                  kernel = "radial", data = df.train,
                  type = "C-classification", probability = TRUE,
                  ranges = list(epsilon = seq(0, 0.2, 0.02), cost = seq(2, 10, 2)))

plot(tune.svm2)
```




```
tune.svm2
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  epsilon cost
##        0    2
##
## - best performance: 0.2038299
```

```
svmF <- svm(Formula0,
            kernel = "radial", data = df.train,
            type = "C-classification", probability = TRUE,
            ranges = list(epsilon = 0.025, cost = 6, probability = TRUE))

svmF
```

```
##
## Call:
## svm(formula = Formula0, data = df.train, kernel = "radial", type = "C-classification",
##     probability = TRUE, ranges = list(epsilon = 0.025, cost = 6,
##         probability = TRUE))
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
```

```
##
## Number of Support Vectors:  375
```

```
Y.train <- df.train$survived
Y.test <- df.test$survived
Ypred.train_svm <- svmF$fitted
Ypred.test_svm <- predict(svmF, df.test, probability = TRUE, decision.value = TRUE)
temp.prob <- attr(Ypred.test_svm, "probabilities")
Yhat.test_svm <- round(temp.prob[,1])
CM.train_svm <- table(Y.train, Ypred.train_svm)
CM.test_svm <- table(Y.test, Yhat.test_svm)
OA.train_svm <- sum(diag(CM.train_svm))/sum(CM.train_svm)
PRF1.train_svm <- PRF1(CM.train_svm)
OA.test_svm <- sum(diag(CM.test_svm))/sum(CM.test_svm)
PRF1.test_svm <- PRF1(CM.test_svm)
```

```
df.PRF1_svm <- rbind.data.frame(PRF1.train_svm, PRF1.test_svm)
colnames(df.PRF1_svm) <- c("Precision.1_svm", "Recall.1", "F1.1_svm",
                           "Precision.0_svm", "Recall.0", "F1.0_svm")
rownames(df.PRF1_svm) <- c("Training", "Test")

df.OA_svm <- rbind.data.frame(OA.train_svm, OA.test_svm)
colnames(df.OA_svm) <- "Overall_Accuracy"
rownames(df.OA_svm) <- c("Training", "Test")
round(df.PRF1_svm, 2)
```

```
##          Precision.1_svm Recall.1 F1.1_svm Precision.0_svm Recall.0
## Training            0.82     0.89     0.85            0.82     0.89
## Test                0.83     0.88     0.86            0.83     0.88
##          F1.0_svm
## Training     0.85
## Test         0.86
```

```
round(df.OA_svm, 2)
```

```
##          Overall_Accuracy
## Training             0.82
## Test                 0.83
```

```
LR1 <- glm(Formula0, family = binomial("logit"), data = df.train)
smre1 <- summary(LR1)

vif1 <- vif(LR1)
min(vif1)
```

```
## [1] 1.115389
```

```
vif1 ## Don't need to drop any variables because all VIF's < 5
```

```
##   pclass      sex      age    sibsp    parch
## 1.409880 1.115389 1.395101 1.213729 1.181009
```

```
LRF <- LR1
CM.train_LR <- table(Y.train, round(LRF$fitted.values))
CM.train_LR
```

```
##
## Y.train   0   1
```

```
##       0 390  76
##       1 101 218
```

```
pred.test_LR <- predict(LRF, df.test, type = "response")
Yhat.test_LR <- round(pred.test_LR)
#head(Yhat.test_LR)
CM.test_LR <- table(df.test$survived, Yhat.test_LR)
CM.test_LR
```

```
##    Yhat.test_LR
##       0   1
##   0 131  22
##   1  27  81
```

```
PRF1.train_LR <- PRF1(CM.train_LR)
PRF1.test_LR <- PRF1(CM.test_LR)
OA.train_LR <- sum(diag(CM.train_LR))/sum(CM.train_LR)
OA.test_LR <- sum(diag(CM.test_LR))/sum(CM.test_LR)
```

```
df.PRF1_LR <- rbind.data.frame(PRF1.train_LR,PRF1.test_LR)
colnames(df.PRF1_LR) <- c("Precision.1_LR", "Recall.1", "F1.1_LR",
                          "Precision.0_LR", "Recall.0", "F1.0_LR")
rownames(df.PRF1_LR) <- c("Training", "Test")

df.OA_LR <- rbind.data.frame(OA.train_LR, OA.test_LR)
colnames(df.OA_LR) <- "Overall_Accuracy"
rownames(df.OA_LR) <- c("Training", "Test")
round(df.PRF1_LR, 2)
```

```
##          Precision.1_LR Recall.1 F1.1_LR Precision.0_LR Recall.0 F1.0_LR
## Training           0.79     0.84    0.82           0.79     0.84    0.82
## Test               0.83     0.86    0.84           0.83     0.86    0.84
```

```
round(df.OA_LR,2)
```

```
##          Overall_Accuracy
## Training             0.77
## Test                 0.81
```