STAT 4400 HW02_03 SVM

Yanjin Li

February 22, 2016

In this code, I uploaded the uspsdata.txt, which contains one data point per row representing a 16 * 16 image of a handwritten number, and uspscl.txt, which contains the corresponding class labels. The data contains two classes- the digits 5 and 6-so the class labels are restored as -1 and +1.

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(e1071)
library(ggplot2)
library(lattice)
library(rpart)
```

0. Read and manage the Usps Data and Usps Scale Data
1. Randomly Select 20% of Data as a Test Set Define 20% of the data as test set

```
test_size <- trunc(0.20 * row_num)
test_ind <- sample(row_num, size = test_size)
test <- data_with_scl[test_ind, ]
train <- data_with_scl[-test_ind, ]
class <- as.matrix(train[ncol(data_with_scl)])
```

2. Train a linear SVM with soft margins Training linear SVM with 10-fold CV on the parameter

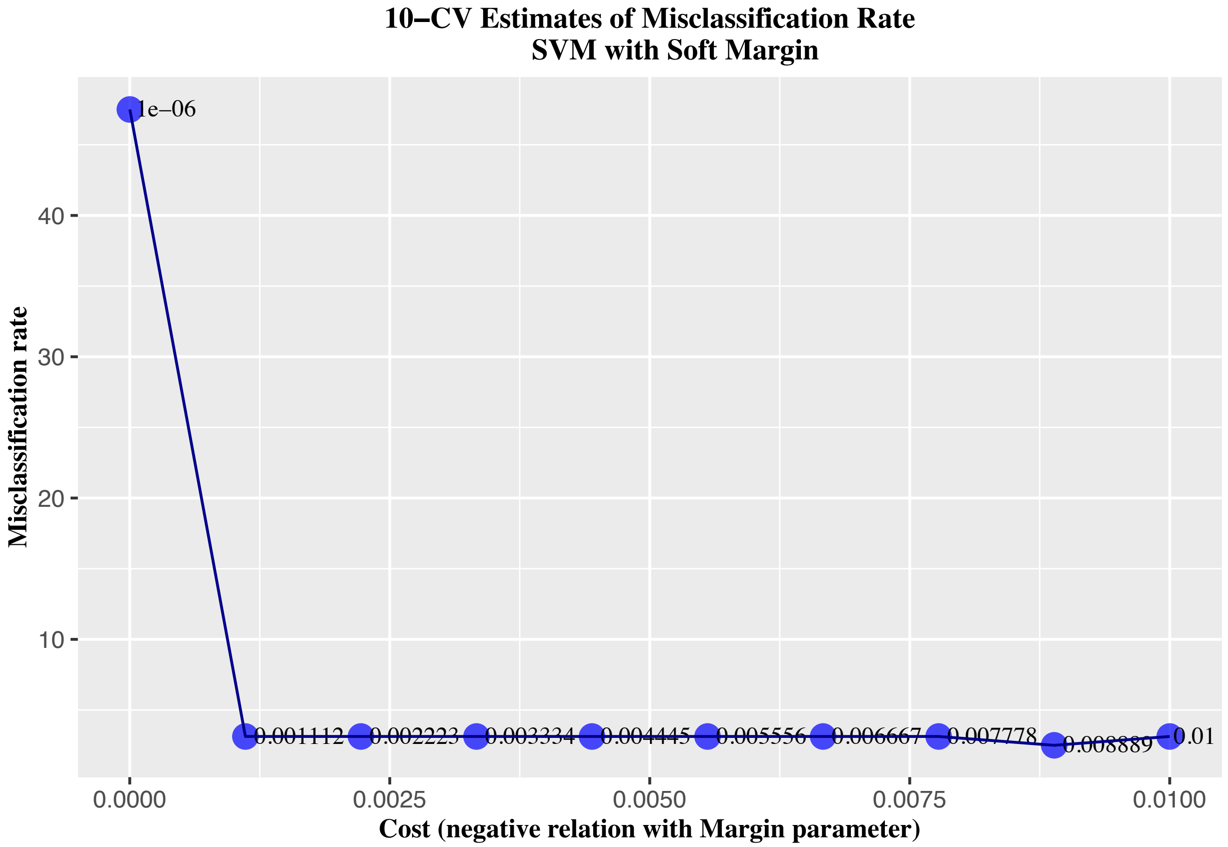
```
list_0 <- seq(0.000001, 0.01, length = 10)
misclass_rate_lnr <-  data.frame(cost_para = integer(),
               correctly_class = integer(), misclass = integer())
for (i in list_0){
  svm_lnr <- svm(class ~ ., data = train, cost = i, kernel = "linear",
                 type="C-classification", cross=10)
  tot <- summary(svm_lnr)$tot.accuracy
  mis <- (100 - tot)
  row <- list(i, tot, mis)
  misclass_rate_lnr <- rbind(misclass_rate_lnr, row)
}
colnames(misclass_rate_lnr) <- c("cost", "correct class",
                                 "misclassification")
```

Order based on misclassification rate

```
order_misclass <- misclass_rate_lnr[order(misclass_rate_lnr$misclassification),]
```

Plot the cross-validation estimates of the misclassification rate:

```
ggplot(data = misclass_rate_lnr, aes(x = cost, y = misclassification,
                                     label = cost)) +
  geom_point(color = "blue", alpha = 0.7, size = 4) +
  geom_text(check_overlap = TRUE, size = 3, hjust = -0.1,
            nudge_y = 0.05, family = "Times") +
  geom_line(color = "dark blue") +
  labs(title = "10-CV Estimates of Misclassification Rate
       SVM with Soft Margin",
       x = "Cost (negative relation with Margin parameter)",
       y = "Misclassification rate") +
  theme(plot.title = element_text(size = rel(1), lineheight = .9,
                                  family = "Times", face = "bold")) +
  theme(axis.title.x = element_text(size = 10, lineheight = .9,
                                    family = "Times",face = "bold")) +
  theme(axis.title.y = element_text(size = 10, lineheight = .9,
                                    family = "Times", face = "bold"))
```



3. Train a radial SVM with soft margins and RBF kernel

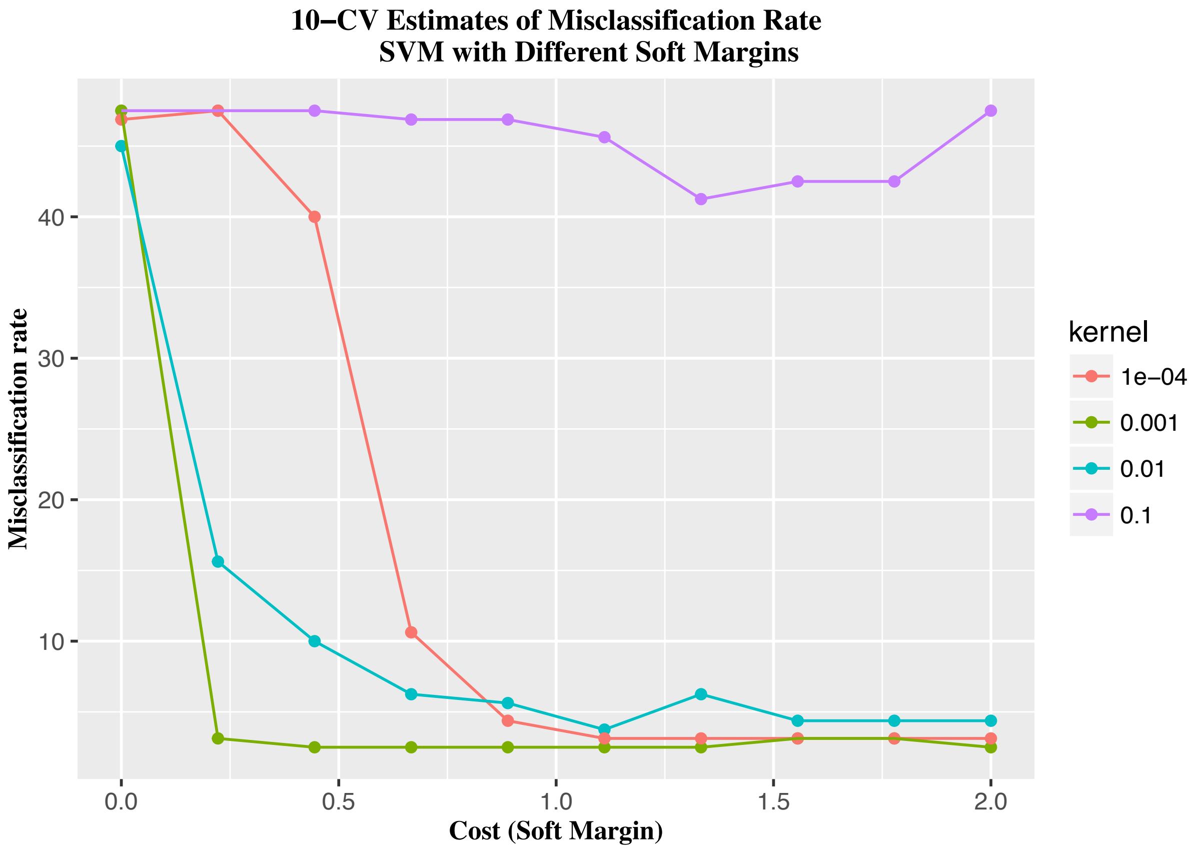
```
list_1 <- seq(0.000001, 2, length = 10)
list_2 <- 10^c(-1, -2, -3, -4)
misclass_rate_rbf <-  data.frame(cost_para = integer(), gamma = integer(),
                      correctly_class = integer(), misclass = integer())
for (i in list_1){
  for (j in list_2){
    svm_rbf <- svm(class ~ ., data = train, kernel = "radial", cost = i,
                   gamma = j, type="C-classification", cross=10)
    tot_2 <- summary(svm_rbf)$tot.accuracy
    mis_2 <- (100 - tot_2)
    row_2 <- list(i, j, tot_2, mis_2)
    misclass_rate_rbf <- rbind(misclass_rate_rbf, row_2)
  }
}

colnames(misclass_rate_rbf) <- c("cost", "kernel","correct class",
                                 "misclassification")
```

Here, I made a plot comparing margin parameters' performance on the dimension of kernel, which is assigned with four different values (0.1, 0.01, 0.001, 0.0001). By comparing the lines of the plots, we will detect the smallest value of misclassification rate, which exists when Kernel parameter gamma euqals to 0.001 and cost approximately equals to 1.5555558. According to this plot. Thus, we choose (cost = 1.5555558, gamma = 0.001) as our selected parameters.

```
misclass_rate_rbf$kernel <- as.factor(misclass_rate_rbf$kernel)

ggplot(data = misclass_rate_rbf,
       aes(x = cost, y = misclassification,
       color = kernel)) + geom_point() + geom_line()+
labs(title = "10-CV Estimates of Misclassification Rate
         SVM with Different Soft Margins",
       x = "Cost (Soft Margin)",
       y = "Misclassification rate") +
  theme(plot.title = element_text(size = rel(1), lineheight = .9,
                                  family = "Times", face = "bold")) +
  theme(axis.title.x = element_text(size = 10, lineheight = .9,
                                    family = "Times",face = "bold")) +
  theme(axis.title.y = element_text(size = 10, lineheight = .9,
                                    family = "Times", face = "bold"))
```



4. Train SVMs on test set Train our Linear SVM on the test dataset

```
selected_model_lnr <- svm(class ~ ., data = train, cost = 0.001112,
                          kernel = "linear", type="C-classification",
                          cross=10)
y_hat_lnr <- predict(selected_model_lnr, test[1:256],
                     decision.values = TRUE)
lnr_SVM_result <- table(pred = y_hat_lnr, true = test[,257])
```

Therefore, the test set estimates of misclassification rates for linear case is shown as following.

```
1-sum(diag(lnr_SVM_result))/sum(lnr_SVM_result)
```

```
## [1] 0.025
```

```
classAgreement(lnr_SVM_result)
```

```
## $diag
## [1] 0.975
##
## $kappa
## [1] 0.9473684
##
## $rand
## [1] 0.95
##
## $crand
## [1] 0.8999408
```

Train our Radial SVM on the test dataset

```
selected_model_rbf <- svm(class ~ ., data = train, cost = 1.5555558,
                          gamma = 0.001, type="C-classification",
                          kernel = "radial", cross=10)
y_hat_rbf <- predict(selected_model_rbf, test[1:256],
                     decision.values = TRUE)
rbf_SVM_result <- table(pred = y_hat_rbf, true = test[,257])
```

Therefore, the test set estimates of misclassification rates for radial case is shown as following.

```
1-sum(diag(rbf_SVM_result))/sum(rbf_SVM_result)
```

```
## [1] 0.025
```

```
classAgreement(rbf_SVM_result)
```

```
## $diag
## [1] 0.975
##
## $kappa
## [1] 0.9473684
##
## $rand
## [1] 0.95
##
## $crand
## [1] 0.8999408
```

5. Conclusion

Based on the results of misclassifications on both two cases (linear and radial), we always derive the same results which are both 0.025. Due to the reasons of computational ease and simpler interpretation, we choose the linear SVM as our final model.