Multivariate Statistical Analysis

Homework 5

Lucas Fellmeth, Helen Kafka, Sven Bergmann

04/04/24

```r
set.seed(42)
```

Problem 1

The pendigit dataset contains digitalized handwritten digits.

```r
pendigit <- read.csv(file = "../Data_csv/pendigits.csv")
pendigit$digit <- factor(pendigit$digit)
```

The variables $x1, y1, \ldots, x8, y8$ are the coordinates of a pen on a writing pad at eight different time points (so, if you want to visualize the written digit, you have to do `plot(c(x1,...,x8),c(y1,...,y8),type='l'`. The variable digit identifies the digit that was written. The goal is to construct a classifier that will identify the handwritten digits as accurately as possible. Split the data into training and test sets (roughly an 80/20 split).

```r
i_test_pendigit <- sample(seq_along(pendigit[, 1]), size = length(pendigit[,
    1]) * 0.2)
pendigit_test <- pendigit[i_test_pendigit, ]
pendigit_train <- pendigit[-i_test_pendigit, ]
```

Fit single-layer neural networks to the training data, with one, two and three hidden nodes (or more if necessary).

```r
library(nnet)
```

```r
num_hidden_nodes_pendigit <- 5
nnets_pendigit <- vector(mode = "list", length = num_hidden_nodes_pendigit)
for (i in 1:num_hidden_nodes_pendigit) {
    nnets_pendigit[[i]] <- nnet(digit ~ . - digit, data = pendigit_train,
        size = i, trace = F)
}
```

Compute the respective misclassification rates on the test set.

```r
mctables_pendigit <- vector(mode = "list", length = num_hidden_nodes_pendigit)
for (i in 1:num_hidden_nodes_pendigit) {
    mctables_pendigit[[i]] <- table(pendigit_test$digit, predict(nnets_pendigit[[i]],
        newdata = pendigit_test, type = "class"))
}
```

What's the lowest misclassification rate attained?

```
for (i in 1:num_hidden_nodes_pendigit) {
    print(paste("Misclassification rate for set ", i, ":", 1 -
        (sum(diag(mctables_pendigit[[i]]))/length(pendigit_test$digit))))
}
```

```
## [1] "Misclassification rate for set  1 : 0.904458598726115"
## [1] "Misclassification rate for set  2 : 0.808007279344859"
## [1] "Misclassification rate for set  3 : 0.709281164695177"
## [1] "Misclassification rate for set  4 : 0.333484986351228"
## [1] "Misclassification rate for set  5 : 0.791628753412193"
```

From the cross-classification tables, which digits have the largest misclassification rates?

Problem 2

The spambase dataset contains data for 4,601 emails which are classified as spam or not spam (as indicated by the variable class);

```
spambase <- read.csv(file = "../Data_csv/spambase.csv")
spambase$class <- factor(spambase$class)
```

58 feature variables are measured on each email. A more detailed description of the data is given on p. 259 of the book. Split the data into training and test sets (roughly an 80/20 split).

```
i_test_spambase <- sample(seq_along(spambase[, 1]), size = length(spambase[,
    1]) * 0.2)
spambase_test <- spambase[i_test_spambase, ]
spambase_train <- spambase[-i_test_spambase, ]
```

Fit a linear support vector machine classifier to the training data, starting with a very large ("infinite") cost, in the event the groups are separable, and progressively lowering the cost if they aren't.

```
library("e1071")
```

We omitted the following code with `cost=Inf` because there was an error message.

```
# svmfit_inf_spambase <- svm(class ~ . - class, data =
# spambase_train, cost = Inf, kernel = 'linear')
# summary(svmfit_inf_spambase) # Error in svm.default(x, y,
# scale = scale, ..., na.action = na.action): NA/NaN/Inf in
# foreign function call (arg 12)
```

```
num_iterations_spambase <- 10
svmfits_spambase <- vector(mode = "list", length = num_iterations_spambase)
# svmfits_spambase[[num_iterations_spambase + 1]] <-
# svmfit_inf_spambase
for (i in num_iterations_spambase:1) {
    svmfits_spambase[[i]] <- svm(class ~ . - class, data = spambase_train,
        cost = 10^i)
}
```

Compute the respective misclassification rates on the test set.

```r
mctables_spambase <- vector(mode = "list", length = num_iterations_spambase)
for (i in 1:num_iterations_spambase) {
    mctables_spambase[[i]] <- table(spambase_test$class, predict(svmfits_spambase[[i]],
        newdata = spambase_test, type = "class"))
}
```

```r
mscrs_spambase <- vector(mode = "list", length = num_iterations_spambase)
for (i in 1:num_iterations_spambase) {
    mscrs_spambase[[i]] <- 1 - (sum(diag(mctables_spambase[[i]]))/length(spambase_test$class))
}
for (i in 1:num_iterations_spambase) {
    print(paste("Misclassification rate for set ", i, ":", mscrs_spambase[[i]]))
}
```

```
## [1] "Misclassification rate for set  1 : 0.00543478260869568"
## [1] "Misclassification rate for set  2 : 0.00543478260869568"
## [1] "Misclassification rate for set  3 : 0.00543478260869568"
## [1] "Misclassification rate for set  4 : 0.00543478260869568"
## [1] "Misclassification rate for set  5 : 0.00543478260869568"
## [1] "Misclassification rate for set  6 : 0.00543478260869568"
## [1] "Misclassification rate for set  7 : 0.00543478260869568"
## [1] "Misclassification rate for set  8 : 0.00543478260869568"
## [1] "Misclassification rate for set  9 : 0.00543478260869568"
## [1] "Misclassification rate for set  10 : 0.00543478260869568"
```

What's the lowest misclassification rate attained?

```r
paste("The lowest misclassificationrate is on set", which.min(mscrs_spambase),
    "with a cost of", 10^which.min(mscrs_spambase))
```

```
## [1] "The lowest misclassificationrate is on set 1 with a cost of 10"
```