HW 2

Ava Klissouras

9/24/2024

This homework is meant to illustrate the methods of classification algorithms as well as their potential pitfalls. In class, we demonstrated K-Nearest-Neighbors using the `iris` dataset. Today I will give you a different subset of this same data, and you will train a KNN classifier.

```
set.seed(123)
library(class)

df <- data(iris)

normal <-function(x) {
  (x -min(x))/(max(x)-min(x))
}

iris_norm <- as.data.frame(lapply(iris[,c(1,2,3,4)], normal))

subset <- c(1:45, 58, 60:70, 82, 94, 110:150)
iris_train <- iris_norm[subset,]
iris_test <- iris_norm[-subset,]

iris_target_category <- iris[subset,5]
iris_test_category <- iris[-subset,5]
```

Above, I have given you a training-testing partition. Train the KNN with $K = 5$ on the training data and use this to classify the 50 test observations. Once you have classified the test observations, create a contingency table – like we did in class – to evaluate which observations your algorithm is misclassifying.

```
set.seed(123)
library(class)

pr <- knn(iris_train,iris_test,cl=iris_target_category,k=5)
tab <- table(pr,iris_test_category)
tab
```

```
##             iris_test_category
## pr           setosa versicolor virginica
##   setosa          5          0         0
##   versicolor      0         25         0
##   virginica       0         11         9
```

```r
accuracy <- function(x){
  sum(diag(x)/(sum(rowSums(x)))) * 100
}
accuracy(tab)
```

```
## [1] 78
```

```r
#Determining why the classification error rate is so high (22% as opposed to ~3.33%):
summary(iris_target_category) #subsetted data
```

```
##     setosa versicolor  virginica
##         45         14         41
```

```r
summary(iris_test_category) #the rest of the data
```

```
##     setosa versicolor  virginica
##          5         36          9
```

```r
summary(iris[,5]) #original data for comparison
```

```
##     setosa versicolor  virginica
##         50         50         50
```

Discuss your results. If you have done this correctly, you should have a classification error rate that is roughly 20% higher than what we observed in class. Why is this the case? In particular run a summary of the `iris_test_category` as well as `iris_target_category` and discuss how this plays a role in your answer.

The classification error rate is higher than the example we did in class because here, our sample is poorly representative of our original dataset. In the original data, there are 50 setosa, 50 versicolor, and 50 virginica, meaning each flower type represents ~33% (1/3) of the total number of observations. However, in the data we subsetted for training (iris_target_category), there are 45 setosa, 14 versicolor, and 41 virginica represented, so setosa makes up 45% of the sample, versicolor makes up 14% of the sample, and virginica makes up 41% of the sample. These percentages are not reflective of the distribution of flower types in the full data set, and this kind of less-than-representative training data causes degradation of KNN performance.

To determine why this sample is not well representative of the entire data set, we can examine the code used to create it: subset <- c(1:45, 58, 60:70, 82, 94, 110:150)

The data set is organized in order of flower kind. So, this sample takes 45 observations from setosa (as the first 50 observations in the data set are setosa), then 14 from versicolor (observations 58, 60-70, 82, and 94), and the remaining 41 from virginica. So, the code directly creates a sample in which the proportions of flower types differ greatly from their proportions in the original data set. In particular, there are far fewer versicolor in the training data set than the testing data set.

Choosing a poorly representative sample degrades the performance of KNN because classification algorithms are optimized on training data, not testing data. What is optimal for training may differ from what is optimal for testing, and the algorithm can fail to generalize as a result. For instance, since the training data contains so few versicolors compared to the testing data, we can notice from the confusion matrix that a higher proportion of versicolors are misclassified in the testing data compared to setosas and virginicas.

Choice of K can also influence this classifier. Why would choosing $K = 6$ not be advisable for this data?

Choosing $K = 6$ is not advisable for this data because 6 is divisible by the number of categories (3). This will create a situation in which "ties" can arise, which leads the computer to classify a given data point with an arbitrary category (since there is an equal chance it will be one of the three categories). This is not ideal because our goal is to actually utilize the data given to classify a point, not just arbitrarily choose a category.

Build a github repository to store your homework assignments. Share the link in this file.

https://github.com/ava742/Stor390-hw/