Homework 3 - Similarity Based Learning

Nelson Corrocher

March 23, 2017

Homework

1. Machine Learning with R - Lantz: Review chapter 3 and understand how to use the R tools described
2. Complete the following problems from the Kelleher book (Chapter 5):

- 1
- 3

```
# Auxiliary normalization function
normalize <- function(x, low, high) {
    return((x - min(x))/(max(x) - min(x)) * (high - low) + low)
}
```

Exercise 1

```
ex1data <- read.csv("ch5ex1.csv")

# Normalizing the inputs
ex1train <- ex1data
ex1train$WAVE_SIZE_.FT. <- normalize(ex1train$WAVE_SIZE_.FT., 0, 1)
ex1train$WAVE_PERIOD_.SECS. <- normalize(ex1train$WAVE_PERIOD_.SECS., 0, 1)
ex1train$WIND_SPEED_.MPH. <- normalize(ex1train$WIND_SPEED_.MPH., 0, 1)

# Create the query set
ex1test <- ex1train[7:9, ]
ex1test <- ex1test[-c(1, 5)]

# Create the class set (knn requires it to be factors)
ex1class <- factor(ex1train[1:6, 5])

# Training set: removing ID and GOOD_SURF and removing the testing sample
# from it
ex1train <- ex1train[1:6, -c(1, 5)]

k <- 3  # Number of neighbors to be used

# Function knn from class:
ex1test <- data.frame(GOOD_SURF = knn(ex1train, ex1test, ex1class, k))
rownames(ex1test) <- c("Q1", "Q2", "Q3")
ex1test
```

```
##    GOOD_SURF
## Q1       yes
## Q2        no
## Q3       yes
```

Exercise 3

```
# For this exercise, the package kknn will be used since it supports
# weighted k-NN algorithm, which is more sophiscated than the regular knn
# from class
ex3data <- read.csv("ch5ex3.csv")
```

a. What value would a 3-nearest neighbor prediction model using Euclidean distance return for the CPI of Russia?

```
kn.a <- kknn(formula = CPI ~ LIFE_EXP. + TOP_10_INCOME + INFANT_MORT. + MIL.SPEND +
    SCHOOL_YEARS, train = ex3data[1:16, -1], test = ex3data[17, -c(1, 7)], k = 3,
    distance = 2, kernel = "rectangular", scale = FALSE)
fitted(kn.a)
```

```
## [1] 4.589133
```

b. What value would a weighted k-NN prediction model return for the CPI of Russia? Use k = 16 (i.e., the full dataset) and a weighting scheme of the reciprocal of the squared Euclidean distance between the neighbor and the query

```
# kn.b <- kknn(formula = CPI ~ LIFE_EXP. + TOP_10_INCOME + INFANT_MORT. +
# MIL.SPEND + SCHOOL_YEARS, train = ex3data[1:16,-1], test =
# ex3data[17,-c(1,7)], k = 16, distance = 2, kernel = 'inv', scale = FALSE,
# use.all = TRUE) fitted(kn.b)
```

I couldn't find a package in R that correctly calculated this case. knn doesn't has any options for weighted knn and kknn randomly selects the nodes when k is set to high, which gives a different results each time. I did the calculation in excel (attached) and found it to be 5.90870754.

c. The descriptive features in this dataset are of different types. For example, some are percentages, others are measured in years, and others are measured in counts per 1,000. We should always consider normalizing our data, but it is particularly important to do this when the descriptive features are measured in different units. What value would a 3-nearest neighbor prediction model using Euclidean distance return for the CPI of Russia when the descriptive features have been normalized using range normalization?

```
# Normalizing the dataset
ex3data_norm <- ex3data
ex3data_norm$LIFE_EXP. <- normalize(ex3data$LIFE_EXP., 0, 1)
ex3data_norm$TOP_10_INCOME <- normalize(ex3data$TOP_10_INCOME, 0, 1)
ex3data_norm$INFANT_MORT. <- normalize(ex3data$TOP_10_INCOME, 0, 1)
ex3data_norm$MIL.SPEND <- normalize(ex3data$MIL.SPEND, 0, 1)
ex3data_norm$SCHOOL_YEARS <- normalize(ex3data$SCHOOL_YEARS, 0, 1)

kn.c <- kknn(formula = CPI ~ LIFE_EXP. + TOP_10_INCOME + INFANT_MORT. + MIL.SPEND +
    SCHOOL_YEARS, train = ex3data_norm[1:16, -1], test = ex3data_norm[17, -c(1,
    7)], k = 3, distance = 2, kernel = "rectangular", scale = FALSE)
fitted(kn.c)
```

```
## [1] 5.968967
```

d. What value would a weighted k-NN prediction model-with k = 16 (i.e., the full dataset) and using a weighting scheme of the reciprocal of the squared Euclidean distance between the neighbor and the query-return for the CPI of Russia when it is applied to the range-normalized data?

```
# kn.d <- kknn(formula = CPI ~ LIFE_EXP. + TOP_10_INCOME + INFANT_MORT. +
# MIL.SPEND + SCHOOL_YEARS, train = ex3data_norm[1:16,-1], test =
# ex3data_norm[17,-c(1,7)], k = 16, distance = 2, kernel = 'inv', scale =
# FALSE) fitted(kn.d)
```

I couldn't find a package in R that correctly calculated this case. knn doesn't has any options for weighted knn and kknn randomly selects the nodes when k is set to high, which gives a different results each time. I did the calculation in excel (attached) and found it to be 6.63466120.

e.The actual 2011 CPI for Russia was 2.4488. Which of the predictions made was the most accurate? Why do you think this was?

The most accurate prediction comes from a), with k = 3 and using Euclidean distance. When I started doing the exercise, I believed that the one that would be most accurate would be the weighted averaged with the entire dataset because that approach gives more weight to the closest points which is important in a continuous result (average of the closes CPIs). After giving some though to it, I believe that there are some important variable not being covered in the test. For example, check Argentina, China and USA: even though they are the closest points to Russia, their CPI differs greatly. Since the units are very different, running the test in a non-normalized set doesn't really make much sense. Thus, a) getting the closest results looks like a coincidence.