kNN Simulation

MA Mingyu 14110562D

4/4/2017

COMP4433 Assignment 2 Question 2 a, b and c

derek.ma@connect.polyu.hk derek.ma

Import Training Data

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
survey <- read.csv("~/Google Drive/_DM/2_Assignments/Ass2/data_q2.csv", stringsAsFactors = FALSE)
colnames(survey) <- c ("customerNo",
                  "averageMonthlyPayment",
                  "averageDurationOfCalls",
                  "totalCallingTime","decision")
```

Basic Function - Calculate Distance

```
distance <- function(vector1, vector2){
  #Euclidean distance
  count <- 0
  for (i in 1:length(vector1)){
    count = count + (vector1[i] - vector2[i])^2
  }
  count^(1/2)
}
```

Set Value of k and Testing Data

```
k <- 5
testData <- c(293.26,16.96,120.25)
```

Preprocessing the Data, Normalization

```
#Normalization
min1 <- min(survey$averageMonthlyPayment)
max1 <- max(survey$averageMonthlyPayment)
min2 <- min(survey$averageDurationOfCalls)
max2 <- max(survey$averageDurationOfCalls)
min3 <- min(survey$totalCallingTime)
max3 <- max(survey$totalCallingTime)
testDataNormalized <- c(
  (testData[1]-min1)/(max1-min1),
  (testData[2]-min2)/(max2-min2),
  (testData[3]-min3)/(max3-min3))
survey <- survey %>%
  mutate(averageMonthlyPaymentNormalized = (averageMonthlyPayment - min1)/(max1-min1)) %>%
  mutate(averageDurationOfCallsNormalized = (averageDurationOfCalls - min2)/(max2-min2)) %>%
  mutate(totalCallingTimeNormalized = (totalCallingTime - min3)/(max3-min3))
survey
```

```
##    customerNo averageMonthlyPayment averageDurationOfCalls
## 1           1                273.43                   8.70
## 2           2                342.10                  12.00
## 3           3                197.54                   4.40
## 4           4                409.86                  17.28
## 5           5                291.94                   9.00
## 6           6                404.43                  17.40
## 7           7                218.24                   3.96
## 8           8                214.72                   8.04
## 9           9                378.62                  18.60
## 10         10                373.78                   9.24
## 11         11                195.36                   5.88
## 12         12                320.32                  14.76
## 13         13                264.11                   5.70
## 14         14                462.44                   2.64
## 15         15                259.16                  11.88
## 16         16                430.44                   3.14
## 17         17                352.00                   8.04
## 18         18                220.66                   2.16
## 19         19                215.16                   6.84
## 20         20                317.68                  10.68
##    totalCallingTime  decision averageMonthlyPaymentNormalized
## 1             98.70 Undecided                     0.292309420
## 2             96.38      Stay                     0.549423394
## 3            147.30      Stay                     0.008162348
## 4            180.50    Switch                     0.803130148
## 5            111.13      Stay                     0.361614498
## 6            171.70    Switch                     0.782799161
## 7            124.88      Stay                     0.085667216
## 8             96.88    Switch                     0.072487644
## 9             83.50 Undecided                     0.686161450
## 10           122.50 Undecided                     0.668039539
## 11           138.88    Switch                     0.000000000
## 12            97.25    Switch                     0.467874794
## 13           107.10    Switch                     0.257413509
## 14           162.38 Undecided                     1.000000000
## 15            82.50      Stay                     0.238879736
## 16           100.74 Undecided                     0.880185712
## 17            56.00 Undecided                     0.586490939
## 18            69.75    Switch                     0.094728171
## 19            41.63      Stay                     0.074135091
## 20           126.38    Switch                     0.457990115
##    averageDurationOfCallsNormalized totalCallingTimeNormalized
## 1                        0.39781022                  0.4109599
## 2                        0.59854015                  0.3942536
## 3                        0.13625304                  0.7609275
## 4                        0.91970803                  1.0000000
## 5                        0.41605839                  0.5004681
## 6                        0.92700730                  0.9366314
## 7                        0.10948905                  0.5994815
## 8                        0.35766423                  0.3978541
## 9                        1.00000000                  0.3015050
## 10                       0.43065693                  0.5823432
## 11                       0.22627737                  0.7002952
## 12                       0.76642336                  0.4005185
## 13                       0.21532847                  0.4714481
## 14                       0.02919708                  0.8695183
## 15                       0.59124088                  0.2943040
## 16                       0.05961071                  0.4256499
## 17                       0.35766423                  0.1034781
## 18                       0.00000000                  0.2024915
## 19                       0.28467153                  0.0000000
## 20                       0.51824818                  0.6102830
```

Question 2a

Assumption: if one testing node has the same count of nodes of specific properties, then the prediction of testing node will depends on the distance between the training nodes and testing node. For example, when k = 5, if two nodes are "switch", two nodes are "stay" and one node is "undecided" among the five nearest nodes, then I will compare the distance between two "switches" and testing node and the distance between two "stay" and testing node. If the sum of distance of two "switch" is smaller, then I will predict "switch" for this testing node.

```r
survey <- survey %>%
  mutate(dist=NA)

for (i in 1:length(survey$customerNo)){
  survey[i,"dist"] <- distance(testDataNormalized,
                      c(survey[i,"averageMonthlyPaymentNormalized"],
                      survey[i,"averageDurationOfCallsNormalized"],
                      survey[i,"totalCallingTimeNormalized"]))
}

surveySorted <- survey[order(survey$dist),]
surveySorted1 <- surveySorted[1:k,]
surveySorted1
```

```
##    customerNo averageMonthlyPayment averageDurationOfCalls
## 12         12                320.32                  14.76
## 2           2                342.10                  12.00
## 20         20                317.68                  10.68
## 9           9                378.62                  18.60
## 15         15                259.16                  11.88
##    totalCallingTime  decision averageMonthlyPaymentNormalized
## 12            97.25    Switch                       0.4678748
## 2             96.38      Stay                       0.5494234
## 20           126.38    Switch                       0.4579901
## 9             83.50 Undecided                       0.6861614
## 15            82.50      Stay                       0.2388797
##    averageDurationOfCallsNormalized totalCallingTimeNormalized      dist
## 12                        0.7664234                  0.4005185 0.2358049
## 2                         0.5985401                  0.3942536 0.3924414
## 20                        0.5182482                  0.6102830 0.3952579
## 9                         1.0000000                  0.3015050 0.4267678
## 15                        0.5912409                  0.2943040 0.4309052
```

```
result <- data.frame(decision=NA, count=NA, sumDist=NA)
i <- 1
for (deci in unique(surveySorted1$decision)){
  temp <- surveySorted1[surveySorted1$decision==deci,]
  result[i,] <- c(deci,nrow(temp),sum(temp$dist))
  i <- i + 1
}
result <- result[order(-rank(result$count), result$sumDist),]
result
```

```
##    decision count           sumDist
## 1    Switch     2 0.631062797616398
## 2      Stay     2 0.823346583581644
## 3 Undecided     1 0.426767751146862
```

We can found that for the five nodes that are closest to the test node, two nodes are “switch”, two nodes are “stay”, and one node is “undecided.”

While the distance of two “switch” nodes are smaller, thus the expected decision of the customer who has an average monthly payment of 293.26, an average duration of calls of 16.96 and a total calling time of 120.25 is “switch”.

Question 2b

Without considering “Decision”, we try quesiont 2b. After get the five nearest nodes, average the distance and get the final result.

```
testData <- c(271.48,184)
max3 <- 184
survey <- survey %>%
  mutate(averageMonthlyPaymentNormalized = (averageMonthlyPayment - min1)/(max1-min1)) %>%
  mutate(averageDurationOfCallsNormalized = (averageDurationOfCalls - min2)/(max2-min2)) %>%
  mutate(totalCallingTimeNormalized = (totalCallingTime - min3)/(max3-min3))
survey <- survey %>%
  mutate(dist=NA)
testDataNormalized <- c(
  (testData[1]-min1)/(max1-min1),
  (testData[2]-min3)/(max3-min3))

for (i in 1:length(survey$customerNo)){
  survey[i,"dist"] <- distance(testDataNormalized,
                      c(survey[i,"averageMonthlyPaymentNormalized"],
                      survey[i,"totalCallingTimeNormalized"]))
}

surveySorted <- survey[order(survey$dist),]
surveySorted1 <- surveySorted[1:k,]
surveySorted1
```

```
##    customerNo averageMonthlyPayment averageDurationOfCalls
## 3           3                197.54                   4.40
## 11         11                195.36                   5.88
## 20         20                317.68                  10.68
## 7           7                218.24                   3.96
## 6           6                404.43                  17.40
##    totalCallingTime decision averageMonthlyPaymentNormalized
## 3            147.30     Stay                     0.008162348
## 11           138.88   Switch                     0.000000000
## 20           126.38   Switch                     0.457990115
## 7            124.88     Stay                     0.085667216
## 6            171.70   Switch                     0.782799161
##    averageDurationOfCallsNormalized totalCallingTimeNormalized      dist
## 3                         0.1362530                  0.7422210 0.3782772
## 11                        0.2262774                  0.6830793 0.4262258
## 20                        0.5182482                  0.5952799 0.4401376
## 7                         0.1094891                  0.5847440 0.4606239
## 6                         0.9270073                  0.9136054 0.5052325
```

```
surveySorted1 %>% select(customerNo,averageDurationOfCalls)
```

```
##    customerNo averageDurationOfCalls
## 3           3                   4.40
## 11         11                   5.88
## 20         20                  10.68
## 7           7                   3.96
## 6           6                  17.40
```

```
mean(surveySorted1$averageDurationOfCalls)
```

```
## [1] 8.464
```

Thus the final expected average duration of calls of a customer whose average monthly payment is 271.48 and total calling time is 184.00 is 8.464.

Question 2c

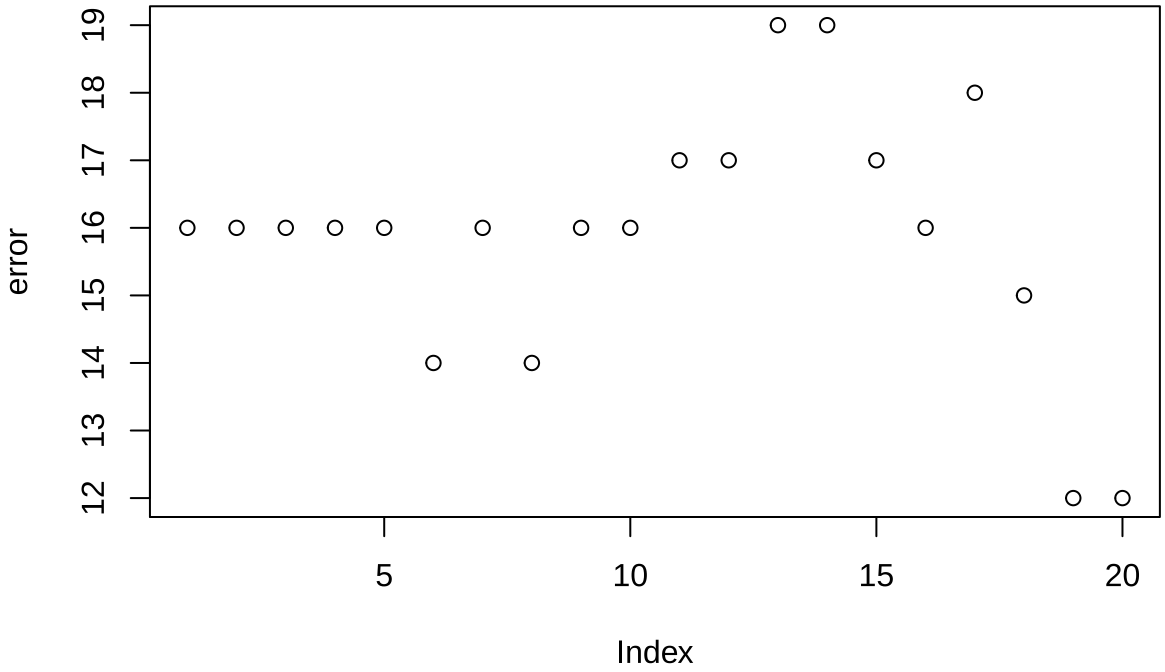
Cross Validation and Choose the Value of k

Simulate the leave-one-out cross validation:

```r
error <- rep(NA,20)
k <- 1
for (k in 1:length(error)){
  errorCount <- 0
  for (testIndex in 1:nrow(survey)){
    testData <- c(survey[testIndex,"averageMonthlyPaymentNormalized"],
                  survey[testIndex,"averageDurationOfCallsNormalized"],
                  survey[testIndex,"totalCallingTimeNormalized"])
    survey <- survey %>%
      mutate(dist=NA)
    for (i in 1:length(survey$customerNo)){
      if(i != testIndex){
        survey[i,"dist"] <- distance(testData,
                            c(survey[i,"averageMonthlyPaymentNormalized"],
                            survey[i,"averageDurationOfCallsNormalized"],
                            survey[i,"totalCallingTimeNormalized"]))
      }
    }
    surveySorted <- survey[order(survey$dist),]
    surveySorted1 <- surveySorted[1:k,]
    result <- data.frame(decision=NA, count=NA, sumDist=NA)
    i <- 1
    for (deci in unique(surveySorted1$decision)){
      temp <- surveySorted1[surveySorted1$decision==deci,]
      result[i,] <- c(deci,nrow(temp),sum(temp$dist))
      i <- i + 1
    }
    result <- result[order(-rank(result$count), result$sumDist),]
    if (result$decision[1] != survey$decision[testIndex]){
      errorCount <- errorCount + 1
    }
  }
  error[k] <- errorCount
}
error
```

```
##  [1] 16 16 16 16 16 14 16 14 16 16 17 17 19 19 17 16 18 15 12 12
```

```r
plot(error)
```



In the plot, the x axis shows the k value and y axis shows the count of error. From the result we can find that when k is equal to 19 the error count is smallest. While when k is 6, 8, the error count is relatively small. They are all good choices for k value.

But due to there are not enough records for choosing k and I only try one method to choose k, this result is just constructive. I think I should try more cross validation methods such as k-fold cross validation and try to observe the best value of k.