Math 656 Midterm

Jeff Gould

10/14/2020

First we load the arrhythmia data set and run J48 under the default arguments, and test under 10 fold cross-validation. We make no cleaning steps to the underlying data. We see that under cross-validation we achieve 64.823% accuracy:

```
arrhythmia_load <- foreign::read.arff("arrhythmia.arff")

arrhythmia_J48_raw <- J48(class ~ ., data = arrhythmia_load, na.action = NULL)
evaluate_Weka_classifier(arrhythmia_J48_raw, numFolds = 10, seed=1)
```

```
## === 10 Fold Cross Validation ===
##
## === Summary ===
##
## Correctly Classified Instances         293               64.823  %
## Incorrectly Classified Instances       159               35.177  %
## Kappa statistic                          0.4702
## Mean absolute error                      0.0595
## Root mean squared error                  0.2208
## Relative absolute error                 56.637  %
## Root relative squared error             96.8344 %
## Total Number of Instances              452
##
## === Confusion Matrix ===
##
##    a   b   c   d   e   f   g   h   i   j   k   l   m   <-- classified as
##  199  10   3   1   9  13   0   2   2   5   0   0   1 |   a = 1
##   15  24   1   0   0   1   0   5   1   3   0   0   0 |   b = 10
##    1   0   1   0   1   1   0   0   0   0   0   0   0 |   c = 14
##    1   1   0   0   1   1   1   0   0   0   0   0   0 |   d = 15
##   10   5   1   1   0   1   1   0   1   1   0   0   1 |   e = 16
##   14   4   1   1   1  20   0   3   0   0   0   0   0 |   f = 2
##    0   0   0   1   0   1  12   0   0   1   0   0   0 |   g = 3
##    3   1   0   0   0   1   0   8   1   0   0   0   1 |   h = 4
##    5   0   0   1   0   0   0   0   7   0   0   0   0 |   i = 5
##    5   4   0   0   0   0   1   0   0  15   0   0   0 |   j = 6
##    1   0   0   0   0   0   1   1   0   0   0   0   0 |   k = 7
##    1   0   0   0   1   0   0   0   0   0   0   0   0 |   l = 8
##    1   0   0   0   1   0   0   0   0   0   0   0   7 |   m = 9
```

We see that while there are a total of 408 missing data points, most of them are in the `J` variable, for which 376 of the 452 observations are missing a data point.

```
findMissing <- colSums(is.na(arrhythmia_load))
findMissing[findMissing>0]
```

```
##         T         P      QRST         J heartrate
##         8        22         1       376         1
```

To fill in the missing values, we use K-Nearest Neighbors imputation. We will use $20 \approx \sqrt{452}$ nearest neighbors, and take the mean of those values to fill in the missing data.

```
preProcValues <- preProcess(arrhythmia_load,
                            method = c("knnImpute"),
                            k = 20,
                            knnSummary = mean)

impute_arryhthmia_info <- predict(preProcValues, arrhythmia_load, na.action = na.pass)

procNames <- data.frame(col = names(preProcValues$mean), mean = preProcValues$mean, sd = preProcValues$s

for(i in procNames$col){
 impute_arryhthmia_info[i] <- impute_arryhthmia_info[i]*preProcValues$std[i]+preProcValues$mean[i]
}
```

Now test J48 on the imputed data, and also adjust the minimum object number to 10. This number just came from some trial and error, as accuracy seemed to increase up until this point, but then began to decrease once I moved it higher. With these changes, we find that we are able to achieve 69.6903% accuracy under 10 fold cross-validation.

```
arrhythmia_J48_cleaned <- J48(class ~ ., data = impute_arryhthmia_info, control = Weka_control(M=10))
crosVal <- evaluate_Weka_classifier(arrhythmia_J48_cleaned, numFolds = 10, seed=1)
crosVal
```

```
## === 10 Fold Cross Validation ===
##
## === Summary ===
##
## Correctly Classified Instances         315               69.6903 %
## Incorrectly Classified Instances       137               30.3097 %
## Kappa statistic                          0.5369
## Mean absolute error                      0.0626
## Root mean squared error                  0.1937
## Relative absolute error                 59.5977 %
## Root relative squared error             84.9315 %
## Total Number of Instances              452
##
## === Confusion Matrix ===
##
##    a   b   c   d   e   f   g   h   i   j   k   l   m   <-- classified as
##  209  10   0   0   0  14   0   3   3   5   0   0   1 |   a = 1
##   11  26   0   0   0   0   0   4   2   5   0   0   2 |   b = 10
##    2   1   0   0   0   0   0   1   0   0   0   0   0 |   c = 14
##    1   0   0   0   0   1   1   1   1   0   0   0   0 |   d = 15
##   10   5   0   0   0   2   2   0   1   2   0   0   0 |   e = 16
```

```
##   17   3   0   0   0  18   0   3   1   1   0   0   1 |   f = 2
##    0   0   0   0   0   1  14   0   0   0   0   0   0 |   g = 3
##    5   0   0   0   0   0   0   9   1   0   0   0   0 |   h = 4
##    3   0   0   0   0   0   0   0  10   0   0   0   0 |   i = 5
##    2   0   0   0   0   0   1   0   0  22   0   0   0 |   j = 6
##    2   0   0   0   0   0   1   0   0   0   0   0   0 |   k = 7
##    1   0   0   0   0   0   0   1   0   0   0   0   0 |   l = 8
##    2   0   0   0   0   0   0   0   0   0   0   0   7 |   m = 9
```

In order to narrow down the vairables to choose for sorting, we followed the followign process:

- Select a random sample of 100 variables and run J48 and cross-validation
- If the variable sample improves our accuracy, store the variables selected, otherwise return a 0 vector
- Repeat for a total of 1000 random variable samples
- Take the $\approx$ 75 most common variables that improved our classification, and re-run the classifier

Note, we observed earlier that the following columns have no variation, so since we know that they will not create any entropy gain, we remove them from our sample ahead of time:

`chDI_SPwave, chAVL_SPwave, chV5_SPwave, chV6_SPwave, chDI_SPwaveAmp, chAVL_SPwaveAmp, chV5_SPwaveAmp, chV6_SPwaveAmp`

```
current_best <- crosVal[["details"]][1]
#baggedCols <- c(sample(c(1:279), 150), 280)
impute_arryhthmia_info <- impute_arryhthmia_info %>%
  select(-chDI_SPwave, -chAVL_SPwave, -chV5_SPwave, -chV6_SPwave, -chDI_SPwaveAmp,
         -chAVL_SPwaveAmp, -chV5_SPwaveAmp, -chV6_SPwaveAmp)

set.seed(123)
randCols <- t(sapply(rep(100, 1000), function(x){c(sample(c(1:271), x), 272)}))


library(parallel)
cl <- makeCluster(detectCores() - 1)


suppressMessages(clusterEvalQ(cl, library(RWeka)))
```

```
## [[1]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[2]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[3]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[4]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
```

```
##
## [[5]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[6]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[7]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[8]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[9]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[10]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[11]]
## [1] "RWeka"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
```

```
clusterExport(cl, c("impute_arryhthmia_info", "current_best"))

baggedCols <- t(parApply(cl, randCols, 1, function(x){
  #require(RWeka)
  J48_bag <- J48(class ~ ., data = impute_arryhthmia_info[, x], control = Weka_control(M=10))
  crosValBag <- evaluate_Weka_classifier(J48_bag, numFolds = 10, seed=1)

  if(crosValBag[["details"]][1] > current_best){
    return(c(x, crosValBag[["details"]][1]))
  }else(return(rep(0, length(x)+1)))

}))


improvedCols <- baggedCols[,-102] %>% unique() %>% as.vector() %>% sort()
improvedCols <- improvedCols[improvedCols!=0]
instances <- sapply(c(1:271), function(x)sum(improvedCols == x))
df <- data.frame(varIndex = c(1:271),
                 instances = instances) %>% arrange(desc(instances))

colSample <- df %>%
  top_n(75, instances) %>%
  select(varIndex) %>% as.vector()

J48new <- J48(class ~ ., data = impute_arryhthmia_info[,c(colSample$varIndex, 272)], control = Weka_con
```

```
evaluate_Weka_classifier(J48new, numFolds = 10, seed=1)
```

```
## === 10 Fold Cross Validation ===
##
## === Summary ===
##
## Correctly Classified Instances         331               73.2301 %
## Incorrectly Classified Instances       121               26.7699 %
## Kappa statistic                          0.5889
## Mean absolute error                      0.0607
## Root mean squared error                  0.1845
## Relative absolute error                 57.713  %
## Root relative squared error             80.9226 %
## Total Number of Instances              452
##
## === Confusion Matrix ===
##
##    a   b   c   d   e   f   g   h   i   j   k   l   m   <-- classified as
##  215  10   0   0   0   8   0   2   5   5   0   0   0 |   a = 1
##   12  30   0   0   0   0   0   1   2   5   0   0   0 |   b = 10
##    2   1   0   0   0   0   0   1   0   0   0   0   0 |   c = 14
##    1   0   0   0   0   0   1   1   2   0   0   0   0 |   d = 15
##   10   4   0   0   0   1   2   1   2   2   0   0   0 |   e = 16
##   15   2   0   0   0  22   1   1   1   1   0   0   1 |   f = 2
##    0   0   0   0   0   1  14   0   0   0   0   0   0 |   g = 3
##    4   0   0   0   0   1   0   9   1   0   0   0   0 |   h = 4
##    3   0   0   0   0   0   0   0  10   0   0   0   0 |   i = 5
##    2   0   0   0   0   0   1   0   0  22   0   0   0 |   j = 6
##    2   0   0   0   0   0   1   0   0   0   0   0   0 |   k = 7
##    2   0   0   0   0   0   0   0   0   0   0   0   0 |   l = 8
##    0   0   0   0   0   0   0   0   0   0   0   0   9 |   m = 9
```

After testing the J48 classification on the $\approx$ 75 (actually 84) most common variables that improved our classification, we find that we were able to improve accuracy to 73.23% under 10 fold cross-validation

Now we try to reiterate the same steps as above, except this time we will pull samples of 50 columns from the $\approx$ 100 most frequent variables that showed up in our samples that improved the classifier.

```
new_best <- evaluate_Weka_classifier(J48new, numFolds = 10, seed=1)[["details"]][1]

top100Vars <- df %>%
  top_n(95, instances) %>%
  select(varIndex) %>% as.vector()

set.seed(123)
rand50 <- t(sapply(rep(50, 1000), function(x){c(sample(top100Vars$varIndex, x),272)}))

clusterExport(cl, c("impute_arryhthmia_info", "new_best"))

baggedCols2 <- t(parApply(cl, rand50, 1, function(x){

  J48_bag <- J48(class ~ ., data = impute_arryhthmia_info[, x], control = Weka_control(M=10))
  crosValBag <- evaluate_Weka_classifier(J48_bag, numFolds = 10, seed=1)
```

```
  if(crosValBag[["details"]][1] > new_best*0.99){
    return(c(x, crosValBag[["details"]][1]))
  }else(return(rep(0, length(x)+1)))

}))
stopCluster(cl)

improvedCols2 <- baggedCols2[,-(51:52)] %>% unique() %>% as.vector() %>% sort()
improvedCols2 <- improvedCols2[improvedCols2!=0]

instances2 <- sapply(c(1:271), function(x)sum(improvedCols2 == x))
df <- data.frame(varIndex = c(1:271),
                 instances = instances2) %>% arrange(desc(instances))

colSample <- df %>%
  top_n(25, instances) %>%
  select(varIndex) %>% as.vector()
```

We were to improve the accuracy from the previous step to 74.7788%, and we were able to do that with just 25 variables. This is significantly fewer and much easier classification. And furthermore, it's about a 10 percentage point increase from running the default classifier on the raw data.

```
J48newest <- J48(class ~ ., data = impute_arryhthmia_info[,c(colSample$varIndex, 272)], control = Weka_
evaluate_Weka_classifier(J48newest, numFolds = 10, seed=1)
```

```
## === 10 Fold Cross Validation ===
##
## === Summary ===
##
## Correctly Classified Instances         338               74.7788 %
## Incorrectly Classified Instances       114               25.2212 %
## Kappa statistic                          0.616
## Mean absolute error                      0.0594
## Root mean squared error                  0.1796
## Relative absolute error                 56.5578 %
## Root relative squared error             78.7546 %
## Total Number of Instances              452
##
## === Confusion Matrix ===
##
##    a   b   c   d   e   f   g   h   i   j   k   l   m   <-- classified as
##  215   9   0   0   0   9   0   2   5   5   0   0   0 |   a = 1
##    8  34   0   0   0   0   0   1   2   5   0   0   0 |   b = 10
##    2   1   0   0   0   0   0   1   0   0   0   0   0 |   c = 14
##    1   0   0   0   0   0   1   1   2   0   0   0   0 |   d = 15
##   10   4   0   0   0   1   2   1   2   2   0   0   0 |   e = 16
##   13   2   0   0   0  25   0   1   1   1   0   0   1 |   f = 2
##    0   0   0   0   0   1  14   0   0   0   0   0   0 |   g = 3
##    4   0   0   0   0   1   0   9   1   0   0   0   0 |   h = 4
##    3   0   0   0   0   0   0   0  10   0   0   0   0 |   i = 5
##    2   0   0   0   0   0   1   0   0  22   0   0   0 |   j = 6
##    2   0   0   0   0   0   1   0   0   0   0   0   0 |   k = 7
```

```
##    2   0   0   0   0   0   0   0   0   0   0   0   0 |   l = 8
##    0   0   0   0   0   0   0   0   0   0   0   0   9 |   m = 9
```

Here are the variables that the classifier ended up using:

```
colnames(impute_arryhthmia_info[,c(colSample$varIndex)])
```

```
##  [1] "heartrate"                   "chV5_TwaveAmp"
##  [3] "chDII_QwaveAmp"              "chAVF_QwaveAmp"
##  [5] "chV4_TwaveAmp"               "chAVR_RPwaveExists"
##  [7] "chV1_JJwaveAmp"              "chV1_RPwaveAmp"
##  [9] "chV2_RPwaveAmp"              "chAVR_DD_RRwaveExists"
## [11] "chV3_Rwave"                  "chV3_Swave"
## [13] "chV2_QRSA"                   "chDIII_DD_RTwaveExists"
## [15] "J"                           "chDI_Qwave"
## [17] "chV5_QRSA"                   "chV5_PwaveAmp"
## [19] "chDI_intrinsicReflecttions" "chV2_Swave"
## [21] "chV3_intrinsicReflecttions" "chDI_QRSA"
## [23] "chAVR_QRSA"                  "chV2_PwaveAmp"
## [25] "chV5_RPwaveAmp"
```

And here is the final decision tree:

```
J48newest
```

```
## J48 pruned tree
## ------------------
##
## chV1_JJwaveAmp <= 2.2
## |   chV3_intrinsicReflecttions <= 0: 3 (19.0/5.0)
## |   chV3_intrinsicReflecttions > 0
## |   |   heartrate <= 58
## |   |   |   chV5_PwaveAmp <= 0.6: 6 (26.0/5.0)
## |   |   |   chV5_PwaveAmp > 0.6: 1 (11.0/6.0)
## |   |   heartrate > 58
## |   |   |   heartrate <= 101
## |   |   |   |   chV1_RPwaveAmp <= 0.9
## |   |   |   |   |   chDII_QwaveAmp <= -1.4: 4 (18.0/7.0)
## |   |   |   |   |   chDII_QwaveAmp > -1.4
## |   |   |   |   |   |   chV4_TwaveAmp <= -0.5: 2 (27.0/4.0)
## |   |   |   |   |   |   chV4_TwaveAmp > -0.5
## |   |   |   |   |   |   |   chV2_RPwaveAmp <= 0: 1 (276.0/48.0)
## |   |   |   |   |   |   |   chV2_RPwaveAmp > 0: 10 (15.0/7.0)
## |   |   |   |   chV1_RPwaveAmp > 0.9: 10 (35.0/8.0)
## |   |   |   heartrate > 101: 5 (14.0/4.0)
## chV1_JJwaveAmp > 2.2: 9 (11.0/2.0)
##
## Number of Leaves  :  10
##
## Size of the tree :   19
```