Angelo Parker
Math 3220 Data Ming Methods Final

Problem Description

In a previous report, Classification methods were discussed and compared, those methods were C5.0 and Rpart. This report will include Support Vector Machines (SVMs) compare it to C5.0 and Rpart and display the differences and any pros or cons associated with each algorithm.

Background

Classification "trees", or methods, are algorithms designed to organize and classify data for analysis and prediction. Usually, a "trained" data set must be used to fit an algorithm to a particular situation in order to be able to interpret new, "untrained" data. Against the trained data, the algorithm will form its own set of classifications based on designated groups that the researcher wants to apply, or that already exist within the data set, and compare them to the original data assignments. The algorithm may create a ratio of how well it was able to classify the data in that particular test. The researcher may go forth with more testing and adjusting to make the algorithm's rules more or less "fitted" to the data set. Previously, a report has been done comparing C5.0 and Rpart, while the documentation for SVM compares SVM to Rpart. This report will look at all three with the three data sets used from the first report.

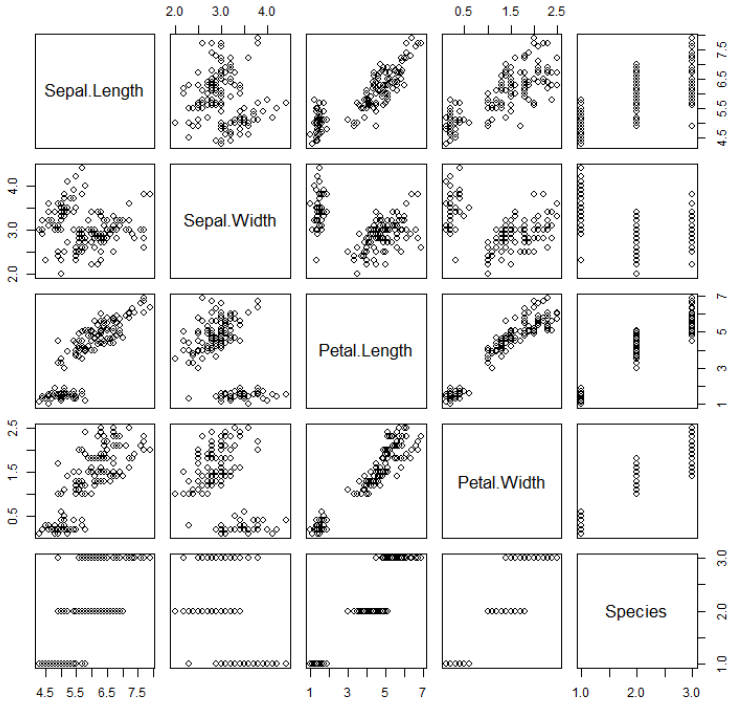
C5.0 is an improvement of the C4.5 algorithm created by Ross Quinlan in 1992. In R and R studio, the data set and a strict matrix of what will be used as classifiers must be given. The basis for C5.0 is Information gain and minimizing entropy or the amount of uncertainty.

RPart, the R equivalent of CART, utilizes an algorithm that uses information gain followed by calculating Gini impurity, the chance of an instance being labeled incorrectly. The idea is to decrease the amount of variance in a particular instance to determine the most significant attributes and in the end, classifications. Cart was developed by four authors Breiman, Friedman, Olshen, and Stone in 1984 (Brieman, 2017).

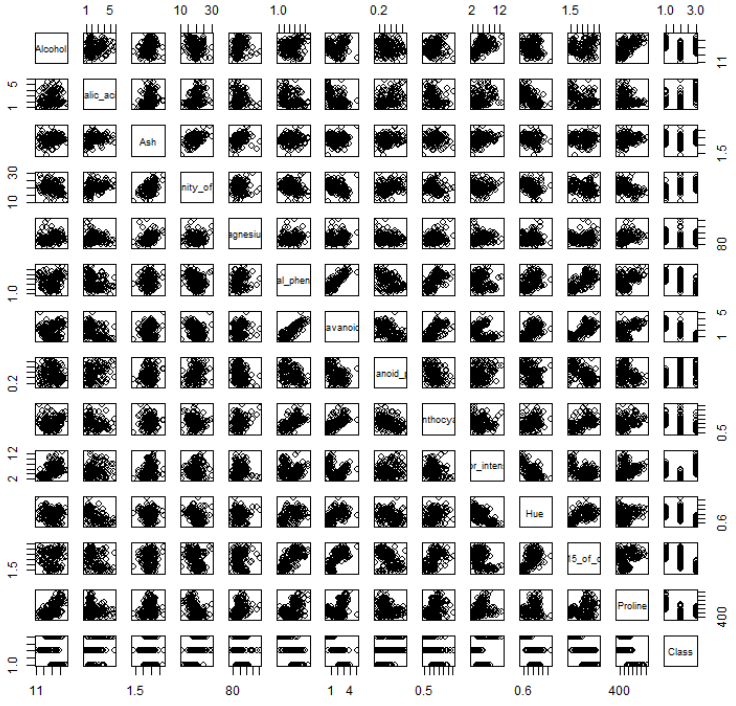
Support Vector Machines (SVMs) are more easily described as a graphical division, where a line is plotted that best splits the data into two with the maximum distance between the data and the created line. SVMs are a binary form of classification, meaning they can only split the data into two groups. Programmable implementations have been designed to project multiple SVMs if the goal is to create more than two classifications, so the extra work does not fall onto the researcher. SVMs do not have to be linear, for which algorithms exist for. The main author for SVMs has been Vladmir N. Vapnik but has seen other authors and editors over each new development.

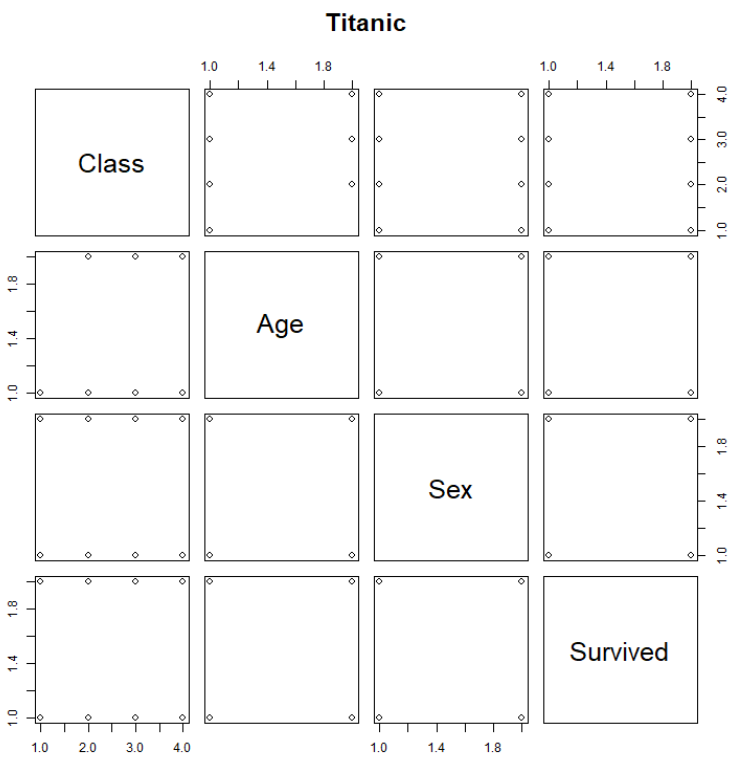
Three fairly standard data sets will be used for this report: the registrar of the Titanic Passengers with four attributes and 2201 instances (despite there being 2224 recorded passengers), 153 types of Wine from 3 different Italian sources with 13 attributes (originally 30), and the classic Iris data of 3 species and the 4 measurements used to define them. Pairs plots for each of the data sets are provided below.

Iris


Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species

Wine


Alcohol
Ash
Hue
Proline
Class



Methodology

SVM will each run with the three data sets in Rstudio and the former results from the experiment with Rpart and C5.0 will be given. Each experiment will be separate with all of the results recorded. The results will be compared, including information that is or is not present in each algorithm, their rule sets and their variations, the misclassifications, graphical representations, etc. The manual inputs, formula length and complexity, and run time will be included if applicable.

Assumptions

Based off past results, C5.0 and Rpart will yield similar results, yet display different information.

Experimental Design

Rscript from previous report:

```
#Iris Rpart
IrisPred = IrisSet$Classification ~ IrisSet$SL+IrisSet$SW+IrisSet$PL+IrisSet$PW
IrisCART= rpart(IrisPred,method = 'class')
summary(IrisCART)
plot(IrisCART)
text(IrisCART)

#Wine C5.0
head(Wine)
WineTree = C5.0(Wine[1:13],Wine[,14])
summary(WineTree)
plot(WineTree)
WineRules = C5.0(Wine[1:13],Wine[,14], rules = T)
summary(WineRules)

#Wine Rpart
WinePred = (Wine$Class ~ Wine$Alcohol+Wine$Malic_acid
```

```
            +Wine$Ash+Wine$Alcalinity_of_ash+Wine$Magnesium
            +Wine$Total_phenols+Wine$Flavanoids+Wine$Nonflavanoid_phenols
            +Wine$Proanthocyanins+Wine$Color_intensity+Wine$Hue
            +Wine$OD280.OD315_of_diluted_wines+Wine$Proline)
WineCart = rpart(WinePred, method = 'class')
summary(WineCart)
plot(WineCart)
text(WineCart)
post(WineCart,title. = 'Rpart Wine Data',filename = 'WineRpart.ps')

#Titanic C5.0
head(TitanicSet)
TitanicTree = C5.0(TitanicSet[1:3], TitanicSet[,4])
summary(TitanicTree)
plot(TitanicTree)
TitanicRules = C5.0(TitanicSet[1:3], TitanicSet[,4], rules = T)
summary(TitanicRules)

#Titanic Rpart
TitanicPred1 = TitanicSet1$Survived ~
TitanicSet1$Class+TitanicSet1$Age+TitanicSet1$Sex
TitanicCart = rpart(TitanicPred1,method = 'class')
summary(TitanicCart)
plot(TitanicCart)
text(TitanicCart)
```

Rscript for SVM and Confusion matrices

```
SVMiris=svm(Species~ ., data=iris)
SVMiris
irispred=fitted(SVMiris)
table(irispred,iris$Species)
irisvalues=predict(SVMiris,iris[1:4],decision.values = T)
irisvalues
predict(SVMiris,iris[1:4])
summary(irisvalues)

read.csv('Titanic-1.csv')
TitanicSet=read.csv('Titanic-1.csv')
SVMTitanic=svm(Survived~ ., data = TitanicSet)
TitanicPred=fitted(SVMTitanic)
table(TitanicPred,TitanicSet$Survived)

WineSet=read.csv('Wine.csv')
SVMWine=svm(Class~ .,data = WineSet)
WinePred=fitted(SVMWine)
table(WinePred,WineSet$Class)

TitanticModel=rpart(Survived~ .,data = TitanicSet)
TitanicRpart=predict(TitanticModel,TitanicSet1[,-4], type = 'class')
table(pred = TitanicRpart,true = TitanicSet1$Survived)


IrisModel=rpart(Species~ .,data = iris)
IrisRpart=predict(IrisModel,iris[,-5], type = 'class')
table(pred = IrisRpart,true = iris$Species)

WineModel=rpart(Class~ .,data = WineSet)
WineRpart=predict(WineModel,WineSet[,-14], type = 'class')
table(pred = WineRpart,true = WineSet$Class)
```

Results

Iris SVM

```
> table(irispred,iris$Species)

irispred     setosa versicolor virginica
  setosa         50          0         0
```

```
  versicolor      0         48         2
  virginica       0          2        48
```

Titanic SVM

```
> table(TitanicPred,TitanicSet$Survived)
```

```
TitanicPred  No Yes
     No  1470 441
     Yes   20 270
```

Wine SVM

```
> table(WinePred,WineSet$Class)

WinePred  Class_1 Class_2 Class_3
  Class_1      47       0       0
  Class_2       0      61       0
  Class_3       0       0      45
```

Wine C5.0

```
            Rules
          ----------------
            No      Errors

             5    1( 0.7%)   <<


           (a)   (b)   (c)    <-classified as
          ----  ----  ----
            47                 (a): class Class_1
                  60     1     (b): class Class_2
                        45     (c): class Class_3
```

Iris C5.0

```
    Decision Tree
          ----------------
          Size      Errors

             4    4( 2.7%)   <<


           (a)   (b)   (c)    <-classified as
          ----  ----  ----
            50                 (a): class Setosa
                  47     3     (b): class Versicolor
                   1    49     (c): class Virginica
```

Titanic C5.0

```
  Decision Tree
          ----------------
          Size      Errors

             3  477(21.7%)   <<


           (a)   (b)    <-classified as
          ----  ----
          1470    20    (a): class No
           457   254    (b): class Yes
```

Titanic Rpart

```
> table(pred = TitanicRpart,true = TitanicSet1$Survived)
     true
pred    No  Yes
  No  1470  441
  Yes   20  270
```

Iris Rpart

```
> table(pred = IrisRpart,true = iris$Species)
            true
pred         setosa versicolor virginica
```

```
  setosa          50          0          0
  versicolor       0         49          5
  virginica        0          1         45
```

Wine Rpart

```
> table(pred = WineRpart,true = WineSet$Class)
         true
pred      Class_1 Class_2 Class_3
  Class_1      43       0       0
  Class_2       4      60       0
  Class_3       0       1      45
```

Based off the confusion matrices above, here are the percentage errors for each method:

SVM Iris: 4/150 (2.67%) Wine: 0/153 (0%) Titanic: 461/2201 (20.95%)

C5.0 Iris: 4/150 (2.67%) Wine: 1/153 (0.65%) Titanic: 477/2201 (21.67%)

Rpart: Iris: 6/150 (4%) Wine: 5/153 (3.27%) Titanic: 461/2201 (20.95%)

Uniquely, SVM is the only method to produce a result better than the other two, that is not to say that SVM is a universally better method. Despite have differing misclassifications with the Iris data set, the percentage error was the same, while SVM had the exact same “misclasses” that Rpart had.

However, Rpart’s final classifications are based on the greatest margin of classifications per rule, while C5.0 creates more rules if more divisions are possible even if it is just between a few instances, and SVM is a more graphic and binary division of all the instances.

To obtain a confusion matrix for Rpart a different computation must be used instead of the standard on that offers the rule “tree”. SVMs don’t have a set of “rules” that the others have, however, a line could be plotted to give a better idea of trends or the characteristics that divides the classified groups. When looking at the rulesets for C5.0 and Rpart, it could almost be argued that they use SVMs as part of their classification process for that each rule divides two groups around a certain value.

Summary

There are various methods to classify data. SVMs are becoming the more popular calculation to do what has been shown in this report. However, it does not mean that SVMs is the best because it scored similar results to C5.0 and Rpart as well as you may want some “misclasses” depending on the type of data. As impressive as SVMs result on Wine is, it could be argued that SVM just fits that data set better and may fall apart if new data was added, this has not be tested in this report.

C5.0 and Rpart do give detailed rules for their methods which can be helpful in certain cases, while SVMs don’t rely on rules. Rpart can also start at different points but will not change the results that much. However, Rpart does not initially give a confusion matrix, which will take addition set up to obtain if that is the desire.

Bibliography

Brieman, F. O. (2017, April 1). *Package 'rpart'*. Retrieved from rpart.pdf

C4.5 Algorithm. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/C4.5_algorithm

Classification and regression trees. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Predictive_analytics#Classification_and_regression_trees_.28CART.29

Decision tree learning. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Decision_tree_learning

ID3 Algorithm. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/ID3_algorithm

Meyer, D. (2017, February 2). *Package 'e1071'.* Retrieved from CRAN: https://cran.r-project.org/web/packages/e1071/e1071.pdf

Meyer, D. (2017, February 1). *Support Vector Machines.* Retrieved from CRAN: https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf

Parker, A. (2017). *Report 2.*

https://archive.ics.uci.edu/ml/datasets/wine

https://archive.ics.uci.edu/ml/datasets/Iris