DATA MINING AND MACHINE LEARNING

Daniel Perkins

Introduction:

The overall task of this report was to use WEKA as a platform for analyzing data using machine learning. Three different data sets were analyzed including the Iris, Diabetes, and Nursery sets from the University of California at Irvine. This process not only was to be familiar with the WEKA interface, but also learn the importance of data mining within the context of data analysis. Three major algorithms were used in order to assist in gathering information from the data set including, J48, JRip, and PART. Furthermore, while in the process of learning data mining and its important algorithms, useful information from the data set was analyzed in order to gain further knowledge about flower type prediction, diabetes risks, and nursery school placement, respectively.

Algorithms Used:

The three algorithms used throughout this report (J48, JRip, and PART) each have a distinct method for analyzing large data sets. Each has a distinct method of learning how to predict the value of data sets.

The J48 Algorithm is a type of Decision Tree that targets one dependent variable against various other attributes of data. JRip creates a decision tree during the training phase in order to predict values during the testing phase. The nodes of the tree are possible attributes of the data, while the branches linking the nodes represent possible values linking to other attributes. The Terminal nodes illustrate the final classification of the dependent variable, ultimately used in the WEKA analysis. In order to create the decision tree, the algorithm must first use training data. The algorithm denotes the variable that identifies the attribute most clearly. Next, the algorithm looks for the next attribute that yields the highest information gain, and creates these as the next branching questions of the decision tree. This is continued until the tree points to a concrete decision of what attributes aligns with what value, or until the algorithm runs out of testing data. Once the tree is created, the algorithm checks the attributes based on the model and selects a prediction value. The prediction is compared to the actual value in order to determine an accuracy percentage.[1]

Unlike J48, The RIPPER algorithm (or JRip) is a rule based algorithm. These types of algorithms use a set of commonly found associations to determine the prediction value. The algorithm was designed by Cohen in 1995. The RIPPER algorithm establishes a set of rules during the training phase to be analyzed against during the testing phase. JRip uses repeated incremental pruning as a form of establishing rules. The algorithm uses association error pruning by splitting the training data into a growing and pruning set. The initial rule set is formed using the growing set, then the rule is repetitively simplified using the pruning data. The major difference of this algorithm is that the RIPPER will re-implement previously learned rules in the context of subsequent rules.[2]

---

[1] http://www.d.umn.edu/~padhy005/Chapter5.html

[2] https://pdfs.semanticscholar.org/f67e/bb7b392f51076899f58c53bf57d5e71e36e9.pdf

The PART algorithm is another form of a rule based algorithm. PART stands for Projective Adaptive Resonance Theory. The PART algorithm uses a separate-and-conquer method to build a partial decision tree in each iteration and designates the most efficient leaf into a new rule.[3] Unlike other algorithms, the PART algorithm does not need to perform global optimization (a form of numerical analysis that optimizes a function based on a set of determined criteria).[4] The PART algorithm creates a decisions list with distinct rules during the testing phase in order to predict values during the testing phase. The PART algorithm combines the strategies of the J48 algorithm and the RIPPER algorithm, by establishing rules off a partial decision tree.

These three distinct algorithms have different strategies, and thus result in different accuracy levels when training on a given set of data. Thus, each algorithm is efficient within a particular context. Decision trees are easy to visualize and interpret, because they mirror human decision-making strategies. They perform well on large data sets and can handle both categorical and numerical data. However, the tree algorithms such as J48 tend to be less accurate than other algorithms and a small change in the training data can result in a large discrepancy in the tree. Trees can also be overly complex at times, which is why some algorithms prune trees. Rules based learning algorithms such as RIPPER are less intuitive for the human user, however they can be more accurate because they don't rely on small discrepancies of the training data.[5]

Program Structure:

The first process of the code is to determine which file the user wants to analyze. The user is prompted to analyze one of three data sets, and selects one through the console. The user is prompted to enter a number until it is a valid option. The code then askes the desired number of k folds based on user input. The scanner prints out instructions to the console and the scanner takes in the user input. If the user input a number less than two, then the user is prompted with an error until the number is greater than two. Once the user enters a valid number, the number is stored as a variable and is used later in the program to tell later methods how many folds to execute.

The program determines the prediction accuracy of the PART, JRip, and J48 algorithms. The first step of the program is to read the data file. The program uses a buffered reader in order to take in all the data from the arff file. Since the class attribute is the last attribute of the arff file, the program sets the class index to the index of the last attribute (see line 91). Next, the program executes the k fold training using the crossValidationSplit() method. In this method, the data is randomized. Then the data is split into training and testing, based on the user input on number of k folds. This process is completed for each of the three algorithms. After the program trains and establishes its respective rules or decision tree, the accuracy is calculated.

The accuracy of the algorithm is calculated using the calculateAccuracy() method. This method compares the predicted value for the testing data to the actual value. If the two values are the same, the method increases the number of correct guesses. The method then loops through all the testing data to determine the total correct out of total predictions. Then the method returns the

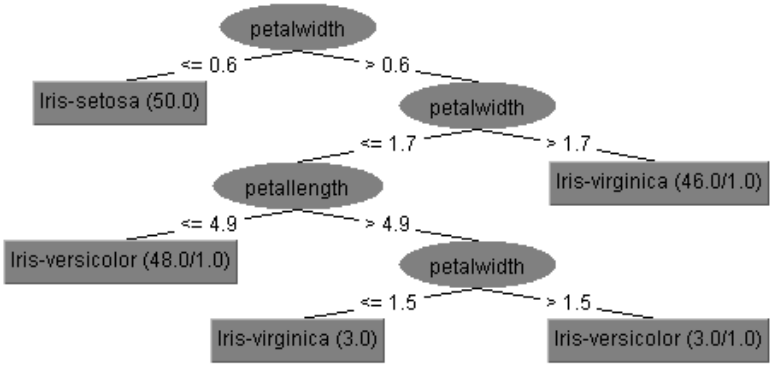
[3] http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/PART.html
[4] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.471.5770&rep=rep1&type=pdf
[5] https://en.wikipedia.org/wiki/Decision_tree_learning#Decision_tree_advantages

accuracy of the algorithm. This process is repeated for all three algorithms and their respective accuracies are printed to the console.

Data Set 1: Iris

The iris data set is often used for programmers learning to use machine learning, and thus it is one of the most commonly seen databases for pattern recognition. The data looks at the Iris plant, which is a flower found throughout America's meadows. The data set has 150 instances and is within the "botanical" domain. The data takes in four attributes. The first two include sepal length and width (the sepal is the green area that protects the flower bud. The data also contains bud length and width. The data is classified into three species: *Iris Setosa*, *Iris Versicolour*, and *Iris Virginica*.

For the J48 algorithm's analysis of the Iris data, one must analyze the decision tree that the algorithm uses. Using the Weka explorer, the major nodes of the decision tree used in determining the data was the petal length and the petal width of the flowers. These rules were determined by the algorithm to be the most effective in predicting the species of the Iris plant. The decision tree was extracted using the WEKA explorer capabilities.[6]The decision tree is shown below:



For the JRip algorithm, the rules generated by the algorithm is most helpful in understanding how the algorithm learned how to predict values within the training stage of the program. The algorithm created a total of three rules based on petal length and petal width in order to determine the species of the flower. A summary of the rules created is shown below.

```
JRIP rules:
===========

(petallength >= 3.3) and (petalwidth <= 1.6) and (petallength <= 4.9) => class=Iris-versicolor (46.0/0.0)
(petallength <= 1.9) => class=Iris-setosa (50.0/0.0)
 => class=Iris-virginica (54.0/4.0)

Number of Rules : 3
```

The PART algorithm is another rules based algorithm, however the algorithm uses a decision list to determine the values to predict during the testing and training stages. In order to best predict the species of the algorithm, PART established a decision list with three rules using the petal length and petal width attributes. The decision list is illustrated below. Although all

[6] https://www.youtube.com/watch?v=l7R9NHqvI0Y

three algorithms uses different strategies to predict the species of the plant, they use the same core attributes that are found to be most efficient.

```
PART decision list
------------------

petalwidth <= 0.6: Iris-setosa (50.0)

petalwidth <= 1.7 AND
petallength <= 4.9: Iris-versicolor (48.0/1.0)

: Iris-virginica (52.0/3.0)

Number of Rules  :      3
```

Data Set 2: Diabetes

The diabetes data comes from the National Institute of Diabetes and Digestive and Kidney Disease. The Diabetes data analyzes the lifestyle choices of certain individuals and predicts if they are prone to diabetes. The data set has 768 instances and is within the "health" domain. All members of the data are females, and thus some health questions pertain to this factor. The data has eight attributes: number of times pregnant, glucose concentration in a two-hour test, diastolic blood pressure, triceps skin fold thickness, 2-hour serum insulin, body mass index, diabetes pedigree function, and age. This data is used against whether or not the participant tested positive or negative for diabetes.[7]

For the J48 algorithm, the program exclusively uses the decision tree in order to predict its values. According to the WEKA explorer, the most effective rules for best predicting diabetes among patients were plasma glucose concentration, Body Mass Index, diabetes pedigree function, and age. Thus, the algorithms would test these attributes first when attempting to predict diabetes among certain patients. If the algorithm could not predict diabetes likelihood based on these attributes, the algorithm would continue down to less effective attributes. The visualization of the decision tree is much more complicated than the iris, due to the larger number of attributes. A small sampling of the decision tree is shown below. The data will follow this hierarchical pattern for all attributes determined to effectively predict diabetes within the patient.

[7] http://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/diabetes.arff

```
plas <= 127
|   mass <= 26.4: tested_negative (132.0/3.0)
|   mass > 26.4
|   |   age <= 28: tested_negative (180.0/22.0)
|   |   age > 28
|   |   |   plas <= 99: tested_negative (55.0/10.0)
|   |   |   plas > 99
|   |   |   |   pedi <= 0.561: tested_negative (84.0/34.0)
|   |   |   |   pedi > 0.561
|   |   |   |   |   preg <= 6
|   |   |   |   |   |   age <= 30: tested_positive (4.0)
|   |   |   |   |   |   age > 30
|   |   |   |   |   |   |   age <= 34: tested_negative (7.0/1.0)
|   |   |   |   |   |   |   age > 34
|   |   |   |   |   |   |   |   mass <= 33.1: tested_positive (6.0)
|   |   |   |   |   |   |   |   mass > 33.1: tested_negative (4.0/1.0)
|   |   |   |   |   preg > 6: tested_positive (13.0)
plas > 127
|   mass <= 29.9
```

JRip algorithm, uses a set of rules found most helpful in predicting diabetes among patients during the testing phase. The algorithm created a total of four rules based on plasma concentration, BMI, diabetes pedigree and insulin levels in order to determine the species of the flower. Although the Iris algorithms used the same attributes, the JRip algorithm has found a different set of attributes to me more effective in determining predictive values. This does not mean that one algorithm is wrong and one is right, but rather this reflects the different strategies of the algorithms in determining the prediction values. A summary of the rules created is shown below.

```
JRIP rules:
===========

(plas >= 132) and (mass >= 30) => class=tested_positive (182.0/48.0)
(age >= 29) and (insu >= 125) and (preg <= 3) => class=tested_positive (19.0/4.0)
(age >= 31) and (pedi >= 0.529) and (preg >= 8) and (mass >= 25.9) => class=tested_positive (22.0/5.0)
 => class=tested_negative (545.0/102.0)

Number of Rules : 4
```

The PART algorithm uses a decision list in order to predict diabetes among patients. Since the decision list has 13 rules, a consolidated version of the list is shown below. Difference in key attributes used even among rules based algorithms such as RIPPER and PART illustrate the differences of the algorithms. PART found plasma, mass and diabetes pedigree to be important in predicting diabetes, but the algorithm also established rules around age and triceps skin fold thickness (as seen in the rules below). This further illustrates the differences of the algorithms ability to analyze data.

```
PART decision list
------------------

plas <= 127 AND
mass <= 26.4 AND
preg <= 7: tested_negative (117.0/1.0)

plas > 154 AND
mass > 29.8: tested_positive (100.0/14.0)

plas <= 99 AND
age <= 25 AND
age <= 22: tested_negative (33.0)
```

Data Set 3: Soybean

The Soybean data predicts the disease of a soybean based on information about previous soybean diagnosis. The data has 683 instances and is under the "botanical" domain. The data has 35 attributes in order to diagnose the soybean disease. The data attributes include: data, plant stand, precipitation, temperature, hail, crop history, area of the plant damaged, severity of the damage, seed, germination, plant growth, multiple attributes analyzing the leaves (including leafspots, size, malformation, and size of abnormalities), stem, lodging, stem cankers, canker lesion, fruiting bodies, decay, mycelium, discolor, sclerotia, fruit pods and spots, mold growth, multiple attributes analyzing the seeds (normality, discoloring, and size), shriveling, and root health. These attributes then diagnose the seed and determine a plethora of probable diseases including Diaporthe stem canker, Rhizocontia root rot, and downy mildew, among others.[8]

This data set had the most attributes, and therefore has the largest decision tree. The algorithm used many different attributes in order to classify the disease of the soybean plant. Some of the major attributes that helped the algorithm predict the best disease were date, plant stand, precipitation, temperature, hail, crop history, area damaged, and severity. A section of the decision tree is shown below. The algorithm will follow the same process for all the attributes in order to predict the disease class of the soybean.

```
leafspot-size = lt-1/8
|   canker-lesion = dna
|   |   leafspots-marg = w-s-marg
|   |   |   seed-size = norm: bacterial-blight (21.0/1.0)
|   |   |   seed-size = lt-norm: bacterial-pustule (3.23/1.23)
|   |   leafspots-marg = no-w-s-marg: bacterial-pustule (17.91/0.91)
|   |   leafspots-marg = dna: bacterial-blight (0.0)
|   canker-lesion = brown: bacterial-blight (0.0)
|   canker-lesion = dk-brown-blk: phytophthora-rot (4.78/0.1)
|   canker-lesion = tan: purple-seed-stain (11.23/0.23)
leafspot-size = gt-1/8
|   roots = norm
|   |   mold-growth = absent
|   |   |   fruit-spots = absent
|   |   |   |   leaf-malf = absent
|   |   |   |   |   fruiting-bodies = absent
|   |   |   |   |   |   date = april: brown-spot (5.0)
|   |   |   |   |   |   date = may: brown-spot (24.0/1.0)
|   |   |   |   |   |   date = june
|   |   |   |   |   |   |   precip = lt-norm: phyllosticta-leaf-spot (4.0)
|   |   |   |   |   |   |   precip = norm: brown-spot (5.0/2.0)
|   |   |   |   |   |   |   precip = gt-norm: brown-spot (21.0)
|   |   |   |   |   |   date = july
|   |   |   |   |   |   |   precip = lt-norm: phyllosticta-leaf-spot (1.0)
|   |   |   |   |   |   |   precip = norm: phyllosticta-leaf-spot (2.0)
|   |   |   |   |   |   |   precip = gt-norm: frog-eye-leaf-spot (11.0/5.0)
```

The JRip algorithm established a total of 29 rules in order to predict soybean diseases. Since the rules list is quite exhausted, a snippet of the complete rules list is shown below. Similar to the J48 algorithm, the JRip algorithm uses a wide variety of attributes in order to predict disease. Unlike diabetes and Iris, there are much more attributes for the algorithms to choose from. Thus, the decision trees, rules, and decision lists for all three algorithms are much larger

[8] http://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/soybean.arff

and complex. Furthermore, the algorithms use a greater discrepancy of strategy in order to predict values.

```
JRIP rules:
===========

(leaf-malf = present) and (stem = abnorm) => class=herbicide-injury (8.0/0.0)
(fruit-pods = few-present) => class=cyst-nematode (14.0/0.0)
(shriveling = present) and (stem-cankers = absent) => class=diaporthe-pod-&-stem-blight (15.0/0.0)
(leaf-malf = present) and (leafspots-halo = absent) => class=2-4-d-injury (16.0/0.0)
(seed-discolor = present) and (canker-lesion = tan) => class=purple-seed-stain (20.0/0.0)
(leaf-malf = present) and (seed = norm) and (leafspot-size = gt-1/8) => class=phyllosticta-leaf-spot (10.0/0.0)
(precip = lt-norm) and (date = june) => class=phyllosticta-leaf-spot (4.0/0.0)
```

As seen before, the PART algorithm's decision list uses different attributes to predict its values. Despite being both rule based algorithms, the PART and RIPPER algorithms predicted attributes using distinct strategies. The RIPPER algorithm only created 29 rules, while the PART algorithm used 40 rules in its decision list. However, much like both previous algorithms tested, the PART algorithm continues to rely on a vast array of different attributes in order to best predict values. Since the decision list contains 40 rules, brief section of the decisions list is shown below, with the rest of the rules following a similar pattern.

```
PART decision list
------------------

leafspot-size = lt-1/8 AND
canker-lesion = dna AND
leafspots-marg = w-s-marg AND
seed-size = norm: bacterial-blight (21.0/1.0)

int-discolor = none AND
plant-growth = abnorm AND
leaves = abnorm AND
stem = abnorm AND
plant-stand = lt-normal AND
area-damaged = low-areas AND
fruiting-bodies = absent: phytophthora-rot (81.29/0.76)

leafspot-size = lt-1/8 AND
canker-lesion = dna: bacterial-pustule (20.31/1.31)
```

Results:

Average Algorithm Accuracy (J48):

		Number of Folds ("k")				
Data Sets	% Accuracy	**3**	**5**	**10**	**20**	**50**
Iris		94.67%	93.33%	94.00%	95.33%	95.33%
Diabetes		73.83%	73.83%	74.35%	74.35%	75.39%
Soybean		91.51%	90.04%	91.95%	92.83%	92.39%

Table 1: *Average Percentage Accuracies of the J48 algorithm after conducting a number of predictive learning experiments on different data sets.*

Average Algorithm Accuracy (PART):

		Number of Folds ("k")				
Data Sets	% Accuracy	**3**	**5**	**10**	**20**	**50**
Iris		94.00%	93.33%	93.33%	94.67%	94.67%
Diabetes		73.31%	74.09%	72.79%	71.09%	71.74%
Soybean		91.65%	90.48%	91.22%	92.95%	92.09%

Table 2: *Average Percentage Accuracies of the PART algorithm after conducting a number of predictive learning experiments on different data sets.*

Average Algorithm Accuracy (JRip):

		Number of Folds ("k")				
Data Sets	% Accuracy	**3**	**5**	**10**	**20**	**50**
Iris		91.33%	93.33%	96.00%	95.33%	94.00%
Diabetes		73.70%	73.96%	76.56%	75.26%	74.87%
Soybean		90.63%	92.83%	92.53%	92.24%	92.53%

Table 3: *Average Percentage Accuracies across RIPPER algorithm after conducting a number of predictive learning experiments on different data sets.*

Conclusion:

Overall, the three algorithms use very different strategies and data analysis in order to learn the patterns of a given dataset. While this may not be extremely obvious when analyzing smaller datasets with less attributes such as iris, the differences become very prevalent in large data sets with multiple calssification possibilities such as soybean. In the Iris dataset, the altorithms used exactly attributes in their decision trees and rules in order to predict the species. However, in datasets with more attributes, such as diabetes and soybean, the algorithms began to split off and rely on different attibutes in order to predict values.

Furthermore, each algorithm has a unique response to the number of folds. J48 tends to increase in accuracy with an increase in the number of folds. PART also generally follows this trend throughout all three data sets. However, the JRip algorithm tends achieve a peak performance around 10 folds. The number of folds not only influences the accuracy of prediction, but also the run time. This is logical, because with an increased number of folds, the algorithms have more tasks to execute.

In addition, the data set size influenced the run time of the program. When there were more instances of data, the algorithms took longer process its decision trees and rules during the training phase. This is logical because with more data instances, the algorithms must execute more training and testing. However, there was no correlation between data size and accuracy. This is true because the accuracy between data sets is a nature of the data itself, rather than the sheer size of the data. For example, the algorithms were less accurate in predicting diabetes among patients because there are many other factors in play for the development of diabetes in people, such as genetics, which are not accounted for in the data (lurking variables). However, for simpler data such as the iris plant, there is greater accuracy because each species has distinct

petal and sepal sizes. These changes are very influential in determining species, and thus the data as a whole was more accurate than diabetes, regardless of folds.

There are outliers in each data set, thus one of the reasons why increased instances of data could influence the predictability of the data would be because outliers are drowned out in larger sets of data. The outliers would have less influence on the algorithms when they set up their rules or decision tree. However, the overall nature of the data itself has more power over its predictability, thus this overshadows the correlation between accuracy and data size. However, if one analyzed a larger number of data sets, there would most likely be a correlation between data size and accuracy. Thus, further tests could be done on more large data sets in order to have a greater understanding between data size and accuracy of the algorithms.

Machine learning has even more useful applications beyond these three data sets and can be applied to many different domains. Thus, understanding how programs are able to predict data throughout machine learning is an increasingly useful skill that has real world application across multiple career paths.