The C4.5 Project:

An Overview of the Algorithm with Results of Experimentation

Toby Seidler

MATH 4500: Foundations of Data Mining

Professor Aleshunas

December 15, 2004

The C4.5 Project:

An Overview of the Algorithm with Results of Experimentation

Executive Summary

The problem is to determine a model that will classify a wine according to certain physical and chemical properties. This time, a new method is used to determine what attributes will best classify a given wine. Results from the new method will then be compared to results from methods previously employed.

This new method called C4.5, classifies 61% of the wine samples using two properties of the wine called proline and flavanoids. The remaining 39% is classified via two other attributes, one attribute for each of two distinct ranges left unclassified by proline and flavanoid value.

C4.5 yields significantly different results than the previously used methods, descriptive statistics and ID3. While the first two methods classified all the samples with only three attributes of the wine, C4.5 uses four. Additionally, C4.5 is less accurate at classifying the wine samples than the other two methods. Descriptive statistics and ID3 were both predicted to be 97% accurate, and in this instance C4.5 has a predicted accuracy of 93.8% and an actual accuracy of 86.6%.

It is recommended that one should not solely rely upon C4.5 as a way to determine a classification model. Although the program found an alternative method of classifying the wine, it proved to be less useful since it performed less accurately and required more chemical measurements. It is recommended that one should use descriptive statistics first to determine a model, and then use C4.5 to check if another, more useful classification procedure exists.

Problem Description

The problem is to determine a model that will classify a wine into one of three classes according to certain physical and chemical properties. Previously, two different methods known as descriptive statistics and ID3 were employed. This time, a program called C4.5 is used to classify the wine. This report explains C4.5 as an improvement on the ID3 algorithm and then compares the classification results from descriptive statistics and ID3 to the results from C4.5.

The wine dataset is a set of 153 instances of wine samples, each containing fourteen numeric attributes including the class to which the instance belongs.; there are three different classes of wine in this set. The commonly used iris dataset contains four numeric attributes and a class indicator for 156 instances of flowers; there are three different classes of flowers. The golf dataset describes fourteen instances of days in terms of weather; it is a small categorical set with two classes and will briefly be mentioned later.

Analysis Technique

Overview

C4.5 is an improvement on an existing classification algorithm known as ID3. ID3 is a decision tree algorithm that decides which attribute of a given instance will best classify the instances in the dataset. It also determines which values of these ranges yield the best classifying results. ID3 uses a measure from information theory called gain (or entropy) to decide how to best classify the data. The classification model is represented as a graph structure known as a tree. Each node of the tree contains a classifying attribute and each branch leaving the node represents a range of values for the attribute assigned to that node. After building the tree from a training dataset, the algorithm receives new data and classifies it. C4.5 improves on the ID3 algorithm in a few important ways.

Firstly, ID3 tends to favor trees containing many leaf nodes since a maximum gain value would be reached if each instance of the training data ended up in its own leaf node (Quinlan, 1993). Ideally, a tree will classify new instances well if there are several instances of data at each leaf node. So in essence, ID3 works well at classifying the data in the training set, but seems to be somewhat useless at classifying new instances of data. Mitchell (1997) shows that for an ID3 generated tree, as the number of nodes grows, the accuracy increases on the training data, but decreases on unseen test cases; this is called overfitting the data. Since the gain measure favors leaf nodes with single instances, a new measure is introduced in the C4.5 algorithm. Gain ratio incorporates the normal gain measure and adjusts this value by taking into consideration how many instances will result in each child node if the given split was executed (Quinlan, 1993). So C4.5 looks for data partitioning splits that will yield more useful results.

In regards to values that are missing for a given attribute of an instance, Quinlan (1993) poses three questions:

1) How does the algorithm select an attribute that is useful for a split if a value for that attribute is missing for a given instance?
2) In which partition of the node will the algorithm place training cases containing missing data?
3) How does the algorithm classify unseen cases that have missing attributes?

ID3 does not address these concerns, but C4.5 does. For the first issue, the gain ratio can be adjusted by multiplying it by a number F which is a ratio of the number of instances with a given attribute that has a known data value over the total number of instances containing that attribute. Thus attributes with existing values are given precedence over attributes with missing values. Next, the problem of training data with missing attributes is addressed by C4.5 by placing them in *both* partitions, using probabilistic weights to determine how likely it is that the instance would fall into either partition of the data. Non-training cases with missing attributes are also placed in multiple leaf nodes and assigned a probability that they fall into each classification.

ID3 tends to produce complex trees that often overfit the data. C4.5 allows for simplifying a complex initial tree and replacing it with a simpler one, making the final tree more accurate on data outside of the training set. This process is known as pruning and there are two ways that C4.5 does this: ". . . subtree raising, replaces a subtree by its most used subtree. . ." and "subtree replacement" replaces a subtree with a leaf node that is frequently reached by instances arriving at that given subtree (Dunham, 2002). C4.5 uses a pessimistic error calculation to predict how accurate the pruned tree will be. If the pruned tree is predicted to be more accurate on unseen cases than the current tree, then the tree is updated. This process begins at the lower subtrees, and continues upward until the tree is considered as a whole.

Another technique included in C4.5--but not ID3--is a technique called windowing. Windowing randomly selects a subset of the training data (called a window) and generates a small tree from that sample. The rest of the test data is then classified by this tree, and some test cases will be misclassified. Next, the subset is enlarged to include every instance that was initially misclassified. A replacement tree is built from the enlarged subset and the process repeats until every instance in the training set is correctly classified by the current tree. This windowing technique generally results in trees more accurate than those produced by the normal process because it captures all the "'interesting' [instances], together with sufficient 'ordinary' cases. . ." (Quinlan, 1993). Another windowing technique creates multiple trees by capturing several random windows and generating a tree from each of them; the most accurate tree is retained.

C4.5 also offers the option of generating rules from a tree that has been created. To start out, a rule is generated for each path in the tree. The set of rules is then reduced using the same pessimistic error prediction that simplifies complex trees. A default class is also specified for each instance of data that does not fit any of the final rules.

Classification Formulae

There are no explicit formulae for classifying new instances of data. The algorithm brings an instance of data down the appropriate path to one (or several) leaf nodes. As each instance with missing attribute values in taken into consideration, it is assigned a weight which is the probability of the instance being in that partition of data (based on known cases). But weighted instances may have to be probabilistically classified again. So an instance from the training set with an unknown outcome at a certain decision node is given weight

$w*P_{Oi}$

where P_{Oi} is the sum of the weights of cases in the training set known to have outcome O_i, divided by the sum of the weights of instances with known outcomes at this decision node (Quinlan, 1993). This same formula is used when classifying non-training cases.

Training Formulae

The following formulae are used when building the tree from the data set:

For a dataset S, that has c different classifications,

$\text{Entropy}(S)$ □ $\sum -p_i * \log_2 p_i$ for i=1 to i=c

where p_i is the ratio of instances from S belonging to class i. For an attribute A from a set S,

$\text{Gain}(S, A)$ □ $\text{Entropy}(S) - \sum (|S_v| / |S|) * \text{Entropy}(S_v)$

where S_v is a subset of S; the second term listed above represents the entropy of the set after using attribute A to classify it.

$$\text{SplitInformation}(S,A) \;\square\; -\sum (|S_i| / |S|) * \log_2(|S_i| / |S|)$$

from i=1 to i=c, where S_i is the subset resulting from splitting the dataset using attribute A.

$$\text{GainRatio}(S,A) \;\square\; \text{Gain}(S,A) / \text{SplitInformation}(S,A).$$

When predicting error rates while pruning the tree, if N is the number of training cases and E are misclassified, CF is the confidence level, and $U_{CF}(E,N) = CF/2$,

$$N*U_{CF}(E,N)$$

gives the predicted error rate of a leaf and the sum of leaf node errors gives the predicted error for a respective subtree.

Presentation of results

The C4.5 algorithm prints the decision tree to the screen of the computer or sends the output to a file which can be opened as a text file. However, the tree does not appear in the conventional graph form, and there are some numbers that need some explanation.

For leaf nodes in the original decision tree, numbers in parentheses following the outcome are listed in the form (N/E), where N is the number of instances that have reached that node and E (if listed) indicates the number of cases misclassified at that leaf node. The numbers listed after pruned tree leaf nodes are listed in the same format but E has a different meaning, namely the predicted number of unseen cases out of N instances that will be misclassified by the given leaf node. Figure 4 in the appendix shows a sample output.

If this output is translated to a conventional tree structure, it will appear as shown in Figures 1, 2, or 3 in the results section. Rules produced by C4.5 are presented in if-then format and listed vertically. The number in brackets following the outcome indicates the predicted accuracy of the respective rule.

Miscellaneous

C4.5 works the same way as ID3 with regards to when the building process stops. As the training dataset is divided into smaller subsets, the algorithm stops once a given node contains instances that belong to only one class. There is no need to further divide such a partition.

Quinlan (1993) states the following regarding the windowing technique, ". . .it rarely leads to a less accurate final tree, it sometimes gives a better final tree, but it usually takes longer. Constructing multiple trees *always* takes longer, but occasionally can be worth the effort. . ."

The C4.5 program comes with a supplementary program called Consulter that will use an existing decision tree to classify new instances of data in real time. The user is prompted to enter values for the various attributes, and will then be told what class the hypothetical instance falls into. This can be helpful to users wanting to classify only a few instances without updating the test set.

Assumptions

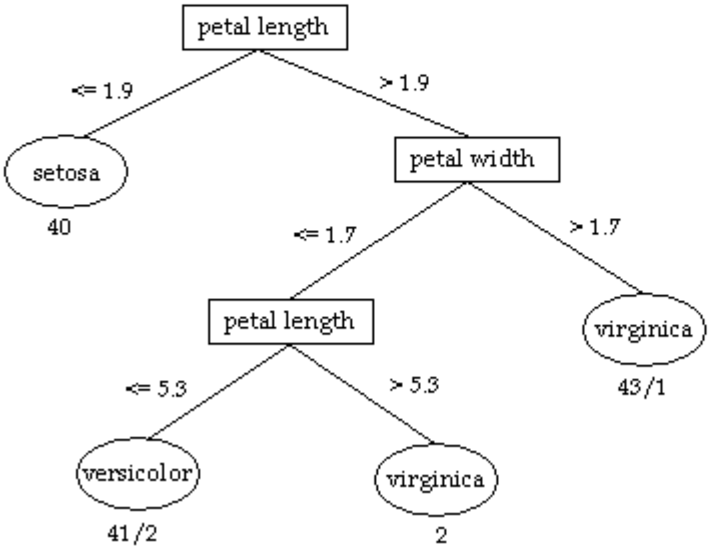
This program assumes that enough data exists for a training set to be created. Furthermore, C4.5 assumes that the data given in the training set will be representative of future data that will be classified. If this is not that case, the decision tree built from that training data will not be very effective. The algorithm also assumes that the data contained has instances with some existing value in most of the attribute fields. Although the algorithm does allow for data with missing attribute values, a tree generated from instances with many missing values may not be helpful in classifying new instances of data. The tree produced would yield merely the probability of a certain instance's classification. The algorithm also assumes that the data is in the proper format (comma delimited) and it assumes that a file exists that gives the names of the classifications, the names of the attributes, and a list of all possible values for attributes. Data can be categorical (golf dataset) or numerical (wine and iris datasets).

Results

C4.5 does a great job of classifying instances in the golf dataset. This is means that users working with categorical data need not convert it to numerical data before analysis.

For the iris dataset, the C4.5 results agree with previously established conclusions. Namely, that the iris-setosa class is easily distinguishable from the other two classes. A graph version of the decision tree created by the program is shown below in Figure 1. Numbers below the leaf nodes correspond to the numbers N and E which have been explained previously in the analysis section.

Figure 1: decision tree produced by C4.5 on iris dataset



The results obtained by using C4.5 on the wine dataset are different than any previous results. The author has listed possible reasons for this in the issues section of this paper. The descriptive statistics and ID3 methods showed flavanoids, hue, and proline to be useful attributes for classifying the wine. Table 1 summarizes previously established results. Correlation numbers were obtained using Microsoft Excel and gain numbers were calculated manually by the author for the attributes with the highest correlation values.

Table 1: gain and correlation numbers

	flavanoid	OD of wines	total phenols	Hue	color intensity	proline	alcohol
ID3 gain	0.184	0.174	0.134	0.238	0.238	0.236	0.181
DS correlation	-0.83	-0.79	-0.70	-0.83	0.80	-0.84	-0.79
	total data set			flavanoid range 1.0 – 1.6		flavanoid range 2.2 - 3.2	

C4.5 finds color intensity, flavanoids, proline, and alcohol to be useful attributes. Both ID3 and C4.5 trees are shown below in Figures 2 and 3. Note that the ID3 tree contains more leaf nodes than the C4.5 tree, but the C4.5 tree contains more decision nodes and more total nodes than the ID3 tree.

Figure 2: decision tree produced by ID3 on wine dataset

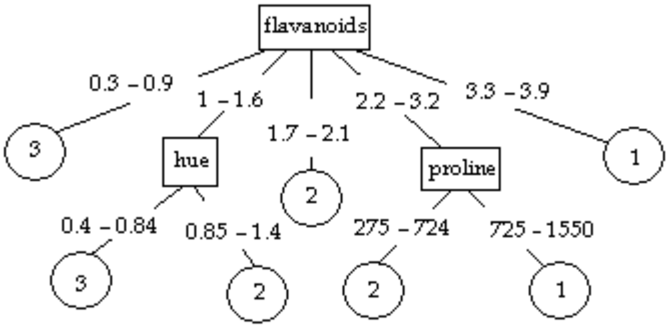
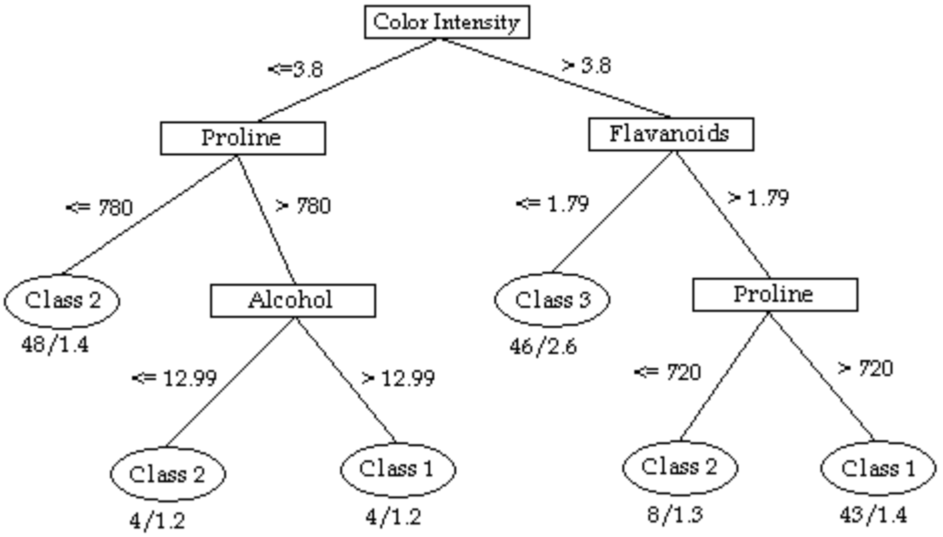


Figure 3: decision tree produced by C4.5 on wine dataset



The accuracy of the descriptive statistics and ID3 tree is higher than the accuracy of the C4.5 tree, 97% and 87% respectively. However, C4.5 does have some advantages. In comparison to descriptive statistics, it is much quicker and may produce a better alternative model. C4.5 also compensates for missing data and prunes the trees it creates, while ID3 does not. In spite of the benefits, C4.5 is recommended to be used only as an alternative means of classifying data, and is not recommended to be solely relied upon for data mining and/or classification. The author would not feel comfortable using C4.5 on a dataset without first using descriptive statistics on it.

Issues

There are some drawbacks to using the C4.5 program to classify data. C4.5 is not 100% accurate. On the wine test dataset of thirty and a default confidence level, C4.5 is 86.7% accurate in classifying test cases while displaying a predicted accuracy of about 94%. But the confidence level parameter can be altered so that the predicted accuracy more closely matches the actual accuracy. The user can then know how accurate the program will be in classifying new instances, although that accuracy may not be as high as desired or necessary.

There is also an issue when working with smaller datasets (approximately 200 or less). Suppose the user wants to set aside a portion of the training dataset to use as a test dataset. The tree produced from the resulting smaller training set will be different than the tree produced from the original full dataset. The author has found this to be true for the wine dataset, although the two trees were not exceptionally different.

When comparing the results from descriptive statistics and ID3, it is important to note that outliers were removed. This may be one factor in why the previous results are so different than C4.5 results. Other possible factors include flawed reasoning in the descriptive statistics method and/or incomplete manual ID3 calculations, the latter of which is currently under more suspicion.

Appendix

Figure 4: sample C4.5 decision tree output

```
C4.5 [release 5] decision tree generator  Sat Nov 27 13:02:17 2004
----------------------------------------

    Options:
      File stem <wine>
      Trees evaluated on unseen cases

Read 123 cases (13 attributes) from wine.data

Decision Tree:

Proline > 985 : Class 1 (30.0)
Proline <= 985 :
|   Flavanoids <= 1.39 :
|   |   Color Intensity <= 3.7 : Class 2 (8.0)
|   |   Color Intensity > 3.7 : Class 3 (35.0)
|   Flavanoids > 1.39 :
|   |   Proline <= 720 : Class 2 (41.0)
|   |   Proline > 720 :
|   |   |   Proline <= 920 : Class 1 (7.0)
|   |   |   Proline > 920 : Class 2 (2.0)


Tree saved


Evaluation on training data (123 items):

       Before Pruning           After Pruning
      ----------------   ---------------------------
      Size      Errors   Size      Errors   Estimate

        11    0( 0.0%)     11    0( 0.0%)    ( 6.2%)   <<

Evaluation on test data (30 items):

       Before Pruning           After Pruning
      ----------------   ---------------------------
      Size      Errors   Size      Errors   Estimate

        11    4(13.3%)     11    4(13.3%)    ( 6.2%)   <<


        (a)  (b)  (c)   <-classified as
       ---- ---- ----
          9    1        (a): class Class 1
          2    8        (b): class Class 2
               1    9   (c): class Class 3
```

References

Dunham, M. D. (2002). Data mining: introductory and advances topics. Upper Saddle River, NJ: Prentice Hall.

Mitchell, T. M. (1997). Machine Learning. New York: WCB/McGraw-Hill.

Quilan, J. R. (1993). C4.5: programs for machine learning. San Mateo, CA: Morgan Kaufmann Publishers.