This article is downloaded from

http://researchoutput.csu.edu.au

It is the paper published as

**Author:** K. Blackmore and T. Bossomaier  
**Title:** Comparison of See5 and J48.PART algorithms for missing persons profiling  
**Conference Name:** 1st International Conference on Information Technology and Applications, ICITA2002  
**Conference Location:** Australia  
**Publisher:** ICITA  
**Year:** 2002  
**Pages:** 1-6p  
**Date:** 25-28 November 2002

**Abstract:** Algorithms to derive rules from data sets can obtain differing results from the same data set. The J48.PART and the See5 schemes use similar methodologies to derive rules, however, differences appear in the number and constitution of rules produced to predict outcomes for missing persons cases. See5 generates fewer rules to obtain the same level of accuracy as J48.PART. Analysis of the input-output space using a measure of concept variation indicates missing persons profiling is characteristic of a difficult classification problem, resulting in fragmentation problems. This provides explanation for the differences that occur in the number and constitution of rules.

**Author Address:** Karen.Blackmore@newcastle.edu.au

**URL:**  
http://researchoutput.csu.edu.au/R/-?func=dbin-jump-full&amp;object_id=6577&amp;local_base=GEN01-CSU01  

**CRO identification number:** 6577
Comparison of See5 and J48.PART Algorithms for Missing Persons Profiling

K. Blackmore, Member, IEEE and T.R.J. Bossomaier

Abstract—Algorithms to derive rules from data sets can obtain differing results from the same data set. The J48.PART and the See5 schemes use similar methodologies to derive rules, however, differences appear in the number and constitution of rules produced to predict outcomes for missing persons cases. See5 generates fewer rules to obtain the same level of accuracy as J48.PART. Analysis of the input-output space using a measure of concept variation indicates missing persons profiling is characteristic of a difficult classification problem, resulting in fragmentation problems. This provides explanation for the differences that occur in the number and constitution of rules.

Index Terms—data mining, decision trees, rule based classifiers, fragmentation problem, concept variation.

I. INTRODUCTION

Mining data to discover “knowledge” can often involve the use of a variety algorithms in an effort to achieve the greatest accuracy over the dataset. Algorithms designed to learn rules from datasets have been extensively researched and compared [5,14]. C5.0 and its Windows (Microsoft) based counterpart See5, the commercial successors of the popular C4.5 decision tree generator [12], have been extended to produce rules from decision trees in an easily understandable “if…then” format. Weka, a public domain Java implementation of common machine learning schemes, includes a modified implementation of the C4.5rules algorithm, J48.PART.

Previous research has highlighted inconsistencies that may appear when using these two algorithms to classify and predict missing persons outcomes [1]. Differences appear in the constitution of and number of overall rules generated by each scheme. Whilst similar accuracy measures may be obtained from the two algorithms, inconsistencies in the derived rules require investigation. This is an important issue given the large array of “off the shelf” data mining software applications available. Products using different algorithms may produce a variety of rule sets with similar accuracy measures however analysis is required to ascertain which of the rules generated most accurately predict patterns in the dataset.

K. Blackmore is with the Faculty of Commerce, Charles Sturt University, Bathurst, NSW 2795 Australia (telephone: 61 2 6338-4083, e-mail: kblackmore@csu.edu.au).

T.R.J. Bossoineraier is with the School of Information Technology, Charles Sturt University, Bathurst, NSW 2975 Australia (e-mail: tbossoineraier@csu.edu.au).

II. BACKGROUND

The J48.PART algorithm combines two common data mining strategies; the divide-and-conquer strategy for decision tree learning with the separate-and-conquer strategy for rule learning. The divide-and-conquer approach [12] selects an attribute to place at the root node and “divides” the tree by making branches for each possible value of the attribute. The process then continues recursively for each branch, “using the only those instances that reach the branch.” [19]. The separate-and-conquer strategy is employed to build rules. A rule is derived from the branch of the decision tree explaining the most cases in the dataset, instances covered by the rule are removed, and the algorithm continues creating rules recursively for the remaining instances until none are left.

The J48.PART implementation differs from standard approaches in that a pruned decision tree is built for the current set of instances, the leaf with the largest coverage is made into a rule and the tree is discarded. By building and discarding decision trees to create a rule rather that building a tree incrementally by adding conjunctions one at a time avoids a tendency to over prune that is characteristic problem of the basic separate and conquer rule learner.

Quinlan’s [12] See5 algorithm derives rules by building a complete decision tree for the data set then transforming the unpruned decision tree into rules by generating one rule for each path from the decision tree root to a leaf. The rules are then pruned by a criterion based on the minimum description length principle [3] and deleted from the whole rule set if to do so reduces the rule set’s error on the training data [5].

Although based on the same algorithm for decision tree generation, the J48.PART implementation offers a more simplistic method for deriving rules [5] by creating only partial decision trees and extracting rules one at a time, rather than employing an optimization process to adjust or discard rules derived from a fully created tree. The J48.PART implementation generates a single set of rules from a tenfold cross-validation whereas See5 generates a set of rules for each fold in a tenfold cross-validation. Whilst these algorithms are similar in methodology, different results can be obtained when applying them to a dataset to discover rules [1,5].
III. DATA COLLECTION AND PREPARATION

The data for analysis was provided by Foy [4] as part of a larger study. The data set contained 19 attributes for 327 missing persons cases derived from the notes for cases where the person had not been quickly located. Attributes included individual characteristics such as age, gender and marital status, and other attributes related to the nature or the disappearance (i.e., time of day of disappearance and where last seen).

Analysis carried out by Foy [4] identified all attributes as having some correlation to the outcome attribute and therefore provide support for inclusion in the dataset for rule generation. Prior research by Henderson, Henderson and Kiernan [7] and Newiss [9] support these results.

The dataset was supplied in a standard spreadsheet format. For classification using J48.PART, the data was converted to ARFF format [19], which provides the attribute definitions and the data instances without specifying the attribute for prediction. Selection of the prediction attribute and filtering of any unwanted attributes is carried out within the Weka scheme. For classification using See5, the spreadsheet was converted to a names file, specifying both the dataset attribute definitions and the attribute for prediction. A separate .data file contains the data instances.

Missing values were present in the dataset, however both algorithms, being tree-based methods, satisfactorily handle missing values without the need for any preprocessing [13]. Five cases included in the original dataset had a missing value in the outcome attribute. Where the outcome attribute value was missing, both algorithms automatically omitted these cases from training sets. Thus, future references in this paper to the number of cases in the dataset will be based on the 322 valid cases used by the classifiers.

IV. METHODS

Evaluation of an algorithm’s predictive ability is best carried out by testing on data not used to derive rules [19] thus comparison of the classifiers was carried out using tenfold cross-validation. Cross-validation is a standard method for obtaining an error rate of the learning scheme on the data [19]. Using the algorithms’ cross-validation option for this dataset, testing occurs on a maximum of 33 cases in each fold, with each case in the dataset used just once as a test case, however, over all folds, all cases in the data are used to train the classifier. We then repeated the tenfold cross-validation to assess the stability of the partitioning.

Tenfold cross-validation splits the data into a number of blocks equal to the chosen number of folds (in this case 10). Each block contains approximately the same number of cases and the same distribution of classes. The algorithms perform an iterative process of constructing a classifier using the cases in 9 of the blocks and testing the accuracy of the derived classification on the cases in a holdout block.

Each case in the data is used just once as a test case and the error rate of the classifier produced from all the cases is estimated as the ratio of the total number of errors on the holdout cases to the total number of cases. Variations in results for each iteration in a cross-validation occur depending on the cases used in the training and holdout folds. Overall error of the classification is then reported as an average of the error obtained during the test phase of each fold. Both the J48.PART and See5 implementations offer tenfold cross-validation as a standard method for error measurement.

Running different single tenfold cross-validation experiments often produce different results because of the effect of random variation in choosing the folds [19]. In order to obtain an accurate error estimate, the tenfold cross-validation process was repeated ten times and the results averaged to obtain an overall error measurement. Repeating the tenfold cross-validation experiments ten times and averaging the results provides a more accurate error measurement than a single tenfold cross-validation [19].

A variety of classifier construction options exist for each algorithm. These options vary between the two algorithms so a subset of the options that were consistent between both were chosen and set to the default values and used for all analysis. The options included global pruning with a 25% confidence factor and a minimum of 2 cases. The confidence factor option affects the way that error rates are estimated and hence the severity of pruning; values smaller than the default (25%) cause more of the initial tree to be pruned, while larger values result in less pruning. The minimum cases option constrains the degree to which the initial tree can fit the data. At each branch point in the decision tree, the stated minimum number of training cases must follow at least two of the branches. This option is a form of pre-pruning; values higher than the default (2 cases) can lead to an initial tree that fits the training data only approximately. Increasing the confidence factor and the minimum number of cases in a branch increases the severity of the decision tree pruning process. Modifying these options, for example using a 10% confidence factor and a minimum of 10 cases, reduced both the number of rules and the predictive accuracy consistently with both algorithms.

V. RESULTS

A. Ten Tenfold Cross-Validation Experiment Results

A summary of the average error and number of derived rules for the ten tenfold cross-validation classifications for the
J48.PART and See5 algorithms is shown in Table 1. Only 2 rules appeared consistently in both of the summary rule sets produced by the algorithms. A copy of the full rule set is available from the authors, however, a tabulation of all rules produced by both algorithms is not included in this discussion due to size.

<table>
<thead>
<tr>
<th>Repition Number</th>
<th>J48.PART</th>
<th>See5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repitition</td>
<td>Average No. Rules</td>
<td>Error</td>
</tr>
<tr>
<td>1</td>
<td>23.8</td>
<td>28.0%</td>
</tr>
<tr>
<td>2</td>
<td>20.6</td>
<td>26.1%</td>
</tr>
<tr>
<td>3</td>
<td>22.1</td>
<td>27.4%</td>
</tr>
<tr>
<td>4</td>
<td>23.2</td>
<td>30.7%</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>28.6%</td>
</tr>
<tr>
<td>6</td>
<td>22.3</td>
<td>27.7%</td>
</tr>
<tr>
<td>7</td>
<td>22.2</td>
<td>29.5%</td>
</tr>
<tr>
<td>8</td>
<td>22.5</td>
<td>27.7%</td>
</tr>
<tr>
<td>9</td>
<td>20.2</td>
<td>27.7%</td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td>28.3%</td>
</tr>
<tr>
<td>Total</td>
<td>22.29</td>
<td>28.2%</td>
</tr>
</tbody>
</table>

Table 1. Results of ten tenfold cross-validations

B. Students T-test

The rule-based classification schemes are able to predict outcomes for the dataset with an average predictive accuracy of 71.8% and 72.2% for the J48.PART and See5 algorithms respectively. A more significant variation occurred in the number of rules generated. An average 22 rules were generated by J48.PART compared to 19 by See5. A two-tailed Students T-test (assuming equal variance, alpha = 5%) was carried on both the number of rules and error level generated from the ten tenfold cross-validations (n=10) to test the hypothesis that the means in both cases are the same. Output from the statistical analysis is shown in Tables 2 and 3.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>22.29</td>
</tr>
<tr>
<td>Variance</td>
<td>1.49</td>
</tr>
<tr>
<td>Observations</td>
<td>10</td>
</tr>
</tbody>
</table>

Test Results

Pooled Variance 1.27
Hypothesized Mean Difference 0
Df 18
p(T<=t) one-tail 5.36
p(T<=t) one-tail 0.00
p(T<=t) one-tail 1.73
p(T<=t) two-tail 0.00
p(T<=t) two-tail 2.10

Table 2. Number of rules per fold - t-Test: Two-Sample Assuming Equal Variances

C. Confusion Matrix

A confusion matrix contains information about actual and predicted classifications done by a classification scheme and is commonly used for evaluating performance. For example, the first row of the confusion matrix in Table 4 shows 149(151) actual runaway cases correctly classified as runaway, 22(20) actual runaway cases incorrectly classified as suicide, and 18(17) runaway cases incorrectly classified as foul play for J48.PART and See5 respectively.

The individual confusion matrices from each tenfold cross-validation for the J48.PART and See5 schemes were combined and averaged to form a single matrix (Table 4). Values on the diagonal represent the average number of cases correctly classified by the J48.PART and See5 schemes respectively. Despite the variation in the number of rules derived by the schemes, analysis of the confusion matrices shows consistency in the classification.
The Weka implementation does not allow identification of the actual cases correctly and incorrectly classified thus a comparison between the two schemes of the individual suicide cases incorrectly classified as runaway was not possible. Future research may require modification of the \textit{J48.PART} source code to include actual case classifications to improve the comparison of methods.

VI. FURTHER TESTING

Experimentation on the data set within the Weka and \textit{See5} schemes using different classification options had varying effects on predictive accuracy. A slight improvement in accuracy (average 2\%) was achieved using boosting [6]. Increasing the confidence factor (as mentioned in Section IV) within \textit{See5} and \textit{J48.PART} to 50\% improved the classification accuracy on average by 2\%. Applying both boosting and a 50\% confidence factor increased the average accuracy by 4\%. Only the Weka implementation included an algorithm for bagging [2]. A 2\% increase in accuracy was achieved using a 25\% confidence factor. Generally, both algorithms achieved the same accuracy for the modified options, and a combination of boosting with an increased confidence factor resulted in the greatest increase in accuracy.

Given that at best a predictive accuracy of 76\% can be expected on the data set using the \textit{J48.PART} and \textit{See5} decision tree classifiers, some explanation of possible causes of error is warranted. One possible explanation for the classification error is the decision tree problem known as the “fragmentation problem”. Decision trees are a data-driven statistical clustering method with a principle limitation of this approach being the need for large amounts of training data representative of the classification problem. The fragmentation problem occurs because “the continuous partitioning of the training set at every tree node reduces the number of examples (i.e. the statistical support) at the lower-level nodes” [17], resulting in too many minor rules which can lead to a loss in accuracy [8]. This problem is further exacerbated by small training sets [15]. An effect of this problem is the replication of subtrees along the output tree [17], providing a plausible reason for the inconsistent constitution of rules produced by the \textit{J48.PART} and \textit{See5} schemes.

A measure known as class or concept variation $\downarrow$ has been used previously [11,16-18] to study the lack of smoothness of the output as a surface in the input space. The presence of high $\downarrow$ has been shown theoretically and empirically to: (a) indicate the presence of fragmentation problems, and (b) adversely affect predictive accuracy of \textit{C4.5rules} and \textit{C4.5trees} [17]. A full discussion of this measure is available in [18].

Research has shown that a measure of $\downarrow > 20\%$ adversely affects the predictive accuracy of decision tree classifiers [16-18]. In calculating the weighting factor within a shell radius $r$, examples close together ($r<2$) contribute higher to the amount of class variation than those far away ($r>3$). The value of $\alpha$ is user defined and modulates the effect of distance. For $\alpha=0$, no weighting applies and for $\alpha=1$ weighting is exponential. The following function was used as the weight factor:

$$W(r)=\frac{1}{2^{\alpha r}}$$

$\downarrow$ for the missing persons data set was calculated at 13\% ($\alpha=1$) and 46\% ($\alpha=0.5$). Analysis of the Euclidean distance matrix reveals a sparse decision surface with the majority of cases occurring at a distance between 3-5, with very few cases occurring close together in the decision space (Fig 1.).

![Image](Fig 1. Histogram of missing persons Euclidean distance matrix shows a unimodal distribution of cases clustered between 3-4.)

Given that the majority of instances in the data set are far apart, the user defined modulator of the effect of distance significantly impacts the calculation of $\downarrow$. Because the data is clustered at large distances it would appear more appropriate to use $\alpha=0.5$ for this data set.

VII. DISCUSSION

The \textit{J48.PART} and \textit{See5} schemes derive rules that predict outcome in missing persons cases with equal accuracy using tenfold cross-validation. The algorithms produce a significantly different number of rules and investigation of the rules generated by each of the algorithms shows major differences and discrepancies in the prediction of suicide cases.
and the individual attributes featured. The major components of the rule sets are the same, and it would appear that the decision trees are largely similar. Essentially, differences occur in the lower portion of the branches.

A possible explanation for the difference in the number of rules produced by the algorithms may lie in the way each algorithm derives rules from, and prunes, decision trees. The strategy employed by J48.PART uses sub-tree replacement rather than sub-tree raising. Witten and Frank [19] raise the conjecture that by building and discarding decision trees to create a rule, rather that building a tree incrementally by adding conjunctions one at a time, avoids a tendency to over prune that is characteristic problem of the basic separate and conquer rule learner. However, it would appear from this research that although similar accuracy is achieved, the sub-tree replacement strategy used by J48.PART results in more complex rule sets.

Results presented in this paper support the views of other researchers [10], [14] in that accuracy measurements often belie actual differences in classifier performance. Comparison of the number and constitution of rule sets derived, and the confusion matrices should form a crucial part of the analysis. Run time also represents another valid dimension for algorithm comparison, however this was not considered in this paper given the relatively small sample size.

Obtaining accurate and reliable predictive accuracy from rule based classifiers should be considered with suspicion, particularly when predictive accuracy is low at around 70%. This issue has been substantiated in this case, having taken two well-regarded, robust packages for deriving rule sets yet only managing to identify two rules in common to the rule sets derived by both algorithms. The similar accuracy measurements would indicate a common predictability level by these methods, however more subtle decision paths exist which lead to these accuracy levels. Additionally,

ACKNOWLEDGMENT

Collection of the data was carried out by Foy [4] under a grant from the Sir Maurice Byers Research Fellowship of the NSW Police Service.

REFERENCE LIST


