Identify Error-Sensitive Patterns by Decision Tree

William Wu

QCIS FEIT, University of Technology Sydney

Abstract. When errors are inevitable during data classification, finding a particular part of the classification model which may be more susceptible to error than others, when compared to finding an Achilles' heel of the model in a casual way, may help uncover specific error-sensitive value patterns and lead to additional error reduction measures. As an initial phase of the investigation, this study narrows the scope of problem by focusing on decision trees as a pilot model, develops a simple and effective tagging method to digitize individual nodes of a binary decision tree for node-level analysis, to link and track classification statistics for each node in a transparent way, to facilitate the identification and examination of the potentially "weakest" nodes and error-sensitive value patterns in decision trees, to assist cause analysis and enhancement development.

This digitization method is not an attempt to re-develop or transform the existing decision tree model, but rather, a pragmatic node ID formulation that crafts numeric values to reflect the tree structure and decision making paths, to expand post-classification analysis to detailed node-level. Initial experiments have shown successful results in locating potentially high-risk attribute and value patterns; this is an encouraging sign to believe this study worth further exploration.

1 Introduction

The ultimate goal of this study is to find the most problematic and error-sensitive part of a classification model, and this would require the collection, identification and comparison of classification statistics of its individual component parts. Decision trees have been selected as the pilot model for this study because it is a well-researched classification model with a simple structure, decisions on attributes and values are clearly displayed in a form of branches and nodes, as shown in Figure 1.

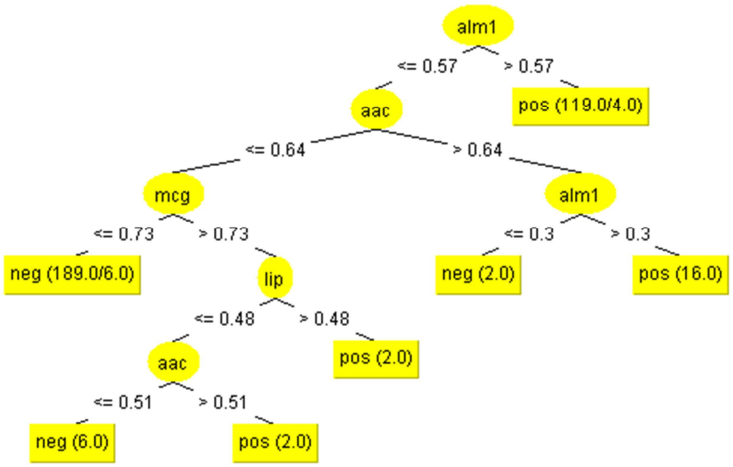


Figure 1 - A decision tree example

Using the first branch of the above decision tree as an example:

alm1 <= 0.57 and aac <= 0.64 and mcg <= 0.73: neg (189.0/6.0)

This branch contains three nodes with three split point values, (1) <= 0.57 for attribute "alm1", (2) <= 0.64 for attribute "aac", and (3) <= 0.73 for attribute "mcg". While all these three split points play a role in leading to the 6 classification errors amongst the 189 instances along this classification path in the form of a decision tree branch, one key question is, which node and its related attribute and value may have been more susceptible to the 6 errors? When expanding this node-specific examination to all branches of the decision tree model, the question can then generalized as – are there some tree nodes more error-prone and error-sensitive than others? If so, can the most error-sensitive nodes and their related attributes and values be identified in a systematic way? These two questions are now the focus of this study.

The rest of this paper is organized as follows. Section 2 describes some initial questions and thoughts that led to the decision tree digitization idea. Section 3 reviews some early influential work that inspired this study. Section 4 outlines the major steps in the decision tree digitization process. Section 5 summarizes experiments on five datasets. Section 6 discusses the experiment results and their implications. Finally, Section 7 concludes the current progress and outlines a plan for future exploration.

2 Initial issues and background

Decision trees provide an easy-to-follow graphical view of the classification process at a glance, outlines each classification rule from root to leaf step by step in the form of node by node. One issue with such visual representation is, when a large dataset is used and the decision tree structure becomes complex, its graphical view can

be clustered and muddled by the full-blown mass of crisscross branches and nodes even when pruning is applied. It can obscure the identification of attributes and values when detailed analysis is required on certain classification rules and components.

Another issue is, when node-level statistics are required in such a detailed analysis, the visual space reserved for each node on a decision tree may not be the most suitable place to present its node-level statistic values, as this would cluster the presentation of existing branches and nodes even further, making the node scanning and visual interpretation process even more difficult.

One possible solution to address these issues is, to provide a unique tag for each tree node, to collect and maintain the node-level classification statistics away from the tree structure, and to link them with their respective node by using the unique node tag as the retrieval key. As a result, classification statistics of each node can be stored and analyzed without any convoluted addition to the existing tree structure.

This decision tree digitization method and error-sensitive pattern analysis may seem trivial when compared to some major published work as outlined in Section 3, but because no such specific analysis has been observed so far, that nevertheless inspired this study and the digitization idea development.

3 Related work

Decision tree classification and attribute selection methodologies are two key research areas closely related to this study. Amongst the vast number of research literature and many innovative algorithms on decision trees, the C4.5 model [1] and CART model [2] are two benchmarks used as the foundation and guidance for the proposed decision tree digitization method. While the C4.5 model utilizes the gain ratio to "divide and conquer" data attributes and values to form a classification tree, the CART model makes use of the Gini impurity measure to split the attributes and data values to build a decision tree.

In the area of attribute selection, many of the renowned methodologies, such as Information Gain, Gain Ratio, Gini Index, RELIEF, SFS and SBE algorithm [1-6], as well as some newly established techniques such as SAGA [7] and UFSACO [8], they have been developed as a pre-process procedure or as an integrated part of the classification process. One key logic shared by these algorithms is to select and prioritize the most informative and differentiating data attributes before or during classification. While these methodologies have improved the classification performance in a holistic and "macro" way, they are not particularly designed to examine attribute and value patterns at an individual node and "micro" level.

Ongoing research and development have resulted new techniques in data sampling and classification process, such as bagging [9], boosting [10] and randomization to reduce bias [11], and provided the ground for individual tree models to be incorporated into an ensemble of tens or hundreds or even thousands of classifiers to achieve better performance. For example, the AdaBoost model [12] that adapts a weak learning algorithm such as a decision tree model as a starting point, then to reweight samples, retrain and rebuild a new tree after each intermediate learning cycle, and to vote

in the best performing tree from "the crowd". RandomForest [13] is another popular ensemble approach; it applies the bagging technique to a subset of attributes as well as training samples that are randomly selected, to generate new trees via iterations and to vote in the best performing tree amongst the peers in "the forest". The AdaTree method [14], the Probabilistic Boosting-Tree method [15], and a combined Bayesian model approach [16] are some new additions to the ensemble trend development.

Compared to standalone decision tree models, ensemble methods provide higher accuracy but at the cost of increased complexity and computational resources, the clarity and ease of result interpretation have also been reduced [17-18]. Amongst the above and other major literatures that have been studied, detailed analysis on error-sensitive attributes and values had not been apparent. One recent evaluation study on error-sensitive attributes (ESA) [19] intended to begin a detailed examination at an attribute level based on three specific terms using binary decision trees. The term "ambiguous value range" describes the "overlapped" value ranges between Positive and Negative instances; the term "attribute-error counter" describes the number of misclassified instances of an attribute with attribute values reside in ambiguous value ranges; and the term "error-sensitive attribute" describes an attribute that is considered to be more prone and risky to cause or associate with errors in a data classification process. All the above work has provided either the inspiration or the basis for the current study, and explained why decision trees have been adopted as the pilot model in this initial phase.

4 The decision tree digitization process

A decision tree model can be transformed into an array of branches and each branch consists of an array of nodes, and each node represents its underlying attribute and value's split point condition. Because each node can be considered as a child node from its immediate parent node, and all levels of parent nodes can be traced back to the root node as the origin, therefore, each node can be uniquely identified by a form of regression or inference process based on its hierarchical position within the tree and using the root as the starting point, and a graphical tree can subsequently be mapped into a matrix of referential and digitized node IDs, which can link and retrieve node level classification statistics for detailed analysis. The following pseudo code outlines the digitization process step-by-step:

Input: a binary decision tree with ***m*** branches and each branch contains a varying number of nodes, and a dataset with ***n*** instances

Output: an enumerated map of individual node IDs and a collection of node level classification statistics

Process: stage-1 is to enumerate each tree node and produce a map of node IDs; stage-2 is to collect classification statistics for individual nodes and using IDs as keys

Stage-1: construct an enumerated decision tree map
for 1 to ***m*** branches of the decision tree
  for all nodes in the current branch

1. if a node is the root node then assign "1" as the starting value of its node ID
2. if a node is an immediate child node from the root node, then first append a "." to the current node ID as a node delimiter, then add x to form 1.x as its 2nd part of the node ID, x denotes the current number of immediate child nodes branching out from the root and increments by 1 by counting from left to right, e.g. 1.1 as the 1st child node, 1.2 as the 2nd child node, and so on
3. if a non-root node has child nodes then first append a "." to the current node ID as a node delimiter, then assign 1 to its 1st child node on the left as its node ID, assign 2 to its 2nd child node on the right as its node ID; a node ID example is: 1.1.2.2.2.1

Stage-2: traverse and collect individual node level classification statistics
for 1 to ***n*** data instances
 for 1 to ***m*** branches in the enumerated matrix map
  for all nodes in the current branch

1. if current instance's attribute value satisfies current node's split point value condition, then continues to next node along current branch
2. if current node's split point condition cannot be satisfied, then advance to the start of next branch in the map
3. if end of current branch is reached and the leaf-node condition is satisfied, then update and store the node-level statistics using the node ID as the key for all nodes of the current branch, and move to the next data instance

On completion of the tree digitization and statistics collection, a simple ranking of the classification error rate by node IDs can then potentially reveal the "weakest" and most error-sensitive node in the tree. The word "potentially" has to be highlighted and emphasized here. Using a node's error rate value instead of its error count number may avoid the bias towards "heavy traffic" nodes; however, this may unduly magnify the "weakness" of some "low traffic" nodes. For example, node-A has been traversed by 10 instances with 5 errors and its error rate is 50%, node-B has been traversed by 100 instances with 48 errors and its error rate is 48%, while node-A is subsequently ranked as a "weaker" node than node-B by comparing error rate, this may not necessarily be true when more data are used for testing. In a later stage of the study, significant test and threshold value control on selection criteria can be implemented as an enhancement measures.

Nevertheless, this decision tree digitization method is another step forward in the study of error-sensitive value patterns in data classification, and results of initial experiments appeared to be supporting this idea.

5 Experiments

During the evaluation study of error-sensitive attributes (ESA) [19], five UCI datasets [20], Ecoli, PIMA Diabetes, Wisconsin Cancer, Liver Disorder and Page Blocks, were used in the evaluation process. These datasets have been used again in

the current study so their experiment results can be analyzed and compared side by side against the ESA evaluation results. Experiments have been conducted using WEKA's [21] C4.5/J48 decision tree classifier with standard configuration, e.g. confidence factor for pruning is 0.25, minimum number of instances per leaf is 2, MDL correction is used and test option is 10-fold cross-validation.

5.1 Digitization reflects decision trees in a concise and effective way

A decision tree model for the Ecoli dataset contains 7 branches and 13 nodes, as shown in Figure 1. On completion of its digitization, the digitized form of branches and nodes is shown in the 1st row of Table 1. Each node is uniquely tagged by a digital ID, and each ID reflects the node's hierarchical location in the tree. Because of its self-structured and self-referenced nature, the ID also encapsulates its preceding nodes of the same branch and presents itself as a compact and enumerated decision path; therefore, a collective display of each branch's leaf node resembles the decision tree model in a simplified and digitized form, as shown in the 2nd row of Table 1.

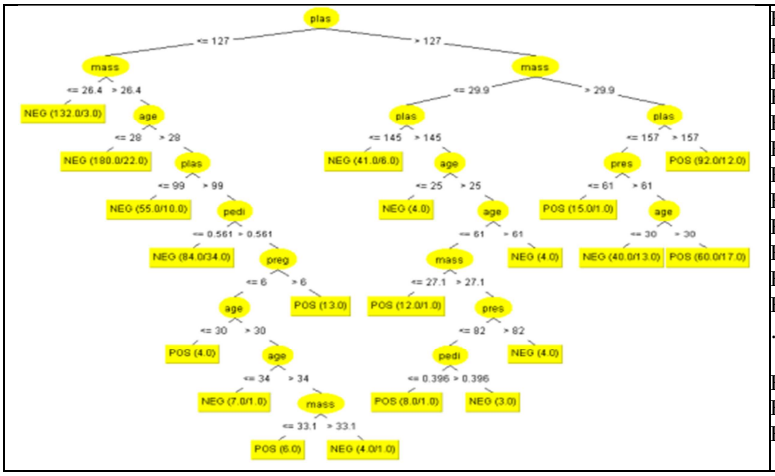
Table 1 - Ecoli dataset's decision tree in digitized form

Numerated tree map showing all node IDs in each decision path	Branch 1: 1.0 -> 1.1 -> 1.1.1 -> **1.1.1.1** Branch 2: 1.0 -> 1.1 -> 1.1.1 -> 1.1.1.2 -> 1.1.1.2.1 -> **1.1.1.2.1.1** Branch 3: 1.0 -> 1.1 -> 1.1.1 -> 1.1.1.2 -> 1.1.1.2.1 -> **1.1.1.2.1.2** Branch 4: 1.0 -> 1.1 -> 1.1.1 -> 1.1.1.2 -> **1.1.1.2.2** Branch 5: 1.0 -> 1.1 -> 1.1.2 -> **1.1.2.1** Branch 6: 1.0 -> 1.1 -> 1.1.2 -> **1.1.2.2** Branch 7: 1.0 -> **1.2**
Leaf-node IDs resemble a simplified and enumerated tree	Branch 1: **1.1.1.1** Branch 2: **1.1.1.2.1.1** Branch 3: **1.1.1.2.1.2** Branch 4: **1.1.1.2.2** Branch 5: **1.1.2.1** Branch 6: **1.1.2.2** Branch 7: **1.2**

In the second example, the Pima diabetes dataset model has 20 branches and 39 nodes, as shown in the left column of Table 2, they have been concisely and effectively represented by their leaf-node IDs, as shown in the right column of Table 2:

Table 2 - Pima dataset's decision tree represented by enumerated leaf-node IDs

Pima diabetes dataset's decision tree model	Leaf-node IDs



Branch 1: 1.1.1
Branch 2: 1.1.2.1
Branch 3: 1.1.2.2.1
Branch 4: 1.1.2.2.2.1
Branch 5: 1.1.2.2.2.2.1.1
Branch 6: 1.1.2.2.2.2.1.2.1
Branch 7: 1.1.2.2.2.2.1.2.2.1
Branch 8: 1.1.2.2.2.2.1.2.2.2
Branch 9: 1.1.2.2.2.2.2
Branch 10: 1.2.1.1
Branch 11: 1.2.1.2.1
Branch 12: 1.2.1.2.2.1.1
…

Branch 18: 1.2.2.1.2.1
Branch 19: 1.2.2.1.2.2
Branch 20: 1.2.2.2

Similarly, the Wisconsin cancer dataset's decision tree which has 14 branches and 27 nodes, the Liver Disorders dataset's decision tree which has 26 branches and 51 nodes, and the Page Blocks dataset's decision tree which has 41 branches and 81 nodes, all have been correctly mapped by their leaf-node IDs respectively in a concise manner, as shown in Table 3:

Table 3 - Another 3 decision trees represented by enumerated lead-node IDs

Wisconsin cancer dataset's enumerated decision tree	Liver Disorders dataset's enumerated decision tree	Page Blocks dataset's enumerated decision tree
Branch 1: 1.1.1	Branch 1: 1.1.1.1	Branch 1: 1.1.1.1
Branch 2: 1.1.2.1	Branch 2: 1.1.1.2.1	Branch 2: 1.1.1.2
Branch 3: 1.1.2.2.1.1	Branch 3: 1.1.1.2.2.1.1	Branch 3: 1.1.2.1.1.1
Branch 4: 1.1.2.2.1.2	Branch 4: 1.1.1.2.2.1.2.1	...
Branch 5: 1.1.2.2.2	Branch 5: 1.1.1.2.2.1.2.2.1	Branch 12: 1.1.2.2.2
Branch 6: 1.2.1.1	Branch 6: 1.1.1.2.2.1.2.2.2	Branch 13: 1.2.1
Branch 7: 1.2.1.2	…	Branch 14: 1.2.2.1.1.1.1.1
Branch 8: 1.2.2.1.1.1	Branch 17: 1.2.1.1.2.1.2.2.1	…
Branch 9: 1.2.2.1.1.2	Branch 18: 1.2.1.1.2.1.2.2.2	Branch 30: 1.2.2.1.2.1.1.2.1.2.1
Branch 10: 1.2.2.1.2.1.1	Branch 19: 1.2.1.1.2.2.1	Branch 31: 1.2.2.1.2.1.1.2.1.2.2
Branch 11: 1.2.2.1.2.1.2.1	Branch 20: 1.2.1.1.2.2.2.1	Branch 32: 1.2.2.1.2.1.1.2.2
Branch 12: 1.2.2.1.2.1.2.2	…	...
Branch 13: 1.2.2.1.2.2	Branch 24: 1.2.2.1.2.1	Branch 39: 1.2.2.2.2.1
Branch 14: 1.2.2.2	Branch 25: 1.2.2.1.2.2	Branch 40: 1.2.2.2.2.2.1
	Branch 26: 1.2.2.2	Branch 41: 1.2.2.2.2.2.2

Once each individual node is uniquely tagged, its classification statistics can then be collected, stored and analyzed at an individual node- and micro-level, as compared to the typical holistic model- and macro-level analysis based on the whole decision tree and its overall result. Some of the statistics collection and comparison results are documented in the following section.

5.2 Node-level statistics comparison and error-sensitive pattern identification

There may be different ways to examine the node-level statistics, for example, to compare the "heaviest" and "lightest" nodes using the highest and lowest counts of instances that traversed through, but the focus of this study is to identify and explore the "weakest" nodes with the highest error rates and the involved value patterns.

As a first step, the attributes and values involved with the top-3 nodes in error rate ranking are compared with the top-3 ranked attributes identified in the error-sensitive attribute (ESA) evaluation [19], to cross-check these two different error-sensitive pattern evaluation methods, as shown in Table 4. It is showing that three datasets - Pima, Wisconsin and Page Blocks, have closely comparable "underscored" error-sensitive attributes, and two datasets - Ecoli and Liver Disorders, have partially comparable "underscored" error-sensitive attributes.

Table 4 - The "weakest" nodes' attributes & values VS. The most error-sensitive attributes

Rank	**Ecoli dataset's enumerated tree node & error-rate … attributes & values involved**	**Attributes identified in ESA evaluation by attribute-error count**
1	**1.2** (3.36%: 4/119) … alm1>0.57	chg (17)
2	**1.1.1.1** (3.17%: 6/189) … alm1<=0.57 & aac<=0.64 & mcg<=0.73	alm1 (15)
3	**1.1.1** (3.02%: 6/199) … alm1<=0.57 & aac<=0.64	lip (14)
Rank	**Pima diabetes dataset's enumerated tree node & error-rate … attributes & values involved**	**Attributes identified in ESA evaluation by attribute-error count**
1	**1.1.2.2.2.1** (40.48%: 34/84) … plas<=127 & mass>26.4& age>28 & plas>99 & pedi<=0.561	plas (83)
2	**1.2.2.1.2.1** (32.50%: 13/40) … plas>127 & mass>29.9 & plas<=157 & pres>61 & age<=30	nmass (70)
3	**1.1.2.2.2** (30.51%: 36/118) … plas<=127 & mass>26.4 & age>28 & plas>99	age (31)
Rank	**Wisconsin cancer dataset's enumerated tree node & error-rate … attributes & values involved**	**Attributes identified in ESA evaluation by attribute-error count**
1	**1.2.2.1.2.1.2.1** (20.00%: 1/5) … UC_Sz>2 & UC_Sh>2 & UC_Sz<=4 & Bare_Nuc>2 & Clump_Th<=6 & UC_Sz>3 & Mg_Adh<=5	UC_Sz - Unif Cell Size (35)
2	**1.2.2.1.2.1.1** (15.38%: 2/13) … UC_Sz>2 & UC_Sh>2 & UC_Sz<=4 & Bare_Nuc>2 & Clump_Th<=6 & UC_Sz<=3	UC_Sh - Unif Cell Shape (35)
3	**1.2.2.1.2.1** (13.04%: 3/23) … UC_Sz>2 & UC_Sh>2 & UC_Sz<=4 & Bare_Nuc>2 & Clump_Th<=6	Clump_Th - Clump Thickness (30)
Rank	**Liver Disorders dataset's enumerated tree node & error-rate … attributes & values involved**	**Attributes identified in ESA evaluation by attribute-error count**
1	**1.2.2.1.2.2** (40.91%: 18/44)… gammagt>20 & drinks>5 & drinks<=12 & sgpt>21 & sgot>22	sgot (22)
2	**1.1.2.2.1.1** (38.10%: 8/21)… gammagt<=20 & sgpt>19 & sgot>20 & drinks<=5 & sgpt<= 26	mcv (16)
3	**1.2.2.1.2** (34.55%: 19/55)… gammagt>20 & drinks>5 & drinks<=12 & sgpt >21	alkphos (10)
Rank	**Page Blocks dataset's enumerated tree node & error-rate**	**Attributes identified in ESA**

	… attributes & values involved	evaluation by attribute-error count
1	**1.2.2.1.1.1.2.1.1.2** (30.00%: 3/10)… height>3 & eccen>0.25 & height <=27 & wb_trans<=7 & p_black<=0.178 & wb_trans>4 & blackpix <=20 & area<=108 & blackpix>7	mean_tr (89)
2	**1.1.2.1.1.1** (28.57%: 2/7)… height<=3 & mean_tr>1.35 & lenght<=7 & height<=2 & blackpix<=7	p_black (43)
3	**1.1.2.2.1.1.1.2** (25.00%: 1/4) … height<=3 & mean_tr>1.35 & lenght> 7 & mean_tr<=4.08 & height<=1 & wb_trans<=2 & mean_tr>3.75	eccen (29)
3	**1.2.2.1.1.1.2.1.1** (25.00%: 3/12)… height>3 & eccen>0.25 & height<= 27 & wb_trans<=7 & p_black<=0.178 & wb_trans>4 & blackpix<= 20 & area <=108	

In a second step, data records associated with the "weakest" nodes identified by the digitization method are removed and a re-test is carried out, and another re-test is carried out on the datasets after certain most error-sensitive attributes are removed as specified in the ESA evaluation study [19]. Initial results confirm improved accuracy in all five datasets after the "weakest" records are removed, and one improved significantly, as shown in Table 5. Also outlined in this table are the ESA removal scenario retest results, three datasets return improved accuracy, and the other two return poorer accuracy, and further analysis on the results is discussed in the Section 6.

Table 5 – Three-way performance comparison after removing the potentially "weakest" records and the most error-sensitive attributes

Ecoli's original dataset of 336 records	**Re-test 217 records after removing 119 "weakest" records**	**Re-test original data after removing top most ESA –** alm1
Accuracy: 94.05% with 20 errors	Accuracy: 97.70% with 5 errors ☑	Accuracy: 92.86% with 24 errors ☒
Pima diabetes' original dataset of 768 records	**Re-test 684 records after removing 84 "weakest" records**	**Re-test original data after removing top 2 most ESAs –** plas & mass
Accuracy: 73.83% with 201 errors	Accuracy: 77.19% with 156 errors ☑	Accuracy: 67.84% with 247 errors ☒
Wisconsin cancer's original dataset of 699 records	**Re-test 694 records after removing 5 "weakest" records**	**Re-test original data after removing top 2 most ESAs -** UC_Sz & UC_Sh
Accuracy: 94.13% with 41 errors	Accuracy: 95.97% with 28 error ☑	Accuracy: 95.71% with 30 errors ☑
Liver Disorders' original dataset of 345 records	**Re-test 301 records after removing 44 "weakest" records**	**Re-test original data after removing top 2 most ESAs –** sgot & mcv
Accuracy: 68.70% with 108 errors	Accuracy: 77.08% with 69 errors ☑	Accuracy: 71.01% with 100 errors ☑
Page Blocks' original dataset of 5473 records	**Re-test 5463 records after removing 10 "weakest" records**	**Re-test original data after removing top most ESA –** mean_tr
Accuracy: 97.19% with 154 errors	Accuracy: 97.36% with 144 errors ☑	Accuracy: 97.24% with 151 errors ☑

6 Experiment analysis

The purpose of decision tree digitization is not simply to convert a graphical decision tree into a digital map of nodes, but rather, to use such a digital map to facilitate the collection of node-level statistics for the purpose of node-level error-sensitive value pattern analysis, to help highlight the potentially "weakest" part of the decision tree and the specific error-sensitive attributes and values involved, to distinguish data records with such risky value patters for further error analysis and the development of error-reduction measure. Results from the initial experiments appeared to be supporting this digitization idea and the identification of the "weakest" node and the related attribute and value patterns. The following sections discuss the results and possible implications.

6.1 Digitized node IDs facilitate node-level analysis

The proposed decision tree digitization method makes the node-level analysis easy by formulating individual node IDs in a unique, numeric and contextual way. For example, the ID 1.2.1.1.2.1.2.2.1 as shown in the Liver Disorders example, is in a numeric text string format and incorporated with its preceding node IDs hierarchically within the same branch starting from the root. Because each ID is unique to the node in the tree, classification statistics can then be collected and stored for individual nodes using their IDs as the keys, and later to locate and retrieve the node-level statistics more efficiently than using the branch and node description text, e.g. "*gammagt > 20 & drinks <= 5 & drinks <= 3 & alkphos > 65 & sgot <= 24 & gammagt > 29 & mcv > 87 & mcv <= 92*", even if such lengthy verbiage is consolidated and simplified as "*gammagt > 29 & drinks <= 3 & lkphos > 65 & sgot <= 24 & mcv between 87 and 92*", it is still awkward. As decision trees grow bigger, such concise node IDs can become more useful because of its systematic and self-referential characteristics.

One way to utilize such node-level statistics is, to ranking the classification error rate from high to low, the top node with the highest error rate may then be considered as the "weakest" node in the tree, and the attributes and values associated with the "weakest" node may be considered more error-sensitive than others, in relative terms.

6.2 Examine the "weakest" nodes and error-sensitive value patterns

To evaluate the effectiveness of the proposed digitization method, the subsequent node-level investigative results can be validated by performance comparison, and one practical but rather non-deterministic measure can be to compare the classification accuracy after some simplistic error-reduction measure is applied. For example, by using the attribute and value patterns associated with the "weakest" node, the potentially "weakest" and most error-sensitive data records, include both the misclassified and correctly classified data instances, can be identified and separated for further examination, and the original dataset becomes smaller in size but potentially higher in reliability and accuracy. Experiment results seemed to confirm the validity of the "weakest" node and the associated error-sensitive value patterns in all five datasets,

one with a significant improvement in accuracy, and others with a modest but consistent level of improvement.

The best example is the Wisconsin cancer dataset with 699 records. After sorting and ranking the classification error rate of individual nodes, node 1.2.2.1.2.1.2.1 (20.00%: 1/5) is identified as the "weakest" node, as shown in Table 4. When the five data records associated with this "weakest" node are identified and separated from the dataset, that is 5/699=0.007% reduction in sample size, the accuracy improves by almost 2% in a re-test, as shown in Table 5. Instead of one less error due to the removal of five error-sensitive records, there are 13 less errors in the re-test.

The other four datasets also show various levels of success in accuracy enhancement. For example, in the Liver Disorders dataset with 345 records, node 1.2.2.1.2.2 (40.91%: 18/44) is identified as the "weakest" node, as shown in Table 4, and there are 44 records associated with this node and 18 of them are errors. A re-test to the updated dataset after the removal of those 44 error-sensitive records shows the actual error reduction is 39 instead of 18, and the overall accuracy has improved from 68.70% in the original dataset to 77.08% in the updated dataset.

One possible explanation to such impressive result is, the inclusions of the "weakest" data records have made "potentially significant" adverse impact to the info-gain (entropy) calculation when constructing C4.5/J48 decision trees because of their error-sensitive attributes and values, which leads to error-prone split point conditions and the consequent "weakest" nodes. If the impact is less significant, then the difference between the original and re-test result may be not so noticeable, as shown in the Page Blocks dataset.

This reasoning may partially explain why ensemble tree models, such as Random Forest, are considered superior to standalone tree models. The Random Forest model selects a portion of the data attributes randomly and generates hundreds and thousands of trees accordingly, and then votes for the best performing one to produce the classification result. The random attribute selection process may have inadvertently generated and voted for trees without some highly error-sensitive attributes, and also with bigger value ranges to split on due to fewer attributes involved, therefore enables ensemble models to produce more accurate results, but on the expense of resources and simplicity.

6.3 Discuss possible contribution, effectiveness and weakness

The evaluation study on error-sensitive attributes [19] has provided some constructive leads for this current study, but this decision tree digitization and node-level examination idea can be considered as another step forward because of its expansion from attribute level evaluation to individual node and split-value level evaluation. While still at an early stage, this latter expansion and study has shown encouraging and consistent experiment results, therefore, this can be considered as a potential contribution to the node-level analysis topic for decision trees.

In terms of effectiveness, this digitization method applies a digital way to tag each individual node of a decision tree uniquely and concisely with contextual reference, to simplify node-level statistics collection and analysis and expand the typical tree-level

"macro" analysis with focus on the whole classification model into the node-level "micro" analysis with focus on specific attributes and values, and in a systematic and transparent way.

Meanwhile, the list of weakness of this study is also long and obvious. First, the successful experiments are based on the removal of the "weakest" data records, which may seem drastic and lacking of formal and theoretical proof; however, this has still highlighted the usefulness in identifying the "weakest" value patterns. This has led to the second major weakness - it is unclear what to do with the "weakest" records. Their removal improves overall accuracy, so a new question is, should a separate model be used to evaluate these error-sensitive records? If the "one size fits all" approach is not recommended, why not introduce a separate model for the "doubtful" data? The third major weakness is, this study is not based on ensemble methods, and ensemble trees are now the preferable classification models due to their superior performance to the standalone decision tree models, this makes the proposed decision tree digitization method less relevant to the latest classification development. Despite more weaknesses are still to be discussed, they have been recognized and will be used as a form of inspiration to broaden and advance this study.

7 Conclusion

This study attempts to address the question - "Is there a way to identify an Achilles' heel of a classification model?", that is, finding a way to locate the 'weakest' and most error-sensitive spot in the model. Towards this goal, the study develops a decision tree digitization method to facilitate the identification and examination of the potentially "weakest" nodes and error-sensitive value patterns in the model using decision trees as a pilot model. Initial experiment have demonstrated successful results when comparing to earlier evaluation study of error-sensitive attributes, but also prompted more questions.

Many of the study's own weak and questionable areas have been recognized, such as the need of formal and theoretical proof, the expansion of evaluation into ensemble methods and non-binary trees etc., and they will form the basis for the next phase of the study, such as a revision of the digitization method to cover ensemble models, and to find a more logical way to understand and utilize the "weakest" data records with error-sensitive value patterns.

Reference

1. Quinlan, J.R.: C4. 5: Programs for Machine Learning. Morgan Kaufmann (1993)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group, Belmont, CA (1984)
3. Alpaydin, E.: Introduction to Machine Learning. The MIT Press, London, England (2004)
4. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco, CA (2006)

5. Saeys, Y., Inza, I., Larranaga, P.: A Review of Feature Selection Techniques in Bio-Informatics. Bioinformatics, vol. 23, no. 19, pp. 2507-2517 (2007)

6. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco, CA (2005)

7. Gheyas, I.A., Smith, L.S.: Feature Subset Selection in Large Dimensionality Domains. Pattern recognition, vol. 43, no.1, pp. 5-13 (2010)

8. Tabakhi, S., Moradi, P., Akhlaghian, F.: An Unsupervised Feature Selection Algorithm Based on Ant Colony Optimization. Engineering Applications of Artificial Intelligence, vol. 32, pp. 112-123 (2014)

9. Breiman, L.: Bagging Predictors. Machine learning, vol. 24, no.2, pp. 123-140 (1996)

10. Schapire, R.E.: The Strength of Weak Learnability. Machine learning, vol. 5, no.2, pp. 197-227 (1990)

11. Ho, T.K.: Random decision forests. In Proceedings of the Third International Conference on Document Analysis and Recognition, 1995, vol. 1, pp. 278-282

12. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of Online Learning and an Application to Boosting. In Computational learning theory, pp. 23-37 (1995)

13. Breiman, L.: Random Forests. Machine learning, vol. 45, no.1, pp. 5-32 (2001)

14. Grossmann, E.: AdaTree: Boosting a Weak Classifier into a Decision Tree. In Computer Vision and Pattern Recognition Workshop, 2004

15. Tu, Z.: Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering. In Tenth IEEE International Conference on Computer Vision, 2005, vol. 2, pp. 1589-1596

16. Monteith, K., Carroll, J.L., Seppi, K., Martinez, T.: Turning Bayesian Model Averaging into Bayesian Model Combination. In the 2011 International Joint Conference on Neural Networks, pp. 2657-2663

17. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons (2004)

18. Yang, P., Yang, Y.H., Zhou, B., Zomaya, A.: A Review of Ensemble Methods in Bioinformatics. Current Bioinformatics, vol. 5, no.4, pp. 296-308 (2010)

19. Wu, W., Zhang, S.: Evaluation of Error-Sensitive Attributes. Trends and Applications in Knowledge Discovery and Data Mining, pp. 283-294, Springer Berlin Heidelberg (2013)

20. Bache, K., Lichman, M.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, School of Information and Computer Science, Irvine, CA (2013)

21. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, vol. 11, no. 1, (2009)