

“© 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Exploring Error-Sensitive Attributes and Branches of Decision Trees

William Wu

Faculty of Engineering and Information Technology
University of Technology
Sydney, Australia

Abstract — Decision trees have a reputation of being efficient and illustrative in classification learning, and majority of the research effort has been focused on making classification improvement in a head-on style with wide-range research topics, such as tree algorithm development and refinement, attribute selection and prioritization, sampling technique improvement, and the addition of cost matrix and other performance-enhancing factors. One less commonly studied topic is about the characteristics of classification errors and how they may be associated with specific attributes due to correlation or causation, and within what value ranges on such attributes when pattern are most likely. This research intends to study this dim area in a sort-of reverse and forensic style as part of post-classification analysis, to analyze the patterns and relationship between errors and attributes, to explore how attributes' risk level in error may play a role in leading to more risky, more error-prone decision tree branches or decision paths. Possible benefits from this study would include raising data stakeholders' awareness of such specific error-sensitive attributes and decision paths, to facilitate better understanding of possible causes and impact of errors and the development of more effective error-reduction measures customized to suit the specific patterns and individual datasets.

Such emphasis on highlighting the specific error-sensitive attributes and decision branches within individual datasets is a reflection of our observation which shared by others - "additional domain-specific knowledge, external to the training set, must be employed to estimate the noise level (... and) the underlying model's complexity ... (because) knowledge-poor tree induction algorithms do not exploit such information." [2]

Keywords – decision tree, error-sensitive attribute, error-sensitive tree branch, feature selection, post-classification analysis

I. INTRODUCTION

The formation of each decision tree branch is different due to the variation of attributes and their split-point values, therefore, the classification result produced by each branch may vary, some branches may have a wider inclusion of sample instances than others, and some branches may lead to a more significant classification path with higher accuracy than others. "How to create a new tree model or modify the current one for better performance?", "What are the most influential attributes that have been used in the process?", and, "What other factors, such as feature-selection routine and cost-sensitive matrix, can be added to enhance the performance?", these are some of forwarding-thinking questions commonly

asked during post-classification analysis. One less favorite question is, "Which tree branch is the weakest and associated with more misclassification errors?", together with another less common question, "What attributes are more likely to cause or associate with those errors?", these error-centric questions are the focus of this "reverse" study.

The study of weak branches of a decision tree can be considered relating to the tree pruning topic. One typical way to handle weak branches is pruning, and a good number of decision tree pruning methodologies have been developed and re-engineered over the years, such as the Error Reduction pruning method [12], the Cost-Complexity pruning method [3], and the Minimum Error pruning method [5]. This study is not about creating a new pruning technique, but to explore an alternative way in achieving the same error reduction objective. The starting point of this exploration process is not before, not during, but after the fact, that is, after the pre-determined feature selection and sub-tree pruning routine have already been completed as part of the classification process, so the exploration can examine the overall classification result, as well as having a particular focus on the most error-prone decision tree branches and possible relationship patterns between these branches and the most error-sensitive attributes.

Our proposed exploration is carried out in connection with the evaluation process of error-sensitive attributes, an idea has recently been discussed [16]. This evaluation idea used three specific terms, "ambiguous value range", "attribute-error counter" and "error-sensitive attribute", to describe how the most error-sensitive attributes could be identified by ordering the attributes' risk level which is based on each attribute's error count within its ambiguous value range. In this proposed exploration, two new routines have been added onto the mentioned evaluation process. The first routine is to rank the resulted decision tree branches according to their associated predicted error counts, and the second routine is to examine the existence of the most error-sensitive attributes and their value ranges on the identified risky branches.

Here is a brief scenario to outline a practical issue. In a binary data dataset with 10,000 sample instances and 50 data attributes, a classification task has been performed by a decision tree model enabled with its own feature selection and pruning routine. The result is a pruned tree with 60 nodes, 30 leaves and 100 misclassification errors. Amongst the 30 root-to-leaf branches, some have higher error counts than others, so

our research question is, what may be the relationship patterns associated with the most error-prone root-to-leaf branches and the most error-sensitive attributes?

Figure-1 is an example of a decision tree with its size as seven, that is, three nodes and four root-to-leaf branches; in comparison, the tree structure generated from the above scenario would be about 10 times bigger than this example.

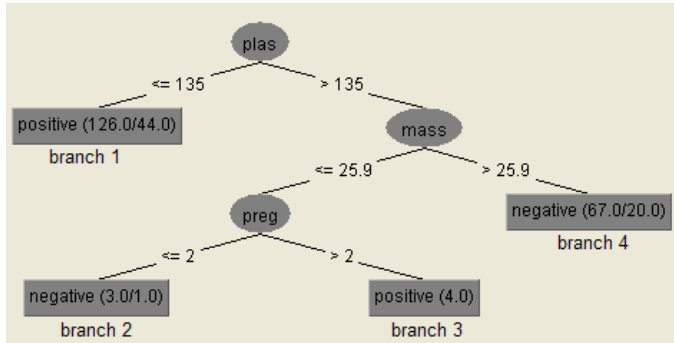


Figure 1. A sample C4.5 tree counting root-to-leaf branches from left to right and each branch is showing its predicted error-rate

The term “tree branch” or “branch” in this paper has been referring to a full branch from the tree root to an end leaf; a portion of such branch or a sub-tree at a node is outside the scope of this study at this stage. It has also been noted that a misclassification error with its one or more attribute values being inside one or more ambiguous ranges may not necessarily be the sole reason that causes the error, but such ambiguity can be assumed to increase the risk level of being misclassified.

The rest of this paper is organized as follows. Section-II reviews some early influential work that inspired and guided the current study, and outlines a recent attribute evaluation process which being utilized as a component of this exploration proposal, section-III describes the exploration process details, section-IV summarizes the experiments on five datasets, section-V compares and analyzes the experiment results, and finally, section-VI concludes the proposed exploration development and outlines a possible plan for future study.

II. RELATED WORK

Decision trees have a reputation of being efficient and illustrative in machine learning and data classification, one prominent decision tree model is Quinlan’s C4.5 model [12, 15], therefore, C4.5 decision tree has been adopted as the underlying classification model for this exploration process.

One key step in conducting decision tree classification is the attribute selection routine when constructing the tree from root to leaves, and Quinlan’s C4.5 model has applied a selection method known as gain ratio. This gain ratio method originated from the information gain concept described in Shannon’s information theory [14], in which it states the expected information required to classify a label for a record in the dataset is the probability for such a record belongs to its class label. However, this information gain approach has a strong bias for selecting attributes with many outcomes. To

address this deficiency, info gain ratio was introduced as the attribute selection criterion in C4.5, to apply “a kind of normalization in which the apparent gain attributable to tests with many outcomes is adjusted”, which means such info gain is to be divided by the number of possible partitions/outcomes for attribute A, so it “represents the proportion of information generated by the split that is useful, i.e., that appears helpful for classification” [12]. As a result, the gain ratio method is to select an attribute which can maximize its gain ratio value. A variety of other attribute selection methodologies have also been developed, such as the Gini index method that used in the CART decision tree model [3], the RELIEF algorithm [7], the Sequential Forward Selection (SFS) method and the Sequential Backward Elimination (SBE) method [8].

In order to improve classification accuracy and to simplify tree models that have “grown-to-fit” the training data in one particular “shape”, and are later found to be “overfitting” with the test data in a different “shape”, various pruning techniques have been developed to make decision tree structure simpler and more efficient. Some of the influential and proven pruning techniques are, the Cost-Complexity Pruning method [3], which selects the superior sub-tree branch from a series of branches based on comparing the error rate and complexity in a bottom-up approach; the Reduced-Error Pruning method [11], which replaces a branch by selecting a leaf representing more instances but with a lower error rate in a bottom-up way; the Pessimistic Pruning method [11], which applies a statistical “continuity correction” and the comparison of one-standard-error before making a sub-tree to leaf replacement in a top-down fashion; and the Critical value pruning method [10], which replaces a sub-tree when its split-point has a lower value than a pre-determined critical threshold value in a bottom-up approach. After pruning, the tree structure is supposed to be more concise and balanced, and accompanied with more accurate classification.

In terms of post-classification analysis, an evaluation process to identify the most error-sensitive attributes has been suggested (Wu & Zhang in press). This evaluation model is currently aiming at binary datasets only, to first locate each attribute’s ambiguous value range which is the overlap area between its Negative and Positive samples, then to calculate each attribute’s attribute-error counter which is the number of misclassified samples with attribute values being within its ambiguous value range, then to rank the counter values from high to low, to identify the possible most error-sensitive attributes because of their association with higher error counts. Figure-2 is an illustration of three simplified situations of a binary attribute’s value range, two being overlap and ambiguous:

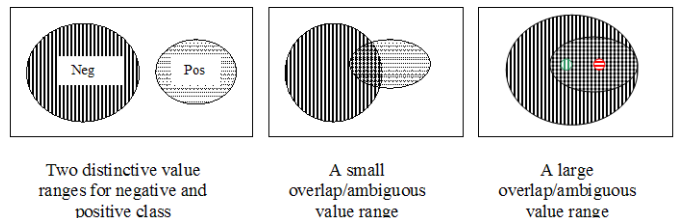


Figure 2. Three situations of a binary attribute’s value range

Many works by other researchers have also been studied for this exploration idea, only some brief highlights are included in this very limited review section.

III. EXPLORATION PROCESS DESCRIPTION

The proposed exploration process can be considered as a new addition in parallel with the evaluation process of error-sensitive attributes, these two processes later combine to form a part of the post-classification analysis, which can be viewed as a new 3rd phase process as illustrated in bold and blue in Figure-3. Based on a conventional 2-phase classification model [9], in which 1st-phase is about the pre-process routines, such as training and test data preparation, attribute validation and prioritization; 2nd-phase is about the core classification process, including tree structure generation and pruning, and this 3rd phase exploration process is about examining the classification results and with a special focus on errors, aiming to develop a more customized and effective error-reduction measure.

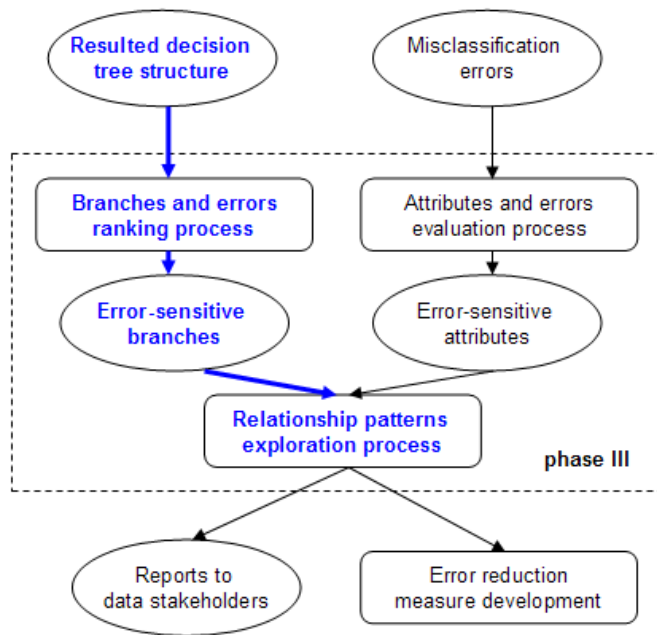


Figure 3 – Error-sensitive branch and attribute ranking join forces in exploring specific patterns during 3rd phase in a 3-phase classification model

On the completion of a typical classification task, this post-classification exploration process would begin. The initial ranking process for the resulted decision tree branches can start simultaneously with the evaluation process for error-sensitive attributes, or one after another, in either order.

The first step of the branch ranking process is to convert all root-to-leaf branches from the left-most branch to the right-most branch into a set of decision rules, with their exact attributes and split-point value details, as well as the associated error count of each branch. These error counts are then ranked and the branches with the highest error counts can be identified. These highly error-sensitive branches are then compared with the most error-sensitive attributes identified by the attributes evaluation process, of which the required steps

have been discussed [16] and are incorporated into the following pseudocode style process summary.

Input: (1) the serialized decision tree structure from the classification result (2) all $(m + n)$ test sample instances of t attributes with their original class labels and their newly classified result labels

Output: (1) a ranking list of all root-to-leaf decision branches in the form of classification rules sorted by each branch’s predicted error rate (2) a ranking list of the t attributes sorted by their attribute-error counters (3) highlighting the possible weakest - most error-sensitive tree branches and the most error-sensitive attributes and their value ranges on those branches

Ranking process for tree branches in two steps:
Step-1: convert the serialized tree structure into a set of exact-match root-to-leaf classification rules from left to right, each rule is attached with its predicted error rate based on the resulted tree model
Step-2: sort the error rates from high to low to identify the possible most error-sensitive branches

Evaluation process for attributes in two steps:
Step-1: work out the ambiguous value range and the attribute-error counter value for each attribute ... refer to [16] for more details
Step-2: sort the attribute-error counter values from high to low to identify the possible most error-sensitive attributes

Exploration process for connection between error-sensitive branches and attributes:
Step-1:
 for 1 to y most error-sensitive branches
 for 1 to z most error-sensitive attributes
 find location of this attribute
 if the attribute is part of the rule
 then compare its split-point value(s) within the branch against the attribute’s ambiguous value range
 if a split-point value is within the ambiguous value range
 then highlight such a split-point as a possible high-risk connection between the branch and the attribute
 end;
Step-2: examine the connection patterns and develop possible more effective error-reduction measures with help from stakeholders and domain experts

To verify whether our proposed exploration process can actually identify any highly error-sensitive tree branches and specific relationship patterns between errors, error-sensitive attributes and branches, initial experiments have been carried out on a number of real world datasets, and their results are analyzed and discussed in the next two sections.

IV. EXPERIMENTS

Five datasets from the UCI Machine Learning Repository [1] have been used to test this exploration idea using Quinlan’s C4.5 algorithm-based J48 classifier in WEKA [15], with all the default settings, such as, applying its underlying feature-selection and C4.5 pruning routine as part of the classification. The conversion from a resulted decision tree structure to a set of classification rules starts from the tree’s left-most root-to-leaf branch to the right-most branch, and the error reduction measure used in this early stage is simply to filter out one or

multiple most or highly error-sensitive attributes. The top-4 ranked error-sensitive attributes, showing with their attribute-error counter values and their ambiguous value ranges, are compared with the GainRatio and InfoGain ranking results. These identified attributes are also highlighted in the top-3 ranked error-sensitive tree branches, together with the error count and the number of samples associated with each branch.

All five datasets in the experiments showed supportive results, although their error-reduction rates are not significant, the improvement is consistent across all datasets, varying from 1.5% to 0.1% accuracy enhancement. This appeared to rather encouraging when comparing to the mixed results produced by the earlier evaluation experiments which involved a different group of UCI datasets [16]. Part of the results are summarize as follows.

A. Connectionist Bench (Sonar, Mines vs. Rocks) Dataset

This binary dataset has 208 instances and 60 attributes. One of the optimal tests showed the accuracy rate increased from **71.15%** to **72.60%** and errors reduced from 60 to 57, and all top-3 error-sensitive branches contained the most error-sensitive attribute field_11, and its values on those branches were either within or close to its ambiguous value range of 0.17~0.29.

Combined-table 1 - Exploration results and performance comparison for the Sonar data dataset

Rank	by Attribute-error counter	by GainRatio	by InfoGain
1	field_11 (26): 0.17~0.29	field_11 (0.2053)	field_11 (0.2014)
2	field_48 (24): 0.07~0.11	field_12 (0.1803)	field_12 (0.1779)
3	field_45 (23): 0.14~0.25	field_9 (0.1634)	field_9 (0.1498)
4	field_9 (21): 0.14~0.21	field_44 (0.1589)	field_10 (0.143)

Rank	Root-to-leaf decision path (classification rule)	Error count	Sample count
1	branch 11: field_11 > 0.197 and field_27 <= 0.8145 and field_54 <= 0.0205 and field_53 <= 0.0166 and field_21 > 0.5959 and field_51 <= 0.0153 and field_23 <= 0.7867: R	1	13
2	branch 14: field_11 > 0.197 and field_27 <= 0.8145 and field_54 <= 0.0205 and field_53 > 0.0166: M	1	12
3	branch 9: field_11 <= 0.197 and field_1 > 0.0392: M	1	8

Original run of J48/C4.5 classification with all attributes	Re-run J48/C4.5 without error-sensitive attributes field_45
=== Stratified cross-validation === === Summary === Correctly Classified Instances 148 71.1538 % Incorrectly Classified Instances 60 28.8462 %	=== Stratified cross-validation === === Summary === Correctly Classified Instances 151 72.5962 % Incorrectly Classified Instances 57 27.4038 %

B. MAGIC Gamma Telescope Dataset

This relatively large binary dataset has 19,020 instances and 10 attributes. One of the optimal tests showed the accuracy rate increased from **85.06%** to **85.17%** and errors reduced from 2,842 to 2,820, all top-3 ranked error-sensitive branches contained two of the top-ranked error-sensitive attributes, and their values were either within or close to their ambiguous value ranges.

Combined-table 2 - Exploration results and performance comparison for the Gamma dataset

Rank	by Attribute-error counter	by GainRatio	by InfoGain
1	fAlpha (865): 18.78~43.99	fLength (0.10211)	fAlpha (0.1771)
2	fAsym (763): -18.29~3.27	fAlpha (0.06127)	fWidth (0.1324)
3	fM3Long (746): -2.85~17.81	fWidth (0.05037)	fLength (0.1158)
4	fWidth (614): 18.59~28.80	fAsym (0.0482)	fM3Long (0.1044)

Rank	Root-to-leaf decision path (classification rule)	Error count	Sample count
1	branch 132: fLength <= 114.586 and fAlpha > 20.2535 and fLength <= 38.5309 ... and fDist <= 217.7938 and fSize <= 2.3133: g	151	1,130
2	branch 169: fLength <= 114.586 and fAlpha > 20.2535 and fLength <= 38.5309 ... and fConc <= 0.5742 and fM3Long > -23.6361: g	131	874
3	Branch 62: fLength <= 114.586 and fAlpha <= 20.2535 and fM3Long > -67.699 ... and fLength <= 99.1202 and fDist > 124.4584: g	130	3,950

Original run of J48/C4.5 classification with all attributes	Re-run J48/C4.5 without error-sensitive attributes [fAsym]
=== Stratified cross-validation === === Summary === Correctly Classified Instances 16,178 85.0578 % Incorrectly Classified Instances 2,842 14.9422 %	=== Stratified cross-validation === === Summary === Correctly Classified Instances 16,200 85.1735 % Incorrectly Classified Instances 2,820 14.8265 %

C. Yeast Dataset

This Multivariate dataset of 1,484 records has been converted from 10 class labels into a binary form of two labels, NUC (nuclear) and non_NUC. One of the optimal tests showed the accuracy rate increased from **75.20 %** to **76.01%** and errors reduced from 368 to 356, all top-3 ranked error-sensitive branches contained at least two top-ranked error-sensitive attributes, nuc and alm, their values on those branches are either within or close to their ambiguous value ranges.

Combined-table 3 - Exploration results and performance comparison for the Yeast dataset

Rank	by Attribute-error counter	by GainRatio	by InfoGain
1	nuc (111): 0.25~0.33	nuc (0.062)	nuc (0.096)
2	gvh (102): 0.46~0.52	pox(0.061)	alm (0.064)
3	alm (92): 0.49~0.53	alm (0.042)	mcg (0.058)
4	mcg (78): 0.45~0.52	mit (0.028)	gvh (0.037)

Rank	Root-to-leaf decision path (classification rule)	Error count	Sample count
1	branch 8: alm > 0.4 and nuc <= 0.24 and nuc > 0.14 and gvh > 0.39: non_NUC	91	562
2	branch 13: alm > 0.4 and nuc > 0.24 and mit <= 0.4 and mcg <= 0.58 and nuc <= 0.31 and alm > 0.52: NUC	42	105
3	branch 12: alm > 0.4 and nuc > 0.24 and mit <= 0.4 and mcg <= 0.58 and nuc <= 0.31 and alm <= 0.52 and vac > 0.46: non_NUC	30	88

Original run of J48/C4.5 classification with all attributes	Re-run J48/C4.5 without error-sensitive attributes gvh & mcg
=== Stratified cross-validation === === Summary === Correctly Classified Instances 1116 75.2022 % Incorrectly Classified Instances 368 24.7978 %	=== Stratified cross-validation === === Summary === Correctly Classified Instances 1128 76.0108 % Incorrectly Classified Instances 356 23.9892 %

D. Cardiotocography Dataset

This Multivariate dataset of 2,126 records has been converted from three class labels into a binary form of two labels, Normal and Abnormal. One of the optimal tests showed the accuracy rate increased from **98.82%** to **98.92%** and errors reduced from 25 to 23, all top-3 ranked error-sensitive branches contained only the most error-sensitive attribute CLASS_Code, and its values on those branches were either within or close to its ambiguous value range.

Combined-table 4 - Exploration results and performance comparison for the Cardiotocography dataset

Rank	by Attribute-error counter	by GainRatio	by InfoGain
1	CLASS_Code (23): 3.39~8.45	CLASS_Code (0.362)	CLASS_Code (0.68101)
2	ASTV (10): 42.47~62.89	DP (0.2322)	ASTV (0.23206)
3	AC (9): 3.13~3.98	DS (0.2201)	MSTV (0.22895)
4	MSTV (9): 0.99~1.43	MSTV (0.1323)	AC (0.20366)

Rank	Root-to-leaf decision path (classification rule)	Error count	Sample count
1	branch 3: CLASS_Code <= 7 and CLASS_Code > 4 and CLASS_Code > 5 and LB <= 145 and DP <= 0.001485: Normal	6	532
2	branch 2: CLASS_Code <= 7 and CLASS_Code > 4 and CLASS_Code <= 5: Abnormal	3	72
3	branch 1: CLASS_Code <= 7 and CLASS_Code <= 4: Normal	2	1097

Original run of J48/C4.5 classification with all attributes	Re-run J48/C4.5 without error-sensitive attributes ASTV & AC
==== Stratified cross-validation ====	==== Stratified cross-validation ====
==== Summary ====	==== Summary ====
Correctly Classified Instances 2101 98.8241 %	Correctly Classified Instances 2103 98.9182 %
Incorrectly Classified Instances 25 1.1759 %	Incorrectly Classified Instances 23 1.0818 %

E. Glass Identification Dataset

This Multivariate dataset of 214 records has been converted from seven class labels into a binary form with two labels, building_windows and non_building_windows. One of the optimal tests showed the accuracy rate increased from **85.51%** to **86.92%** and errors reduced from 31 to 28, all top-3 ranked error-sensitive branches contained two or three top-ranked error-sensitive attributes, and their values on those branches were either within or close to their ambiguous value ranges.

Combined-table 5 - Exploration results and performance comparison for the Glass ID dataset

Rank	by Attribute-error counter	by GainRatio	by InfoGain
1	Sodium (17): 13.17~13.91	Magnesium (0.3351)	Potassium (0.2905)
2	Aluminum (15): 1.29~1.76	Barium (0.3241)	Magnesium (0.2889)
3	Magnesium (6): 1.44~3.27	Sodium (0.2658)	Barium (0.1973)
4	refractive_index (5): 1.518~1.519	Aluminum (0.2658)	Aluminum (0.1972)

Rank	Root-to-leaf decision path (classification rule)	Error count	Sample count
1	branch 5: Magnesium > 2.68 and Sodium <= 13.31: building_windows	5	92
2	branch 8: Magnesium > 2.68 and Sodium > 13.31 and Barium <= 0.11 and Calcium > 8.31 and refractive_index > 1.518: building_windows	5	27
3	branch 1: Magnesium <= 2.68 and refractive_index <= 1.5241 and Iron <= 0.07: non_building_windows	2	45

Original run of J48/C4.5 classification with all attributes	Re-run J48/C4.5 without error-sensitive attributes Aluminum & refractive_idx
==== Stratified cross-validation ====	==== Stratified cross-validation ====
==== Summary ====	==== Summary ====
Correctly Classified Instances 183 85.514 %	Correctly Classified Instances 186 86.9159 %
Incorrectly Classified Instances 31 14.486 %	Incorrectly Classified Instances 28 13.0841 %

V. EXPERIMENT ANALYSIS

It has been acknowledged that the current error-reduction measure of filtering out the most error-sensitive attributes is a simplistic one, it does not consider their possible close relationship with the remaining attributes, and therefore such simple filter-out/removal may introduce unexpected negative impact and inaccuracy to the re-run of classification process.

Such implied negative impact may explain why the improvement from our initial experiments have been marginal; and even though the improvement on all five datasets are consistent, they have been based on specific selection of the most error-sensitive attributes of each dataset. Some of the issues and suggestions can be outlined as follows.

- Filter-out the most error-sensitive attribute may not necessarily reduce the misclassification errors; in many situations, its removal actually increased the errors. For example, in the Sonar dataset, field_11 was identified as the most error-sensitive attribute, but it was also the most significant attribute for GainRatio, the feature selection method behind the C4.5 decision tree algorithm, so when field_11 was removed, the errors increase because the negative impact on the underlying feature selection routine outweighed the possible error-reduction measure
- When the highly error-sensitive attributes are also the highly ranked GainRatio attributes, it is an intrigue task to select the right attribute to remove, to ensure the right balance between the intend of error-reduction and the negative impact on the underlying feature-selection routine. The Sodium attribute in the Glass Id dataset is the most error-sensitive attribute and 3rd-ranked significant attribute for GainRatio, its removal caused the errors to increase, and it shared the same reason as the previous point; in contrast, the fAsym attribute is the 2nd most error-sensitive attribute in the Gamma dataset, and it also ranked 4th in GainRatio, but its removal enhanced the accuracy rate and the errors reduced from 2,842 to 2,820
- Highly ranked error-sensitive attributes may be associated with higher risk level of errors merely because their wide and inclusive ambiguous value ranges and other factors, such as dependence and correlation, and they should not be considered as the main contributor or influencer of the errors; therefore their removal would not guarantee reduction of errors. One such example is the field_48 attribute in the Sonar dataset, it is the 2nd most error-sensitive attribute so it is considered highly risky, and is ranked 10th for

GainRatio and is supposed to be less influential for GainRatio and C4.5, however, its removal made no difference to the accuracy

It is natural to assume that the most error-sensitive decision tree branches are closely related to the most error-sensitive attributes. The effort of this study has been to highlight the links between these high-risk branches and the high-risk attributes, to explore any specific relationship patterns between them. Experiment results appeared to have verified such obvious assumption but only at a vague level; therefore, the exploration process should be enhanced to provide further details on such vague level of patterns information. Some related thoughts are outlined below:

- The proposed identification method of the most error-sensitive tree branches is only based on the predicted error-rate produced when the tree structure was formed based on training data, but the identification of the most error-sensitive attributes is based on the test data. Depending on the level of similarity between the test data and training data, the reliability level of relationship patterns highlighted in the current process may vary widely. One such example is in the Sonar dataset, all top-3 most error-sensitive tree branches each has only one error reported; when using the test data under stratified cross-validation, these top-3 branches yielded six, five and one error(s) respectively, only 3rd ranked branch maintained the same error-rate
- Value ranges of some most error-sensitive attributes on the error-sensitive branches appeared less specific than expected; such relationship patterns are too vague and uninformative. One such example is in the Glass Id dataset, Sodium is the most error-sensitive attribute with ambiguous value range as 13.17~13.91, and Magnesium is 3rd most error-sensitive with ambiguous value range as 1.44~3.27, they both have been correctly highlighted in the most error-sensitive tree branch of “*branch 5: Magnesium > 2.68 and Sodium <= 13.31: building_windows*”, but their value ranges on the branch appeared to be open-ended and uninformative

Much work is required to make the current tree branch exploration process more sophisticated and more specific, and on this note, the importance of having domain expertise joining this post-classification analysis has to be reiterated. Only the stakeholders of the data and the domain experts can genuinely examine and advise on the correctness and effectiveness of the proposed findings, and can eventually make practical use of the proposed processes. Without close collaboration with the stakeholders and in-depth field knowledge, such a study can be seemed as only playing with numbers, and building a “white elephant” or a “pet project”.

VI. CONCLUSION

This study has provided a systematic way to explore and identify the most error-sensitive decision tree branches and attributes, despite its simplicity and still at its early stage, results from initial experiments appeared to be supportive for such an exploration process. Further study has been planned to refine the error-sensitive evaluation components and the tree branch ranking routine, to take on a risk level weighting approach rather than the removal of the most error-sensitive attributes as an error-reduction test measure, to develop the current exploration process into a more sophisticated and creditable model.

REFERENCES

- [1] Bache, K. & Lichman, M. 2013, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
- [2] Bredensteiner, E.J. & Bennett, K.P. 1998, 'Feature Minimization within Decision Trees', Computational Optimization And Applications, vol. 10, no. 2, pp. 111-126.
- [3] Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. 1984, Classification and Regression Trees, Wadsworth International Group, Belmont, Calif.
- [4] Breslow, L.A. & Aha, D.W. 2001, 'Simplifying decision trees: a survey', The Knowledge Engineering Review, vol. 12, no. 01, pp. 1-40.
- [5] Cestnik, B. & Bratko, I. 1991, 'On estimating probabilities in tree pruning', Machine Learning - EWSL-91, vol. 482, pp. 138-150.
- [6] Guyon, I. & Elisseeff, A. 2003, 'An introduction to variable and feature selection', The Journal of Machine Learning Research, vol. 3, p. 25.
- [7] Kira, K. & Rendell, L.A. 1992, 'A practical approach to feature selection', Proceedings of the ninth international workshop on Machine learning, Morgan Kaufmann Publishers Inc., pp. 249-256.
- [8] Kittler, J. 1978, 'Feature set search algorithms', Pattern recognition and signal processing, vol. 41, p. 60.
- [9] Liu, H. & Yu, L. 2005, 'Toward Integrating Feature Selection Algorithms for Classification and Clustering', IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 4.
- [10] Mingers, J. 1989, 'An empirical comparison of pruning methods for decision tree induction', Machine learning, vol. 4, pp. 227-243.
- [11] Quinlan, J.R. 1987, 'Simplifying decision trees', International Journal of Man-Machine Studies, vol. 27, no. 3, p. 14.
- [12] Quinlan, J.R. 1993, C4.5: programs for machine learning, Morgan Kaufmann.
- [13] Saeys, Y., Inza, I. & Larranaga, P. 2007, 'A review of feature selection techniques in bioinformatics', Bioinformatics, vol. 23, no. 19, p. 10.
- [14] Shannon, C.E. 1948, 'A Mathematical Theory of Communication', The Bell System Technical Journal, vol. 27, pp. 379-423, 623-656.
- [15] Witten, I.H. & Frank, E. 2005, Data Mining: Practical machine learning tools and techniques, 2nd edn, Morgan Kaufmann, San Francisco
- [16] Wu, W. & Zhang, S. in press, 'Evaluation of Error-Sensitive Attributes', paper presented to the third QIMIE workshop, 2013, Gold Coast, Australia