PIDT: A Novel Decision Tree Algorithm Based on Parameterised Impurities and Statistical Pruning Approaches

Daniel Stamate[1], Wajdi Alghamdi[1*], Daniel Stahl[2],
Doina Logofatu[3] and Alexander Zamyatin[4]

[1] Data Science & Soft Computing Lab, and Department of Computing, Goldsmiths, University of London, UK, Email d.stamate@gold.ac.uk.
[2] Department of Biostatistics & Health Informatics, Institute of Psychiatry, Psychology and Neuroscience, King's College London, UK.
[3] Department of Mathematics and Computer Science, Frankfurt University of Applied Sciences, Germany.
[4] Faculty of Informatics, Department of Applied Informatics, National Research Tomsk State University, Russia.

[*] Joint first author

Abstract. In the process of constructing a decision tree, the criteria for selecting the splitting attributes influence the performance of the model produced by the decision tree algorithm. The most well-known criteria such as Shannon entropy and Gini index, suffer from the lack of adaptability to the datasets. This paper presents novel splitting attribute selection criteria based on some families of parameterised impurities that we proposed here to be used in the construction of optimal decision trees. These criteria rely on families of strict concave functions that define the new generalised parameterised impurity measures which we applied in devising and implementing our PIDT novel decision tree algorithm. This paper proposes also the S-condition based on statistical permutation tests, whose purpose is to ensure that the reduction in impurity, or gain, for the selected attribute is statistically significant. We implemented the S-pruning procedure based on the S-condition, to prevent model overfitting. These methods were evaluated on a number of simulated and benchmark datasets. Experimental results suggest that by tuning the parameters of the impurity measures and by using our S-pruning method, we obtain better decision tree classifiers with the PIDT algorithm.

Keywords: Machine Learning, Decision trees, Parameterised impurity measures, Concave functions, Optimisation, Preventing overfitting, Statistical pruning, Permutation test, Significance level

1 Introduction

The decision tree algorithm is a highly efficient algorithm used in machine learning and data mining; the model the algorithm produces is easy to understand and interpret, and

the algorithm offers accurate results in abbreviated time. Different versions of the decision tree algorithm have been introduced in the last few decades, and it remains an attractive research domain within the field of machine learning. Such algorithms are useful in numerous contexts within pattern recognition and machine learning applications. In the medical field, for instance, decision trees have been employed to diagnose heart disease patients [1] and to predict patients who may suffer from psychosis [2].

A decision tree algorithm simulates a tree assembly [3]. A decision tree consists of nodes that are connected via branches. The decision tree begins with a single root node and ends with a number of leaf / decision nodes; the nodes in between are the internal nodes.

In classification trees, each leaf node is labelled with a particular class. Each node that is not a leaf node applies a test on a certain attribute, and each branch represents a result of the test. The nodes are selected from the top level based on the attribute-selection measure [4]. For example, ID3 algorithm [5] and its extended version C4.5 [4] use information gain (which is based on Shannon entropy) to construct the decision tree; the element with the highest gain is taken as the root node, and the dataset is divided based on the root element values. Again, the information gain is calculated for all the internal nodes separately, and the process is repeated until leaf nodes are reached.

Unlike most machine learning algorithms, decision trees perform local feature selection on different sets of features. The selected feature should be the feature that reduces the uncertainty at the node the most [6]. The dataset may then be partitioned accordingly into sub-nodes. This procedure is applied recursively until it meets any stopping criterion, such as the minimum number of instances or the maximum tree depth. Choosing the splitting and stopping criteria are two open problems in decision tree algorithms.

To address the first issue, many decision tree algorithms have proposed different impurity measures as a splitting criterion. Most decision tree algorithms are based on the information gain function for choosing the best attribute for splitting the data at each node that is not a leaf node. For instance, the ID3 and C4.5 algorithms are based on Shannon entropy [6], while the classification and regression tree CART algorithm is based on the Gini index [7]. However, one drawback in this kind of approach is that these types of impurity measures are only based on one fixed concave function for assessing the impurity in the datasets' class distributions, which means they suffer from a lack of adaptability to various datasets.

Many studies have investigated the importance of the split criterion [8], [9]. These studies have concluded that the choice of impurity measure does have some influence on the decision tree's efficacy. Inspired by these studies, we have proposed several novel splitting criteria based on parameterised families of strict concave functions that may be used as impurity measures. As such, we propose new parameterised impurities including parameterised entropy (PE), parameterised Gini (PG), parameterised Tsallis (PT), parameterised Renyi (PR), as well as parameterised AlphaBeta impurity (ABI) and parameterised GiniEntropy (GE) impurity. Their purpose will consist of being mostly reduced in a node after a split, which will dictate the choice of the most suitable attribute in that node. These methods indeed provide an innovative approach to improved decision tree performance, as this work shows.

As for the second problem, most practical decision tree implementations use a 'greedy' approach to grow the tree. Such algorithms would usually suffer from overfitting the dataset [3], and additional mechanisms are needed to be put in place to prevent this. Several stopping criteria have been introduced to overcome this issue, such as setting the minimum value of the information gain to grow the tree with a C4.5 algorithm for instance [4]. A number of recent papers have used permutation tests for different machine learning problems, such as studying the classifier performance [10], or in the feature selection process [11]. With the model overfitting problem in mind, we proposed in this paper the S-condition based on statistical permutation tests, whose purpose is to ensure that the reduction in impurity, or gain, for the selected attribute in a node of the decision tree is statistically significant, and that the observed gain is unlikely to be at least that high just by chance. Moreover, we implemented the S-pruning procedure based on the S-condition, to prevent model overfitting.

We integrate the use of our novel families of parameterised impurities for the attribute selection, with the S-pruning procedure, and with the optimisation of the parameters of the impurity via cross-validation according to the accuracy performance, in a new decision tree algorithm that we call PIDT, whose name stands for Parameterised Impurity Decision Tree.

The rest of this paper is organised as follows. Section 2 introduces the mathematical formulations and the general requirements for the impurity measures, as well as the novel parameterised impurity measures that we propose to be used in selecting the splitting attributes in our PIDT algorithm. Section 3 introduces our S-condition and S-pruning procedure based on permutation tests, which enhance the PIDT algorithm to prevent model overfitting. Section 4 experimentally investigates the proposed parameterised impurity measures and compares them with conventional impurity functions, based on the performances obtained by the PIDT and conventional decision tree algorithms on a number of benchmarks and generated datasets. Finally, section 5 presents conclusions and offers directions for future work.

2 Impurity measures

As mentioned above, a decision tree algorithm splits the dataset sample (at each node that is not a leaf node) into two or more sets based on the attribute that scores the highest gain (i.e. reduction in impurity) [12]. In the previous section, we mentioned two conventional impurities mostly used in decision tree algorithms, namely Shannon entropy and Gini index. But there are also other impurities which are presented in the literature such as Tsallis [13], and Renyi [12]. A different work proposed also a generalisation of the conditional entropy [14]. Considering these different studies based on various impurity measures suggests that the choice of the impurity measure influences the decision tree's effectiveness. In the following sub-sections, we provide the mathematical formulations of and the criteria for functions defined on discrete probabilistic distributions, to be impurity measures.

2.1 Mathematical formulations

Let X be an $n \times m$ data matrix. We denote the r-th row vector of X by X_r, and the c-th column vector of X by X^c. Rows are also called *records* or *data points*, while columns are also called *attributes* or *features*. Since we do not restrict the data domain of X, the scale of this domain's features can be categorical or numerical. For each data point X_r, we have a class label y_r. We assume a set of known class labels Y, so $y_r \in Y$. Let D be the set of labelled data $D = \{(X_r, y_r)\}_{r=1}^{n}$. During the classification task, the goal is to predict the labels of new data points by training a classifier on D. Now, let k be the total number of data entries in a node, and k_i be the number of data entries classified as class i. Then $p_i = {}^{k_i}/_{k}$ is the ratio of instances classified as i and estimates the probability of class i in the dataset in that node.

The primary purpose of the impurity measures is to express the degree of mixture of various classes in a dataset and then to help to define how well the classes are separated via a split in a node. As such, in general, an impurity measure should satisfy specific requirements. Breiman [8] suggested that an impurity measure is a function ***Imp*** whose argument is a vector of probabilities from a discrete probability distribution (given by the class proportions in a dataset), which satisfies the following properties:

Property A: Strict concavity $Imp'' < 0$.

Property B: Maximality $Imp' = 0 \; for \; (p_i = {}^{1}/_{k}) \; for \; i = 1, \ldots, k$.

Property C: Minimality $Imp = 0 \leftrightarrow \exists i \, | pi = 1$.

These properties state that the impurity function should be a strictly concave function; they also express what the maximum and minimum points of the function are. Both Shannon entropy and Gini index, which are defined below, meet the impurity-based criteria:

$$\text{Entropy (D)} = \text{E(D)} = -\sum_{i=1}^{k} p_i * \log(p_i) \tag{1}$$

$$\text{Gini (D)} = \text{G(D)} = 1 - \sum_{i=1}^{k} p_i^2 \tag{2}$$

Several authors compared the behaviour of Gini index and Shannon entropy to determine which performs better; they concluded that it is not possible to decide which one leads to higher accuracies of the produced decision trees since the two measures have only about 2% disagreement in most cases [9]. Note that both Gini index and Shannon entropy are based on one strict concave function each, and as such they might not have the flexibility in adapting to various datasets. We have also considered Renyi entropy and Tsallis entropy, both of which generalising Shannon entropy. They are described by the following formulas, respectively:

$$\text{Renyi (D)} = \text{R(D)} = \frac{1}{1-\gamma} * \log(\sum_{i=1}^{k} p_i^{\gamma}) \qquad \text{where } \gamma > 0 \text{ and } \gamma \neq 1 \tag{3}$$

$$\text{Tsallis (D)} = \text{T(D)} = \frac{1-\sum_{i=1}^{k} p_i^{\gamma}}{1-\gamma} \qquad \text{where } \gamma > 0 \text{ and } \gamma \neq 1 \qquad (4)$$

In the next subsection, we propose several families of generalised parameterised impurity measures based on the requirements suggested by Breiman [8] and outlined above, and we introduce our new PIDT algorithm employing these impurities.

2.2 Parameterised impurity measures

As mentioned, the novel parameterised impurity measures that we propose in what follows, are used to select the attribute that mostly reduces the impurity by splitting the dataset in a node of the decision tree.

Our first proposed family of parameterised impurities is the parameterised entropy PE, which is formulated below, and is illustrated in Figure 1 for the case of the 2 class problems (the x-axis represents the probability of one class).

$$\text{PE (D)} = \text{E(D)}^{\alpha} \qquad \text{where } \alpha \in (0, 1] \qquad (5)$$

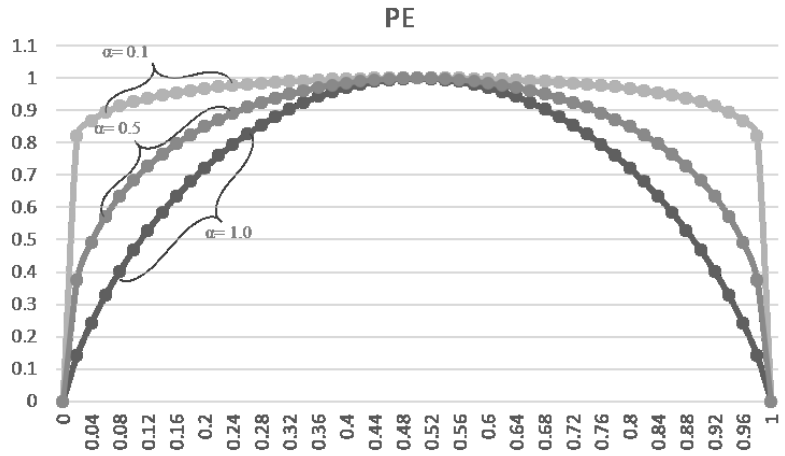

Fig. 1: Parameterised entropy (PE) with different values for α.

The interval of variation for the parameter α, i.e. (0,1], was chosen to allow, on the one hand, a large diversity of shapes of the graph of the impurity PE, and on the other hand, to mathematically ensure the concavity of the impurity (proof not included here due to lack of space). The other requirements inspired by Breiman's work [8], to which we referred in the previous subsection, are also met.

Figure 1 illustrates the impact of α on the shape of the PE curve. In particular, α =1 corresponds to the conventional Shannon entropy, while smaller positive values for α

have an effect of diminishing the curvature of the PE curve around its middle (the second derivative's absolute value tends to decrease in that area), and of gradually transforming the curve and make it tends to a plateau for small values of the parameter (for illustration see the curve for $\alpha = 0.1$ in Figure 1). Intuitively, these changes in the shape of the PE curve suggest potential changes in choosing attributes in a split node of the decision tree, and this was confirmed experimentally when we implemented our framework. This situation happens because the process may give preference to different class probability distributions in the data subsets that are issued from the split. Parameter α clearly influences which splits will be created in the decision tree, and as such it influences the model learnt from the data and allowed it to have more flexibility in adapting to the data than in the case of a fixed impurity such as the conventional Shannon entropy.

In the same manner, parameterised Gini, parameterised Renyi, and parameterised Tsallis are defined by using the following formulas:

$$\mathrm{PG}\,(\mathrm{D}) = \mathrm{G}(\mathrm{D})^{\alpha} \qquad \text{where } \alpha \in (0, 1] \tag{6}$$

$$\mathrm{PR}\,(\mathrm{D}) = \mathrm{R}(\mathrm{D})^{\alpha} \qquad \text{where } \alpha \in (0, 1] \tag{7}$$

$$\mathrm{PT}\,(\mathrm{D}) = \mathrm{T}(\mathrm{D})^{\alpha} \qquad \text{where } \alpha \in (0, 1] \tag{8}$$

Note that since the concave functions that define the conventional Shannon entropy and Gini index are generalised by the proposed families of parameterised impurities *PE* and *PG* respectively, the use of these families of impurities is expected, roughly speaking, to produce comparable or better decision trees in most cases than those based on the conventional entropy and Gini index.

We now define two more families of parameterised impurities based on two parameters α and β this time.

$$\mathrm{GE}\,(\mathrm{D}) = \mathrm{G}(\mathrm{D})^{\alpha} + \mathrm{E}(\mathrm{D})^{\beta} \qquad \text{where } \alpha \text{ and } \beta \in (0, 1] \tag{9}$$

$$\mathrm{ABI}\,(\mathrm{D}) = \sum_{i=1}^{k} p_i^{\propto} * (1 - p_i)^{\beta} \qquad \text{where } \alpha \text{ and } \beta \in (0, 1] \tag{10}$$

Note that *GE* combines arbitrary positive and not larger than 1 powers of the Gini index and of the conventional Shannon entropy, generalising these impurities, and offering further flexibility by using two parameters. By the use of the two parameters, *ABI* family generalises the Gini index and also offers further flexibility in expressing various shapes of impurity. Note also that both *GE* and *ABI* fulfil, mathematically speaking, the requirements of impurity inspired by Breiman [8] (proof omitted here due to lack of space).

Figure 2 illustrates, for the case of 2 class problems, the parameterised families of impurities PE and PG for various values of parameter α (see the top half), and the parameterised family of impurities GE for various values of parameters α and β (see the bottom half).

The above parameterised impurity families are used in our novel decision tree algorithm which we call PIDT, whose name stands for Parameterised Impurity Decision

Trees. In particular, the impurities define the criterion for selecting the best attributes in the nodes of the decision tree, based on the largest decrease in impurity from the dataset in the parent node to the datasets in the child nodes. This difference is the so-called gain, and will be precisely defined in the next section when the statistical S-condition will be introduced. The PIDT algorithm uses one single selected family of parameterised impurities for a tree induction, and optimises the parameters of the impurity in a cross-validation fashion with respect to the accuracy performance.

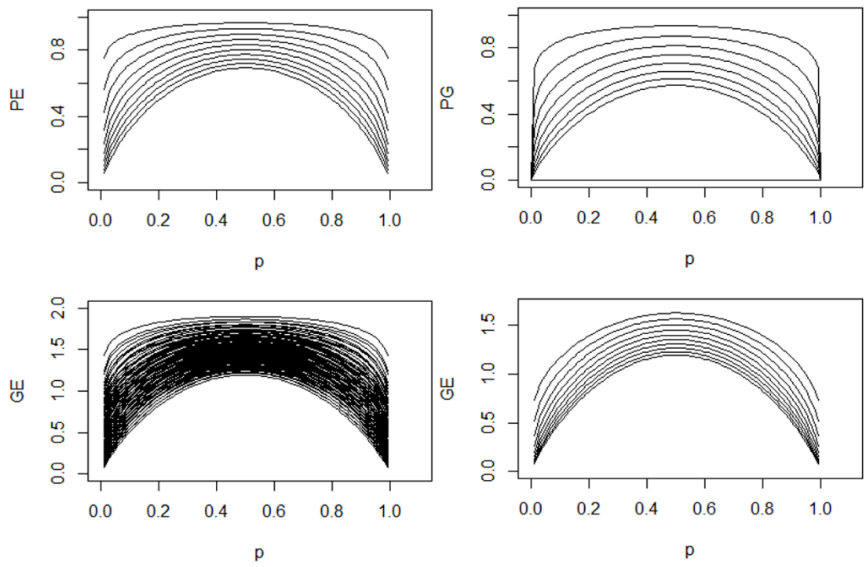

Fig. 2: Novel parametrised impurity measures PE, PG (top), and GE (bottom)

In the next section, we develop an enhancement of the process of growing the decision tree with the PIDT algorithm, based on a novel statistical pruning procedure S-pruning that we introduce here as a useful tool to prevent overfitting problems.

3 S-pruning

Roughly speaking, the novel S-pruning procedure we describe here terminates some of the branches of the decision tree based on the outcome of a statistical test. In particular, this pruning method only allows the attributes that have a significant predictive power to split the node and grow the tree. Stopping the development of a branch is based on a certain condition, named here the S-condition.

S-condition:

Let X^c be the attribute with the highest gain G in a node N. Roughly speaking, G is expressed by the reduction in impurity after the split with attribute X^c in the node N. More precisely, the gain is defined in the same way as the information gain for the conventional Shannon entropy in C4.5 algorithm [4]. The impurity is measured in the dataset before the split, and in the resulting data subsets for each child after the split. The impurities in all these data subsets are averaged with weights derived as the fractions represented by the data subsets out of the dataset before the split. The impurity weighted average is then subtracted from the impurity of the dataset before the split, and the result defines the gain G mentioned above. The gain is non-negative for all attributes due to the concavity property of the impurity. Moreover, a higher gain may indicate a higher predictive power for an attribute. However, we want to ensure that a higher gain does not occur by chance. The S-condition defined here is a statistical mechanism to check this.

Let D be the dataset in node N. Shuffle (i.e. randomly permute) the labels in dataset D and measure again the gain for X^c. Do this t times so that a vector V of t gain values is built. *The S- condition is satisfied if and only if G is smaller than the q quantile of vector V*. When the S-condition is satisfied, the branch in node N stops growing and N becomes a terminal node. *This defines the S-pruning procedures*.

Overall, the logic behind the S-condition is that if the gain G is smaller than the q quantile (for instance for a value q such as 0.95 or 0.9) of a vector V of t gain values (for instance $t = 1000$) obtained for X^c using random labels (since they are shuffled or randomly permuted), then X^c is not considered to have predictive power according to the data D in that node N. The values of t and $s = 1\text{-}q$ must be specified by the user, where t is the number of label permutations (and thus equal to the number of gain values collected), and the value of s is the significance level (such as in the statistical tests). A smaller s will encourage more pruning. Intuitively, s indicates how likely the gain of the selected attribute X^c would have been acceptably high just by chance. Another relevant quantity here is the p-value, defined experimentally as the fraction of cases in which the gain obtained with the random labels was higher than or equal to the gain obtained with the original labels of the records in D. Therefore, if the p-value is small enough (e.g. the p-value is smaller than or equal to the significance level $s = 0.1$ or 0.05), then we can say that the gain of the selected attribute in the original data is indeed significantly better and, in consequence, that the gain is too high to have occurred just by chance. That is, the null hypothesis of the permutation test is rejected in this case. As such the attribute X^c is considered to have significant predictive power, and the split takes place. Note that the S-condition does not hold in this case.

On the other hand, if the p-value is larger than the significance level s, or in other words the S-condition holds, this means that the gain for the selected attribute is not large enough to indicate predictive power, so the development of that branch is stopped.

Note also that higher q (or equivalently smaller s) results in oversimplified trees, whereas the opposite results in reduced pruning and larger trees. As a result of using the S-pruning procedure, fewer nodes are expanded during the building phase, and thus

constructing the decision tree is simplified. In addition, the decision tree has the advantage of avoiding overfitting while it is being built.

4 Comparison of decision tree classifiers with various impurity measures

We now compare several impurity measures with respect to their impact on the decision tree induction, including the conventional impurities such as Shannon entropy and Gini index, and also the new parameterised families of impurities introduced here. We argue that the conventional impurities mentioned above have their flexibility limitations when used with various datasets. We also argue that, due to their flexibility, the parameterised families of impurities are better suited for the purpose of class separation. We also test our novel S-pruning procedure introduced in the previous section. Finally, we demonstrate empirically that the proposed PIDT algorithm indeed produces better decision trees than the algorithms that use simply the conventional entropy and Gini index impurity measures.

This section also investigates the performance of decision trees as a result of parameter optimisation. In order to investigate the usefulness of the novel parameterised impurity functions, we tested them on different datasets and compared them with the conventional impurities mentioned above. In order to optimise the parameters of an impurity family, a grid search over a parameter space with 5-fold cross-validation, were used to select the best parameters' values.

4.1 Experimental analysis

We chose the open-source library Weka (Waikato Environment for Knowledge Analysis) [17] as a starting point in implementing our PIDT algorithm with the S-pruning method option, and parameter optimisation for the families of parameterised impurities above. In particular, the tree builder code was modified and extended to support the conventional impurities Shannon entropy, Gini index, as well as Tsallis, and Renyi, and of course we implemented also the new families of parameterised impurity measures introduced in this paper. The S-pruning method was also added. The PIDT software allows users to specify the family of impurities and values for their relevant parameters, or choose the optimisation of these parameters. It also allows specifying the significance level s and the number of permutations t when the S-pruning method is enabled.

Each experiment used 5-fold cross-validation and was performed with and without the S-pruning method. Finally, the minimum number of nodes was set to 7 in all experiments. Each of the techniques was applied to 7 datasets, of which 5 were real datasets and 2 were simulated datasets with different characteristics.

The real datasets from the University of California–Irvine (UCI) machine learning repository [18] that were provided to illustrate the performance of different impurity measures, included the diagnostic Wisconsin breast cancer dataset, the diabetes dataset, the glass identification dataset, and a medical dataset for hepatitis and primary tumours [19]. Two datasets were also generated using simulation techniques, in particular based

on Guyon's proposed approach employed in various researches [19, 20, 21, 23]. The simulated datasets contain a few thousand samples and different numbers of classes.

The PIDT algorithm was run for different impurity measures and values for α, β, γ parameters (whichever apply), and significance level *s*. The parameter space for α and β was 0.05, 0.1, ..., 0.95, 1.0; for γ the values were 0.1, 0.2, … ,0.9,1.5, 2.0, …, 5.0; and the considered significance level *s* values were 0.01, 0.05, and 0.1. Finally, the best-performing models with their parameters were chosen for the final comparison on the separate test datasets. Table 2 shows a summary of the models built with the chosen optimised parameters, while Table 1 provides the summary of the models built by using conventional impurities. Bold fonts in Table 2 show the best results scored regarding the chosen dataset. The results demonstrate that the parameterised entropy (PE) could be used to construct more efficient decision trees compared with the conventional entropy impurity and Gini index impurity. In particular, PE led to better results when it was applied with the S-pruning method on most datasets. By looking at Table 1 and Table 2, we observe that the accuracy generally improved, and the number of nodes decreased for the models produced by the PDIT algorithm.

In particular, it is interesting to observe that the accuracy tended to improve depending on the dataset, thus confirming that this performance could be affected by the method used for selecting attributes during the tree construction. In terms of tree size, this was diminished for most datasets. The best reduction was achieved for the Pima diabetes database, where the size of the tree was reduced ten times compared to the standard tree algorithm – which used entropy (as shown in Table 1) – and was comparable to the tree size discussed in [14]. We also note that our results for the hepatitis dataset produced more accurate and smaller tree compared to the results presented in [14]. Overall, PE and PR impurities, in conjunction with activating the S-pruning procedure, produce more accurate results and yield much smaller trees for most of the datasets.

Table 1: Assessing decision trees built with conventional impurity performances

Dataset	Decision tree with entropy		Decision tree with Gini	
	Accuracy	No. nodes	Accuracy	No. nodes
Breast cancer	0.654	67	0.654	76
Pima diabetes	0.736	119	0.724	135
Hepatitis	0.807	21	0.794	25
Primary tumour	0.434	60	0.363	57
Glass	0.626	39	0.556	55
Simulated data 1	0.721	33	0.668	67
Simulated data 2	0.612	188	0.601	157

Table 2: Assessing decision trees built with the PIDT algorithm with parameter optimisation, and with and without S-pruning procedure activated. "-" means values do not apply.

Dataset	PIDT								
	Accuracy	No. nodes	Parameters						
			Impurity	α	β	γ	S-pruning	*s*	Permutations
Breast cancer	**0.731**	91	PG	0.5	-	-	no	-	-
	0.720	**29**	PR	1	-	0.5	yes	0.05	1000
Pima diabetes	0.734	**11**	PE	0.5	-	-	yes	0.05	1000
Hepatitis	**0.839**	23	PE	0.3	-	-	no	-	-
	0.807	**7**	PE	0.3	-	-	yes	0.05	1000
Primary tumour	0.434	60	PE	1	-	-	no	-	-
Glass	**0.636**	**27**	PE	0.6	-	-	no	-	-
Simulated data 1	0.721	**7**	PE	0.8	0	0	yes	0.05	1000
Simulated data 2	**0.693**	**157**	GE	1	0.4	-	no	-	-

5 Conclusion and directions for future work

This paper proposed and tested an approach to building optimised classification trees using novel parameterised impurity measures which generalise conventional impurities such as Shannon entropy and Gini index. The experiments were conducted on five real datasets as well as on two simulated datasets. The results show that by building decision trees using parameterised impurity measures with optimal values for their parameters, the predictive models primarily led to better performance in terms of accuracy, than those built with traditional entropy impurity and Gini impurity.

A novel S-pruning method based on permutation tests was also introduced here to overcome the overfitting problem and to produce smaller decision trees. The proposed impurity measures gained significance and produced much smaller trees when they were applied with the S-pruning procedure enabled. However, if the significance level *s* for S-pruning is set too small, it may result in oversimplified trees.

One direction of extending this work is related to investigating novel impurity measures with flexibility capabilities in adapting to and working well with class-unbalanced problems. This direction is currently under investigation.

References

1. Shouman, M., Turner, T., et al.: Using decision tree for diagnosing heart disease patients, Proc. of 9th Australasian Data Mining Conference, pp. 23–30, (2011).
2. Alghamdi, W., Stamate, D., et al.: A Prediction Modelling and Pattern Detection Approach for the First-Episode Psychosis Associated to Cannabis Use, Proc. of 15th IEEE International Conference on Machine Learning and Applications, Anaheim, CA, pp. 825-830, (2016).
3. Han, J., Kamber, M., Pei, J.: Data Mining Concepts and Techniques, pp. 279-328, (2011).
4. Witten, I., Frank, E., et al.: Data mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann Publisher, (2016).
5. J. Quinlan: Induction of decision trees, Machine Learning, pp. 81– 106, (1986).
6. Tan, P., Michael, S., Vipin, K.: Introduction to Data Mining, (2005).
7. Breiman, L., Friedman, J., et al.: Classification and regression trees, Machine Learning, (1984).
8. Buntine, W., Niblett, T.: A further comparison of splitting rules for decision-tree induction, Machine Learning, pp. 75–85, (1992).
9. Liu, W., White, A.: The importance of attribute selection measures in decision tree induction, Machine Learning, (1994).
10. Ojala, M., Garriga, G.: Permutation tests for studying classifier performance, Journal of Machine Learning Research, vol. 11, pp. 1833–1863, (2010).
11. Good, P.: Permutation tests: a practical guide to resampling methods for testing hypotheses; Springer series in statistics, Springer, vol. 2nd, (2000).
12. Maszczyk, T., Duch, W.: Comparison of Shannon, renyi and tsallis entropy used in decision trees," Artificial Intelligence and Soft Computing–ICAISC 2008, pp. 643–651, (2008).
13. Raileanu, L., Stoffel, K.: Theoretical comparison between the gini index and information gain criteria," Annals of Mathematics and Artificial Intelligence, vol. 41, p. 7793, (2004).
14. Tsallis, C., Mendes, R., et al.: The role of constraints within generalised non-extensive statistics," Physica 261A, pp 534–554, (1998).
15. Jaroszewicz, D., Szymon, S.: A Generalization of Conditional Entropy, (2018).
16. Kuhn, M., Johnson, K.: Applied Predictive Modelling. Springer, (2013).
17. Frank, E., Hall, M., Witten, I.: The WEKA Workbench, Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, Fourth Edition, (2016).
18. UCI machine learning repository: Datasets, https://archive.ics.uci.edu/ml/datasets.html, 2017-1-1.
19. Guyon, I., Li, J., Mader, T.: Competitive baseline methods set new standards for the nips 2003 feature selection benchmark, Pattern recognition letters, vol. 28, no. 12, pp. 1438–1444, (2007).
20. Guyon, I.: Design of experiments of the nips 2003 variable selection benchmark, (2003).
21. Guyon, I., Gunn, S., et al.: Result analysis of the nips 2003 feature selection challenge, in Advances in neural information processing systems, 2005, pp. 545–552, (2005).
22. Osman, H.: Correlation-based feature ranking for online classification, in Systems, Man and Cybernetics, 2009, IEEE International Conference on. IEEE, pp. 3077–3082, (2009).
23. Guyon, I., Elisseeff, A: An introduction to feature extraction, Feature extraction, pp. 1–25, (2006).