Interpretable Decisions Trees via Human-in-the-Loop-Learning

Interpretable decisions trees via human-in-the-loop-learning

Vladimir Estivill-Castro[1][0000−0001−7775−0780], Eugene Gilmore[2], and René Hexel[2][0000−0002−9668−849X]

[1] Universitat Pompeu Fabra, Barcelona 08018, Spain vladimir.estivill@upf.edu
[2] Griffith University, Nathan 4111, Australia
eugene.gilmore@griffithuni.edu.au, r.hexel@griffith.edu.au

Abstract. Interactive machine learning (IML) enables models that incorporate human expertise because the human collaborates in the building of the learned model. Moreover, the expert driving the learning (human-in-the-loop-learning) can steer the learning objective, not only for accuracy, but perhaps for discrimination or characterisation rules, where isolating one class is the primary objective. Moreover, the interaction enables humans to explore and gain insights into the dataset, and to validate the learned models. This requires transparency and interpretable classifiers. The importance and fundamental relevance of understandable classification has recently been emphasised across numerous applications under the banner of explainable artificial intelligence. We use parallel coordinates to design an IML system that visualises decision trees with interpretable splits beyond plain parallel axis splits. Moreover, we show that discrimination and characterisation rules are also well communicated using parallel coordinates. We confirm the merits of our approach by reporting results from a large usability study.

Keywords: Interactive Machine Learning · Human-In-The-Loop-Learning · Parallel Coordinates · Explainable AI · Characteristic rules.

1 Introduction

Humans' trust in expert systems has required explanations in human understandable terms [5]. It could be argued that machine learning (ML) was fuelled by the need to transfer human expertise into decision support systems and to reduce the high cost of knowledge engineering. *"It is obvious that the interactive approach to knowledge acquisition cannot keep pace with the burgeoning demand for expert systems; Feigenbaum terms this the 'bottleneck problem'. This perception has stimulated the investigation of ML as a means of explicating knowledge"* [19]. Early reviews on the progress of ML show that the understandability (then named comprehensibility) of the classification was considered vital. *"A definite loss of any communication abilities is contrary to the spirit of AI. AI systems are open to their user who must understand them"* [14]. There is so much to gain by incorporating Human-In-the-Loop-Learning (HILL) in ML tasks, for

instance validation or new knowledge elicitation [7, 26]. Today, combining fast heuristic search for classifiers and HILL has received the name of IML [1] because not only are datasets the source of knowledge, but IML also captures the human experience [8]. Supervised learning applications offer extraordinary predictive power but have sacrificed transparency and interpretability of the predictions. Now, new criteria besides predictive accuracy are considered [9, 11]; particularly in drought analysis [6] medicine, credit scoring, churn prediction, and bio-informatics [10].

Convolutional Neural Networks (CNNs) are regarded as superior for object classification, face recognition, and automatic understanding of handwriting. Similarly, Support Vector Machines (SVMs) are considered extremely potent for pattern recognition. CNNs, ensembles, and SVMs output "black box" models, since they are difficult to interpret by domain experts [9, 17]. Thus, delivering understandable classification models is an urgent research topic [9, 11]. The most common approach is the production of accurate black-box models followed by extracting explanations [22]. There are two lines of work for delivering explainable models. (1) to build interpretable surrogate models that learn to closely reproduce the output of the black-box model, while regulating aspects such as cluster size for explanation [3]. (2) to produce an explanation for classification of a specific instance [21] or to identify cases belonging to a subset of the feature space where descriptions are suitable [16]. However, there are strong arguments that suggest that real interpretable models must be learnt from the beginning [20].

Learning decision trees from data is one of the pioneer methods that produce understandable models [9, 13]. Decision tree learning is now ubiquitous in big data, statistics, and ML. C4.5 (a method based on a recursive approach incorporated into CLS and ID3) is first among the top 10 most used algorithms in data mining. CART (Classification and Regression Trees) is another decision-tree learning method among the top 10 algorithms in data mining. Parallel coordinates (PC) [12] have been used for HILL, but there are few empirical evaluations of its merits. Perhaps the most relevant is an in-depth evaluation of the WEKA [25] package for IML which, however, does not consider the use of PC.

We incorporate PC for exploration of datasets and HILL. We deploy a prototype for the IML of decision-tree classifiers (DTCs). We discuss how this prototype exhibits improvements over many other HILL systems. We emphasise that our prototype enables (1) understanding of learnt classifiers, (2) exploration and insight into datasets, and (3) meaningful exploration by humans. We present how PC can provide a visualisation of specific rules and support the operators' interaction with the dataset to scrutinise specific rules. This enables the construction of characterisation and discrimination rules, which focus on one class above the others. We show that users gain understanding through visualisation by presenting results of a detailed usability study.

In Section 2, we review salient HILL systems, where learning classifiers involves dataset visualisations. The advantages of using PC for accuracy have been highlighted elsewhere [15]. But, our review of HILL systems reveals that there is almost no experimental evaluation of the effectiveness of HILL. The largest study

is a reproduction with 50 users, while the original WEKA `UserClassifier` paper reported a study with only 5 subjects [25]. Section 3 explains our algorithms and system for HILL. We provide key details of our study that consists of three experiments in Section 4. Then, Section 5 reports results with over 100 users on our proposed system. We highlight how our system overcomes a number of the shortcomings of the HILL systems we review in Section 2.

2 Learning classifiers involving dataset visualisations

Perhaps the earliest system to profit from the interpretability of decision trees for HILL was `PCB`'s visualisation of an attribute as a coloured bar [2]. This bar is constructed such that each instance of the sorted attribute values is a pixel, coloured corresponding to its class. Users shall visually recognise clusters of a class on an attribute. A DTC is visualised by showing bars with cuts to represent a split on an attribute. Each level of the tree can then be shown as subsets of an attribute bar with splits. A user can participate in learning the tree using this visualisation by specifying where on a bar to split an attribute. The HILL process has some algorithmic support to offer suggestions for splits and to finish subtrees. This visualisation appears particularly effective at showing a large dataset in a way that does not take much screen real-estate. However, the bar representation removes important human domain knowledge; for instance all capability to see actual values of attributes (or the magnitude of difference in values) disappears. This blocks experts from incorporating their knowledge. Moreover, the bar representation restricts classification rules to tests consisting of strictly univariate splits. There is no visualisation of attribute relationships (correlations, inverse correlations, or oblique correlations).

Nested Cavities (NC) [15], is an approach to IML, based on PC [12]. Unlike most other visualisation techniques, PC scale and are not restricted to datasets with a small number of dimensions. Parallel coordinates with 400 dimensions have been used. More attributes are displayed by packing their axes on the side. But decisions being based on over 100 variables are hardly interpretable and understandable. Thus, our prototype uses PC and ML metrics to recommend attributes (and their order) in a visualisation. The operator still can select their preferred number of parallel axes to display. The construction of classifiers with NC is similar to decision-trees, because both approaches follow recursive refinement that results in a decision-tree structure. But, to the best of our knowledge, there are no user-focused evaluations of IML with NC.

When using star-coordinates for dataset visualisation and DTC construction [23, 24], each attribute is drawn as an axis on a 2D plane starting from the centre and projected outwards. However, star-coordinates displays suffer similar drawbacks as bar visualisations: users are unable to find subsets of predictive attributes, or ways to discriminate classes. In star-coordinates, the location for visualisation of an instance depends on the value of all attributes, making it impossible to identify boundaries between classes. In contrast, with PC, such separations are readily apparent. With star coordinates, experts cannot explore and interchange attributes with other attributes, even if aware of subsets of

predictive attributes. Users can only chose a projection emphasising influential attributes, losing any insight of one attribute's interaction with other attributes. There is no natural interaction with the star-coordinates visualisation where a user can also determine exactly what attribute or attributes are contributing the most to the position of a point in the visualisation. `PaintingClass` [24] extends `StarClass` [23] so the expert can use visualisations of categorical attributes with PC. But the restriction persists for numerical attributes. `PaintingClass` uses evenly distributed categorical values of a categorical attribute along a PC axis. This produces a visualisation with unintended bias. Because `PaintingClass` provides no ML support, and building the classifier is completely human-driven, it could be argued that it is not HILL.

`iVisClassifier` [4] profits from PC, but, to reduce the attributes presented to the user, the dataset is presented after using linear discriminant analysis (LDA). The visualisation uses only the top LDA vectors. But using these new LDA features blocks the user's understanding of the visualisation since each LDA feature is a vector of coefficients over all the original attributes (or dimensions). Heat-maps are displayed in an attempt to provide interpretation of the component features, but they could only possibly have some semantics in the particular application of front-human-portrait face-recognition.

Some empirical evaluation is reported in WEKA's [25] `UserClassifier`. `UserClassifier` is a IML and HILL system for DTCs that shows a scatter plot of only two (user selectable) attributes at a time. A display of small bars for each attribute provides some assistance for attribute relevance. The attribute bar presents the distribution of classes when sorted by that attribute. The user can review the current tree as a node-link diagram in one display, select and expand a node. WEKA's `UserClassifier` is the only one reporting usability studies and it involved only five participants [25]. Later, it was evaluated with 50 university students who had completed 7 weeks of material on ML and DTCs. This study confirmed a number of limitations of the WEKA's `UserClassifier`.

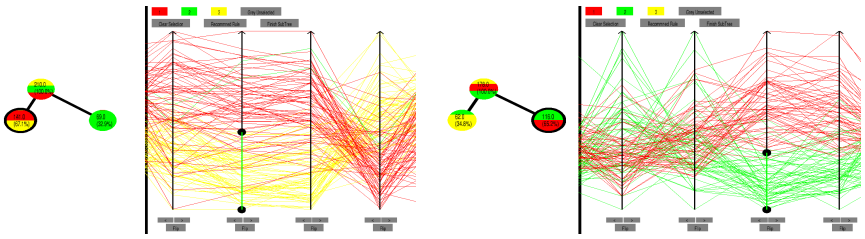
3 Experts iteratively construct decision trees

We propose a HILL system that addresses the shortcomings of earlier systems, because it uses PC to visualise the training set, the DTC, and also specific rules.

3.1 Using Parallel Coordinates

A PC visualisation draws a vertical axis for each dimension (each attribute). An instance $v = (a_1, a_2, \ldots, a_d)$ (a point in Cartesian coordinates) corresponds to a poly-line in PC that visits value a_n on the axis for attribute A_n (for $n \in \{1, \ldots, d\}$). For HILL, each class has a colour, and the labelled instances of the training set are painted using this colour. Figure 1 shows examples of PC visualisations and the corresponding partially-built decision-tree.

If there are many attributes, a window with a projection onto a subset of the attributes is displayed. On-line Analytical Processing (OLAP) tools enable



(a) Exploring the `seed` dataset. (b) Exploring the `wine` dataset.

Fig. 1: Two examples, each exploring a different dataset.

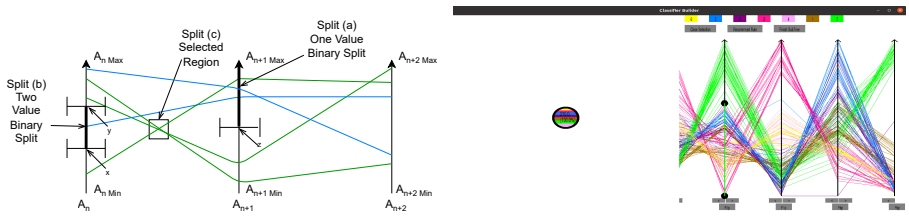
business intelligence practitioners to analyse multidimensional data interactively. Our use of colour concentration allows rapid selection and application of OLAP-type operations on the visible window. For instance, removing one attribute from visualisation and adding another one is a pivot operation. The information gain on an attribute or range within an attribute is used to suggest relevant attributes to the human operator. The user will interactively build a DTC following Hunt's recursive construction: the user picks a leaf-node T to refine the current rule that terminates at T. But, our system provides support for this growth of the tree from leaf T.

1. We colour the corresponding leaf T to illustrate the purity of T (this directly correlates with the classification accuracy of the rule terminating at T). For instance, the left leaf in the tree in Figure 1a indicates it contains an almost even split of two classes. However, the right leaf is practically pure. The depth of the leaf T inversely correlates with the applicability and generality of that rule terminating at T. Understandability and interpretability also inversely correlate with leaf depth.
2. The system allows the user to select whether to display values of predictability power of attributes, such as the information gain.

All trees are classifiers because the decision at a leaf is a Naïve Bayes decision.

3.2 The splits the user shall apply

A split on one axis alone is commonly a range and this is familiar to DTCs construction as this split belongs to C4.5. However, the PC visualisation allows an *oblique* split that involves two attributes. Thus, the oblique test is interpretable, particularly, because of the point-line duality in PC [12]. For instance, a rule that uses a point between two attributes in PC splits instances into two groups, those that closely follow some linear correlation between two attributes and those that do not. If we use a rectangular region for the split, then we regulate a margin for the above mentioned linear correlation. Figure 2a illustrates the types of splits users can introduce in our system to further a leaf and interactively refine a decision tree. Split (a) and split (b) are familiar from standard decision tree construction. But a rectangular split (c) is an excellent trade-off between interpretability and oblique splits. Besides the interpretability of the splits based on



(a) The splits that the user can apply. (b) A human would chose a better split.

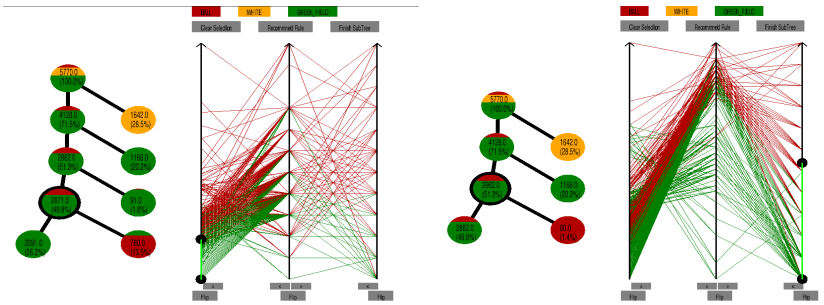
Fig. 2: Visualising decisions as splits.

rectangles between two parallel axes, these splits constitute a richer language to define DTCs than the standard splits of classical machine learning algorithms. And although not as powerful as the full oblique splits, this is appropriate, as full oblique splits are extremely hard to comprehend by humans. DTCs that use oblique splits are also called multivariate decision trees. Although learning multivariate decision trees results in shorter trees, they are rarely used because the test (and thus their rules) are incomprehensible by humans.

3.3 Information that supports interaction

Our HILL system supports interaction in several ways. For instance, when the user selects a node in the tree, the visualisation restricts the instances displayed in the PC-canvas to those that satisfy the splits of the selected node's ancestors. The user can elect the criteria for ordering attributes in the PC-canvas, which by default are sorted by a criteria of discriminative power, and among these, the default is information gain. When the user selects an attribute for the next split, the system also offers suggestions for the split on the axis, and diverse algorithms are available (again, information gain, gini-index, etc.). The user can opt to ask the system to propose a rectangular split. The use can accept or modify the rectangle suggested by the system. The automatic construction can be restricted to a node or to a sub-tree. As the user explores proposed sub-trees, the user gains an understanding of the attributes, and interaction between attributes. These algorithms support HILL and provide adequate balance between number-crunching and machine learning support, and user's intervention and interaction to incorporate human expertise, or for users to discover new insights in the data.

Human pattern spotting has a crucial role on the split selections. Figure 2b represents a setting where humans easily chose better splits than ML indicators (such as information gain, gini-index). More purifying are the second attribute that isolates the brown class and with a robust gap, and the fourth attribute, which also isolates the green class with also a wide gap. Thus, HILL can deliver better models. The user can also intervene when ML indicators are offering similar values, but some attributes are easier to capture or much more readily available.



(a) High recall DTC (b) High precision DTC

Fig. 3: Illustrations of the HILL systems used as a IML to build DTCs with either a high precision or high recall for the black class (coloured red).

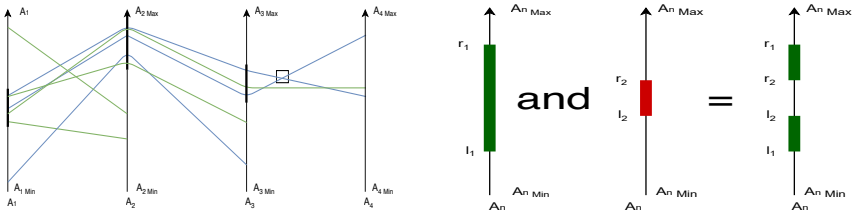
3.4 Visualising the Tree

Figure 1 and Figure 3 illustrate an interactive visualisation of the tree under construction with the PC-canvas. As we mentioned, the nodes are coloured with a histogram that informs the user about the number of instances of each class that reached the node. This swiftly informs the user of the purity (and thus accuracy) of rules reaching a node. The user can also obtain feedback information about the support (% of the training set instances that reach the node) and confidence (% of correct-class classification) for the rule at the node.

3.5 Visualising Rules

We showed how our HILL system uses PC to effectively assist a user in interactively constructing a DTC. We now demonstrate how we use PC to allow a user to understand a particular classification made by a learnt DTC.

Our approach here supports explorative data mining, where experts are seeking to find characteristic rules or discriminant rules. One of the features of DTCs is their ability to be converted into a decision list. Such a decision list is composed of a series of if-else rules that can be followed to determine the classification of a particular instance. We use a similar idea to visualise the decision path for a single leaf in a DTC. Instead of a textual representation of such a rule however, we can again use PC to graphically represent this decision path. We argue that PC are ideally suited to this task as we can use a series of axes to visualise each component of the rule in the one visualisation. Depending on the depth of the rule and available screen real estate, we may even be able to visualise the entire decision path. Not only this, in our graphical representation, we can visualise the training data and the effect that each component of the decision path has on the resulting subset of selected data. When used in conjunction with our visualisation of the entire DTC and its accompanying statistics for each node, we argue that this gives a human user a profound intuitive and interpretable explanation of a DTC's classification.



(a) Our HILL system visualises the decision path of a single leaf in a DTC.

(b) Disconnected sections on a single axis by condensing splits that include negation.

Fig. 4: Visualising rules.

Figure 4a shows an example of how we use PC to visualise the decision path to a leaf node in a DTC. Here the leftmost axis is used to represent the split of the root node in the DTC, which will always be the first component in any decision path. In this example, the split for the root node of the decision tree is a simple, single-attribute split on attribute A_1 and is visualised as such with the highlighted range. Between the first and second axes, the start of poly-lines for every instance in the training set is shown. From the second axis, only instances in the training set continue on that matched the split from the previous axis. This allows the user to clearly see what subset of data is selected by each split. The visualisation continues in this manner with poly-lines being terminated once they no longer match a split in the decision path. In this example, the final split in the decision path is a PC region split and is visualised using the last two axes.

Another advantage of visualising a leaf in this way is that users can assess the likelihood that a specific classification is accurate. While performance metrics such as accuracy, ROC, and F1 score capture the performance of the entire tree, it is possible that sections of the model are more (or less) accurate than others. Using this visualisation, users view the amount of the majority class that arrives at a leaf node. When looking at the classification of an instance, users can also see how close to the margins of each split that instance is.

Condensing the decision path In cases where an attribute (or attribute pair in the case of our PC region splits) appears more than once in a decision path, we have the opportunity to reduce the number of attributes required to display the decision path by taking the union of both splits. If using only a single-attribute split, we can create an upper bound on the number of axes required to visualise any decision path equal to the number of attributes in the dataset. When not condensing the decision path in this way, the ordering of axes in PC allows the user to see the depth of each rule in the DTC. For this reason, we provide the user with the option of turning this feature on or off in our HILL system.

Visualising negated splits Negated splits are important since half of all path components are negated splits, i.e. we traverse to the other child node that represents not matching the node's split. For single-attribute splits of the form

$l \leq A_n$ the negated split becomes $A_n < l$ and can be represented on a PC axis by simply swapping the highlighted versus the not highlighted region. For a single-attribute test with two split points of the form $l \leq A_n \leq r$, our negated split becomes $A_n < l \vee r < A_n$. This again swaps the highlighted region with the region not highlighted. Although the representation of this two-value split may seem complicated (particularly, when we consider condensing a decision path that contains the same attributes multiple times), swapping of highlighted regions is simplified using De Morgan Laws. Consider a decision path that includes two components using attribute A_n. The first component is of the form $l_1 \leq A_n \leq r_1$ and the second is the negated form of the split $l_2 \leq A_n \leq r_2$, i.e. $A_n < l_2 \vee r_2 < A_n$. Suppose we have the situation where $l_1 < l_2 \wedge r_1 > r_2$. In this case our condensed split becomes $l_1 \leq A_n \leq l_2 \vee r_2 \leq A_n \leq r_1$. To represent this condensed split on a PC axis, we now need to highlight multiple, disconnected sections of the axis. Figure 4b illustrates this situation. Here the green selection on the first axis and the negation of a selection (illustrated as the red region on the second axis) results in two selected ranges on the third axis. In a case where the same attribute is used many times in a decision path, the resulting axis consists of several highlighted disconnected sections (these are disjunctions of intervals). Interestingly, this effect is only possible when using tests containing two split points. We argue that in a HILL system it is only natural that a user will want to use these splits containing two split points to isolate certain sections of data and as such, our system supports visualisation of such condensed rules.

4 Design of the Usability Evaluation

We now report on the evaluation of our prototype by using the online surveying platform Prolific [18]. We obtain quantitative data on the effectiveness of our new visualisation techniques from 104 participants who performed timed tasks. Engagement with the survey requires reviewing 3 concepts applicable to HILL: 1) DTC 2) PC and 3) scatter-plot visualisations. This refresher practice ensures we can rank participants' expertise with ML and exclude anyone not fluent with DTC construction. Moreover, we used Prolific's capability to focus on the following demographics correlated with experience in ML.

- **Which of the following best describes the sector you primarily work in?** IT, Science, Technology, Engineering & Mathematics
- **What is your first language?** English
- **Which of these is the highest level of education you have completed?** Undergraduate degree (BA/BSc/other)
- **Do you have computer programming skills?** Yes

Our survey consists of 3 different experiments. The platform offers participants interaction (and animations) as if they were performing tasks in a system for HILL. Not only does the system display videos and allows users to click on images, and experiment, while moving forward and back through explanations, but it also enables interactive functionality in all components of building a DTC, including its display, the selection of nodes, and their expansion. In addition, the system allows user configuration of these visualisations.

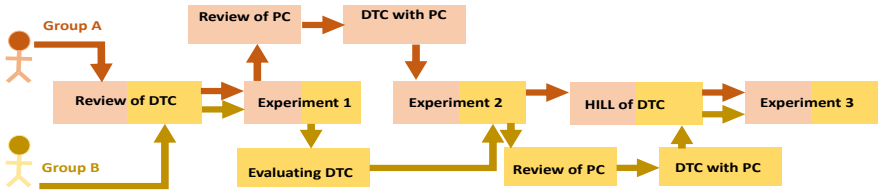


Fig. 5: Composition of survey for participants in Group A and Group B.

Experiment 1 examines the effectiveness of visualising a DTC using a node-link diagram with coloured nodes. Experiment 2 evaluates participants' ability to understand the classification of individual instances. This experiment compares the traversal of a DTC against the PC-based visualisation of the path to a leaf. Experiment 3 exhibits video-pairs contrasting two different HILL systems. Showing videos removes the participants' need to gain sufficient expertise with GUI aspects that are not the core of the visualisation. Nevertheless, participants can evaluate how well a system supports users to perform a HILL task.

With probability 1/2, each participant is randomly placed in one of two groups (Group A and Group B). Each group is shown slightly different visualisation when completing tasks in Experiment 1 and Experiment 2 with the aim of contrasting these visualisations' effect on the participants' ability to complete the tasks. Thus, the order of the review material is slightly different. Figure 5 shows the order (work-flow) for each group.

Experiment 1 — DTC node colouring. In Experiment 1 participants are shown several different DTCs and are required to estimate the accuracy of each DTCas well as select the leaf node that they believe *'is most in need of further refinement to improve the accuracy of the classifier'*. Figure 6 shows the layout for this experiment. The aim is to scrutinise the following hypotheses.

Hypothesis 1*:* (H 1) The technique of colouring nodes will allow a user to estimate the predictive power of a DTC more accurately.

Hypothesis 2*:* (H-2) The technique of colouring nodes will allow a user to more quickly estimate the predictive power of a DTC.

Hypothesis 3*:* (H-3) The technique of colouring nodes will allow a user to more accurately identify the most impure nodes in a DTC.

Hypothesis 4*:* (H-4) The technique of colouring nodes will allow a user to more quickly identify the most impure nodes in a DTC.

Since accuracy is easier to calculate than the F-measure, asking for accuracy ensures that the cognitive load on the participant is reduced. To quantify the accuracy of a user's choice for a node in need of improvement, we use a metric $RImp(n)$ describing the impurity of a node in a DTC relative to the most impure node in the tree. Let n be defined as a leaf node in a DTC containing N leaf nodes. Further, let $I(n)$ be defined as the number of instances of training data that reach n whose class is not the majority class of instances reaching n. Our evaluation metric is precisely defined as follows.

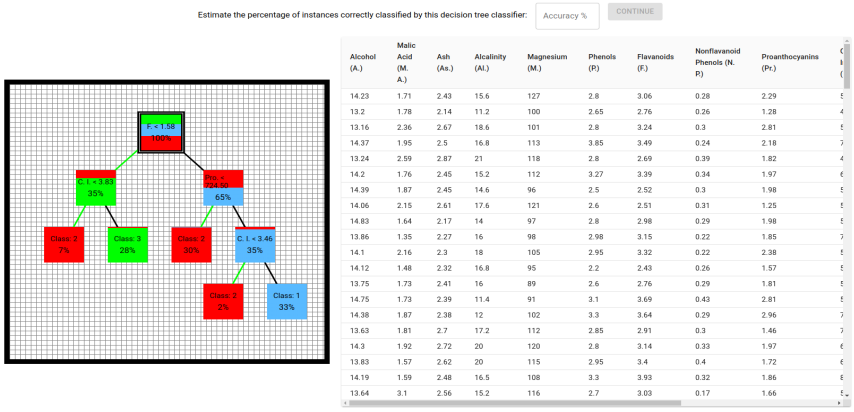


Fig. 6: Layout of survey for Experiment 1.

Definition 1. We define $I_{max}(N)$ as $I_{max}(N) = \max\{I(n) \mid n \in N\}$.

Definition 2. We define *the relative impurity* $RImp(n)$ of a node by $RImp(n) = (I_{max}(N) - I(n))/I_{max}(N)$.

Each participant is shown 8 different DTCs that were constructed from three datasets available in the UCI-repository (Wine, Cryotherapy, and Seeds). These three datasets exhibit the following relevant properties.

- Reasonably accurate ($> 90\%$ accuracy) DTC can be learnt for each dataset with small trees sizes that remain interpretable to a human.
- The attributes have humanly understandable names and semantic meaning.

From the datasets, eight DTC were built with a range of different accuracies for two reasons. First, having different accuracies ensures that there are no patterns that participants can use to help them assess the accuracy of any tree. Second, having a range of accuracies ensures that participants' ability to assess the accuracy of a DTC is not dependent on the DTC having a particularly low or high accuracy. For each participant, we record the following information.

- Their prediction of the accuracy of each DTC.
- The time taken to predict the accuracy of each DTC.
- The leaf node selected by the participant as most impure for each DTC.
- The time taken to select the most impure leaf node in each DTC.

Experiment 2 — Rule visualisation. Figure 7 shows the two different visualisations for Group A and Group B. Group A's visualisation has a DTC on the left and a table on the right. The DTC is a simple node-link diagram with the split criteria for each internal node represented textually. The table contains a single row with the attribute values of one instance. Group A is required to traverse the DTC for the instance shown in the table and determine whether it arrives at a leaf selected by the green arrow.

Fig. 7: Layout of Experiment 2: Group A (left) and Group B (right).

Group B is shown a PC visualisation of the path to a leaf in a DTC. This PC visualisation shows the entire dataset as well as ranges on several of the PC axes to represent each of the univariate splits on the path to the leaf. The polyline for one instance is shown as a distinct, thick black line. Group B is required to determine if this instance arrives at the leaf being visualised using PC.

Both groups use the same set of instances and DTC. The DTCs used are the same as those in Experiment 1. For each DTC, participants must evaluate three instances. Participants answer whether they think this instance reaches the selected leaf node. This experiment tests the following hypotheses.

Hypothesis 5*:* (H-5) Participants visualising the path to a node using PC will more accurately determine whether an instance reaches a particular leaf node.

Hypothesis 6*:* (H-6) Participants visualising the path to a node using PC will more quickly determine whether an instance reaches a particular leaf node.

Experiment 3 — Human-in-the-Loop Video Survey. For each pair of videos, one video illustrates a particular HILL task being performed using a system based on the techniques proposed earlier. The other video shows the same task, but carried out using Weka's `UserClassifier`. After viewing both videos, participants are required to express their preference on a five-point Likert scale (Table 1). The survey system randomly decides which HILL system's video will be the first shown. This order remains the same for all video pairs per participant. In total, five different pairs of videos demonstrate a variety of HILL tasks. After answering the Likert-scale questions for each of the five pairs of videos, the survey system asks participants Q6.

Table 1: Questions for Experiment 3.

ID	Question
Q1	Which system do you believe provides a better method of finding splits to build a decision tree classifier?
Q2	Which system allows you to better navigate and understand the current state of a decision tree classifier as it is being constructed?
Q3	Which system allows you to more easily determine how often a tree will predict the class correct class of an instance?
Q4	Which system allows you to more easily find nodes in a decision tree classifier that need additional splits?
Q5	Which system would provide better assistance to you when constructing a decision tree?
Q6	Based on the videos you've seen which system would you prefer to use to build a decision tree classifier?

5 Results

Experiment 1 Table 2a shows the average results for each group. When estimating the accuracy of a DTC, Group B appears to be less accurate than Group A but makes estimates much faster. Similarly, when selecting the most impure leaf in a DTC, the mean relative impurity of leaves selected by Group B was slightly higher, however, the time to choose this leaf was quicker than Group A.

Since the results are not normally distributed by the Shapiro–Wilk test, we use the Wilcoxon-Mann-Whitney test to check for statistically significant differences (as opposed to a T-Test). Performing this statistical test on the accuracy differences and *Rimp* results in p-values of 0.196 and 0.311, respectively. As such, we cannot reject the null hypotheses that Group A and Group B perform equally and must reject H-1 and H-3. Performing statistical tests on the accuracy time and the leaf selection time results in p-values of 3.65×10^{-10} and 0.006, respectively. Here the differences in time for both tasks are statistically significant: we can accept H-2 and H-4. Although the experiment shows that colouring nodes assist users to more quickly assess the accuracy of a DTC and find leaves requiring refinement, node colouring appears to have little impact on how accurately subjects perform these tasks.

Experiment 2 Table 2b shows the average result (as a % of correctly answered questions) from Group A and Group B. These average accuracies show clear differences in the performance between the two groups. Using the PC-based visualisation, Group B answered 86.7% of questions correctly. This is in contrast to Group A, who only achieved 77.5%. Group B also determines whether an instance reaches a leaf faster than Group A. On average, Group B only required approximately one-quarter of the time that Group A needs to determine whether an instance reaches a leaf. Using the Wilcoxon-Mann-Whitney Test, the average accuracy and time differences are statistically significant with p-values of 1.51×10^{-9} and $< 2.2 \times 10^{-16}$, respectively. As such, H-5 and H-6 are accepted. These results demonstrate a clear advantage to the use of PC to allow humans to interpret DTC. Using PC dramatically decreased the mistakes made when interpreting the splits for a DTC as well as allowing participants to more quickly interpret the series of splits leading to the classification of an instance.

Experiment 3 Figure 8 visualises the distribution of responses received for each question. We can see from these results that there is a clear preference for the PC-based system for all HILL tasks examined. For each of the five pairs of videos, between 66.3% and 79.8% of participants had some preference for the PC-based system. In addition, participant responses from the last question showed 79.8% of participants had an overall preference for the PC-based system.

5.1 Validity threats

Since participants could not be observed, time measurements may be affected by distractions. Although intended for subjects with limited experience with PC, some may have had previous experience using PC. Finally, all subjects used their

Table 2: Results for the first two experiments.

(a) Results for Experiment 1.

Group	Mean accuracy difference	Mean accuracy time (seconds)	Mean $RImp$	Mean leaf choice time (seconds)
A	17.8%	80.2	0.259	17.6
B	19.0%	49.3	0.282	17.1

(b) Results for Experiment 2.

Group	Mean accuracy	Mean time per leaf (seconds)
A	77.5%	24.0
B	86.7%	6.7

monitors, and available screen real-estate may have affected their performance. Regarding external validity, participation was restricted to a bachelor's level, and working in STEM fields. Participants with a higher level of expertise in ML may have found the visualisation techniques more intuitive and performed better.

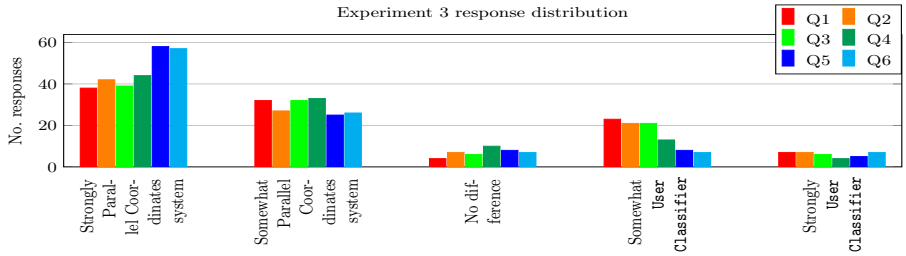


Fig. 8: Distribution of response from Experiment 3.

6 Conclusion

Many have argued around the advantages of HILL for machine learning tasks. Our approach here shows how an effective visualisation can involve a human expert in guiding the construction of interpretable classification models. Moreover, our IML system not only contributes to explainable AI by producing understandable models, it also allows a user to revise the objective of classification accuracy to other important objectives. We have also emphasised that while decision trees rank high as a model for HILL classification, we can strike a suitable balance between multi-variate decision tress and uni-variate decision trees. Users can propose bi-variate splits that remain understandable in our PC-canvas. We have proposed how to visualise and elaborate on characterisation and discrimination rules where users are interested in one class above the others.

References

1. Amershi, S., Cakmak, M., Knox, W.B., Kulesza, T.: Power to the people: The role of humans in interactive machine learning. AI Magazine **35**(4), 105–120 (2014).
2. Ankerst, M., Ester, M., Kriegel, H.P.: Towards an effective cooperation of the user and the computer for classification. 6th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining. pp. 179–188. KDD ACM, NY, USA (2000)

3. Blanco-Justicia, A., Domingo-Ferrer, J.: Machine learning explainability through comprehensible decision trees. Machine Learning and Knowledge Extraction. pp. 15–26. Springer (2019)
4. Choo, J., *et al.*: iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction. IEEE VAST. pp. 27–34 (2010)
5. Darlington, K.: Aspects of intelligent systems explanation. Universal J. of Control and Automation **1**(2), 40 – 51 (2013)
6. Dikshit, A., Pradhan, B.: Interpretable and explainable AI (XAI) model for spatial drought prediction. Science of The Total Environment **801**, 149797 (2021).
7. Estivill-Castro, V.: Collaborative knowledge acquisition with a genetic algorithm. 9th ICTAI'97. pp. 270–277. IEEE CA, USA (1997)
8. Fails, J.A., Olsen, D.R.: Interactive machine learning. 8th Int. Conf. on Intelligent User Interfaces. p. 39–45. IUI '03, ACM, , NY, USA (2003).
9. Freitas, A.A.: Comprehensible classification models: a position paper. SIGKDD Explorations **15**(1), 1–10 (2013)
10. Freitas, A.A., Wieser, D., Apweiler, R.: On the importance of comprehensible classification models for protein function prediction. IEEE/ACM Trans. Comput. Biology Bioinform. **7**(1), 172–182 (2010)
11. Guidotti, R., *et al.*: A survey of methods for explaining black box models. ACM CSUR **51**(5), 1–42 (2018)
12. Inselberg, A.: Parallel Coordinates : Visual Multidimensional Geometry and its Applications. Springer, NY (2009)
13. Kingsford, C., Salzberg, S.L.: What are decision trees? Nature biotechnology **26**(9), 1011–1013 (2008)
14. Kodratoff, Y.: Chapter 8: Machine learning. Knowledge Engineering Volume I Fundamentals. pp. 226–255. McGraw-Hill, USA (1990)
15. Lai, P.L., Liang, Y.J., Inselberg, A.: Geometric divide and conquer classification for high-dimensional data. DATA . pp. 79–82. SciTePress (2012)
16. Lakkaraju, H., *et al.*: Faithful and customizable explanations of black box models. AI, Ethics, and Society. p. 131–138. AIES '19, ACM, NY, USA (2019)
17. Moore, A., Murdock, V., Cai, Y., Jones, K.: Transparent tree ensembles. 41st SIGIR'18 p. 1241–1244. ACM, NY, USA (2018)
18. Palan, S., Schitter, C.: Prolific.ac—a subject pool for online experiments. Behavioral and Experimental Finance **17**, 22–27 (2018)
19. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986).
20. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**, 206–215 (2019)
21. Samek, W., *et al.*: Evaluating the visualization of what a deep neural network has learned. IEEE T. Neural Networks & Learning Sys. **28**(11), 2660–2673 (2017)
22. Samek, W., Müller, K.R.: Towards explainable artificial intelligence. Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. pp. 5–22. Springer, Cham (2019)
23. Teoh, S.T., Ma, K.: Starclass: Interactive visual classification using star coordinates. SIAM Int. Conf. on Data Mining. vol. 112, pp. 178–185. (2003)
24. Teoh, S.T., Ma, K.L.: Paintingclass: Interactive construction, visualization and exploration of decision trees. 9th SIGKDD'03, ACM, NY, USA (2003)
25. Ware, M., *et al.*: Interactive machine learning: letting users build classifiers. Int. J. Hum.-Comput. Stud. **55**(3), 281–292 (2001)
26. Webb, G.I.: Integrating machine learning with knowledge acquisition through direct interaction with domain experts. Knowledge-Based Sys. **9**(4), 253–266 (1996).