Interpretable and Reliable Rule Classification based on Conformal Prediction

Husam Abdelqader[1,2,3], Evgueni Smirnov[1,4], Marc Pont[2,5], and Marciano Geijselaers[2,6]

[1] Maastricht University, Maastricht, The Netherlands
[2] Integrin, Geleen, The Netherlands
[3] h.husamfuadsalehabdelqader@student.maastrichtuniversity.nl
[4] smirnov@maastrichtuniversity.nl
[5] marc.pont@intergrin.nl
[6] marciano.geijselaers@intergrin.nl

Abstract. This paper deals with the challenging problem of simultaneously integrating interpretablility and reliability into prediction models in machine learning. It proposes to combine the interpretable models of decision rules with the reliable models based on conformal prediction. The result is a new technique of conformal decision rules. Given a test instance, the technique is capable of providing a point prediction, an explanation, and a confidence value for that prediction plus a prediction set. The experiments show when and how conformal decision rules can be used for interpretable and reliable machine learning.

Keywords: Interpretable Machine Learning · Reliable Machine Learning · Decision Rules · Conformal Prediction.

1 Introduction

Machine learning in critical domain applications needs to provide predictions that are both interpretable and reliable [7]. Following [8] we informally define, interpretablility of a prediction as the degree that the cause for the prediction can be understood by a user. Analogously, we define reliability of a prediction as the degree that a user can trust the prediction [13]. Thus, the acceptance process of a prediction can be facilitated using additional information on the interpretablility and reliability of the prediction.

Integrating interpretable and reliable machine learning is usually implemented using the Mondrian scheme summarized in [1]. The scheme consists of two steps:

(1) train an interpretable prediction model (e.g. a decision tree) on the available data T and view that model as a taxonomy that partitions the data into categories r (i.e. through leaf nodes).
(2) train a reliable prediction model on the data T_r of each category r.

In this context, when a test instance is processed, it is first handled by the interpretable prediction model that provides a point prediction plus a cause

for that prediction. In addition, the model identifies the category that fits the instance and calls the reliable prediction model that corresponds to that category. The latter outputs a confidence value in the point prediction and/or a region prediction, i.e. a set of labels that with a high probability contains the true label of the test instance.

To provide a data-distribution free guarantee integrating interpretable and reliable machine learning is realized using the conformal prediction framework [13, 14]. This framework provides a set of techniques for establishing precise level of confidence in new predictions in the presence of finite training data and without any assumption on data distribution. It allows computing valid region predictions, i.e. regions that contain the true labels of test instances within a user-acceptable error probability.

In general, the conformal prediction framework operates as follows [14]. Given a test instance x, it first provisionally labels x with label y; i.e. it considers hypothetically labeled instance (x, y). Then the (confidence) p-value p_y for label y is calculated as the proportion of the instances in $T \cup \{(x, y)\}$ whose nonconformity scores α are greater than or equal to that of the instance (x, y). If $p_y > \epsilon$ for a chosen significance level ϵ, label y is added to the region prediction set Γ for test instance x.

To apply conformal prediction we need to compute for each instance nonconformity score α that indicates how untypical the instance is w.r.t. the rest of the data. This computation is realized by a nonconformity function A that is trained on the data. There are different scenarios for this based on different validation procedures which result in different conformal predictors.

Conformal prediction was integrated with interpretable prediction models for regression and classification using variations of the Mondrian integration scheme presented above [11, 5, 6, 1]. The interpretable prediction models used were regression/decision trees while the reliable prediction models were conformal predictors. The proposed integrations employed a *global* approach to train conformal predictors. The regression/decision tree trained is viewed as a taxonomy that imposes a partition P_h on training data T. Each element $T_r \subset T$ of this partition corresponds to a concrete leaf node r. Conformal predictors are trained, one for each node r, however, in a *global* manner. This means that first each T_r is split into a proper training set T_r^t and a calibration set T_r^c. Then the global proper training set $\sum_r T_r^t$ is used to train the global nonconformity function A shared by all the conformal predictors. The conformal predictor for each leaf node r employs the global function A to compute the nonconformity scores of the calibration training instances in T_r^c associated with that leaf node. Thus, each test instance receives label p-values and region prediction from the conformal predictor of the leaf node in which it arrives.

The *global* approach to train conformal predictors is based on the assumption that larger data result in more accurate nonconformity functions that in turn decrease the sizes of the region prediction sets. However, in this paper we argue that the *global* approach has a fundamental problem that concerns integrating interpretable and reliable machine learning following the Mondrian scheme. This

problem is a *label-imbalanced problem*: the probability distributions that generate the global proper training set $\sum_r T_r^t$ and leaf-node calibration subsets T_r^c can be very different since the trees are learned by minimizing the class entropy or output-variable variance in leaf nodes. This implies that the global nonconformity function can be inaccurate on test data that arrive in a particular leaf node. This is due to the fact that this function is trained on the global proper training set $\sum_r T_r^t$ while the test data is generated from the distribution similar to that of subset T_r associated with that node.

In this paper we propose a *local* approach to train conformal predictors to address the label-imbalanced problem. The key idea is to train the nonconformity functions of the conformal predictors locally, i.e. on the proper training subsets T_r. We show that this approach has a potential to improve integrating interpretable and reliable machine learning for large data.

The second contribution of our paper is that we propose to combine decision rules [4] and conformal prediction according to the Mondrian integration scheme, i.e., we continue the research line of conformal interpretable models in classification as outlined in [11, 5]. Following the criteria for model interpretablility in [9] we motivate our choice for decision rules as follows. First, decision rules are more interpretable than decision trees [4]. On a model level decision rules are usually shorter, i.e. more general, than the rules encoded by decision trees [7]. This implies that for the same classification problem we need less decision rules; i.e. we need less modules for global interpretability (in terms of [9]). On a prediction level decision rules provide individual prediction explanations. For the reason given before these explanations are usually shorter than those of decision trees. Thus, (again in terms of [9]) the local interpretability for a single prediction is better. Finally, we note that while still disputable decision rules are algorithmically more transparent than decision rules. We believe that it is easier to explain the separate-and-conquer strategy of decision rules than the divide-and-conquer strategy of decision trees [4] (check the pseudo-code in Algorithm 1).

The rest of the paper is organized as follows. In the next section we formalize the classification task in the context of point estimation and prediction-set estimation. In Section 3, we present decision rules. The conformal prediction and its basic set predictors are presented in Section 4. In Section 5, we propose our approach and explain the underlying algorithms. The experiments and results are provided in Section 6. Section 7 concludes the paper.

2 Classification

Let X be an instance space, Y be a finite discrete class variable, and P be a probability distribution over $X \times Y$. Training data set T is a multi set of M instances $(x_m, y_m) \in X \times Y$ drawn from the distribution P under the randomness assumption. In this context, we can define two possible tasks: point classification task and region classification task.

[7] The decision tree rules partition the data which assumes these rules are longer; i.e. more specific.

The point classification task is to find an estimate $y \in Y$ of the true class for a test instance $x \in X$ according to P. To solve the task we first learn a point predictor h in a hypothesis space H using training data T. The predictor h is a function of type $h : X \to Y$. It first computes for test instance x a distribution of posterior scores $\{s_y\}_{y \in Y}$ over all the classes in Y. Then, h outputs class y with the highest score s_y as the estimated class for test instance x.

The region classification task is to estimate a prediction set $\Gamma(x) \subseteq Y$ that contains possible classes for a test instance $x \in X$ according to P. To solve the task we need a class set predictor. The two most desired properties of such predictor are validity and informational efficiency. A class set predictor is said to be valid iff the probability that the prediction set $\Gamma^{\epsilon}(x) \subseteq Y$ does not contain the class for the test instance x is at most the chosen significance level $\epsilon \in [0, 1]$. A class set predictor is said to be informationally efficient if the prediction set $\Gamma^{\epsilon}(x) \subseteq Y$ is non-empty and small. In Section 4 we briefly introduce the conformal framework that is used for designing valid set predictors [14].

3 Decision Rules

Decision rules form an approach to point classification [4]. They are "if-then" rules that can be learned from training data T. The antecedent of any rule r is a condition that can be tested for any instance $x \in X$. The consequent part of r consists of a single class value $y \in Y$ that is assigned to any test instance $x \in X$ as a class point estimate. The final point predictor h is a set of decision rules r.

Decision rules can be used for descriptive and classification tasks. For descriptive tasks they provide interpretations/summarization of the training data w.r.t. class information. For classification tasks decision rules provide class predictions plus their explanations based on the conditions in the rule antecedents. This makes decision rules an important tool in interpretable machine learning.

The separate-and-conquer learning algorithm of decision rules is given in Algorithm 1. In an iterative manner it executes the following steps. First, the algorithm learns one rule r from T. If rule r is acceptable (e.g. a high TPr rate for the class assigned by r), it is added to point predictor h and subset T_r of training instances covered by r is removed from T. In this way the algorithm focuses only on those training instances in each new iteration that have not been covered so far. The iteration process ends when a stopping criterion is met. The criterion can be a threshold on the percentage of covered data, the validation performance of the final point predictor h etc. Once the criterion holds, the algorithm adds the default rule r that holds when all other rules logically fail.

To use point predictor h of decision rules r, a classification procedure has to be defined. We assume that the rules are ordered in decreasing order of their performance on a separate validation data. A test instance x receives a class value of that rule $r \in h$ that matches x first in the order.

There are several techniques for implicit regularization of decision rules due their sensitivity to over-fitting. One of the most accurate of those is Incremental Reduced Error Pruning (IREP) given in [4]. The pseudo-code of IREP is provided

Algorithm 1: Decision Rule Learning

Input: Training set T;
Output: Point predictor h of decision rules;
Set h equal to empty set $\emptyset$;
repeat
  Learn rule r from T;
  if *rule r is acceptable* **then**
    Add rule r to point predictor h;
    Remove set T_r of instances covered by r from T;
until *stopping criterion is met*;
Add default rule r to point predictor h;
return point predictor h.

Algorithm 2: Incremental Reduced Error Pruning (IREP)

Input: Training set T;
Output: Point predictor h of decision rules;
Set h equal to empty set $\emptyset$;
repeat
  Split T into growing set T^g and prune set T^p;
  Learn rule r from T^g;
  Prune r on T^p;
  if *rule r is acceptable* **then**
    Add rule r to point predictor h;
    Remove instances covered by r from T;
until *stopping criterion is met*;
Add default rule r to point predictor h;
return point predictor h.

in Figure 2 and it is very similar to that of decision rule learning. The only difference is the manner of learning new rules (steps 3 to 5). IREP first splits the current training data T into growing set T^g and prune set T^p. Then it trains a new rule r on T^g and subsequently prunes that rule on T^p. Since the data covered by rule r are removed from T, the next rule will have a small overlap with r on instance space X if at all. If we extrapolate this finding over the whole sequence of rules r in the final point predictor h, we may conclude that IREP minimizes the overlap between the (subsequent) rules. This in turn reduces the number of decision rules r in h compared with any other technique for decision rule pruning. Thus, IREP is an excellent candidate for prediction interpretability.

The order of decision rules r in final point predictor h, that we have assumed for classification purposes, imposes a partition P_h on training set T. Each element $T_r \subset T$ of this partition corresponds to a concrete decision rule r and, thus, it is biased toward class $y \in Y$ that r assigns. This implies that the probability distributions that generate sets T and T_r can be very different.

In addition, we note that partition P_h can be viewed as a rule-induced taxonomy. The categories of this taxonomy are intensionally represented by rules r while extensionally by training sets T_r. This property is used for combining decision rules and conformal prediction following the Mondrian integration scheme.

4 Conformal Prediction

This section provides a short intro to conformal prediction. First, it considers transductive and inductive conformal prediction. Then, it proceeds with Mondrian conformal prediction.

4.1 Transductive and Inductive Conformal Prediction

The conformal prediction framework [12, 13] allows us to train class set predictors that are automatically valid. They operate as follows. Given a test instance $x_{M+1} \in X$, to decide whether to include a class $y \in Y$ in prediction set $\Gamma^{\epsilon}(x_{M+1}) \subseteq Y$, the labeled instance (x_{M+1}, y) is provisionally considered. Then the nonconformity scores α_m of all the instances (x_m, y_m) in $T \cup \{(x_{M+1}, y)\}$ are computed. The p-value p_y of class y for test instance x_{M+1} is computed as follows:

$$p_y = \frac{\#\{(x_m, y_m) \in T | \alpha_m > \alpha_{M+1}\} + \tau \#\{(x_m, y_m) \in T | \alpha_m = \alpha_{M+1}\}}{M+1} \quad (1)$$

where α_{M+1} is the nonconformity score of (x_{M+1}, y) and τ is an uniformly distributed random variable in $[0, 1]$.

Once we have fixed significance level ϵ, class y is included in prediction set $\Gamma^{\epsilon}(x_{M+1})$ of test instance x_{m+1} if $p_y > \epsilon$. Thus, in a long run we get validity: the error e when prediction sets do not include the true classes is bounded from below by ϵ.

The art to apply conformal prediction is to decide how to compute nonconformity scores. A nonconformity score α_m for any instance (x_m, y_m) is a score that indicates how untypical is (x_m, y_m) w.r.t. the instances in data $(T \cup \{(x_{M+1}, y)\}) \setminus \{(x_m, y_m)\}$. To compute such a score we need a nonconformity function A. Formally, this function is of type $A : (X \times Y)^{(*)} \times (X \times Y) \rightarrow \mathbb{R}^+ \cup \{+\infty\}$ [8]. Given a data set $T \subseteq X \times Y$ and an instance $(x_m, y_m) \in (X \times Y)$, it returns a nonconformity score $\alpha_m \in \mathbb{R}^+ \cup \{+\infty\}$ indicating how untypical the instance (x_m, y_m) is for the instances in $(T \cup \{(x_{M+1}, y)\}) \setminus \{(x_m, y_m)\}$. An example of function A is the general nonconformity function applicable for any point predictor $h(x)$ [14]. Given an instance (x_m, y_m), the function outputs $\sum_{y \neq y_m} s_y$, i.e. the sum of the scores s_y of all the classes $y \in Y$ computed by h without that of y_m. This makes the conformal prediction predictor-agnostic.

We note that in general the nonconformity score α_m for any instance (x_m, y_m) is w.r.t. all the remaining instances in data $(T \cup \{(x_{M+1}, y)\}) \setminus \{(x_m, y_m)\}$.

[8] $(X \times Y)^{(*)}$ denotes the set of all multi sets defined over $X \times Y$.

Thus, computing the nonconformity scores α_m for all the instances (x_m, y_m) in $T \cup \{(x_{M+1}, y)\}$ is realized by a leave-one-out process implemented in so-called transductive conformal predictors (TCPs). To reduce the computational complexity of TCPs [10] proposed inductive conformal predictors (ICPs). ICPs use a hold-out process and thus they split the training data set T of size M into the proper training set $T^t \subseteq T$ of size $L < M$ and the calibration set $T^c \subseteq T$ of size $M - L$. Set T^t is used to train the nonconformity function A. The function is then applied over all the instances in data $T^c \cup \{(x_{M+1}, y)\}$ to compute their nonconformity scores. The p-value p_y of class y for test instance x_{M+1} is computed in a similar manner, however, over nonconformity scores of instances in $T^c \cup \{(x_{M+1}, y)\}$ only; i.e.,

$$p_y = \frac{\#\{(x_m, y_m) \in T^c | \alpha_m > \alpha_{M+1}\} + \tau \#\{(x_m, y_m) \in T^c | \alpha_m = \alpha_{M+1}\}}{M - L + 1} \quad (2)$$

where α_{M+1} is the nonconformity score of (x_{M+1}, y) and τ is an uniformly distributed random variable in $[0, 1]$.

We note that ICPs are computationally more efficient than TCPs. However, their informational efficiency (prediction set size) is usually lower than that of TCPs. Still, in the rest of the paper will be using ICPs.

4.2 Mondrian Conformal Prediction

Assume that we have a taxonomy P of disjointed categories. P partitions T into disjointed subsets T_r intensionally represented by categories r from P. Due to the disjointedness the probability distributions behind sets T and T_r can be very different. In this case any conformal predictor trained on T is valid for any data set generated by the probability distribution that generates T. However, it may be invalid for data sets generated by the probability distributions that generate subsets T_r for some categories r in P. To guarantee predictor validity within the categories, Mondrian conformal prediction was introduced in [14].

The key idea is to train a separate conformal predictor for each subset T_r. In case of ICP this is realized as follows. First, each subset T_r is split into proper training set T_r^t and calibration set T_r^c. Then, a global proper training set T^t is formed equal to $\bigcup_{r \in P} T_r^t$ to train the global nonconformity function A. The function is used to compute the nonconformity scores of the instances in calibration set T_r^c of each ICP$_r$. Once this process is complete, we receive individual ICP$_r$ for each category r in taxonomy P.

The process of region classification is simple. Given a test instance x_{M+1}, we first determine category r from taxonomy P that matches x_{M+1}. Then we apply the corresponding ICP$_r$ on x_{M+1} to compute a prediction set $\Gamma^\epsilon(x_{M+1})$. We note that each ICP$_r$ is a valid conformal predictor on data sets generated by the probability distributions that generate the corresponding subset T_r. Thus, we receive a local validity within the categories and, thus, a global validity of the conformal predictors.

5 Conformal Decision Rules

In this section we propose to integrate decision rules and conformal prediction. The key idea is simple: decision rules imposes a taxonomy on training data, and, thus, we integrate by training a Mondrian conformal predictor on the taxonomized data. This implies that a conformal decision rule is a decision rule r with its own ICP_r and the final predictor is a set of conformal decision rules.

The learning algorithm of conformal decision rules is given in Algorithm 3. The algorithm input consists of the training set T, calibration set ratio c, and Boolean variable *global*. First, the algorithm trains point predictor h of decision rules r on training set T using Algorithm 1 (step 1). Since rules r are ordered, they represent intensionally a taxonomy that can be employed for Mondrian conformal prediction. Therefore, the algorithm uses the rules to partition training data T into disjointed subsets $T_r \subseteq T$ s.t. each rule r covers exactly one T_r (steps 3-4). Then, to prepare the data for training ICPs, all the subsets T_r are divided in a *class-stratified manner* into proper training sets T_r^t and calibration sets T_r^c according to calibration set ratio c in a class-stratified manner (step 5).

In steps 7-16 a Mondrian conformal predictor is trained on the partitioned data; i.e. an ICP_r is trained for each rule r. Two strategies are employed to train ICPs: global (steps 7-11) and local (steps 12-15). The global strategy is similar to that from [1]: a global nonconformity function A is trained on the union of proper training sets T_r^t over all the rules r, and each ICP_r employs A on its own calibration set T_r^c. The local strategy is a new strategy that we propose: each ICP_r gets its own local nonconformity function A_r trained on its own proper training set T_r^t and this function is applied on its own calibration set T_r^c.

Once all the ICPs have been trained, the algorithm outputs point predictor h of all decision rules and the set of all ICPs. Thus, each conformal decision rule is given a rule r and its corresponding ICP_r.

The classification procedure is straightforward. Given a test instance x_{M+1}, the decision rules $r \in h$ are visited in the order imposed on h (see the explanation of Algorithm 1). If x_{M+1} matches the antecedent of the current rule r, its receives a point (class) prediction $y \in Y$ associated with r, an explanation (of how x_{M+1} matches the antecedent) plus a prediction set $\Gamma^\epsilon(x_{M+1}) \subseteq Y$ provided by ICP_r on a given significance level ϵ.

We note that conformal decision rules are valid class set predictors; i.e. the probability that the prediction set $\Gamma^\epsilon(x) \subseteq Y$ does not contain the class for the test instance x is at most ϵ. This is due to the fact that they are essentially Mondrian conformal predictors (see above).

The global and local strategies for setting up ICPs of decision rules are rather different. The global strategy trains global nonconformity functions A that are accurate on data generated by the original data distribution P. However, the calibration sets T_r^c of decision rules r might come from different distributions since the rules usually cover subsets that are class biased. Thus, *the label imbalanced problem* might be present. As a result, global nonconformity functions A can be less accurate on these sets which can result in less accurate nonconformity functions (which in turn will decrease the informational efficiency).

Algorithm 3: Conformal Decision Rule Learning

Input: Training set T, calibration set ratio $c \in (0, 1.0)$, and Boolean variable *global*;
Output: Point predictor h of decision rules r and set $\{ICP_r\}_{r \in h}$;

```
Train point predictor h of decision rules r on training set T;
for each rule r ∈h do
    Determine training subset T_r ⊆ T covered by rule r;
    T := T \ T_r;
    Split T_r into proper training set T_r^t and calibration set T_r^c according to c;
if global then
    T^t := ⋃_{r∈h} T_r^t ;
    for each rule r ∈h do
        Set up inductive conformal predictor ICP_r using T^t and T_r^c;
else
    for each rule r ∈h do
        Set up inductive conformal predictor ICP_r using T_r^t and T_r^c;
Output Point predictor h of decision rules r and set {ICP_r}_{r∈h}.
```

The local strategy does not suffer from the label-imbalanced problem above: the local nonconformity functions A_r are trained on the proper training sets T_r^t and process calibration sets T_r^c that if stratified can be viewed coming from the same data distribution. Thus, the functions A_r can be accurate on T_r^c which can result in accurate nonconformity functions (which in turn will boost the informational efficiency). However, this happens only if the proper training sets T_r^t and calibration sets T_r^c are not small. Due to the nature of decision rule learning, the size of the covered set T_r of each new rule usually decreases. This implies that the local strategy has to be used for relatively large data.

6 Experiments and Results

This section presents our experiments. The data sets used for this research are described in Subsection 6.1. The experimental setup is provided in Subsection 6.2. The results are given in Subsection 6.4.

6.1 Data sets

In the experiments, we consider 10 binary classification data sets from the UCI machine learning repository [3]. The sets are summarized in Table 1. We note they are pre-processed where necessary: missing values are replaced by mean for numeric features and by mode for discrete features.

Table 1: Public data sets characteristics

Data set	Short Hand	Instances	Majority class
Heart Cleaveland	HC	303	54%
Heart VA	HV	200	74%
Haberman	HM	306	74%
Spam base	SB	4601	61%
Australian Credit Card	AC	1372	56%
Cancer	C	569	63%
Ionosphere	I	351	64%
Hepatitis	H	155	79%
German Credit	GC	1000	70%
Indian Liver	IL	583	71%

6.2 Experimental Settings

We experiment with two types of conformal set predictors: pure ICP and conformal decision rules based on IREP and ICP denoted by IREP-ICP. IREP-ICP employs the local strategy and global strategy denoted by IREP-ICP(L) and IREP-ICP(G), respectively. The minimal number of training instances per rule is set to 30 for IREP. The pure ICP predictors and ICP predictors in IREP-ICP use the nearest-neighbor nonconformity function from [14]. This function outputs for any instance (x, y) a nonconformity score α equal to $\frac{D_K^y}{D_K^{-y}}$, where D_K^y (D_K^{-y}) is the sum of distances between x and Knearest neighbors of x that do (not) belong to class y. For all ICPs $\frac{2}{3}$ of the training data is used for the proper training set and $\frac{1}{3}$ for the calibration set.

The set predictors are tested using a stratified 5-fold cross validation procedure. We employ several metrics to estimate the performance of the models. To test experimentally the validity of a conformal set predictor we use the error rate e. The error rate e for a significance level ϵ is defined as proportion of test instances whose predicted prediction-sets Γ^ϵ do not contain the correct class. To show experimentally that a conformal set predictor is valid, we need to show that for any significance level $\epsilon \in [0, 1]$ we have $e \leq \epsilon$.

To test experimentally the informational efficiency of a set predictor on significance level ϵ we employ three main metrics: rate r^e of empty prediction sets, rate r^s of single prediction sets, and rate r^m of multiple prediction sets. The empty prediction sets, single prediction sets, and multiple prediction sets have their own errors. Rate r^e of empty prediction sets is an error, since the correct classes are not in the prediction sets. Error rate e^s (e^m) on single (multiple) prediction sets is defined as the proportion of the single (multiple) prediction sets that do not contain the correct classes.

6.3 Algorithm Output

For any test instance the output of conformal decision rules consists of a point prediction, an explanation, confidence values for all possible predictions plus a

Name	Description	Name	Description
Age	Age of the patient	Sgpt	Alamine Aminotransferase
Gender	Gender of the patient	Sgot	Aspartate Aminotransferase
TB	Total Bilirubin	TP	Total Protiens
DB	Direct Bilirubin	ALB	Albumin
Alkphos	Alkaline Phosphotase	A/G	Albumin and Globulin Ratio

Table 2: Indian Liver Data Set Input Variables

Instance	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	A/G	Class
58	48	Female	0.9	0.2	175	24	54	5.5	2.7	0.9	*no liver problem*
206	45	Male	2.5	1.2	163	28	22	7.6	4	1.1	*liver problem*

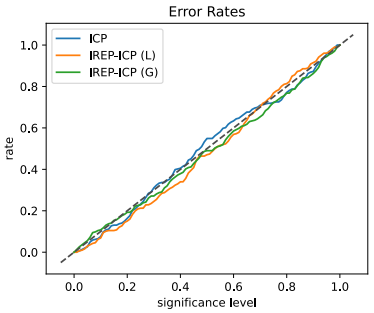
Table 3: Indian Liver Data Set Examples

prediction set for a chosen significance level ϵ. We provide the output of IREP-ICP(L) on the Indian Liver data. This data consists of 583 liver patients records divided into two classes, patients with a *liver problem* and patients with *no liver problem*. The input variables are presented in Table 2. IREP-ICP(L) was trained on the data and tested on two instances given in Table 3. The output for these instances for significance level $\epsilon = 0.05$ is as follows:

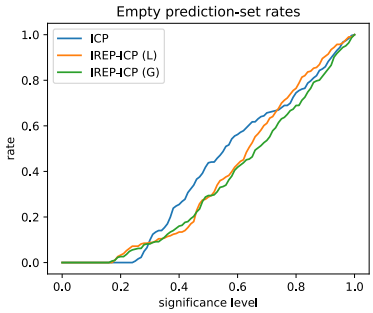
- **Instance 58:**
  - **Point Prediction:** *liver problem*
  - **Explanation:** *liver problem* since *Alkphos* is between 21.0 and 25.0
  - **p-value** of *no liver problem* is 0.49; **p-value** of *liver problem* is 0.51
  - **Prediction Set for** $\epsilon = 0.05$**:** { *no liver problem*, *liver problem* }
- **Instance 206:**
  - **Point Prediction:** *liver problem*
  - **Explanation:** *liver problem* since *TB* is between 0.88 and 1.6
  - **p-value** of *no liver problem* is 0.04; **p-value** of *liver problem* is 0.84
  - **Prediction Set for** $\epsilon = 0.05$**:** { *liver problem* }

6.4 Results

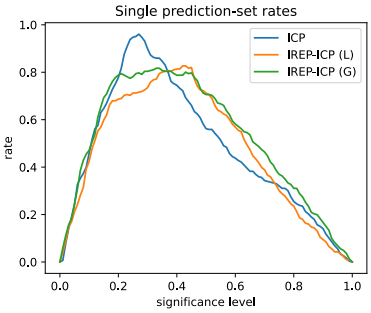
Illustrative Comparison In this sub-subsection we study pure ICP versus IREP-ICP as well as the local strategy versus the global strategy of IREP-ICP. The performance of the rules created by IREP has been studied in [2].The results are presented in Figures 1 and 2 for the Haberman data and Spam base data. Figures 1(a) and 2(a) show that ICP, IREP-ICP(L) and IREP-ICP(G) are valid set predictors. Their informational efficiency, however, are different. For the Haberman dataset ICP is more informationally efficient than both IREP-ICP predictors. Figure 1(e) shows that rate r^m of ICP decreases faster with significance level ϵ while Figure 1(c) shows that the max rate r^s of ICP is 0.96 against 0.71 and 0.79 of IREP-ICP(L) and (G), respectively. In addition, we note



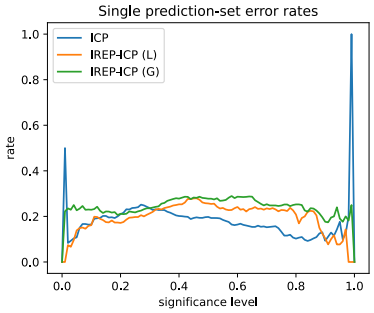
(a) Error Rates



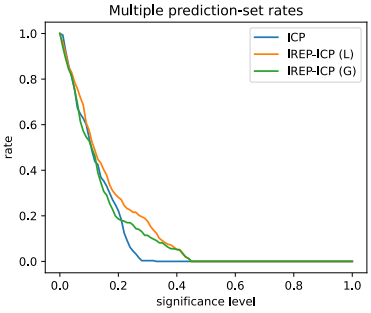
(b) Empty prediction-set rates



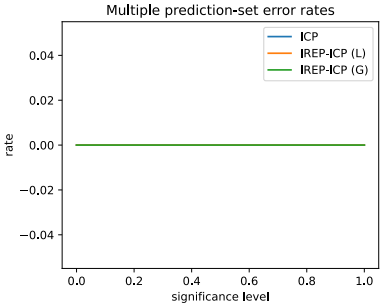
(c) Single prediction-set rates



(d) Single prediction-set error rates

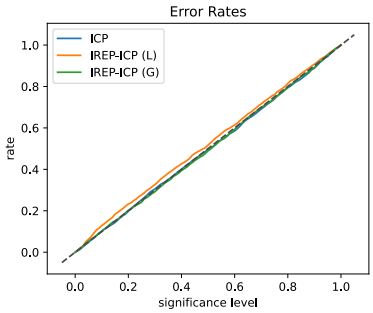


(e) Multiple prediction-set rates

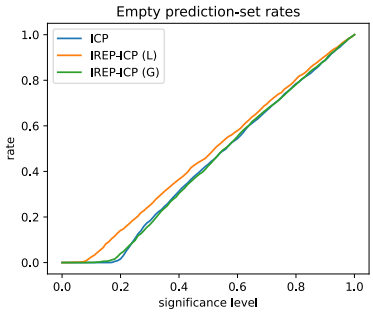


(f) Multiple prediction-set error rates

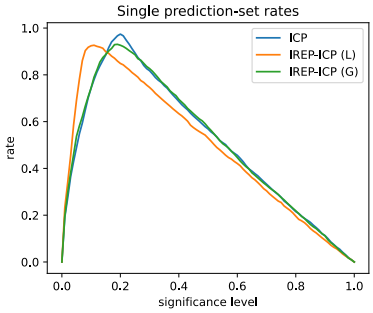
Fig. 1: Error and Prediction-Set Size Plots for the Haberman dataset



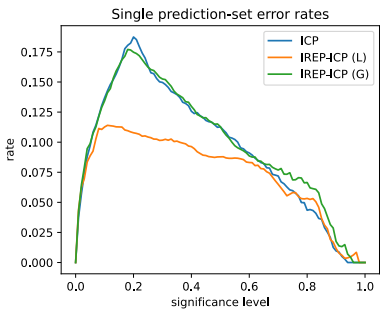
(a) Error Rates



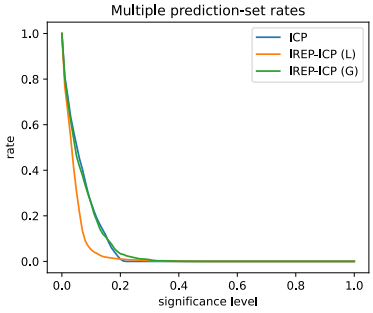
(b) Empty prediction-set rates



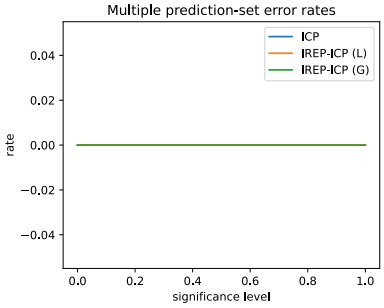
(c) Single prediction-set rates



(d) Single prediction-set error rates



(e) Multiple prediction-set rates



(f) Multiple prediction-set error rates

Fig. 2: Error and Prediction-Set Size Plots for the Spambase data set

that in Figure 1(d) error rate e^s of ICP is always lower than those of IREP-ICP predictors.

For the Spam base dataset IREP-ICP(L) is more informationally efficient than ICP and IREP-ICP(G) for significance level $\epsilon < 0.2$. Figure 2(e) shows that rate r^m of IREP-ICP(L) decreases faster with significance level ϵ while Figure 2(c) shows that the rate r^s of IREP-ICP(L) is 0.93 against 0.78 of ICP and IREP-ICP(G), respectively. In addition, we note that in Figure 2(d) error rate e^s of IREP-ICP(L) is always lower than those of ICP.

The results from Figures 1 and 2 can be explained as follows. The informational efficiency of ICP is usually better than that of the IREP-ICP(L) and (G) since ICP employs all the available data for training nonconformity functions A and calibration. However, there are cases similar to one we observed for the Spam base data when IREP-ICPs are better. This happens when decision rules impose taxonomies that make easier learning local nonconformity functions A_r.

Information efficiency of IREP-ICP(G) depends on the extent the distributions of the global proper training set T^t and calibration sets T_r^c of the ICP_r predictors are close. For the Haberman data the distributions are close (e.g. the majority class is positive over all the sets). This observation and a relatively small size of the data make the global nonconformity function A more accurate than the local nonconformity functions A_r. As a result IREP-ICP(G) has a better performance than IREP-ICP(L) on the Haberman data. However, for the Spam base data the situation is rather different: the distributions of the global proper training set T^t and calibration sets T_r^c are not close (e.g. the positive class is the majority class for calibration sets T_r^c and is the minority class for the global proper training sets T^t). This implies that the global nonconformity function A is not very accurate which explains why the performance of IREP-ICP(G) is worse than that of IREP-ICP(L). The latter keeps the distribution of the local proper training set T_r^t and calibration sets T_r^c through stratified splitting, and, thus, the local nonconformity functions A_r are more accurate on the Spam data.

Results on Ten UCI Data Sets Table 4[9] contains the experimental results for ICP, IREP-ICP (L), and IREP-ICP (G) on all data sets from Table 1. From the tables we observe that for significance level $\epsilon \in \{0.01, 0.05, 0.1\}$

- the error rate e is smaller than or equal to ϵ for ICP, IREP-ICP (L), and IREP-ICP (G) up to some statistical fluctuations;
- the rates r^e of empty prediction sets for IREP-ICP (L) and IREP-ICP (G) are usually greater than or equal to those of ICP;
- the rates r^s of single prediction sets for IREP-ICP (L) are usually higher than those of ICP and IREP-ICP (G), especiallly for larger data sets;
- the rates r^m of multiple prediction sets for IREP-ICP (L) and ICP are usually lower than those of IREP-ICP (G);
- the error rates e^s of single prediction sets for IREP-ICP (L) and IREP-ICP(G) are usually lower than those of ICP.

[9] Multiple prediction set error rate e^m is excluded from the table as it equals 0.0.

Table 4: Public Data Sets Results

Set	ϵ	ICP					IREP-ICP (L)					IREP-ICP (G)				
		e	r^e	r^s	r^m	e^s	e	r^e	r^s	r^m	e^s	e	r^e	r^s	r^m	e^s
HC	0.01	0.01	0.0	0.069	0.931	0.143	0.003	0.0	0.05	0.95	0.067	0.003	0.0	0.026	0.974	0.125
	0.05	0.026	0.0	0.195	0.805	0.136	0.036	0.0	0.267	0.733	0.136	0.046	0.0	0.172	0.828	0.269
	0.1	0.086	0.0	0.386	0.614	0.222	0.099	0.0	0.538	0.462	0.184	0.096	0.0	0.386	0.614	0.248
HV	0.01	0.005	0.0	0.005	0.995	1.0	0.005	0.0	0.04	0.96	0.125	0.005	0.0	0.055	0.945	0.091
	0.05	0.04	0.0	0.13	0.87	0.308	0.035	0.0	0.255	0.745	0.137	0.06	0.0	0.285	0.715	0.211
	0.1	0.075	0.0	0.305	0.695	0.246	0.095	0.0	0.53	0.47	0.179	0.09	0.0	0.5	0.5	0.18
HM	0.01	0.003	0.0	0.007	0.993	0.5	0.0	0.0	0.039	0.961	0.0	0.013	0.0	0.059	0.941	0.222
	0.05	0.026	0.0	0.242	0.758	0.108	0.029	0.0	0.212	0.788	0.138	0.056	0.0	0.245	0.755	0.227
	0.1	0.075	0.0	0.458	0.542	0.164	0.069	0.0	0.425	0.575	0.162	0.108	0.0	0.471	0.529	0.229
SB	0.01	0.007	0.0	0.191	0.809	0.037	0.01	0.0	0.233	0.767	0.044	0.008	0.0	0.196	0.804	0.042
	0.05	0.047	0.0	0.489	0.511	0.097	0.063	0.002	0.692	0.307	0.089	0.053	0.0	0.537	0.463	0.099
	0.1	0.102	0.0	0.744	0.256	0.137	0.128	0.024	0.924	0.051	0.113	0.1	0.0	0.744	0.255	0.134
AC	0.01	0.01	0.0	0.049	0.951	0.206	0.01	0.0	0.177	0.823	0.057	0.01	0.0	0.071	0.929	0.143
	0.05	0.048	0.0	0.206	0.794	0.232	0.051	0.0	0.557	0.443	0.091	0.042	0.0	0.223	0.777	0.188
	0.1	0.104	0.0	0.443	0.557	0.235	0.107	0.016	0.87	0.114	0.105	0.109	0.0	0.438	0.562	0.248
C	0.01	0.007	0.0	0.568	0.432	0.012	0.007	0.0	0.23	0.77	0.031	0.004	0.0	0.16	0.84	0.022
	0.05	0.053	0.005	0.928	0.067	0.051	0.032	0.0	0.547	0.453	0.058	0.032	0.002	0.489	0.51	0.061
	0.1	0.093	0.033	0.967	0.0	0.062	0.074	0.04	0.938	0.021	0.036	0.1	0.044	0.926	0.03	0.061
I	0.01	0.006	0.0	0.473	0.527	0.012	0.003	0.0	0.425	0.575	0.007	0.003	0.0	0.41	0.59	0.007
	0.05	0.043	0.0	0.689	0.311	0.062	0.037	0.0	0.664	0.336	0.056	0.026	0.0	0.661	0.339	0.039
	0.1	0.074	0.0	0.769	0.231	0.096	0.1	0.014	0.849	0.137	0.101	0.094	0.011	0.849	0.14	0.097
H	0.01	0.0	0.0	0.032	0.968	0.0	0.0	0.0	0.045	0.955	0.0	0.0	0.0	0.071	0.929	0.0
	0.05	0.045	0.0	0.271	0.729	0.167	0.052	0.0	0.477	0.523	0.108	0.026	0.0	0.361	0.639	0.071
	0.1	0.058	0.0	0.452	0.548	0.129	0.097	0.0	0.665	0.335	0.146	0.052	0.0	0.503	0.497	0.103
GC	0.01	0.01	0.0	0.077	0.923	0.13	0.011	0.0	0.077	0.923	0.143	0.01	0.0	0.064	0.936	0.156
	0.05	0.038	0.0	0.214	0.786	0.178	0.048	0.0	0.273	0.727	0.176	0.045	0.0	0.27	0.73	0.167
	0.1	0.079	0.0	0.422	0.578	0.187	0.093	0.0	0.488	0.512	0.191	0.087	0.0	0.453	0.547	0.192
IL	0.01	0.003	0.0	0.017	0.983	0.2	0.007	0.003	0.021	0.976	0.167	0.009	0.003	0.036	0.961	0.143
	0.05	0.039	0.007	0.276	0.717	0.118	0.034	0.005	0.144	0.851	0.202	0.036	0.005	0.178	0.816	0.173
	0.1	0.089	0.007	0.484	0.509	0.17	0.079	0.015	0.381	0.604	0.167	0.11	0.029	0.419	0.552	0.193

From the above we may conclude that for significance level $\epsilon \in \{0.01, 0.05, 0.1, 0.2\}$ on the experimental data:

- ICP, IREP-ICP(L), and IREP-ICP(G) are valid class set predictors; i.e. they comply with the theory of conformal prediction.
- ICP is more informationally efficient than IREP-ICP(G).
- IREP-ICP(L) is more informationally efficient than IREP-ICP(G). Its superiority grows with the size of the data.
- IREP-ICP(L) is comparable with ICP in terms of informational efficiency (i.e. there is no clear winner although IREP-ICP has more wins).

From the above we provide the following recommendations:

- ICP and IREP-ICP(L) can be used interchangeably for reliable prediction. However, if interpretability is need, IREP-ICP(L) has to be employed.
- IREP-ICP(G) can be used for relatively small data sets when the number of final rules is small. If this is not the case IREP-ICP(L) has to be preferred.

7 Conclusion

This paper used the Mondrian scheme to integrate decision rules and conformal prediction. The result is a new technique of conformal decision rules capable of providing a point prediction, an explanation, and confidence values for all possible predictions plus a prediction set for any test instance.

An analysis of the Mondrian integration scheme showed that the global approach for computing the nonconformity scores can cause the label imbalance problem. To address this problem we proposed a local approach. We experimentally compared both approaches using conformal decision rules and showed when they can be used.

References

1. Boström, H., Johansson, U.: Mondrian conformal regressors. In: Proceedings of the 9th Symposium on Conformal and Probabilistic Prediction with Applications, COPA 2020. Proceedings of Machine Learning Research, vol. 128, pp. 114–133. PMLR (2020)
2. Cohen, W.W.: Fast Effective Rule Induction. In: In Proceedings of the Twelfth International Conference on Machine Learning. pp. 115–123. Morgan Kaufmann (1995)
3. Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
4. Furnkranz, J., Gamberger, D., Lavrac, N.: Foundations of Rule Learning. Springer (2012)
5. Johansson, U., Linusson, H., Löfström, T., Boström, H.: Conformal prediction using decision trees. In: Proceedings of the 13th IEEE International Conference on Data Mining. pp. 330–339. IEEE Computer Society (2013)
6. Johansson, U., Linusson, H., Löfström, T., Boström, H.: Interpretable regression trees using conformal prediction. Expert Syst. Appl. **97**, 394–404 (2018)
7. Johansson, U., Sönströd, C., Löfström, T., Boström, H.: Rule extraction with guarantees from regression models. Pattern Recognition **126**, 1–9 (2022)
8. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Journal ofArtificial intelligence **267**, 1–38 (2019)
9. Molnar, C.: Interpretable Machine Learning: A Guide for Making Black Box Models Explainable (2022)
10. Papadopoulos, H., Proedrou, K., Vovk, V., Gammerman, A.: Inductive Confidence Machines for Regression. In: Proceedings of 13th European Conference on Machine Learning (ECML 2002). vol. 2430, pp. 345–356. Springer (2002)
11. van Prehn, J., Smirnov, E.N.: Region classification with decision trees. In: Proceedings of the 8th IEEE International Conference on Data Mining Workshops. pp. 53–59 (2008)
12. Shafer, G., Vovk, V.: A tutorial on conformal prediction. arXiv:0706.3188 [cs, stat] (Jun 2007), `http://arxiv.org/abs/0706.3188`, arXiv: 0706.3188
13. Toccaceli, P.: Introduction to conformal predictors. Pattern Recognition **124**, 108507 (Apr 2022)
14. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic learning in a random world. Springer, New York (2005)