# CRC: Consolidated Rules Construction for Expressive Ensemble Classification

Manal Almutairi[1], Frederic Stahl[2(✉)], and Max Bramer[3]

[1] Department of Computer Science, University of Reading, Reading, UK
manal.almutairi@pgr.reading.ac.uk
[2] DFKI Niedersachsen, Marine Perception, German Research Center for Artificial Intelligence GmbH (DFKI), Oldenburg, Germany
frederic_theodor.stahl@dfki.de
[3] School of Computing, University of Portsmouth, Portsmouth, UK
Max.Bramer@port.ac.uk

**Abstract.** Predictive modelling is one of the most important data mining tasks, where data mining models are trained on data with ground truth information and then applied to previously unseen data to predict the ground truth of a target variable. Ensemble models are often used for predictive modelling, since ensemble models tend to improve accuracy compared with standalone classification models. Although ensemble models are very accurate, they are opaque and predictions derived from these models are difficult to interpret by human analysts. However, explainability of classification models is needed in many critical applications such as stock market analysis, credit risk evaluation, intrusion detection, etc. A recent development of the authors of this paper is ReG-Rules, an ensemble learner that aims to extract a classification (prediction) committee, which comprises the first rule from each base classifier that fired. The rules are interpretable by humans, thus ReG-Rules is a step towards explainable ensemble classification. Since there is a set of matching rules presented to the human analyst for each prediction, there are still numerous rules that need to be considered for explaining the model to the human analyst. This paper introduces an extension of ReG-Rules termed Consolidated Rules Construction (CRC). CRC merges individual base classification models into a single rule set, that is then applied for each prediction. Only one rule is presented to the human analyst per prediction. Therefore, CRC is more explainable than ReG-Rules. Empirical evaluation also shows that CRC is competitive with ReG-Rules with respect to various performance measures.

**Keywords:** Ensemble learning · Rule-based classification · Explainable classifiers · Data mining

## 1 Introduction

Ensemble classification is the training of individual and diverse base classifiers and the combination of their predictive models into a unified classification model.


M. Bramer and F. Stahl (Eds.): SGAI-AI 2022, LNAI 13652, pp. 138–152, 2022.
https://doi.org/10.1007/978-3-031-21441-7_10

Ensembles are known to be generally more accurate than their individual models [7,12,13,15,23]. This is explained by the notion that combining the predictions of multiple learners can effectively remove high variance or high bias in predictions [9,15]. However, predictive learning models are required to be not only reliable and accurate, but also comprehensible to avoid the risk of irreversible misclassification, especially in many critical applications such as medical diagnoses, financial analysis, terrorism detection, etc. The use of ensemble approaches decreases the level of comprehensibility of the classification, as the human analyst is presented with a large number of different classification models [12,23]. This challenges the ability of decision makers to understand how a predictive ensemble system makes its predictions.

Therefore, this paper's contribution is a predictive ensemble learner that is both accurate and expressive at the same time. This is achieved by transforming the ensemble classification model into a consolidated expressive rule set, while preserving the predictive accuracy of the ensemble it is derived from. The level of expressiveness of the individual base learners is an important factor for improving the ensemble's explainability. This was one of the main reasons for choosing predictive rule learning approaches, as they are highly expressive and closer to 'white box models' than most other techniques. Another important reason is related to their ability to abstain from classifying a new instance when the algorithm is uncertain about a prediction. This aspect is needed to prevent costly false classifications. Nevertheless, the lower the abstaining rate, the better for most applications. Measuring the expressiveness of a rule-based learner often depends on the complexity of its rule set. A rule set is considered more expressive when it produces fewer rules with less complex terms per rule.

This paper is organised as follows: Sect. 2 discusses related work and summarises the authors' previous work on the ReG-Rules ensemble learner. Section 4 describes the proposed Consolidated Rules Construction (CRC) ensemble learner as a more explainable variation and extension of the ReG-Rules. Section 5 provides an empirical evaluation of the CRC ensemble learner, and concluding remarks including ongoing and future work are presented in Sect. 6.

## 2 Related Work

Ensemble methodology consists of a collection of base learners each trained on a different training subset and produces a single prediction (vote). Combining these individual predictions (decisions) using a some kind of voting approach is likely to create an ensemble with a higher level of overall predictive accuracy than its base learners [9,15]. The base learners are generated sequentially or hierarchically. The sequential paradigm leverages the concept of dependence between the individual classifiers. *Boosting*, in particular *AdaBoost algorithms* [11], is a well-known variant of this paradigm. In addition, numerous sequential ensemble techniques, such as the *Vote-boosting* algorithm [20] or the *SEL* framework [22], have recently been proposed. The parallel ensemble paradigm, on the other hand, relies on the independence and diversity of the base learners, because combining their separate decisions can effectively reduce the classification error [24]. The parallel ensemble paradigm is used in this study because it is parallelisable due to the base classifiers being independent, which can make the ensemble rule-based model more powerful in practice. *Bagging*, which stands for **B**ootstrap **aggr**egat**ing**, is a widely used parallel technique proposed by Breiman in [8]. Bagging aims to increase the stability and predictive performance of a composite classifier. It entails random sampling of data with replacement. Each classifier learns from a sample of instances that is statistically estimated to comprise 63.2% of the training data and gives one vote to the data instance is classifying The remaining 36.8% serve as test data. The final classification is usually determined by a vote, such as a majority vote or a weighted majority vote. The capacity to eliminate bias and variance in data is the main benefit of bagging [8,9].

*Random Forest* is also a popular independent ensemble method [9] based on decision trees. It can be considered as an extended version of Bagging. Random Forest essentially incorporates the basic Random Decision Forest approach, which is introduced by Ho in [14], with Bagging method [21,24]. The Random Forest algorithm builds multiple decision trees. Each tree is constructed using the whole training dataset in sub-spaces selected randomly from the feature space. *Random Prism* [21], is an ensemble learner not based on trees but on rule sets produced by PrismTCS algorithm [6]. It follows the parallel ensemble learning approach and takes a bootstrap sample by randomly selecting $n$ instances with replacement from the training dataset, where $n$ is the total number of training instances available. Random Prism outperforms its standalone base classifier in terms of accuracy and noise tolerance, as seen in [21]. However, although Random Prism generates highly explainable base learners, the analyst must manually review each base learner's rule set to obtain insight into the classifications.

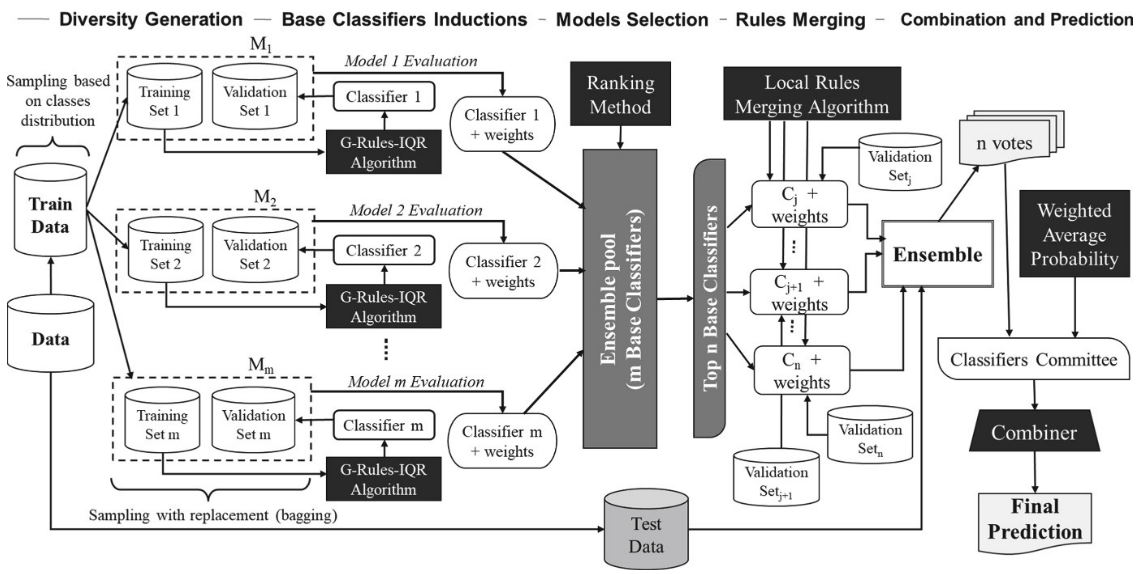


**Fig. 1.** The ReG-Rules learner framework

## 3 The ReG-Rules Ensemble Learner

ReG-Rules, a previous development of the authors of this paper, stands for Ranked ensemble G-Rules [3]. 'G' stands for Gaussian probability density distribution, which is used to build expressive base classifiers in G-Rules-IQR [2]. G-Rules-IQR [2] has been specifically developed for ReG-Rules and has shown in empirical experimentation to outperform similarly expressive rule based learners [1,4] in terms of accuracy, F1 score, tentative accuracy while producing more compact and easier to interpret rules. All the learners were evaluated against 5 metrics which are (1) the number of rules induced, (2) abstaining rate: the ratio of instances that remain unclassified, (3) F1 score: the harmonic mean of precision and recall, (4) accuracy: the ratio of correctly classified instances, (5) tentative accuracy: the ratio of correctly classified instances excluding abstained instances. G-Rules-IQR assumes normally distribute attributes and performs data transformations for non-normally distributed attributes. ReG-Rules provides an extract of the most relevant rules for each individual prediction, while preserving the predictive power of ensemble classifiers. ReG-Rules consists of 5 Stages as can be seen in Fig. 1:

**Stage 1: Diversity Generation:** The set of base classifiers should be diverse to assure producing uncorrelated errors and then obtain a more accurate ensemble [18,19,24]. Bagging [8] is applied to the training data to build local training and validation datasets to induce diverse base classifiers. The test data is only used to evaluate the final entire ReG-Rules ensemble.

**Stage 2: Base Classifiers Inductions:** $M$ G-Rules-IQR base classifiers are induced. A value of 100 for $M$ has performed well in ReG-Rules' empirical evaluation [3]. The out-of-bag samples produced by bagging are used to weight the performance of each individual base classifier. ReG-Rules uses a combination of metrics to calculate the weights: rule set size, number of correctly used rules (CUR), abstaining rate, accuracy and tentative accuracy.

**Stage 3: Models Selection:** Here three of the in Stage 2 mentioned metrics, namely tentative accuracy, CUR and abstaining rate, are used as ensemble selection criteria by ranking all the base classifiers accordingly. Only the top 20 base classifier models are retained, since 20 models produced consistently the best results in the empirical evaluation in [3].

**Stage 4: Rules Merging:** There is a possibility that for each rule set (of the 20 top ranked classifiers) some rules overlap. However, overlapping rules are generally unnecessary as they need to be tested at prediction stage and thus incur unnecessary additional testing cost [10]. ReG-Rules addresses this by providing a local rule merging method. This method is repeatedly applied to each base classifier in ReG-Rules.

**Stage 5: Combination and Prediction:** Combining classification results in ReG-Rules is based on weighted majority voting, however, not on a classifier level like most ensemble learners, but on an individual rule level. For this, ReG-Rules builds a committee of rules, termed Classification Committee (see Fig. 1), comprising the first rule that fires from each of the selected top ranked base classifiers. This committee derives a score for each possible classification as a combination of tentative accuracy, CUR, classifier vote frequency for certain classes.

In an empirical evaluation, ReG-Rules was compared against various rule based classifiers and exhibited a much better accuracy, tentative accuracy and low abstaining rate. The results also show that the local rule merging approach is very effective in lowering the total number of rules. A qualitative analysis revealed that ReG-Rules requires the human analyst to only examine a small set of relevant rules for each prediction, the classification committee [3].

## 4 The CRC Ensemble Learner

Although the committee in Stage 4 of ReG-Rules is much smaller than all the rules combined in ReG-Rules, the analyst still has to examine about 20 rules to extract information about the decision. Also, in ReG-Rules there is still the possibility that there are overlapping rules in the committee since rule merging is limited to local base learners. This section proposes an extension of ReG-Rules termed 'CRC', which is stand for Consolidated Rules Construction. The general structure of CRC as shown in Fig. 2 consists of five stages: (1) Diversity Generation, (2) Base Classifier Inductions, (3) Models Selections, (4) Stacking and Consolidation, (5) Prediction. The stages 1–3 are identical to the predecessor ReG-Rules and are summarised in Sect. 3. For a detailed description of Stages 1–3 the reader is referred to [3]. The new replaced Stages 4–5 are presented in the following sections by referring to the general framework (Fig. 2) and to the lines of code in Algorithm 1, which describe the CRC framework.

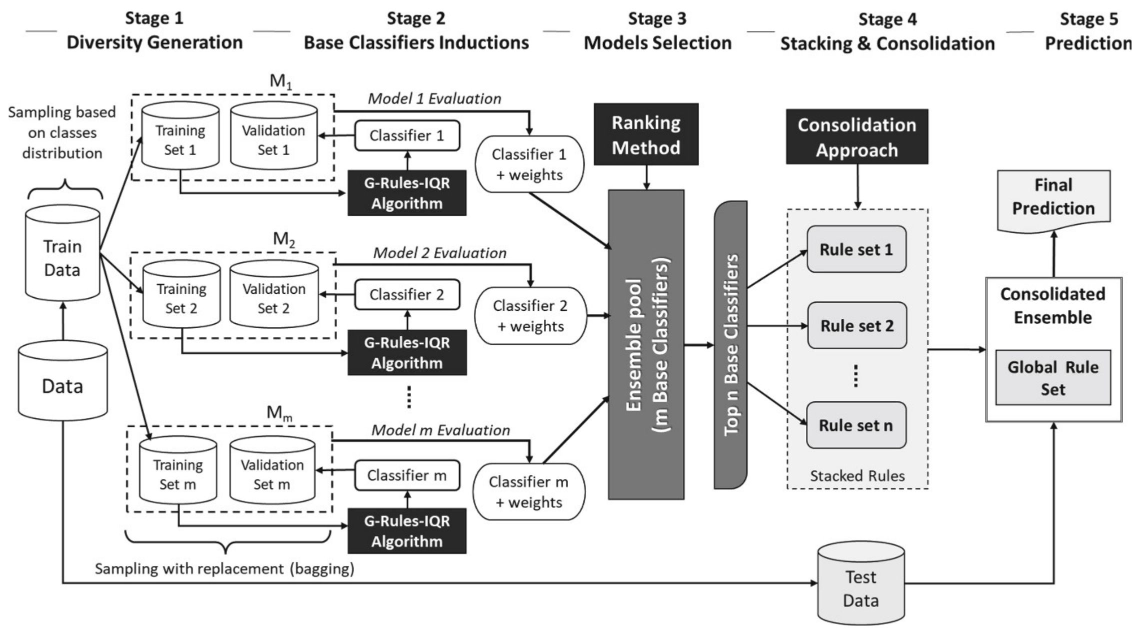


**Fig. 2.** The Consolidated Rules Construction (CRC) learner framework

**Algorithm 1:** Consolidated Rules Construction: CRC

**Notations**: $M$: Number of models, $S$: Training dataset, $V$: Validation dataset, $R$: rule set, $BC$: base classifier, $E_{pool}$: Ensemble Pool

1 Randomly sample dataset without replacement into train and test datasets (*train*, *test*) **for** $i = 1 \rightarrow M$ **do**
2 $\quad s_i \leftarrow$ a random sample of *train* dataset generated by Bagging method (sample with repalcement)
3 $\quad v_i \leftarrow$ out-of-bag set
4 $\quad$ Generate a base classifier $BC_i$ by applying Algorithm (G-Rules-IQR)[2] on $s_i$ dataset and learn a rule set $\rightarrow R_i$
5 $\quad$ Evaluate $BC_i$ performance by applying $R_i$ on $v_i$ dataset
6 $\quad$ Calculate a weight for each rule induced in previous line
7 $\quad$ Send $BC_i$ including its rule set weights to the ensemble pool $E_{pool}$
8 **end**
9 Rank all the base classifiers $BC$ collected in $E_{pool}$ according to the criteria described in Section 3 (stage 3)
10 Eliminate weak $BC$ by selecting the $n$ top models ($topBC$) ranked in the previous step according to the following if statement:
11 **if** *models selection type* = *defualt* **then**
12 $\quad n \leftarrow 20\%$ $M$ models
13 **else**
14 $\quad n \leftarrow$ selected models size defined by user
15 Select the top $n$ $BC$ models in line 9
16 $SR \leftarrow$ stack all the rule sets induced by the $n$ top models ($topBC$) in one large set
17 **Apply Algorithm 2** to the rule sets in $SR$ and produce a single consolidated rule set
18 Sort the individual rules in the consolidated set according to their quality
19 **return** *CRC Classifier*

### 4.1 Stacking and Consolidation Stage

In Stage 4, Rules 'Stacking and Consolidation', the issue of overlapping rules between all top ranked base classifiers is addressed in order to eliminate the problem of the analyst being confronted with potentially unnecessarily overlapping and redundant rules. In this new method, CRC learner compresses the top base classifiers into a single global rule-based model instead of locally merging each rule set independently. This is expected to enhance the expressiveness of the ensemble CRC learner to the point where it is similar to a common predictive rule-based classifier. The approach consists of the following two steps:

1. *Stacking*: The CRC learner collects all the top base classifiers' rule sets together into one large set. This is depicted in Fig. 2 as'stacked rules,' and is referred to as'$SR$' in Algorithm 1 (line 16). The essential concept behind stacking is to simply collect rule sets in the same order as their original ranked base classifiers, with no optimisation or filtering applied to the rules. As a result, there is no longer a requirement to preserve the base classifiers, and they are simply deleted at this level.
2. *Consolidation*: CRC learner combines a consolidation mechanism to perform the global merging process and provides a consolidated rule set as highlighted in Algorithm 1 (line 17). The method is termed CRC Consolidator.

*CRC Consolidator* - after removing the base classifiers and stacking their rules into one large set, the quality of each rule determines whether it will be preserved, improved, or even eliminated (individual weight). The process as shown

**Algorithm 2:** Consolidation Approach: CRC Consolidator

```
1  Initialise new GlobalRules set
2  for (i = 1 → SR ) do
3      OverlappedRules ← SR_i ;
4      for (j = 1 → SR [− OverlappedRules] ) do
5          if (SR_i and SR_j are identical rules) then
6              Skip current SR_j
7          else
8              OverlapExist ← Apply Algorithm 3 (Overlaps Checking) on SR_i and
               SR_j
9              if (OverlapExist = True ) then
10                 OverlappedRules ← ADD (SR_j)
11             end
12         end
13     end
14     if (OveralppedRules list contains rules other than SR_i) then
15         ConsoR ← empty          // a new consolidated rule intialisation
16         foreach ( α in OveralppedRules list) do
17             if (attribute α is categorical) then
18                 Create a rule-term α_j in the form (α = v) ;
19             else if (attribute α is continuous) then
20                 x ← smallest lower bound of α ;
21                 y ← largest upper bound of α ;
22                 Create a rule-term in a fom of (x < α ≤ y)
23             end
24             Append a rule-term built in lines 18 or 22 to the new consolidated rule
               ConsoR
25         end
26         GlobalRules set ← ADD (ConsoR)
27     end
28 end
29 return new GlobalRules set
```

in Algorithm 2 (CRC Consolidator) begins by initialising a new global rule set (line 1). Then each rule in $SR$ is checked against the replications and the overlaps. If two rules (e.g. $SR_1$, $SR_2$) are identical, one of them will be removed (line 6). Otherwise, $SR_1$ and $SR_2$ will be to considered as candidate overlapped rules. This is conditioned by the decision returned from Algorithm 3 (Overlap Checking), which is invoked by the CRC Consolidator in line 8 to carry out the examination. A decision (true/false) about the current rules examination is returned to the CRC Consolidator.

The CRC Consolidator then proceeds to line 10, where the current overlapped rules are examined for the final consolidation process, and a new iteration is initiated to examine two more additional rules until all of the rules in the stacked rule sets ($SR$) have been examined. Then, in line 14, a process of creating a new consolidated rule from a number of overlapped rules begins. The overlapping rules are first categorised by terms. The procedure will then continue, depending on the type of attribute in each term. First, the overlapped rules are grouped by terms. Then, depending on the type of attribute in each term, the process will continue. In case of categorical attributes, a new term is generated in the form $(\alpha = v)$ where $\alpha$ is the attribute name and $v$ is a discrete value that occurs in all the current overlapped terms. If the attribute type is continuous, a new term is generated in the form $(x < \alpha \leq y)$ where $x$ is the smallest lower bound presented in all the current overlapped terms and $y$ is the largest upper bound presented in the same overlapped terms. After the term is created, it will be appended to the new consolidated rule (line 24). Then, a new iteration of the next term will be started. Finally, in line 26, all the consolidated rules are added to the global rule set. The weight of each consolidated rule is estimated by averaging the weights associated to all the overlapped rules used in its generation.

**Algorithm 3:** Overlap Checking

```
1  Input: Rule1 (current rule), Rule2 (another rule)

2  if ( class label in Rule1 = class label in Rule2) and
3      ( all attributes α in Rule1 = all attributes α in Rule2) then
4      foreach attribute α ∈ Rule1 and Rule2 do
5          switch the type of α do
6              case Continuous
7                  if (lower bound of one rule includes the lower bound of the other and
8                  upper bound of one rule includes the upper bound of the other) then
9                      OverlapExist ← True
10                 else
11                     OverlapExist ← False
12
13             case Categorical
14                 if ( discrete value in Rule1 = discrete value in Rule2) then
15                     OverlapExist ← True
16                 else
17                     OverlapExist ← False
18
19             endsw
20         endsw
21         if (OverlapExist = False ) then
22             Exit the loop in line 4
23         end
24     end
25 else
26     OverlapExist ← False
27 end
28 return OverlapExist
```

## 4.2 Prediction Stage

As discussed in Sect. 2 combining multiple individual models' predictions promises a considerable increase in predictive accuracy compared with a single classifier. However, as mentioned in Sect. 4, ReG-Rules has a number of potential issues in training and testing phases. These are the number of base classifiers that need to be employed at prediction, and this consumes more processing overhead time prior to voting than applying a single model. Also the resulting vote on the prediction is harder to explain and justify by a human since there are several rules that need to be considered, i.e. in ReG-Rules the classification committee. In other words, the expressive power of ReG-Rules depends on the size of the classification committees and the complexity of the datasets. Both issues are removed or simplified in CRC, since CRC's learned model replaces the classification committees derived for each prediction by a single and global rule set re-used for each prediction. The first rule that matches the data instance to be classified is used to label the data instance, and at the same time this single rule serves as an explanation for the human analyst.

## 5 Empirical Evaluation of CRC Learning Model

The goal of the empirical evaluation is to evaluate the performance of the proposed CRC learner compared with ReG-Rules and G-Rules-IQR, which the stand-alone classifier used as base learner for both, ReG-Rules and CRC.

### 5.1 Experimental Setup

All the experiments were performed on a 2.9 GHz Quad-Core Intel Core i7 machine with memory 16 GB 2133 MHz LPDDR3, running macOS Big Sur version 11.4. All the 24 datasets used in the experiments were chosen randomly from the UCI repository [16], the only conditions being that they contain continuous attributes and involve classification tasks. The specifications of the datasets are highlighted in Table 1. Datasets 15, 16 and 24 included few missing values in continuous attributes. Missing values were replaced with the average value of the for the concerning attribute. Both ReG-Rules and CRC, and their base learning algorithm (G-Rule-IQR) have been implemented in the statistical programming language R [17]. The source code used to implement CRC algorithm is similar to that for ReG-Rules differing only in the methodological aspects described in Stages 4 and 5 described in Sect. 4.

The source code is available in a public online repository at https://github.com/ManalAlmutairi/PhD_Project_Codes/tree/v1.0.0 and is also archived at https://doi.org/10.5281/zenodo.5557590 [5].

All the algorithms are evaluated against 6 metrics for classifiers, which are: Number of Rules, abstaining rate, F1 score, accuracy, tentative accuracy and execution time. Execution comprises the time needed to complete all the training stages and to produce the final decisions. The remaining metrics were already described in Sect. 3. Please note that there is a relationship between accuracy, tentative accuracy and abstaining rate. Tentative accuracy simply ignores abstained instances, and accuracy treats abstained instances as potential misclassifications. Hence, the more a algorithm abstains, the higher the tentative accuracy and the lower the accuracy. The methodology used for experimentation with the 24 datasets is hold-out procedure; each dataset was randomly sampled without replacement into train and test datasets. While the 70% of the data instances were used to train and build the ensemble classifier, the remaining 30% were used as a testing dataset. In case of the ensemble models (ReG-Rules and CRC), the training dataset is used to generate multiple base classifiers using bagging, whereas the test set is used only once to assess the general performance of the classification models.

**Table 1.** Characteristics of the datasets used in the experiments

| No. | Dataset | No. attributes | No. classes | No. instances |
|---|---|---|---|---|
| 1 | Iris | 5 (4 cont) | 3 | 150 |
| 2 | Seeds | 8 (7 cont) | 3 | 210 |
| 3 | Wine | 14 (13 cont) | 3 | 178 |
| 4 | Blood transfusion | 6 (5 cont) | 2 | 748 |
| 5 | Banknote | 6 (5 cont) | 2 | 1,372 |
| 6 | Ecoli | 9 (7 cont, 1 name) | 8 | 336 |
| 7 | Yeast | 10 (8 cont, 1 name) | 10 | 1,484 |
| 8 | Page blocks | 11 (10 cont) | 5 | 5,473 |
| 9 | User modelling | 6 (5 cont) | 4 | 403 |
| 10 | Breast tissue | 11 (10 cont) | 6 | 106 |
| 11 | Glass | 11 (10 cont, 1 id) | 7 | 214 |
| 12 | HTRU2 | 10 (9 cont) | 2 | 17,898 |
| 13 | Magic gamma | 12 (11 cont) | 2 | 19,020 |
| 14 | Wine quality-white | 13 (12 cont) | 11 | 4,898 |
| 15 | Breast cancer | 12 (10 cont, 1 id) | 2 | 699 |
| 16 | Post operative | 10 (1 cont, 9 categ) | 3 | 90 |
| 17 | Wifi localization | 8 (7 cont) | 4 | 2,000 |
| 18 | Indian liver patient | 12 (10 cont, 1 categ) | 2 | 583 |
| 19 | Sonar | 62 (61 cont) | 2 | 208 |
| 20 | Leaf | 17 (15 cont, 1 name) | 40 | 340 |
| 21 | Internet firewall | 12 (cont) | 4 | 65,532 |
| 22 | Bank marketing | 17 (6 cont, 10 categ) | 2 | 45,211 |
| 23 | Avila | 11 (10 cont) | 12 | 20,867 |
| 24 | Shuttle | 10 (9 cont) | 7 | 58,000 |

### 5.2 Results and Interpretation

In each table, the # symbol refers to the index of the dataset in Table 1. The best result(s) in the tables for each dataset and metric are highlighted in bold letters. Table 2 shows the number of rules induced by each algorithm. Table 3 shows the comparison between CRC and ReG-Rules while Table 4 presents the comparison between CRC and G-Rules-IQR in these metrics, which will be discussed. Regarding the ‘number of induced rules’ and the ‘abstaining rates’ metrics listed in Table 2, it is not fair to compare the ensemble learners against the base classifier G-Rules-IQR. Therefore, CRC learner is only compared with ReG-Rules ensemble. As shown in the table, the number of consolidated rules produced by CRC is considerably smaller than the total number of rules produced by ReG-Rules in all datasets. In most cases, the size of the rules generated by CRC is reduced by 90%. Abstaining from classification, a typical problem of rule-based classifiers, was almost non-existent in both ensembles (ReG-Rules and CRC) compared with the stand-alone G-Rules-IQR’s abstaining rates, which were higher by more than 10% on several datasets compared with ReG-Rules and CRC. In four datasets (9, 10, 18 and 20) the abstaining rate in G-Rules-IQR reaches 30%, 19%, 18% and 40% respectively.

**Comparing with ReG-Rules Ensemble Learner:** Table 3 shows the comparison of the performance of CRC and ReG-Rules. The results of F1 score reveals that CRC performs equal or better than ReG-Rules in 13 out of 24 datasets. Also, CRC was very competitive on 4 out of the remaining datasets, on which it only underperformed by a maximum difference of 3%. Please note that the comparison between CRC and ReG-Rules in terms of overall accuracy and tentative accuracy are very similar. The results show that with respect to both metrics, CRC performs at the same level as ReG-Rules in 14 out of 24 cases. On 5 out of the remaining 10 datasets (2, 6, 11, 18 and 19) where CRC

**Table 2.** Number of rules and abstaining rates for CRC compared with ReG-Rules and G-Rules-IQR

| # | Number of rules | | | Abstaining rate | | |
|---|---|---|---|---|---|---|
| | G-Rules-IQR | ReG-Rules | CRC | G-Rules-IQR | ReG-Rules | CRC |
| 1 | *18* | 342 | **44** | 0.07 | **0.00** | **0.00** |
| 2 | *22* | 386 | **64** | 0.03 | **0.00** | **0.00** |
| 3 | *13* | 250 | **28** | 0.06 | **0.00** | **0.00** |
| 4 | *20* | 321 | **38** | **0.00** | **0.00** | **0.00** |
| 5 | *89* | 1630 | **128** | 0.02 | **0.00** | **0.00** |
| 6 | *37* | 649 | **107** | 0.02 | **0.00** | **0.00** |
| 7 | *99* | 1648 | **289** | 0.04 | **0.00** | **0.00** |
| 8 | *131* | 2348 | **570** | 0.02 | **0.00** | **0.00** |
| 9 | *57* | 901 | **162** | 0.30 | **0.00** | 0.01 |
| 10 | *28* | 485 | **87** | 0.19 | **0.00** | **0.00** |
| 11 | *30* | 505 | **72** | 0.11 | **0.02** | **0.02** |
| 12 | *35* | 521 | **57** | **0.00** | **0.00** | **0.00** |
| 13 | *79* | 1388 | **251** | **0.00** | **0.00** | **0.00** |
| 14 | *126* | 2289 | **243** | 0.02 | **0.00** | **0.00** |
| 15 | *11* | 186 | **28** | **0.00** | **0.00** | **0.00** |
| 16 | *29* | 451 | **105** | 0.11 | **0.00** | **0.00** |
| 17 | *59* | 955 | **159** | 0.01 | **0.00** | **0.00** |
| 18 | *47* | 996 | **368** | 0.17 | **0.00** | **0.00** |
| 19 | *16* | 270 | **30** | 0.13 | **0.00** | **0.00** |
| 20 | *124* | 2015 | **393** | 0.40 | **0.01** | 0.03 |
| 21 | *21* | 402 | **49** | **0.00** | **0.00** | **0.00** |
| 22 | *115* | 1977 | **428** | 0.01 | **0.00** | **0.00** |
| 23 | *158* | 3272 | **699** | 0.09 | **0.01** | **0.01** |
| 24 | *27* | 428 | **38** | **0.00** | **0.00** | **0.00** |
| **Average** | 58 | 1026 | 185 | 0.07 | 0.00 | 0.00 |
| **SD** | 45 | 846 | 186 | 0.102 | 0.004 | 0.01 |

did not achieve the highest accuracy and tentative accuracy, it was still very close within 3% of the best results compared with ReG-Rules learner. On one dataset (#5), the accuracy and tentative accuracy of CRC were lower than ReG-Rules by about 15%. However, this is also the dataset where the highest compression in the size of the rules is taking place. Regarding the learning time, Table 3 demonstrates that CRC learner is faster than ReG-Rules on all datasets. The decrease in learning times was up to 45% in some cases.

**Comparing with Stand-Alone G-Rules-IQR Learner:** The performance of CRC learning model is also compared with its stand-alone inducer (G-Rules-IQR algorithm). Table 4 shows the results of this comparison using F1 score, accuracy and tentative accuracy. CRC achieves the best F1 scores in 14 out of 24 datasets.

**Table 3.** Comparison of the performance of CRC and ReG-Rules using F1 score, Overall Accuracy, Tentative Accuracy and Learning Time

| # | F1 score | | Accuracy | | Tentative accuracy | | Learning time (sec.) | |
|---|---|---|---|---|---|---|---|---|
| | ReG-Rules | CRC | ReG-Rules | CRC | ReG-Rules | CRC | ReG-Rules | CRC |
| 1 | **0.93** | **0.93** | **0.93** | **0.93** | **0.93** | **0.93** | 112.8 | **94.8** |
| 2 | **1.00** | 0.97 | **1.00** | 0.97 | **1.00** | 0.97 | 232.2 | **192.6** |
| 3 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | 166.2 | **148.8** |
| 4 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | 358.8 | **322.2** |
| 5 | **0.99** | 0.87 | **0.99** | 0.84 | **0.99** | 0.84 | 3251.4 | **2939.4** |
| 6 | **0.91** | **0.91** | **0.95** | 0.92 | **0.95** | 0.92 | 384 | **360** |
| 7 | **0.91** | 0.83 | **0.97** | **0.97** | **0.97** | **0.97** | 3262.8 | **3096** |
| 8 | **0.87** | 0.82 | **0.98** | **0.98** | **0.98** | **0.98** | 10512 | **9612** |
| 9 | **0.95** | 0.87 | **0.94** | 0.86 | **0.94** | 0.87 | 733.2 | **631.2** |
| 10 | **0.97** | 0.91 | **0.97** | 0.91 | **0.97** | 0.91 | 245.4 | **228.6** |
| 11 | **0.93** | 0.90 | **0.95** | 0.94 | **0.97** | 0.95 | 264 | **247.2** |
| 12 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | 12600 | **12168** |
| 13 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | 18288 | **17100** |
| 14 | 0.87 | **0.99** | **1.00** | **1.00** | **1.00** | **1.00** | 12240 | **11880** |
| 15 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | 174.6 | **156** |
| 16 | **0.77** | **0.77** | **0.63** | **0.63** | **0.63** | **0.63** | 193.2 | **189** |
| 17 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | 2833.2 | **2635.8** |
| 18 | **0.84** | 0.82 | **0.73** | 0.71 | **0.73** | 0.71 | 2095.8 | **1951.8** |
| 19 | **0.97** | 0.96 | **0.97** | 0.95 | **0.97** | 0.95 | 897.6 | **864.6** |
| 20 | **0.67** | 0.61 | **0.56** | 0.46 | **0.57** | 0.47 | 2388 | **2125.8** |
| 21 | **0.90** | **0.90** | **1.00** | **1.00** | **1.00** | **1.00** | 71316 | **39276** |
| 22 | **0.99** | **0.99** | **0.98** | **0.98** | **0.98** | **0.98** | 41400 | **32940** |
| 23 | **0.93** | 0.86 | **0.92** | 0.86 | **0.92** | 0.86 | 119232 | **68292** |
| 24 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | 20808 | **17964** |
| **Average** | 0.93 | 0.91 | 0.94 | 0.91 | 0.94 | 0.91 | 13499.55 | 9392 |
| **SD** | 0.083 | 0.096 | 0.119 | 0.135 | 0.118 | 0.13 | 27874 | 16426 |

In the cases where CRC did not outperform G-Rules-IQR, its scores were only marginally lower. For example, the differences in 5 cases (1, 2, 7, 18 and 23) were less than 4%. In terms of overall accuracy, as can be seen in Table 4, CRC achieves the highest results in most cases (21 out of the 24 datasets). Moreover, CRC achieves the highest tentative accuracies in 14 out of 24 datasets compared with G-Rules-IQR classifier. CRC was also very competitive with G-Rules-IQR, in 7 out of the remaining 8 datasets their results are very close. On only one dataset (# 20), CRC's tentative accuracy was much lower than the stand-alone G-Rules-IOR. However, this dataset also causes the highest abstaining rate for G-Rules-IQR in the current experiments, and therefore it had been classified using the majority class label method. As mentioned in Sect. 5.1 the abstained instances are not considered in the tentative accuracy.

**Table 4.** Comparison of the performance of CRC and G-Rules-IQR using F1 score, overall accuracy and tentative accuracy

| # | F1 score | | Accuracy | | Tentative accuracy | |
|---|---|---|---|---|---|---|
| | G-Rules-IQR | CRC | G-Rules-IQR | CRC | G-Rules-IQR | CRC |
| 1 | **0.96** | 0.93 | 0.91 | **0.93** | **0.95** | 0.93 |
| 2 | **1.00** | 0.97 | **0.97** | **0.97** | **1.00** | 0.97 |
| 3 | 0.98 | **1.00** | 0.94 | **1.00** | 0.98 | **1.00** |
| 4 | 0.98 | **1.00** | 0.97 | **1.00** | 0.97 | **1.00** |
| 5 | **0.98** | 0.87 | **0.98** | 0.84 | **0.99** | 0.84 |
| 6 | **0.99** | 0.91 | **0.96** | 0.92 | **0.97** | 0.92 |
| 7 | **0.87** | 0.83 | 0.93 | **0.97** | 0.96 | **0.97** |
| 8 | **0.93** | 0.82 | **0.98** | **0.98** | **0.99** | 0.98 |
| 9 | **0.95** | 0.87 | 0.72 | **0.86** | **0.95** | 0.87 |
| 10 | 0.77 | **0.91** | 0.66 | **0.91** | 0.81 | **0.91** |
| 11 | 0.86 | **0.90** | 0.86 | **0.94** | **0.97** | 0.95 |
| 12 | 0.99 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| 13 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| 14 | 0.96 | **0.99** | 0.97 | **1.00** | 0.99 | **1.00** |
| 15 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| 16 | 0.49 | **0.77** | **0.67** | 0.63 | **0.67** | 0.63 |
| 17 | **1.00** | **1.00** | 0.99 | **1.00** | **1.00** | **1.00** |
| 18 | **0.83** | 0.82 | **0.71** | **0.71** | **0.71** | **0.71** |
| 19 | 0.95 | **0.96** | 0.87 | **0.95** | 0.94 | **0.95** |
| 20 | **0.75** | 0.61 | 0.39 | **0.46** | **0.65** | 0.47 |
| 21 | **0.90** | **0.90** | **1.00** | **1.00** | **1.00** | **1.00** |
| 22 | **0.99** | **0.99** | **0.98** | **0.98** | **0.98** | **0.98** |
| 23 | **0.89** | 0.86 | 0.85 | **0.86** | **0.88** | 0.86 |
| 24 | 0.99 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| **Average** | 0.92 | 0.91 | 0.89 | 0.91 | 0.93 | 0.91 |
| **SD** | 0.12 | 0.10 | 0.15 | 0.14 | 0.11 | 0.13 |

## 6 Conclusion

The current work aims to increase the expressive power of rule-based ensemble learning models while maintaining the key advantage of the ensemble learners, which is the high predictive accuracy compared with the stand-alone classifiers. A new approach was presented in this paper to compress the ensemble ReG-Rules [3] learner into a single global classifier, which can be used directly in predictions without the need to combine multiple classifiers' votes on every classification attempt. The best ranked classifiers are consolidated in a single global rule set. The proposed ensemble learner is therefore called Consolidated Rules Construction (CRC). CRC was empirically evaluated and compared with the ensemble ReG-Rules classifier and the stand-alone G-Rules-IQR classifier. Compared with ReG-Rules, CRC achieved its overall aim and outperformed ReG-Rules in all cases and in terms of number of rules that have been constructed and used for predictions. In most cases the reduction of rules reached 90%. Abstaining of classification was almost non existent in both ensemble classifiers. CRC exhibited a similar F1 score, overall accuracy and tentative accuracy compared with ReG-Rules. CRC outperformed ReG-Rules in terms of learning time in all cases, which reaches up to 45% learning time reduction in some cases. CRC also achieved the highest results in most cases in terms of F1 score, overall accuracy and tentative accuracy. Ongoing work comprises incorporating further diversification techniques by initialising base classifiers with different learning parameters. CRC is also highly parallelisable, since its base classifiers are induced independently. Therefore, future work comprises the development of a parallel CRC ensemble classification framework to scale up CRC to large datasets.

**Acknowledgements.** The research in this paper is partly supported by the Ministry for Science and Culture, Lower Saxony, Germany, through funds from the Niedersächsische Vorab (ZN3480).

## References

1. Almutairi, M., Stahl, F., Bramer, M.: Improving modular classification rule induction with g-prism using dynamic rule term boundaries. In: Bramer, M., Petridis, M. (eds.) SGAI 2017. LNCS (LNAI), vol. 10630, pp. 115–128. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71078-5_9
2. Almutairi, M., Stahl, F., Bramer, M.: A rule-based classifier with accurate and fast rule term induction for continuous attributes. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 413–420. IEEE (2018)
3. Almutairi, M., Stahl, F., Bramer, M.: Reg-rules: an explainable rule-based ensemble learner for classification. IEEE Access **9**, 52015–52035 (2021). https://doi.org/10.1109/ACCESS.2021.3062763
4. Almutairi, M., Stahl, F., Jennings, M., Le, T., Bramer, M.: Towards expressive modular rule induction for numerical attributes. In: Bramer, M., Petridis, M. (eds.) Research and Development in Intelligent Systems XXXIII, pp. 229–235. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47175-4_16

5. Almutairi, M.K.: ManalAlmutairi/PhD_Project_Codes: G-Rules-IQR. ReG-Rules and CRC, October 2021. https://doi.org/10.5281/zenodo.5557590
6. Bramer, M.: An information-theoretic approach to the pre-pruning of classification rules. In: Musen, M.A., Neumann, B., Studer, R. (eds.) IIP 2002. ITIFIP, vol. 93, pp. 201–212. Springer, Boston, MA (2002). https://doi.org/10.1007/978-0-387-35602-0_18
7. Bramer, M.: Principles of Data Mining, vol. 530. Springer, London (2016). https://doi.org/10.1007/978-1-4471-7307-6
8. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
9. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
10. Cendrowska, J.: PRISM: an algorithm for inducing modular rules. Int. J. Man Mach. Stud. **27**(4), 349–370 (1987)
11. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
12. Fürnkranz, J., Gamberger, D., Lavrač, N.: Foundations of Rule Learning. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-540-75197-7
13. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, Amsterdam (2011)
14. Ho, T.K.: Random decision forests. In: Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 1, pp. 278–282. IEEE (1995)
15. Johnston, B., Mathur, I.: Applied Supervised Learning with Python: Use Scikit-Learn to Build Predictive Models from Real-World Datasets and Prepare Yourself for the Future of Machine Learning. Packt Publishing, Birmingham (2019). https://books.google.co.uk/books?id=I_eVDwAAQBAJ
16. Lichman, M.: UCI machine learning repository (2013). https://archive.ics.uci.edu/ml
17. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2020). https://www.R-project.org, ISBN 3-900051-07-0
18. Rokach, L.: Ensemble-based classifiers. Artif. Intell. Rev. **33**(1–2), 1–39 (2010). https://doi.org/10.1007/s10462-009-9124-7
19. Rokach, L.: Ensemble Learning: Pattern Classification Using Ensemble Methods, vol. 85. World Scientific, Singapore (2019)
20. Sabzevari, M., Martínez-Muñoz, G., Suárez, A.: Vote-boosting ensembles. Pattern Recogn. **83**, 119–133 (2018)
21. Stahl, F., Bramer, M.: Random PRISM: a noise-tolerant alternative to random forests. Expert. Syst. **31**(5), 411–420 (2014)
22. Vong, C.M., Du, J.: Accurate and efficient sequential ensemble learning for highly imbalanced multi-class data. Neural Netw. (2020)
23. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2016)
24. Zhou, Z.H.: Ensemble Methods: Foundations and Algorithms. Chapman and Hall/CRC, Boca Raton (2012)