Building a Competitive Associative Classifier

Nitakshi Sood and Osmar Zaiane[0000-0002-0060-5988]

Alberta Machine Intelligence Institute
University of Alberta, Edmonton AB T6G2R3, Canada
{nitakshi,zaiane}@ualberta.ca

Abstract. With the huge success of deep learning, other machine learning paradigms have had to take back seat. Yet other models, particularly rule-based, are more readable and explainable and can even be competitive when labelled data is not abundant. However, most of the existing rule-based classifiers suffer from the production of a large number of classification rules, affecting the model readability. This hampers the classification accuracy as noisy rules might not add any useful information for classification and also lead to longer classification time. In this study, we propose SigD2 which uses a novel, two-stage pruning strategy which prunes most of the noisy, redundant and uninteresting rules and makes the classification model more accurate and readable. To make SigDirect more competitive with the most prevalent but uninterpretable machine learning-based classifiers like neural networks and support vector machines, we propose bagging and boosting on the ensemble of the SigDirect classifier. The results of the proposed algorithms are quite promising and we are able to obtain a minimal set of statistically significant rules for classification without jeopardizing the classification accuracy. We use 15 UCI datasets and compare our approach with eight existing systems. The SigD2 and boosted SigDirect (ACboost) ensemble model outperform various state-of-the-art classifiers not only in terms of classification accuracy but also in terms of the number of rules.

1 Introduction

Associative classifiers combine the concept of association rule mining and classification to build a classification model. In an associative classifier, for a rule in the form X $\rightarrow$Y, we choose the consequent(Y) of the rule to be the class label and the antecedent set(X) is a set of attribute-value pairs for the associated class label. In the literature, various associative classifiers have been proposed till now namely, CBA [11], CMAR [10], CPAR [14] etc. Although these classifiers are easily understandable, flexible and do not assume independence among the attributes, they require prior knowledge for choosing appropriate parameter values (support and confidence). Furthermore, the rules generated may include noisy and meaningless rules, which might hinder the classification. A rule is said to be noisy if it does not add any new information for prediction and instead misleads the classification model. In other terms, a noisy rule would participate more often in misclassifications than in correct classifications. The authors in [9] proposed SigDirect, an associative classifier which mines statistically significant rules without the need for the support and confidence values. However, in this paper, we propose SigD2 where we introduce a more effective two stage pruning strategy to obtain a more accurate classification model. The proposed method reduces the number of rules to be used for classification without compromising on the prediction performance. In fact, the performance is improved. Most of the prevalent supervised classification techniques like Artificial Neural Networks (ANN), Support Vector Machines (SVM) etc, although provide very high classification accuracy, they act as a black box. The models produced by such classifiers are not straight forwardly explainable. However, the proposed associative classifier makes the model more explainable by producing only a minimal set of classification association rules (CARs). The proposed technique finds its immense usage in various health-care related applications, where the explanation of proposed models along with the classification accuracy are highly significant. In health-care, incorrect predictions may have catastrophic effect, so doctors find it hard to trust AI unless they can validate the obtained results. Furthermore, we also propose ACboost, which uses an ensemble of classification models obtained from the weak version of SigDirect, for boosting. Our goal is to strengthen the classifier using less number of rules for prediction. Since, SigDirect is a strong learner and produces already a lesser number of rules for prediction, we form a weak version of SigDirect called wSigDirect, by further reducing the number of rules to be used for classification as explained later in Section 3. We also propose ACbag which is defined as bagging on an ensemble of wSigDirect classifiers. With the use of this strategy of combining weak learners, the goal is to decrease the variance in the prediction and improve the classification performance henceforth. It was found that for most of the datasets ACboost performs better than SigD2, ACbag, SVM, or ANN; ANN which performs similarly to deep neural network(DNN) on these reasonably sized datasets. The main aim of this study is to make associative classifiers more competitive and to highlight their significance as opposed to the other machine learning based classifiers like neural networks which do not produce explainable predictions. Our contribution in this study is as follows:

- We propose SigD2, an associative classifier, which uses an effective two stage pruning strategy for pruning the rules to be used for classification. Using the proposed approach, the number of rules used for classification are reduced notably, without compromising on the classification performance.
- We propose ACbag, an ensemble based classifier founded on wSigDirect.
- We also propose ACboost, which is boosting the wSigDirect classifier, to improve the classification accuracy with an explainable base model. Therefore, making SigDirect more competitive for classification tasks.

The rest of the paper is organized as follows: Section 2 gives a literature review about some previously proposed associative classifiers, Section 3 explains the methodologies we have adapted in SigD2, ACbag and ACboost, Section 4 shows the evaluation results of our proposed classifier on UCI datasets and lastly, Section 5 gives the conclusion of the work and directions about future investigations.

2 Related Work

In this section, we briefly describe some related work on associative classification. Stemming from association rule mining, associative classifiers have been extensively studied in the last two decades. Liu et al. first proposed the classification based on association (CBA) technique in [11] and showed that the association rule mining techniques could be applicable to classification tasks. CBA uses the Apriori algorithm to generate CARs and database coverage for pruning the noisy rules. It uses the highest ranked matching rules as the heuristic for classification. Inspired by the idea of CBA, many authors came up with more efficient versions of associative classifiers. CPAR proposed by Yin and Han uses a dynamic programming based greedy strategy that generates association rules from the training dataset [14]. It prevents repeated calculation in rule generation and also selects best k rules in prediction. The associative classifiers have the ability to provide a readable classification model. The study done by Zaiane et al. in [15] focuses on the significance of obtaining a minimal set of CAR's without jeopardising the performance of the classifier. They propose a pruning strategy to reduce the number of rules in order to build an effective classification model without seriously compromising on the classification accuracy. The authors also propose heuristics to select rules which obtain high accuracy on the plot of correct/incorrect classification for each rule on the training set for effective rule pruning combined with the database coverage technique based on the given dataset. Tuning values for support and confidence parameters is an arduous task as it varies with the change in dataset. Li and Zaïane in [9] overcome this limitation by proposing SigDirect that tunes only one parameter that is the p-value, which computes the statistical significance of rules using Fisher's exact test. The authors proposed an instance centric rule pruning strategy for pruning the non statistically significant rules. Although SigDirect has proved to be quite competitive in terms of prediction, there are still noisy rules that can compromise the accuracy. Furthermore, ensemble models are widely used for enhancing the accuracy of the classification models using a combination of weak learners. The SAMME algorithm proposed by Hastie et al. in [8] is a multi-class extension of the binary Adaboost algorithm [7].

3 Methodology

In this section, we introduce the details about the proposed effective pruning technique as used in SigD2. Further, we extend our work to perform bagging and boosting over the ensemble of wSigDirect associative classifier.

3.1 SigD2

The aim of an associative classifier is to find knowledge from data in the form of association rules associating conjunctions of attribute-value pairs with class labels, and then use these rules for prediction. SigD2 processes the learning of rules in rule generation and rule pruning phases. It further uses these rules for prediction in the classification phase.

Rule Generation phase: In this phase, we use the same approach proposed by Li and Zaïane for SigDirect in [9]. SigD2 also generates statistically significant CARs using the p-value from Fisher's exact test, of the rule in the form $X \rightarrow c_k$. The complete explanation of the generation process can be found in [9].

Algorithm 1: Algorithm for Two-Stage Pruning Strategy used in SigD2

```
Input: T: Pruning transaction database, R: Initial rule list from rule
generation phase, R_mid: Rule list being formed after pruning the insignificant
rules from R, conf_threshold: Confidence threshold value.
Result: R_new: Classification association rules to be used for prediction
while rules exist in R do
    Sort the rules in R in descending order of their confidence values
    Select the rule r_i with highest confidence from R and add to the R_mid
    if conf(r_i) < conf_threshold then
        break
    Find all applicable instances in T that match the antecedent of rule r_i
    if r_i correctly classifies a pruning instance in T then
        Mark r_i as a candidate rule in the classifier
        Remove all instances in T covered by r_i
    Update the confidence values, based on the remaining transactions
    Remove the rule r_i from the R
end
for each instance t in the original transaction database T do
    Scan the CARs from R_mid to find the matching CAR r_i, with highest
     confidence value
    if r_i ∉ R_new then
        R_new.add(r_i)
        r_i.count=1
    else
        r_i.count+=1
    end
end
```

Rule Pruning Phase: The rule generation phase may produce many CARs which are noisy and would not only slow down the process of classification but also lead to incorrect classification. Originally, SigDirect only performs instance based rule pruning on generated rules. It was observed that, although the previous strategy produces globally best CARs, the rules were still noisy and could be further reduced. So the question is, how can we prune more rules without actually jeopardising the accuracy of the associative classifier?

We propose a two stage strategy for pruning, wherein we randomly divide the training set into train set and prune set in the ratio of 2:1. The rules are generated in the rule generation phase using the train set. However, for pruning, only the prune set is used. We sort the CARs in descending order according to confidence values. The proposed pruning process, consists of matching the CAR with highest confidence and scanning over all the transactions in the pruning dataset to see if they match. If the rule applies correctly on the transactions, it is marked and is selected to be used for classification and subsequently the matching transactions are removed from the pruning set. We re-calculate the confidence values of the remaining rules, each time using the remaining transactions in the pruning set and arrange them in the descending order. This process is repeated until either the rules or transactions have been covered or until the confidence threshold is reached. It is assumed that for a rule, if the confidence value in each iteration is less than the threshold, then that rule can be pruned as it is not able to cover at least few instances in the prune set. After this step, we obtain the rules which might be useful for classification. However, we still need to find the globally best CARs. So, further we apply the instance based pruning step as proposed in SigDirect [9]. For every instance in the pruning transaction database, the complete set of CARs generated from the previous step are scanned. The aim here is to find the matching CARs with the highest confidence value, such that, the class label of the rule and the transaction matches and the antecedent of the rule is the subset of the transaction. Furthermore, the count of how many times the rule has been selected in the pruning instances is maintained. This is later used in order to perform weighted classification using the number of times the CAR was selected in the pruning phase. Using the proposed approach only high quality rules with high confidence values are kept. This pruning strategy also avoids over-fitting on the data.

Classification Phase: After the pruning phase, the minimal set of statistically significant rules is obtained. Further we make predictions on the new instances from the test set. For a given new instance, the classification process would search the subset of the CARs that match the new instance in order to predict its class label. The three heuristics used for classification are such that, for all the matching CARs, each class's group should be ordered on the basis of sum of ln(p-value), sum of confidence value and the sum of ln(p-value).confidence. Explanation of the classification process can be found in [9]. Furthermore, we can also use two stage classification as proposed by Sood et al. in [13], to learn with a NN in a second phase to predict the the classification rules to use.

3.2 Bagging and Boosting on wSigDirect

In this section, we perform bagging and boosting on the weak version of SigDirect, we call wSigDirect. While SigDirect is already a strong learner, we chose it over CBA as it gives a smaller number of rules. But we need to make it weaker to be used for ACbag and ACboost. We do this by further reducing the number of rules to be used for classification. The strategy for rule generation and rule pruning stays similar to that of the original SigDirect. However, for all the association rules obtained from the pruning phase for classification, we divide these rules as per the class label. Further, we chose the top η rules on the basis of highest confidence values from each class label group. The classification model thus obtained is called weak as it does not involve all the significant rules. We perform bagging and boosting on the ensemble model of wSigDirect over different trained datasets for prediction.

Bagging: ACbag is motivated by the approach proposed in [1]. The weak classifiers are learnt in parallel by picking instances randomly with replacement from the training data. Each wSigDirect model is learnt independent of each other. In bootstrap sampling, every observation has equal probability of appearing in the training dataset. Finally, we perform a majority voting over the results of the weak learners and predict the class label for each testing sample. Since, the base models are explainable, the ACbag can explain the responses of each learner, and the explanation of the ensemble would be the set of rules that were voted on by the ensemble. Furthermore, it was observed that the results obtained after performing bagging on wSigDirect are very comparable or slightly better than those achieved by bagging on the original SigDirect.

Boosting: Boosting is a process of improving the performance of a weak learning algorithm. It is done under the assumption that, the performance of the weak learner is at least slightly better than random guessing on different observations. In this phase, we propose ACboost which iteratively calls wSigDirect. This weak learner is converted to a strong learner either by weighted average of the predictions from weak learners or by considering prediction with majority voting. Given a training set, with features and class labels, we initialize the weights of our samples as one divided by the number of training instances. For the number of weak learners to be used sequentially, we train the first base learner using wSigDirect and obtain the misclassification error of the model. Further, the weight of the classifier is calculated based on its performance on the training data. Finally, the weight of each sample is updated, such that samples that were correctly classified are given less weights whereas the samples which were incorrectly predicted are given more weights. This would force the learner to pay more attention towards the incorrect predictions done by the previous learner. The iteration is continued till the maximum number of estimators (pre-set number of weak learners) are reached or a low training error is achieved. Finally, the prediction is done by using the weights of each classifier calculated previously to perform weighted prediction. This sequential learning of models helps in reducing the training error. We have used the methodology proposed for multi-class classification in SAMME algorithm [8], an extension of adaboost, which adds up a log term to the weight of the classifier making the boosting algorithm applicable for both two-class and multi-class classification tasks. Furthermore, since the rules produced by the base classifier are explainable therefore, there is a possibility of interpretation of results.

4 Experimental Results

We evaluate our SigD2 associative classifier on 15 UCI datasets [6]. We discretize the datasets as proposed in [2], so the classification accuracy might be marginally different from the previously reported results. We report the results after performing the average over 10 fold cross validation on each dataset. We use 90% of the total data as the train set and further divide the train set into train set and prune set in the ratio of 2:1.

Table 1: Comparison of classification accuracy of SigD2 with other rule-based classifiers

Datasets	**#cls**	**#rec**	**C4.5**	**CBA**	**CMAR**	**CPAR**	**RIPPER**	**SigDirect**	**SigD2**
Adult	2	48842	78.8	**84.2**	81.3	77.3	**84.1**	**84.1**	83.59
Anneal	6	898	76.7	94.5	90.7	95.1	**98.32**	96.99	**97.21**
Breast	2	699	91.5	**94.1**	89.9	93	**95.42**	91.7	92.7
Flare	9	1389	82.1	84.2	**84.3**	63.9	72.13	**84.23**	**84.3**
Glass	7	214	65.9	68.4	**71.1**	64.9	68.69	**70.56**	69.17
Heart	5	303	**61.5**	57.8	56.2	53.8	53.97	58.49	**59.81**
Hepatitis	2	155	84.1	42.2	79.6	75.5	78.06	**85.83**	**86**
Horse	2	368	70.9	78.8	82.3	81.2	**84.23**	81.23	**85.03**
Iris	3	150	91.3	93.3	94	94.7	**95.33**	94	**96**
Led7	10	3200	**73.9**	73.1	73.2	71.3	69.15	73.78	**73.81**
Mushroom	2	8124	92.5	46.5	**100**	**98.5**	**100**	**100**	**100**
PageBlocks	5	5473	92	90.9	90.1	**92.5**	**96.83**	91.21	92.18
Pima	2	768	70.5	74.6	74.4	74	66.36	**75.25**	**74.86**
Wine	3	178	71.7	49.6	92.7	88.2	91.57	**92.71**	**93.2**
Zoo	7	101	91	40.7	**93**	**94.1**	87.12	91	89.18
Average			79.62	71.52	83.52	81.2	82.75	**84.73**	**85.13**

Note- #cls indicates number of class labels and #rec indicate the number of records in dataset.

4.1 Classification Accuracy

We compare the performance of the proposed classifiers on 15 UCI datasets, with other rule-based classifiers like CBA, CMAR, CPAR, RIPPER, C4.5 and the original SigDirect, in terms of classification accuracy and number of classification rules in the final model. Further, we also compare ACboost with ANN and SVM in Table 2. We use the best parameters as stated by the authors in original respective papers as well as stated in [9]. In CBA and CMAR the parameters are tuned such that the minimum confidence values is set to be 50% , minimum value of support is set as 1%, the maximum number of CARs are limited to 80,000 and the size of number of antecedent items are limited to 6. The best parameters for RIPPER [3] are taken from [14]. The best parameters as stated in [9] are used for CPAR, C4.5 [12], SVM [4] and SigDirect, in order to have a fair comparison. For SigD2, we have performed a sensitivity analysis on the confidence threshold and it was found that threshold value lower than 30% or higher than 50%, does not lead to best results for all the considered datasets. Hence, we chose to vary the confidence threshold in the range of 30-50% depending on the dataset. For ANN, we use a shallow network with one hidden layer. The number of nodes in the hidden layer are set as the average of number of input and output nodes. The architecture may vary slightly with dataset, but we use ReLU (Rectified Linear Units) or sigmoid functions for activation and around 200 training epochs with a learning rate of 0.1. For ACboost and ACbag, the value of η is tuned in the range of 5-15 for every dataset. The number of estimators are varied in the range of 15-100 for each fold in every dataset and we report the best results. The value for parameters η and the number of estimators have been concluded after performing a sensitivity analysis on each of them. Table 1 shows that SigD2 performs quite

Table 2: Comparison of classification accuracy of ACboost with ACbag, SigD2, SigDirect, ANN and SVM

Datasets	**SVM**	**ANN**	**DNN**	**SigDirect**	**SigD2**	**ACbag**	**ACboost**
Adult	75.8	75.66	**85.35**	84.1	83.59	84.74	**85.23**
Anneal	85	93.964	**97.6**	96.99	97.21	**97.43**	97.31
Breast	95.7	**96.83**	**96.48**	91.7	92.7	93.86	92.62
Flare	73.8	**84.61**	70.3	84.23	84.3	84.31	**85.35**
Glass	68.6	70.148	66.9	70.56	69.17	**72.01**	**76.96**
Heart	55.4	56.72	55.6	58.49	59.81	**61.33**	**63.74**
Hepatitis	79.3	82.89	83.07	85.83	**86**	85.18	**90.89**
Horse	72.5	81.321	80.9	81.23	85.03	**85.3**	**85.7**
Iris	94.6	**98.09**	95.8	94	96	94.66	**97.33**
Led7	73.6	69.64	68.63	73.78	73.81	**74.84**	**75.21**
Mushroom	**99.8**	**100**	**100**	**100**	**100**	**100**	**100**
PageBlocks	91.2	**95.42**	95.08	91.21	**92.18**	91.24	92.13
Pima	74	**75.95**	75.15	75.25	74.86	75.53	**75.55**
Wine	94.9	91.662	**97.62**	92.71	93.2	94.04	**98.85**
Zoo	92.2	93.192	89.94	91	89.18	**94.28**	**98.9**
Average	81.76	84.406	83.89	84.738	85.136	**85.91**	**87.71**

Table 3: SigD2 compared with other algorithms based on number of rules

Datasets	**C4.5**	**CBA**	**CMAR**	**CPAR**	**SigDirect**	**SigD2**	**Difference with Average # of rules**
Adult	1176.5	691.8	2982.5	84.6	91.2	53.62	951.7 (94.67%)
Anneal	17	27.3	208.4	25.2	41.7	29.2	34.72 (54.31%)
Breast	8.8	13.5	69.4	6	10.9	7	14.72 (67.65%)
Flare	54.4	115.1	347.1	48.1	75.8	25.7	102.4 (79.93%)
Glass	14.8	63.7	274.5	34.8	55.6	23.1	65.58 (73.9%)
Heart	23.9	78.4	464.2	44	80.2	27.7	110.44 (77.3%)
Hepatitis	8.1	2.3	165.7	14.3	33.3	16	28.74 (64.23%)
Horse	25.6	116.4	499.9	19	90.4	41.5	108.76 (72.38%)
Iris	8.4	12.3	63.4	7.4	6.2	4.8	14.74 (75.43%)
Led7	63.2	71.2	206.3	31.7	104.3	54.4	40.94 (42.94%)
Mushroom	121.2	2	102.6	11.1	106.4	48.9	19.76 (28.77%)
PageBlocks	16.3	7.6	80.6	29.9	31.1	13.2	19.9 (60.12%)
Pima	24.4	43.2	203.3	21.7	36.6	11.3	54.54 (82.83%)
Wine	12.8	4.7	122.7	15.2	29.3	16.3	20.64 (55.87%)
Zoo	5.3	2	35	16.9	16.2	9	6.08 (40.31%)

well as compared to other rule-based and associative classifiers. The average performance over 15 datasets of SigD2 is better than all the other rule-based classifiers. Although, the difference between SigDirect and SigD2 on the basis of classification accuracy is marginal, when we compare the number of rules, we show that SigD2 outperforms SigDirect. In order to have a fair comparison, among different algorithms on various datasets, we analyse how many times did an algorithm win and how many times it was a runner up as shown in Table 4. The proposed pruning strategy is found to give quite promising results as compared to the other rule-based and associative classifiers. SigD2 outperforms

RIPPER on 10 out of 15 datasets. Furthermore, Table 2 shows that ACboost outperforms all the classifiers including SigDirect, SigD2, ANN and SVM. We have also tried to compare our approach with DNN with 5 hidden layers. Since most of the considered datasets are not big enough to be used for DNN, the results might not be conclusive.

4.2 Number of Rules

The main advantage of the associative classifiers over the other machine learning supervised classifiers is its ability to build a model which is human readable. Noisy, redundant and uninteresting rules lead to longer classification time, reduce the performance of the classifier and also make it tedious for humans to analyse the model. Ideally, we want to achieve maximum accuracy with a minimum possible set of rules. Table 3 shows the comparison among different classifiers on the basis of number of rules generated. The two stage pruning technique is found to give a minimum number of rules without compromising the classification performance. Table 5 clearly shows that out of 15 datasets, on average SigD2 outperforms most of the contenders for at least 10 datasets with some ties in few cases as well. CBA is found to have less rules for some datasets but it is unable to provide a high accuracy in such cases. Our proposed strategy outperforms CMAR on all datasets, the original SigDirect on all but one dataset and CPAR, C4.5 on 8 datasets. The number of rules is found to be appropriate enough to provide information about the classification model without compromising on the performance. In Table 3, we take the difference of the average of number of rules over all the other classifiers and the proposed classifier in the last column. It is found that the difference is substantial which essentially shows the significance of the proposed pruning strategy. We also compute the percentage decrease of the number of rules on average in Table 3. Furthermore, SigD2 is found to outperform RIPPER in terms of accuracy for most of the datasets, however, RIPPER obtains less rules comparatively. This is majorly because RIPPER greedily modifies the generated rules using the Minimum Discription Length (MDL) principle. RIPPER produces a kind of superset of rules covering all information required for classification in the form of intervals. This indicates that there is potential for further improvements. Furthermore, ACboost is said to be explainable as the base model called wSigDirect produces meaningful and readable rules. The ensemble model helps in determining the attributes which are of most indicative to determine a class. Consider the example of mushroom dataset, the rule produced will be in the format -: (habitat = leaves) and (cap-color = white) $\rightarrow$ (class = poisonous), where feature name 'habitat' has value 'leaves' and feature name 'cap-color' has value equal to 'white'. This rule along with other similar rules can be further used in the classification phase to determine whether a mushroom is poisonous or not. Similarly for ACbag, the readable rules from the base classifiers can help in interpreting the results.

4.3 Statistical Analysis

For better understanding the performance over various datasets, we use Demsar's method [5] to perform statistical tests in order to compare different algo-

Table 4: Best and runner-up counts comparison from (a) Table 1 and (b) Table 2 on the basis of classification accuracy

(a)

Classifiers	Best	Runner-up
C4.5	2	0
RIPPER	5	2
CBA	1	1
CMAR	3	1
CPAR	1	2
SigDirect	2	5
SigD2	6	4

(b)

Classifiers	Best	Runner-up
SVM	0	1
ANN	4	1
DNN	3	2
SigDirect	1	0
SigD2	1	2
ACboost	9	3
ACbag	1	6

rithms over different datasets. We perform non parametric Friedman's test for comparing the contenders with the proposed approaches. The Friedman's test on algorithms in Table 1 and Table 2 gave significant results as the p-value obtained is less than alpha (=0.05), which shows that at least one of the samples is significantly different from other samples. Furthermore, we also perform Wilcoxon's signed-ranks test which is another non-parametric statistical hypothesis test to compare the performances of proposed algorithms and the contenders in a pairwise manner. The results in Table 5 show that, SigD2 is significantly better than C4.5, CBA, CMAR, CPAR and SVM. However, the performance when compared with the original SigDirect seems to be quite similar and the p-value comes out to be greater than 0.05. We assume that, although there might not be difference in terms of classification accuracy, however, the new pruning strategy of SigD2 is more substantial and promising as it has reduced the number of rules to a small number as compared to the original SigDirect. The results from ACboost are found to be statistically significant than those of SigD2, ANN ,DNN and SVM as p-value is less than the significance level of 0.05. Thus, the results obtained in this section highlight the significance of the explainable models over the ones that are hard to interpret (ANN, DNN & SVM). SigD2 and ACboost are almost at par with other strong learners like neural network in terms of classification accuracy along with its ability to be interpreted using a limited number of rules.

Table 5: Statistical analysis of Table1 and Table 2

Classifiers	Wins	Losses	Ties	p-value
SigD2 vs C4.5*	12	3	0	0.005
SigD2 vs RIPPER	10	4	1	0.074
SigD2 vs CBA*	13	2	0	0.005
SigD2 vs CMAR*	11	2	2	0.033
SigD2 vs CPAR*	12	3	0	0.008
SigD2 vs SigDirect	10	4	1	0.272
SigD2 vs SVM*	12	3	0	0.041
SigD2 vs ANN	7	7	1	0.510
SigD2 vs DNN	7	7	1	0.510

Classifiers	Wins	Losses	Ties	p-value
ACbag vs SigD2*	11	3	1	0.064
ACbag vs SVM*	12	3	0	0.005
ACbag vs ANN	9	5	1	0.140
ACbag vs DNN	8	6	1	0.140
ACboost vs SigD2*	12	2	1	0.002
ACboost vs SVM*	14	1	0	0.002
ACboost vs ANN*	10	4	1	0.016
ACboost vs DNN*	10	4	1	0.022

(*) indicates statistically significant results with a p-value of 0.05.

5 Conclusion and Future Work

In this paper, we present a competitive associative classifier, which builds a rule-based model that is explainable, readable and minimalist. The classifier initially performs a rule generation step followed by a two phase rule pruning step to obtain the classification rules. The proposed rule pruning strategy reduces the rule set to a significantly small number.The proposed approaches are at par with the other supervised classifiers like ANN and SVM, which do not provide interpretable classification models. Furthermore, ACboost algorithm uses an ensemble of wSigDirect, to build a strong learner that boosts the prediction performance. The results obtained are very encouraging; we intend to use our proposed approach on various health-care related applications where explanation of prediction is required. Furthermore, since SigD2 produces human readable rules, we would like to study the possibility of injecting human expert knowledge to the obtained rules in order to further improve the prediction performance.

References

1. Breiman, L.: Bagging predictors. Machine learning **24**(2), 123–140 (1996)
2. Coenen, F.: The lucs-kdd software library (2004), http://cgi.csc.liv.ac.uk/∼frans/KDD/Software/
3. Cohen, W.: Fast effective rule induction. In: International Conference on Machine Learning, pp. 115–123. Elsevier (1995)
4. Cortes, C. and Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
5. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine learning research **7**, 1–30 (2006)
6. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
7. Freund, Y. and Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning. vol. 96, pp. 148–156 (1996)
8. Hastie, T., Rosset, S., Zhu, J. and Zou, H.: Multi-class adaboost. Statistics and its Interface **2**(3), 349–360 (2009)
9. Li, J. and Zaiane, O.R.: Exploiting statistically significant dependent rules for associative classification. In Intelligent Data Analysis **21**(5), 1155–1172 (2017)
10. Li, W., Han, J. and Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In: Intl. Conf. on Data Mining. pp. 369–376 (2001)
11. Liu, B., Hsu, W. and Y.: Integrating classification and association rule mining. In: International Conference on Knowledge Discovery and Data Mining (1998)
12. Quinlan, J.R.: C4.5: Programs for machine learning. Machine Learning **16**(3), 235–240 (1994)
13. Sood, N., Bindra, L., Zaiane, O.: Bi-level associative classifier using automatic learning on rules. In: Intl. Conf. on Database and Expert Systems Applications (2020)
14. Yin, X. and Han, J.: Cpar: Classification based on predictive association rules. In: SIAM International Conference on Data Mining. pp. 331–335 (2003)
15. Zaïane, O.R and Antonie, M.L.: On pruning and tuning rules for associative classifiers. In: Intl. Conf. on Knowledge-Based and Intelligent Information and Engineering Systems. pp. 966–973 (2005)