Iterative Reordering of Rules for Building Ensembles without Relearning [⋆]

Paulo J. Azevedo[1] and Alípio M. Jorge[2]

[1] Departamento de Informática, Universidade do Minho, Portugal pja@di.uminho.pt
[2] LIACC, Fac. de Economia, Universidade do Porto, Portugal amjorge@fep.up.pt

Abstract. We study a new method for improving the classification accuracy of a model composed of classification association rules (CAR). The method consists in reordering the original set of rules according to the error rates obtained on a set of training examples. This is done iteratively, starting from the original set of rules. After obtaining N models these are used as an ensemble for classifying new cases. The net effect of this approach is that the original rule model is clearly improved. This improvement is due to the ensembling of the obtained models, which are slightly better than the original one. This ensembling approach has the advantage of running a single learning process, since the models in the ensemble are obtained by self replicating the original one.

1 Introduction

The use of association rules for classification has proved to be a promising path in terms of improving predictive performance by enabling a wider search in the set of patterns supported by the data [12, 13, 15]. Given a set of association rules, using them in the best possible way to perform classification is a challenge proportional to the enormous number of rules that can be produced with reasonable computational resources. Recent work has exploited the use of low-cost ensemble learning (with a single learning process) to further improve the results of association rule classifiers [9]. The idea is to generate a first set of rules and then to obtain replications of this set by sampling it in a manner similar to bootstrap. The replications are then used as an ensemble.

In this paper we study another approach for generating ensembles using a single rule generation step. The main idea is to obtain the models by iteratively reweighting/reordering the rules of the original rule set. The initial rule model M_0 is obtained using a learning algorithm. In this initial model, each rule has an associated predictive value, which can be used to sort the rules for classification. Model M_1 is obtained by reweighting the rules on the training set. This reweighting can lead to a different rule ordering, if a decision list approach is used for model evaluation. Each model in the sequence M_i is obtained from the previous one in the same way, until we obtain N models. The ensemble $\{M_i, i = 1..N\}$

[⋆] Supported by Fundação Ciência e Tecnologia, Project Site-o-matic, FEDER e Programa de Financiamento Plurianual de Unidades de I & D.

is used to classify new cases. The intended effect is that rule ordering is recomputed taking into account global effects on accuracy, instead of local ones. We call this approach Iterative Reordering Ensembling (IRE).

As referred above, one particular feature of this ensemble approach is that the learning process that generates the rules runs only once. The sequence of models is obtained by finding close alternatives to the initial rule ordering. This process has some similarities to boosting [7, 19], where a sequence of models is generated from iteratively reweighted sets of examples. In boosting, the weights of the examples are changed, so that misclassified examples get higher weights.

In Iterative Reordering Ensembling, a new model is generated by changing the order of the rules, where rules with more errors go down. Thus, misclassified examples improve their chance of being well classified. The main advantage w.r.t. boosting is the fact that one single learning step is used, whereas in boosting there as many learning steps as models in the ensemble.

In the remaining of the paper we revisit the research done on classification with association rules and also on ensemble learning. We describe in detail this new approach and present an empirical evaluation. The results obtained indicate that IRE improves the predictive accuracy of classification with association rules mainly by reducing the bias component of the classification error.

2 Classification with AR

Association rules have been proposed for the first time as complete and competitive classification models by Liu *et al.* in 1998 [13]. In simple terms, the produced classifier was a decision list, and each new case was classified by the best rule that applied to it, i.e., the rule with highest confidence. Later, Li et al. [12] proposed the use of multiple rules, instead of just one, to classify each new case. The subset of rules that apply to the new case are grouped by anwered class, and each of these groups is assessed with a weighted χ^2 heuristic that tried to identify the strongest group. Meretakis and Wüthrich [15] suggested a well founded procedure to combine multiple rules by using the confidence of the rules to determine the most likely class for each case, in a kind of naïve Bayes approach with less independence assumptions. Jovanoski and Lavrac [10] have studied the effect of simple voting and other simple strategies to improve the prediction ability of a set of association rules. Jorge and Azevedo [9] have proposed an ensemble strategy based on multiple sets of association rules. The work presented here is a follow-up of that general approach.

2.1 Obtaining classifiers from Association Rules

We can regard classification from association rules as a particular case of the general problem of model combination. Either because we see each rule as a separate model or because we consider subsets of the rules for combination. We first build a set of rules R. Then we select a subset M of rules that will be used in classification, and finally we choose a prediction strategy π that obtains a decision for a given unknown case x. To optimize predictive performance we can fine tune one or more of these three steps.

Strategy for the generation of rules: A standard approach is to employ a sort of coverage strategy [13]. All association rules are derived. Then, one chooses the best rule, removes the covered cases and repeat the selection of rules until all cases are covered. In [12] this standard coverage strategy is generalised to allow more redundancy between rules. A case is only removed from the training data when it is covered by a pre-defined number of rules. In our work, we build the set of rules separately using the *Carenclass* system [9]. Carenclass is specialized in generating association rules for classification and employs a bitwise depth-first frequent patterns mining algorithm.

Choice of the rule subset: We can use the whole set of rules for prediction, and count on the predictive strategy to dynamically select the most relevant ones. Selection of rules is based on some measure of quality, or combination of measures. The structure of rules can also be used, for example for discarding rules that are generalizations of others. Discarding rules that are potentially irrelevant or harmful for prediction is called *pruning* [12, 13].

Strategy for prediction: Most of the previous work on using association rules for classification has been done on this topic. The simplest approach is to go for the rule with the highest quality, typically measured as confidence, sometimes combined with support [13]. Other approaches combine the rules by some kind of *committee method*, such as voting [10], or weighted voting [12].

Rule selection, or pruning, can be done right after rule generation. However, most of the rule selection techniques can be used before, when the rules are being generated. Pruning techniques rely on the elimination of rules that do not improve more general versions. For example, rule $\{a, b, c\} \rightarrow g$, may be pruned away if rule $\{a, c\} \rightarrow g$ has similar or better predictive accuracy. CBA [13] uses pessimistic error pruning. Another possibility is to simply use some measure of *improvement* [3] on a chosen rule quality measure. At modeling time we can still reduce the set of rules by choosing only the N-best ones overall, or the N-best ones for each class [10], where N is a user provided parameter. This technique may reduce the number of rules in the model dramatically, but the choice of the best value for N is not clear.

2.2 Combining the decisions of rules

In this section we describe the two simplest strategies for using association rule sets as classification models. In the discussion we assume we have a static set R of classification association rules, and a predefined set of classes G and that we want to classify cases with description x, where the description of a case is a set of statements involving independent attributes. The set of rules that apply to the case, or that fire upon the case with description x will be $F(x)$ defined as $\{(x' \rightarrow class = g) \in R \mid x' \subseteq x, g \in G\}$.

Best rule This strategy classifies using one single rule $bestrule_x$:

$$bestrule_x = arg \max_{r \in F(x)} measure(r) \tag{1}$$

The *measure* used is a function that assigns to each rule a value of its predictive power. *Confidence* is the natural choice when it comes to prediction. It estimates the posterior probability of C given A, and is defined as $confidence(A \rightarrow C) = sup(A \cup C)/sup(A)$.

Conviction is another interest measure [5] somewhat inspired in the logical definition of implication and attempts to measure the degree of implication of a rule. Conviction is infinite for logical implications (confidence 1), and is 1 if A and C are independent, and it sometimes outperforms confidence in terms of prediction [9]. It is defined as $conviction(A \rightarrow C) = (1 - sup(C))/(1 - confidence(A \rightarrow C))$.

The prediction given by the best rule is the best guess we can have with one single rule. When the best rule is not unique we can break ties maximizing support [13]. A kind of best rule strategy, combined with a coverage rule generation method, provided encouraging empirical results when compared with state of the art classifiers on some datasets from UCI [16].

Our implementation of Best Rule prediction follows closely the rules ordering described in CMAR [12]. Thus, R_1 is earlier than R_2 is defined as:

$$\begin{array}{l} R_1 \prec R_2 \quad if \\ \qquad int(R_1) > int(R_2) \quad or \\ \qquad int(R_1) == int(R_2) \wedge sup(R1) > sup(R2) \quad or \\ \qquad int(R_1) == int(R_2) \wedge sup(R1) == sup(R2) \wedge ant(R1) < ant(R2). \end{array}$$

where *int* is the used interest measure and *ant* is the length of the antecedent.

Weighted voting This strategy combines the rules $F(x)$ that fire upon a case x. The answer of each rule is a *vote*, and the final decision is obtained by assigning a specific weight to each vote, according to its perceived quality. In the case of association rules, this can be done using one of the above defined measure.

$$prediction_{wv} = arg \max_{g \in G} \sum_{x' \in antecedents(F(x))} vote(x', g). \max measure(x' \rightarrow g) \tag{2}$$

3 Iterative reordering

One possible way for increasing the accuracy of a CAR set is to re-evaluate the interest and support of a rule according to the its performance on a specific dataset. This new evaluation works by running the rules on the training set. Then, rule's interest is redefined according to its accuracy on this set. Rule's support is also redefined but as a measure of rule's usage in classification. The redefinition yields a new ordering on the original set of rules. This process can be applied iteratively, yielding a set of rule models that can be aggregated.

```
Input: training_set=$D$, max iterations = $MaxI$
Generate rule set $R$ from $D$;
Trial generation $Trial_0 = R$;
foreach $i$ in 1 to $MaxI$ do
    foreach $x$ in $D$ do
        using bestrule_measure approach see which rule r in $Trial_{i-1}$ fires;
        recomputes interest and support measures of winning rule in $R$ based on
        usage and hits;
    end
    $Trial_i$ = rules used from $Trial_{i-1}$, with new interest and support + rules
    from $Trial_{i-2}$ not used in $i-1$ + rules from $Trial_0$ not used in either $i-1$
    or $i-2$;
    (interest and support measures of rules from trials $i-2$, $i-1$ and 0 are the
    ones calculated there);
    $accuracy_i$ = accuracy of $Trial_i$ on $D$;
    If $accuracy_i < 0.5$ or $accuracy_i > 0.99$ break for;
end
Output: $Trials$
```

Algorithm 1: Iterative reordering trial generation

3.1 Ensemble generation

BestRule prediction is applied to the training dataset using the original CARules. From this application, rule's measures (support and interest) are updated according to usage and accuracy.

For instance, if confidence is used in BestRule prediction in $Trial_{i-1}$ then in $Trial_i$, the confidence of rule $A \to C$ is:

$$conf(A \to C, Trial_i, D) = \frac{hits(A \to C, Trial_{i-1}, D)}{usage(A \to C, Trial_{i-1}, D)}$$

where

$$hits(A{\to}C,Trial,D){=}\#\{x{\in}D|\ (A{\to}C){=}{=}BestRule(Trial,x) : x{\sqsupseteq}A \wedge x{\sqsupseteq}C\}$$

$$usage(A{\to}C,Trial,D){=}\#\{x{\in}D|\ (A{\to}C){=}{=}BestRule(Trial,x) : x{\sqsupseteq}A\}$$

and $BestRule(Trial, x)$ represents the best rule in $Trial$ that applies to x.

Other interest measures can be defined referring to $conf$ and support. For instance, conviction is defined as:

$$conv(A \to C, Trial_i, D) = \frac{1 - sup(C, D)}{1 - conf(A \to C, Trial_{i-1}, D)}$$

Notice that the minimal required information to represent trials (rule models) is the *usage* and *hits* associated with each rule. A matrix with $2 \times n$ (for n trials) is enough to represent the ensemble.

Prediction on test cases using the ensemble is obtained using BestRule for each case on each trial. The overal prediction is obtained by weighted trial voting. Different ensemble predictions can be obtained using different weighing strategies. For instance, the weight can be the global accuracy of the trial on the training set. Alternatively, the vote can be the interest measure of the best rule within the trial. In the sequel, the latter will be referred as $IRE.BR.int$ and the former as $IRE.V.Acc.int$, where *int* is either confidence or conviction.

On each trial, only rules with $usage > 0$ are considered. During ensemble formation it may be the case that there is no rule with positive usage in a given trial that covers a specific case. In that situation the BestRule prediction consults previous trials looking for rules that may cover the case. Line 8 includes this contingency mechanism.

Similarly to AdaBoost [7] trial construction, in line 11 the algorithm stops earlier either if a very good or very bad trial accuracy is achieved. Algoritm 1 summarizes the IRE ensemble construction process. Given n (number of trials) and a training set, the algorithm derives $Trial_0$ as the set of CAR rules through an association rule engine. Then, it iteratively derives $Trial_i$ applying BestRule to the training set using the rules from $Trial_{i-1}$. After n iterations, the ensemble construction is complete. A test set is evaluated by weight voting using best rule prediction on each trial.

4 Experimental validation

We have conducted experiments comparing the predictive performance of the *ensemble* approach with *bestrule with AR*, using different prediction measures (for assessing the net effect of this kind of ensembling) and state-of-the-art algorithms (for controlling the results). We have used 17 UCI datasets [16]. The datasets are described in Table 1. As a reference algorithm, we used the decision tree inducer *c4.5* [17]. Due to its availability and ease of use we have also compared the results with *rpart* from the statistical package *R* [18]. *Rpart* is a CART-like decision tree inducer [4].

Table 1. Datasets used for the empirical evaluation

Dataset	#examples	#classes	#attr	#numerics
australian	690	2	14	6
breast	699	2	9	8
pima	768	2	8	8
yeast	1484	10	8	8
flare	1066	2	10	0
cleveland	303	5	13	5
heart	270	2	13	13
hepatitis	155	2	19	4
german	1000	2	20	7
house-votes	435	2	16	0
segment	2310	7	19	19
vehicle	846	4	18	18
adult	32561	2	14	6
lymphography	148	4	18	0
sat	6435	6	36	36
shuttle	58000	7	9	9
waveform	5000	3	21	21

For the single model association rule classifiers, we used four carenclass variants, by combining two strategies: "Best rule" and "Weighted Voting" with two measures (confidence and conviction). Minimal support was set to 0.01 or 10 training cases. The only exception was the *sat* dataset, where we used 0.02 for computational reasons. Minimal improvement was 0.01 and minimal confidence 0.5. We have also used the χ^2 filter to eliminate potentially trivial rules. For each combination we ran carenclass with and without IRE-ensembles. Numerical attributes have been previously discretized using CAREN's implementation of Fayyad and Irani's supervised discretization method [6].

An estimation of the error of each algorithm (and carenclass variant) was obtained on each dataset with stratified 10-fold cross-validation (Table 2). From the estimated errors we ranked the algorithms separately for each dataset, and used mean ranks as an indication of global rank (Table 3). Besides that, we have tested the statistical significance of the results obtained.

Table 2. Average error rates obtained with the algorithms on the datasets (min. sup.=(0.01 or 10 cases, except sat with 0.02), min. conf.=0.5, imp.=0.01). Key: BR=best rule, V=Voting, IRE=Iterative Reordering, cf=confidence, cv=conviction, Acc=trial accuracy voting

	rpart	c4.5	BR.cf	BR.cv	V.cf	V.cv	IRE.BR.cf[3]	IRE.BR.cv	IRE.V.Acc.cf[4]	IRE.V.Acc.cv
aus	0.1623	0.1392	0.1378	0.1378	0.1871	0.1552	0.1318	0.1392	0.1333	0.1348
bre	0.0615	0.0500	0.0457	0.0428	0.0386	0.0386	0.0386	0.0500	0.0386	0.0357
pim	0.2472	0.2436	0.2278	0.2212	0.2277	0.2264	0.2329	0.2355	0.2316	0.2316
yea	0.4327	0.4427	0.4214	0.4194	0.4301	0.4240	0.4294	0.4435	0.4327	0.4395
fla	0.1773	0.1744	0.1914	0.2025	0.1810	0.1894	0.1932	0.1950	0.1932	0.2026
cle	0.4616	0.5004	0.4570	0.4570	0.4570	0.4570	0.4570	0.4587	0.4570	0.5021
hea	0.2000	0.2109	0.1778	0.1815	0.1741	0.1815	0.1778	0.1963	0.1741	0.1778
hep	0.2600	0.2132	0.1999	0.1870	0.1420	0.1878	0.1682	0.1823	0.1682	0.1761
ger	0.2520	0.3020	0.2820	0.2620	0.2570	0.2630	0.2710	0.2650	0.2730	0.2540
hou	0.0487	0.0325	0.0786	0.0786	0.1266	0.1334	0.0646	0.0668	0.0622	0.0622
seg	0.0831	0.0321	0.1190	0.1190	0.2030	0.1242	0.0779	0.0978	0.0801	0.0801
veh	0.3176	0.2596	0.3673	0.3662	0.3331	0.3342	0.3176	0.3272	0.3222	0.3234
adu	0.1555	0.1361	0.1873	0.1549	0.1735	0.1617	0.1599	0.2113	0.1592	0.1605
lym	0.2527	0.2307	0.1729	0.1800	0.2666	0.1989	0.1595	0.1729	0.1667	0.1667
sat	0.1904	0.1397	0.1975	0.1939	0.3514	0.2309	0.1523	0.1848	0.1510	0.1563
shu	0.0053	0.0005	0.0251	0.0075	0.0569	0.0083	0.0304	0.0168	0.0310	0.0048
wav	0.2664	0.2273	0.1774	0.1770	0.1990	0.1822	0.1654	0.2336	0.1650	0.1654

Table 3. Ranks for each algorithm on each dataset (1 is best, x.5 is a draw). Table lines are sorted by the mean rank, which can be found in the first column.

	mean	aus	bre	pim	yea	fla	cle	hea	hep	ger	hou	seg	veh	adu	lym	sat	shu	wav
IRE.Br.conf	4	1	3.5	7	4	6.5	3.5	4	2.5	7	5	2	2.5	5	1	3	8	2.5
IRE.Vote.Acc.conf	4.06	2	3.5	5.5	6.5	6.5	3.5	1.5	2.5	8	3.5	3.5	4	4	2.5	2	9	1
IRE.Vote.Acc.conv	4.5	3	1	5.5	8	10	10	4	4	2	3.5	3.5	5	6	2.5	4	2	2.5
Best.rule.conv	5.21	4.5	6	1	1	9	3.5	6.5	6	4	7.5	7.5	9	2	6	7	4	4
c4.5	5.53	6.5	8.5	9	9	1	9	10	9	10	1	1	1	1	8	1	1	8
Voting.conv	6.09	8	3.5	2	3	4	3.5	6.5	7	5	10	9	8	7	7	9	5	6
Voting.conf	6.15	10	3.5	3	5	3	3.5	1.5	1	3	9	10	7	8	10	10	10	7
Best.rule.conf	6.21	4.5	7	4	2	5	3.5	4	8	9	7.5	7.5	10	9	4.5	8	7	5
rpart	6.24	9	10	10	6.5	2	8	9	10	1	2	5	2.5	3	9	6	3	10
IRE.Br.conv	7.03	6.5	8.5	8	10	8	7	8	5	6	6	6	6	10	4.5	5	6	9

4.1 Analysis of results

The first strong observation is that the iterative reordering (IRE) approaches rank high, when compared to the other approaches. Of the 10 algorithms tested, the first three employ IRE. Separate experiments, not shown here, indicate that the IRE gains advantage through the ensemble strategy, rather than the filtering of rules. Of the two possibilities for combining the rules in the ensemble, the bestmodel approach works well with confidence but poorly with conviction. Of the two predictive measures used, confidence seems to be preferable for the top strategies, but not in general.

In terms of statisticall significance, the IRE ensemble approaches are clearly better than the single model AR classifiers. In a t-test with a significance of 0.01, IRE.Br.conf has 4 significant wins against 0 of Best.rule.conf. When compared to Best.rule.conv, the advantage is of 3 significant wins. Using the same statistical test, we see that the single model AR classifiers tend to be worse than rpart (Best.rule.conv looses 3/1), but the IRE ensemble best strategies beat rpart (2/0). c4.5 beats the best IRE strategy, in terms of significant wins, by 3/2.

By using Friedman's test on all the data on Table 2, we may reject the hypothesis that all the approaches have equal performance with a confidence of 0.05 (p-value is 0.033).

4.2 Method behavior

To understand why IR-ensembling improves the results of a bestrule classifier we have performed a bias-variance analysis as described in [11]. For each dataset we proceed as follows. We divide the examples in two sets D and E. This last set is used for evaluation and is a stratified sample, without replacement, with half the size of the original dataset. From the set D we generate 50 simple random samples, without replacement. Each one of these samples is used as training, and the results of the obtained models on E are used to estimate the contribution of the bias and of the variance to the global error. For each dataset we decompose the error into bias and variance for both strategies: bestrule and IR-ensembling. The parameters used were the same as in the experiments reported above, except for the minimum support. In this case, since the training sets were smaller (25% of the original set), we have lifted the admissible support of the rules to values that guarantee that at least 5 cases are covered (instead of 10, as we used above). In any case, support never goes below 0.01.

Figure 1 shows the results of the bias-variance analysis for 12 datasets. Each dataset has two bars, the left for best rule and the right one for IRE. The grey part of each bar corresponds to the bias component and the white part to the variance. In terms of the bias-variance decomposition, we can see that for 10 of the 12 datasets the bias component of the error visibly decreases. For the other 2 cases (flare and heart) there is at most a small increase in bias. The variance component tends to increase although not in the same proportion as bias.

We can thus hypothesize that the error reduction caused by IRE is mostly due to the reduction of the bias component. Since the variance component will

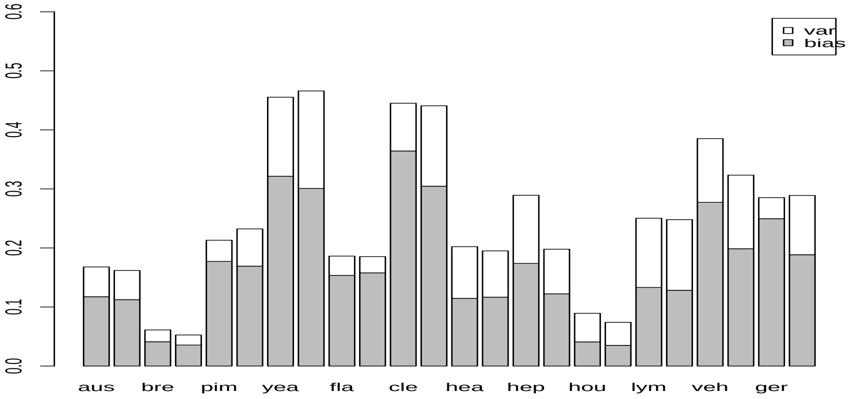

Fig. 1. The decomposition of bias and variance for 12 of the datasets. For each dataset, the bar on the left corresponds to the best rule approach and the bar on the right to the IRE.

converge to zero with the size of the datasets, IRE seems advantageous for large datasets. We should note that dealing with large datasets is not particularly complicated since the generation of association rules grows linearly with the number of examples [1]. The process of rule reweighting also grows linearly with the number of examples.

In another set of experiments, we have observed how the answers of the models in the ensemble compare with the answers given by the single model. In Figure 2 we can see the result for the *yeast* dataset. The xx axis represents the test examples and the yy axes the percentage of correct answers given by the two strategies for each case. In the case of the best rule, this percentage is either 0 (failure) or 1 (succes). In the case of the ensemble approach we have the percentage of models in the ensemble that gave the correct answer. The examples in the xx axis are sorted by the success of the best rule and than by the percentage of successes of the ensemble.

With this analysis we can see that there is a good number of “easy cases” and of “hard cases”. These are the ones at the right and left end of the plot, respectively. The cases in the middle are in a grey area. These are the ones that can be more easily recovered by IRE. To be successful, the IRE approach must recover more examples (improve the answer of the best rule) than the ones it loses (degrades the answer of the best rule). In the case of *yeast*, we can see that many case are recovered (crosses above the 0.5 horizontal line, and to the left of the vertical solid line), although some others are lost (below the horizontal line and to the right of the vertical one). In the case of the *heart* dataset (figure 2), similar observations can be made. Notice the small number of test examples that perform worst in the ensemble method than in the best rule prediction.

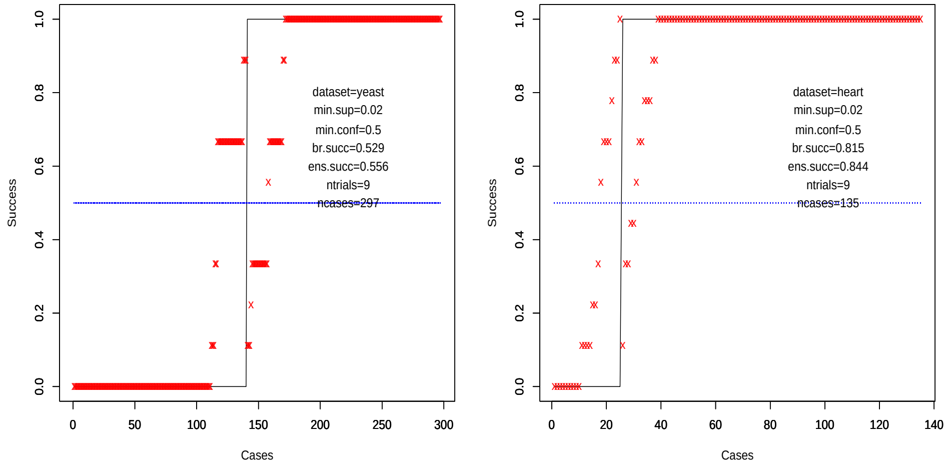

Fig. 2. Left: percentage of correct answers per test case for the *yeast* dataset (crosses) shown against the correct answers given by the best rule approach (solid crisp line). Right: Similar analysis for the *heart* dataset.

5 Discussion

Iterative Reordering Ensembling is a technique that produces replications of an original model without relearning. Each replication is a variant not very different from the original one. The obtained ensemble is thus more than homogeneous. Its elements are rather a result of jittering the original model in the version space. Still, the combined effect of these very similar models reduces the error consistently, when compared to using the single model, and in many datasets the reduction is significant.

The study of the bias variance decomposition indicates that IRE tends to reduce the bias component of the error, more than the variance. This is similar to what happens in boosting and in contrast to the case of bagging, where the reduction of the error is mainly due to the reduction in variance [2].

The intuitive explanation for the reduction of the bias component is that the single model best rule approach is tied to a particular rule ordering, and it is hard to find an ordering that maximizes the number of examples correctly classified. This constraint seems to be softened by combining similar versions of the rule set with different orderings.

6 Related Work

Ensemble learning has concentrated a large number of proposals in the literature. In [2] a study on the performance of several voting methods (including

Bagging and Boosting) was presented. A careful analysis of the bias/variance error decomposition is described as means to explain the error reduction yielded by the different voting methods variants.

A novel version of model aggregation obtained from bagging is described in [14]. The main idea is to derive an ordering on the models and to consider the top of the order. This is obtained by halting the bagging process earlier. Only a small part of the models is selected (15% to 30% of the total). This fraction of models are expected to perform best when aggregated.

A similar idea to ours is [8]. The author proposes an iterative version of Naive Bayes. The aim is to boost accuracy by iteratively updating the distribution tables yield from Naive Bayes to improve the probability class distribution associated with each training case. The end product is a single model rather then an ensemble. Our iterative updating of each rule predictive measure within each trial can be seen as a form of improving probability class distribution.

The work in [20] investigates the hypothesis that combining effective ensemble learning strategies leads to the reduction of the test error is explained by the increase of diversity. These authors argue that by trading a small increase in individual test error, a reduction in overall ensemble test error is obtained.

The Post-bagging ensemble method proposed in [9] employs a similar general strategy. Like IRE, Post-bagging derives replications that jitter around the original model. However, the combined effect of the similar models minimizes the test error but mostly due to a reduction on the variance component.

7 Conclusions

Classificaton using association rules can be improved through ensembling. We have proposed Iterative Reordering Ensembling (IRE), which is a procedure that generates multiple models with one single learning step. First, a rule set is obtained from the data. Then, replications of this initial set are obtained by iteratively recalculating the predictive measures of the rules in the set.

Experimental results with 17 datasets suggest that this ensembling technique improves best rule prediction and is competitive when compared to *rpart* and *c*4.5. The bias-variance decomposition indicates that most of the improvement is explained by a reduction of the bias component. This is possibly explained by the ability of the ensembling technique avoiding being tied to one particular ordering of the rules.

This kind of ensemble approach obtains multiple models by perturbing an original one. The resulting models are computationally unexpensive and atend to be similar to each other. Despite that low variety, their combination results in an effective improvement with respect to the single model.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB '94: Proceedings of the 20th International Conference on*

Very Large Data Bases, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

2. E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.
3. R. J. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. In *ICDE*, pages 188–197. IEEE Computer Society, 1999.
4. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, 1984.
5. S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In J. Peckham, editor, *SIGMOD Conference*, pages 255–264. ACM Press, 1997.
6. U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, pages 1022–1029, 1993.
7. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In P. M. B. Vitányi, editor, *EuroCOLT*, volume 904 of *Lecture Notes in Computer Science*, pages 23–37. Springer, 1995.
8. J. Gama. Iterative bayes. *Theor. Comput. Sci.*, 292(2):417–430, 2003.
9. A. Jorge and P. J. Azevedo. An experiment with association rules and classification: Post-bagging and conviction. In A. G. Hoffmann, H. Motoda, and T. Scheffer, editors, *Discovery Science*, volume 3735 of *Lecture Notes in Computer Science*, pages 137–149. Springer, 2005.
10. V. Jovanoski and N. Lavrac. Classification rule learning with apriori-c. In P. Brazdil and A. Jorge, editors, *EPIA*, volume 2258 of *Lecture Notes in Computer Science*, pages 44–51. Springer, 2001.
11. R. Kohavi and D. Wolpert. Bias plus variance decomposition for zero-one loss functions. In *ICML*, pages 275–283, 1996.
12. W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In N. Cercone, T. Y. Lin, and X. Wu, editors, *ICDM*, pages 369–376. IEEE Computer Society, 2001.
13. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *KDD '98: Proceedings of the fourth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 80–86, New York, NY, USA, 1998. ACM Press.
14. G. Martínez-Muñoz and A. Suárez. Pruning in ordered bagging ensembles. In W. W. Cohen and A. Moore, editors, *ICML*, pages 609–616. ACM, 2006.
15. D. Meretakis and B. Wüthrich. Extending naïve bayes classifiers using long itemsets. In *KDD*, pages 165–174, 1999.
16. C. J. Merz and P. Murphy. Uci repository of machine learning database. http://www.cs.uci.edu/∼mlearn, 1996.
17. J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.
18. R Development Core Team. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2004. ISBN 3-900051-00-3.
19. R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.
20. G. I. Webb and Z. Zheng. Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques. *IEEE Trans. Knowl. Data Eng.*, 16(8):980–991, 2004.