An Experiment with Association Rules and Classification: Post-Bagging and Conviction*

Alípio M. Jorge and Paulo J. Azevedo

[1] LIACC, Faculdade de Economia, Universidade do Porto, Rua de Ceuta, 118,
4050-090, Porto, Portugal
amjorge@fep.up.pt
[2] Departamento de Informática, Universidade do Minho, Portugal
pja@di.uminho.pt

Abstract. In this paper we study a new technique we call post-bagging, which consists in resampling parts of a classification model rather then the data. We do this with a particular kind of model: large sets of classification association rules, and in combination with ordinary best rule and weighted voting approaches. We empirically evaluate the effects of the technique in terms of classification accuracy. We also discuss the predictive power of different metrics used for association rule mining, such as confidence, lift, conviction and χ^2. We conclude that, for the described experimental conditions, post-bagging improves classification results and that the best metric is conviction.

1 Introduction

One can use an association rule discovery strategy to obtain a large set of rules from a given dataset, and subsequently combine a subset of the rules to obtain a classification model. This two-step training process is typically heavier than building directly a model, such as a decision tree. The motivation for going the long way lies on the possibility for delaying heuristic decisions in model building, while maintaining the scalability of the process. On the other hand, association rules can be seen as Bayesian statements about the data, and can be combined using Bayesian principles in a justified way.

As an example of the power of association based classifiers we can resort to a variant of the well known XOR two class problem, with three independent attributes (x, y and z) and one dependent attribute $class$, all taking values 0 or 1. The value of $class$ is 1 if and only if x and y have different values. Attribute z introduces noise. A heuristic method, such as decision tree induction, will tend to choose z as the root variable, and then possibly fail to discover the correct answer in all the branches. A technique based on association rules can discover the 4 rules that are needed to correctly classify a new example, independently of the values of z.

* Supported by the POSI/SRI/39630/2001/Class Project (Fundação Ciência e Tecnologia), FEDER e Programa de Financiamento Plurianual de Unidades de I & D.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 137–149, 2005.

Since at least 1997, some proposals have appeared that employed association rules to obtain classification models [2] [15][17][18][20]. Such classifiers have been empirically shown as competitive in terms of predictive power (although, in the case of the works cited above, no indication of the statistical significance of the results has been provided). In this paper we explore a variant of bagging [6] to obtain a classification model from a set of association rules. In classical bagging, a number of bootstrap samples are obtained from the given training examples. For each sample, a classification model is learnt, and new cases are classified by combining the decisions of the resulting models for the new case. Bagging is therefore an ensemble method that requires a single training data set and a single model generator algorithm.

We propose and empirically evaluate a *post-bagging* method. From the training data, we obtain one set of association rules, and from that single set of rules we build a number of (partial) classification models using a bootstrap sampling approach on the set of rules.

We compare this approach with the single best rule and voting approaches using different rule characterization metrics, and also with two decision tree methods (*c4.5* and *rpart*) on 12 datasets. The empirical results provide some evidence on the average predictive power of post-bagging.

2 Classification from Association

An association rule discovery algorithm such as APRIORI [1], takes a set of transactions $D = \{T \mid T \text{ is a set of items } i\}$, a minimal support threshold σ and a minimal confidence threshold ϕ, and outputs all the rules of the form $A \rightarrow B$, where A and B are sets of items in D and $sup(A \cup B) \geq \sigma$ and $sup(A \cup B)/sup(A) \geq \phi$. $sup(X)$ is the support or the relative frequency of an item set X observed in D.

Association rule discovery can be directly applied to tabular datasets, such as the typical UCI dataset, with one column for each attibute by regarding each example as a set of items of the form $< attribute = value >$. Likewise, continuous attributes can be dealt with if discretized in advance.

Despite the fact that an association rule algorithm finds ALL rules that satisfy σ and ϕ, the discovery process can be relatively fast and discovery time grows linearly with the number of examples (clearly shown in [1] for the algorithm AprioriHybrid). This provides a scalable heuristic-free process that makes possible to avoid greedy methods such as decision trees.

The discovery of association rules can then be seen as a step preceding model building, or a computationally feasible way of having a quasi-complete search on the space of rules. A classification rule model built from such an unrestrained set of rules can potentially be more accurate than another using a greedy search approach [17,18,20].

Which is the best way of obtaining a classification model from a set of association rules is, however, not entirely clear. One can look at the set of association rules as a large decision list ordered by confidence and support [18], or by some other metric. Rules can also be combined to classify new examples through some kind of voting [17] or by using Bayesian principles [20].

In the following, we state the problem of finding a good Classification model from Association Rules.

2.1 The Problem

The problem we approach in this paper consists in obtaining a classifier, or a discriminant model M, from a set of association rules R. The rules are generated from a particular data set D of cases T, where each case T is a set of pairs $< attribute = value >$, where *value* can be categorical or numerical. One of the attributes is the *class* attribute, ranging over a finite, and typically small, set G of classes. All the rules have exactly one item on the consequent involving the *class* attribute.

We want the model M to be successful in the prediction of the classes of unseen cases taken from the same distribution as D. A Bayesian view of the success of a classifier defines that the optimal classifier M_{Bayes} maximizes the probability of predicting the correct class value $g \in G$ for a given case x [11].

$$M_{Bayes}(x) = \max_{g \in G} \Pr(g \mid x) \qquad (1)$$

The success of a model M in estimating M_{Bayes} will depend on how the model is obtained from R and on how it is used to classify new cases from R. Given a case with description x, the confidence ϕ of an association rule $x' \rightarrow class = g$, with x' covering x, estimates the conditional probability $\Pr(g \mid x')$, and in the lack of more information it is a good estimator of $\Pr(g \mid x)$. The coverage relation is defined as: x covers x' iff $x' \subseteq x$ when x and x' are sets of items.

Previous work on classification from association rules has confirmed the predictive power of confidence. In this paper we provide empirical indication that another metric, *conviction*, obtains better results.

When we have a set R of association rules, we can expect to obtain more predictive power by combining different rules that apply to the same case. How to select the rules from R and how to use them is not trivial. In other words, given a rule set R, how do we obtain and use a classification model M?

3 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: The simplest choice is to run APRIORI [1] once over the data D. The choice of minimal support and confidence is not trivial. Constraints on other rule charateristics can be used. A more sophisticated approach is to employ a sort of coverage strategy [18]: Build all the association rules, choose the best, remove the covered cases and repeat until all cases are covered. In [17] 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 [4]. Carenclass is specialized in generating association rules for classification and employs a bitwise depth-first frequent patterns mining algorithm. It resembles the ECLAT algorithm proposed in [25], which is also a depth first algorithm that makes use of a vertical representation of the database.

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 their quality, or combination of measures. The structure of rules can also be used, for example for discarding rules that are generalizations of others. The general effort of discarding rules that are potentially irrelevant or harmful for prediction is called *pruning* [17][18].

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, where quality is typically measured as the confidence of the rule, sometimes combined with support [18]. Other approaches combine the rules by some kind of *committee method*, such as simple voting [14], or weighted voting [17]. In this paper we explore another possibility inspired in *bagging* [6].

4 Rule Generation

Typically, the generation of association rules is done after the identification of frequent itemsets. For efficiency purposes, it is desirable to push the rules generation task into the frequent pattern mining phase. Frequent itemset identification is typically done as follows: first, all frequent items are identified, and then candidate itemsets are generated following an imposed order. In the case of [1] this is a lexicographic order. Other, like [25], use a support oriented order. When we interleave frequent itemset counting and rule generation, as soon as a frequent itemset is counted and checked as valid (for instance, that it contains the required consequent item), rule generation for that itemset can be triggered. However, depth-first approaches to itemset mining face a problem. It may happen that subsets of the itemset in question are not yet determined due to unfavourable ordering. Thus, we might have a rule ready to be derived (because it already contains a consequent item) but that does not have its antecedent support already counted.

Carenclass has a simple and elegant approach to this problem. Since it knows in advance which items it will generate rules for (they will occur in the consequent) it imposes an itemset ordering that keeps the itemsets involving consequent items at the end. This ensures two things: first, consequent items appear at last in an itemset; secondly, when about to generate a rule, the subset of the itemset (without the consequent item) is already counted.

5 Rule Selection

Rule selection, or pruning, can be done right after rule generation. However, most of the rule selection techniques can be used earlier 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 [18] uses pessimistic error pruning. Another possibility is to simply use some measure of *improvement* [5] on a chosen rule quality metric. Using the same example as above, if we set a minimal confidence improvement of 0.1, we may discard $\{a, b, c\} \rightarrow g$ if its confidence is less than $confidence(\{a, c\} \rightarrow g) + 0.1$. In general, $improvement(A \rightarrow B)$ can be defined as $\min(\{metric(A \rightarrow B) - metric(A_s \rightarrow B) \mid A_s \subseteq A\}$, where *metric* is a rule characterization metric such as confidence.

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 [14], 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. The rule selection method *RC* [15] builds a decision list by traversing the generalization lattice of the rules and by looking at the training error of the rules. It starts with the most general rules, which will be at the bottom of the decision list. After that, it moves to the next level of the generalization lattice and chooses the rules that better handle the exceptions of the more general rules, while discarding the other rules at the same generalization level. This is done iteratively until the bottom of the lattice is reached.

6 Combining the Decisions of Rules

In this section we will analyze how association rules have been, and can be used for classification purposes, by studying the quality of the decisions produced. 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 $\{(x' \rightarrow class = g) \in R \mid x' \subseteq x, g \in G\}$.

Given a new case x to classify, we can use some prediction strategy to combine the rules in R.

6.1 Best Rule

This strategy tries to solve the problem with one single rule $bestrule_x$ obtained with:

$$bestrule_x = arg \max_{r \in F(x)} metric(r) \tag{2}$$

The *metric* used is a function that assigns to each rule a value of its predictive power. In this paper we study interest metrics typically used in association rule discovery: $confidence$, $conviction$, $lift$ and χ^2.

Confidence is the natural choice when it comes to prediction. It estimates the posterior probability of B given A, and is defined as $confidence(A \rightarrow B) = sup(A \cup B)/sup(A)$.

Lift is sometimes also called *interest* [8] and is a ratio between the observed support of $A \cup B$ and its expected support under the assumption that A and B are independent, $lift(A \rightarrow B) = sup(A \cup B)/(sup(A).sup(B))$. Under this assumption, the expected support is given by $sup(A).sup(B)$. Lift measures the deviation from independence of A and B. If lift is close to 1, A and B are independent, and the rule is not interesting.

Conviction is another interest metric [8] that also measures the independence of A and B, but goes a little bit further. Contrarily to lift, conviction is sensitive to rule direction $(lift(A \rightarrow B) = lift(B \rightarrow A))$. Conviction is 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 B are independent.

$$conviction(A \rightarrow B) = \frac{1 - sup(B)}{1 - confidence(A \rightarrow B)} \tag{3}$$

Another way of measuring the independence of antecedent and consequent of a rule is by testing that hypothesis with a χ^2 test [19]. If the value of the statistic (equation 4) is close to zero the hypothesis of independence is accepted. How close it must be to zero depends on the level of the significance of the test.

$$\chi^2(A \rightarrow B) = \mid D \mid \sum_{X \in \{A, \neg A\}, Y \in \{B, \neg B\}} \frac{(sup(X \cup Y) - sup(X).sup(Y))^2}{sup(X).sup(Y)} \tag{4}$$

where $\mid D \mid$ is the database size.

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 [18]. 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 [21].

However, the decision of a single rule is optimal only if we have a rule $x \rightarrow class = g$ that uses all the information in the description of the case. In general such a ‘complete’ rule has a very low support, most likely zero, and will not be available, or is not reliable. Therefore, we can expect to improve the quality of the prediction by using rules that use different sets of attributes in the antecedent. In [20] different rules have been combined to better approximate a Bayesian estimate of the probability of each class.

6.2 Voting

These strategies combine the rules $F(x)$ that fire upon a case x. A *simple voting* strategy takes all the rules in $F(x)$, groups the rules by antecedent, and for each antecedent x' obtains the class corresponding to the rule with highest confidence. We will denote the class voted by an antecedent x' with a binary function $vote(x', g)$ which takes the value 1 when x' votes for g, and 0 for the other classes.

$$prediction_{sv} = arg \max_{g \in G} \sum_{x' \in antecedents(F(x))} vote(x', g) \qquad (5)$$

6.3 Weighted Voting

This strategy is similar to voting, but each vote is multiplied by a factor that quantifies the quality of the vote [16]. In the case of association rules, this can be done using one of the above defined metric.

$$prediction_{wv} = arg \max_{g \in G} \sum_{x'} vote(x', g). \max metric(x' \rightarrow g) \qquad (6)$$

Carenclass implements these and other prediction strategies efficiently by keeping in an appropriate data structure [3].

In the next section we describe a technique for rule combination inspired in bagging.

7 Bagging Association Rules for Classification

Bagging is the generation of several models from bootstrap samples of the same original dataset D [6]. The prediction given by the set of resulting models for one example e is done by averaging the predictions of the different models. Bagging has the effect of improving the results of an unstable classifier by reducing its variance [11]. Domingos [9] suggests that, in the case of decision trees, bagging works because it increases the probability of choosing more complex models.

In the case of classification from association, we obtain a large set of rules R that contain many alternative possible models. So what we propose is the technique we call *post-bagging.* It consists in sampling repeatedly the set of rules a posteriori to obtain an ensemble of models similarly to bagging. The models in a particular ensemble will be similar, but their differences will tend to reflect the variability of rule sets obtained from the same source of data.

New cases are classified by obtaining the prediction of each of the models in the ensemble (and this can be done with any strategy), and using simple voting to combine those predictions. Experimental evaluation indicates that this technique can obtain good results when compared to a bestrule or a voting approach, or even to decision tree learners, such as *c4.5* [23] and *rpart* [13].

We will now describe the BAGGAR (Bootstrap Aggregation of Association Rules) algorithm (Algorithm 1) in detail. After obtaining a set of association rules R from a dataset D, we build a number of *baggs* from R. Each bagg is a sample with a pre-defined size of the rule set. Sampling is performed with replacement. The number of baggs ($n.baggs$) is 30 by default, and the size T of each bag is, in general, 10%. These defaults have been set in preliminary experiments and should not be regarded as necessarily ideal.

```
Algorithm 1. Baggar Algorithm, training
Given: a set $E$ with labelled examples; $n.bags$: the number of baggs (default 30);
$T$: size of each bag (default $\min(| R |, \max(50, 0.1\times | R |))$)
Do:
    1. Build a set $R$ of Association Rules
    2. For $i$ in 1 to $n.bags$
    3.     $S_i \leftarrow$ sample with replacement from $R$ of size $T$
Output: the set of baggs $\{S_i\}$
```

The classification of a single example e using a set of baggs $\{S_i\}$ is done by applying a chosen prediction strategy π to each of the baggs. The most voted class is then output as the overall prediction.

8 Empirical Evaluation

To test the value of post-bagging, we have compared different variants of *carenclass*, corresponding to different prediction strategies, on 12 UCI datasets [21]. To serve as a state of the art reference, we used the decision tree inducer *c4.5* [23]. Due to its availability and ease of use we have also compared the results with *rpart* from the statistical package *R* [13]. *Rpart* is a CART-like decision tree inducer [7].

We used eight carenclass variants, by combining two strategies: “Best rule” and “Weighted Voting” with four metrics (confidence, conviction, lift and χ^2). Minimal support was set to 0.02 and minimal improvement to 0.01. For each combination we ran carenclass with and without post-bagging. Numerical attributes have been previously discretized using Weka's [24] implementation of Fayyad and Irani's supervised discretization method [10].

An estimation of the error of each algorithm (and carenclass variant) was obtained on each dataset with a 10 × 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. Besides that, we have studied the statistical significance of the results obtained.

8.1 Post-Bagging Ranks High

The first empirical observation is that 3 post-bagging variants rank high among the four top places (Table 3). Compared to c4.5 and rpart, 5 carenclass variants

Table 1. Datasets used for the empirical evaluation

Dataset	#examples	#classes	#attr
australian	690	2	14
breast-wisconsin	699	2	9
cleveland	303	5	13
diabetes	768	2	8
flare	1066	2	10
heart	270	2	13
hepatitis	155	2	19
house votes	435	2	16
german	1000	2	20
segment	2310	7	19
vehicle	846	4	18
yeast	1484	10	8

Table 2. Average error rates obtained with the algorithms on the datasets (minimal support=0.02 and improvement=0.01)

	austr	breas	diabe	flare	cleve	yeast	house	germa	vehic	heart	hepat	segme
rpart	0.1504	0.0611	0.2572	0.1831	0.4566	**0.4276**	0.0481	0.2611	0.3213	0.1856	0.3044	0.0831
c4.5	0.1493	0.0512	0.2599	**0.1804**	0.4939	0.4408	**0.0343**	0.2862	**0.2690**	0.2205	0.2122	**0.0324**
Bestrule.conf	0.1432	0.0438	0.2279	0.1884	0.4587	0.4439	0.0770	0.2961	0.3968	0.1767	0.1794	0.1731
Bestrule.lift	0.3096	0.1890	0.4158	0.2179	0.6545	0.5237	0.4457	0.5865	0.4358	0.2270	0.5819	0.1752
Bestrule.conv	0.1409	0.0413	0.2236	0.2039	0.4466	0.4408	0.0770	0.2801	0.3961	0.1715	0.1794	0.1729
Bestrule.chi	0.1449	0.0635	0.2798	0.1978	0.4409	0.4558	0.0498	0.3059	0.4893	0.2522	0.3006	0.3315
Voting.conf	0.1800	0.0372	0.2301	0.1857	0.4306	0.4591	0.1236	0.2558	0.3590	0.1759	0.2314	0.2808
Voting.lift	0.1686	**0.0300**	0.2365	0.1936	0.4565	0.4448	0.1417	0.2592	0.3586	0.1707	0.3663	0.2805
Voting.conv	0.1542	0.0376	0.2244	0.1856	0.4272	0.4453	0.1101	**0.2465**	0.3437	**0.1596**	0.2108	0.1971
Voting.chi	0.1448	0.0388	0.2400	0.1913	0.4285	0.4397	0.1423	0.2683	0.3872	0.1767	0.2401	0.2914
Bag.Bestrule.conf	0.1351	0.0378	0.2271	0.1880	0.4345	0.4480	0.0723	0.2883	0.3696	0.1696	**0.1663**	0.2533
Bag.Bestrule.lift	0.2035	0.0571	0.3278	0.2104	0.5065	0.4764	0.2767	0.4322	0.3831	0.1681	0.5099	0.2515
Bag.Bestrule.conv	**0.1345**	0.0329	0.2246	0.1984	0.4361	0.4426	0.0739	0.2699	0.3702	0.1648	0.1672	0.2533
Bag.Bestrule.chi	0.1480	0.0499	0.2582	0.1968	**0.4176**	0.4488	0.1457	0.2961	0.4186	0.1800	0.2554	0.3196
Bag.Voting.conf	0.1810	0.0376	0.2283	0.1853	0.4326	0.4582	0.1220	0.2562	0.3597	0.1737	0.2314	0.2819
Bag.Voting.lift	0.1703	**0.0300**	0.2381	0.1939	0.4518	0.4427	0.1393	0.2567	0.3589	0.1707	0.3622	0.2823
Bag.Voting.conv	0.1394	0.0342	**0.2219**	0.1850	0.4287	0.4469	0.0778	0.2528	0.3437	0.1659	0.2048	0.2592
Bag.Voting.chi	0.1535	0.0399	0.2424	0.1906	0.4318	0.4435	0.1425	0.2636	0.3876	0.1778	0.2414	0.3043

rank higher than those. Although this is a good indication of the predictive power of post-bagging, we still have to discriminate its effect from the effect of the metric, and test its statistical significance.

To perceive the specific effect of post-bagging, we can observe that 5 (Voting.conv, Bestrule.conv, Bestrule.conf, Bestrule.chi and Bestrule.lift) against 3 (Voting.conf, Voting.lift, Voting.chi) of the carenclass variants benefit from post-bagging. The improvement is more visible on the simple Bestrule approach, rather than Voting. This may be explained by the fact that Voting is already a multi rule method.

We should note that the segment data set appears as a particularly difficult task for our association rule approaches. This is the data set, out of these 12, where the tree approaches perform visibly better. Moreover, it is also the only data set where post-bagging does not improve the results of best rule with confidence as a metric. In fact, post-bagging obtains very bad results. The segment data set has seven equally balanced classes. However, the number of rules per class tend to be unbalanced, which may be the reason for the higher error of the

Table 3. Ranks obtained (minimal support=0.02 and improvement=0.01)

	mean	austr	breas	diabe	flare	cleve	yeast	house	germa	vehic	heart	hepat	segme
Bag.Voting.conv	4.79	3	4	1	3	4	11	8	2	3.5	3	5	10
Voting.conv	5.42	12	6.5	3	5	2	10	9	1	3.5	1	6	6
Bag.Bestrule.conv	6.21	1	3	4	15	9	5	5	10	10	2	2	8.5
Bag.Bestrule.conf	6.88	2	8	5	7	8	12	4	13	9	5	1	8.5
Bestrule.conv	7.79	4	11	2	16	11	3.5	6.5	11	14	8	3.5	3
c4.5	8.04	9	14	15	1	16	3.5	1	12	1	16	7	1
rpart	8.17	10	16	13	2	14	1	2	7	2	15	14	2
Voting.conf	8.88	15	5	8	6	5	16	11	3	7	10	8.5	12
Bag.Voting.conf	9	16	6.5	7	4	7	15	10	4	8	9	8.5	13
Bestrule.conf	9.08	5	12	6	8	15	8	6.5	14.5	15	11.5	3.5	4
Voting.chi	9.38	6	9	11	10	3	2	14	9	12	11.5	10	15
Voting.lift	9.5	13	1.5	9	11	13	9	13	6	5	6.5	16	11
Bag.Voting.lift	9.5	14	1.5	10	12	12	6	12	5	6	6.5	15	14
Bag.Voting.chi	10.92	11	10	12	9	6	7	15	8	13	13	11	16
Bag.Bestrule.chi	12.62	8	13	14	13	1	13	16	14.5	16	14	12	17
Bestrule.chi	13.67	7	17	16	14	10	14	3	16	18	18	13	18
Bag.Bestrule.lift	14.42	17	15	17	17	17	17	17	17	11	4	17	7
Bestrule.lift	16.75	18	18	18	18	18	18	18	18	17	17	18	5

best rule approach: classes with more rules tend to get more votes. Similarly, the rule distribution per class is unbalanced in the produced baggs.

8.2 Conviction Ranks High

Four of the five top places are taken by carenclass variants that use conviction as a rule value metric. Moreover, conviction always has higher mean ranks than all the other metrics with respect to all the variants, and always higher than c4.5 and rpart. This predictive power of conviction is somewhat surprising and deserves to be better explained in the future. One possibility for the apparently good predictive performance of this metric may be due to the fact that it tends to favour less frequent classes. In particular, given two rules with the same confidence, conviction prefers the one whose consequent has lower support (Equation 3). The results with segment corroborate this intuition. This is a problem with 7 equally frequent classes. As a result, confidence and conviction practically have the same results. This is also observed on datasets with an almost balanced number of classes (australian, heart, vehicle).

The second best metric is clearly confidence. χ^2 and lift seem more or less equivalent in terms of results with a slight advantage to χ^2. Note that these two metrics are symmetric w.r.t. antecedent and consequent of the rule, contrarily to confidence and conviction.

8.3 Statistical Significance of Results

Although the average ranks provide a good overall picture of the results, these should be verified in terms of statistical significance. Our claims are based on statements of the form “algorithm x is better than algorithm y”, and “the tested algorithms perform equally”. To assess the statistical significance to such statements we will use paired *t-tests* and the *Friedman rank sum* test [22]. The *t-tests* are used as follows. For each partition of a dataset, we average the 10 error values obtained with a given algorithm from 10-fold cross validation. Since we have 10 different partitions we obtain 10 average errors. To compare two algorithms we perform an hypothesis test where the two samples are the average errors of each algorithm on the same dataset. The null hypothesis is that the algorithms perform equally. The alternative hypothesis is accepted if the p value for the t statistic is lower than 0.001. *Friedman* tests the hypothesis that all the methods have equal performance.

Table 4. Statistically significant ($\alpha = 0.001$) wins

	Bestrule				Voting			
	conv		conf		conv		conf	
	Bagging	Simple	Bagging	Simple	Bagging	Simple	Bagging	Simple
c4.5	3/4	4/3	3/5	3/2	3/5	3/5	4/5	5/5
rpart	3/3	4/3	5/3	4/3	3/3	2/3	5/3	5/3

If we compare directly post bagging and single model variants using *t-test*, we observe that statistically significant wins are not outstanding (Table 4). We mostly observe a near-draw with a slight advantage in favour of the post-bagging variants. However, if we separately compare post-bagging and the respective single model variant with c4.5, we observe a higher number of statistically significant wins of the post-bagging approach. With respect to rpart, post-bagging tends to improve the results of the single model variants. Compared with post-bagging, we observe an advantage of rpart, despite the fact that the direct comparison between rpart and c4.5 is favourable to the latter (4 wins against 2).

By using Friedman's test on all the data on Table 2, we reject the hypothesis that all the approaches have equal performance with very high confidence (p-value is lower than 10^{-7}). However, if we take out the carenclass variants that use lift and χ^2, p-value goes up to 0.13. Despite the good indications given by the ranking and the t-tests, and despite the fact that similar rankings are observed when the parameters of post-bagging are changed (number of baggs=30, 50, 70, 200; size of baggs=50%, minsup=0.01), we cannot firmly claim that there is a highly significative advantage in using post-bagging.

9 Conclusions

We have presented the technique of post-bagging, which produces an ensemble of rule classification from a single set of association rules. Post bagging has the advantage that a single model is built from the dataset and bootstrap models are built from this one. Empirical experiments indicate that post-bagging outranks on average standard decision tree techniques and tends to improve the results of bestrule, for the metrics considered. The effect of post-bagging on voting is only marginally positive, using confidence and conviction. We hypothesize that this is probably due to the fact that voting is already a multi rule decision method, and post-bagging has little room for improvement.

In terms of metrics, conviction tends to give better results than confidence, which is the second best metric. This is probably because class frequency is taken into account by conviction but not by confidence. The other two metrics (lift and χ^2) have been included for the sake of completeness but are far from being competitive.

We also observe that a simple Bestrule approach (generate rules-use best rule) gives competitive results: slightly better than c4.5 with conviction, slightly worse with confidence.

In conclusion, we can say that it is worthwhile to proceed with the research on post-bagging, and to better study the reasons for failure and success according to data set and rule set characteristics (number of classes, class distribution, number of rules, number of rules per class). This could lead us to improved classification accuracy and a better insight of the classification problem itself.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. I.: Fast Discovery of Association Rules, Advances in Knowledge Discovery and Data Mining: 307-328, (1996).
2. Ali, K., Manganaris, S. and Srikant, R.: Partial classification using association rules, Proceedings of the Third ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-97, ACM, 115-118, (1997).
3. Azevedo, P.J., A Data Structure to Represent Association Rules based Classifiers Technical Report, Universidade do Minho, (2005).
4. Azevedo P.J., Jorge, A.M., The CLASS Project, http://www.niaad.liacc.up.pt/~amjorge/Projectos/Class/
5. Bayardo, R.J., Agrawal, R., Gunopulos, D., Constraint-Based Rule Mining in Large, Dense Databases, Data Mining and Knowledge Discovery, Volume 4, Issue 2 - 3, Pages 217 - 240, (2000)
6. Breiman, L.: Bagging Predictors, Machine Learning, Vol. 24, No. 2, pp. 123-140, (1996).
7. Breiman, L., Friedman, J.H., Olshen, R. A., Stone, C. J. : Classification and Regression Trees. Wadsworth, (1984).
8. Brin, S., Motwani, R., Ullman, J. D., and Tsur,S., Dynamic itemset counting and implication rules for market basket data, Proceedings of th ACM SIGMOD International Conference on Management of Data, (1997).
9. Domingos, P., Why does bagging work? A Bayesian account and its implications, Proceedings of the Third ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-97, ACM, 115-118, (1997).
10. Fayyad, U.M., Irani, K. B.: Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, in Proceedings of the 13th International Joint Conference on Artificial Intelligence, R. Bajcsy (Ed.): Chambry, France, Morgan Kaufmann, pp. 1022-1029, (1993).
11. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference and Prediction, Series in Statistics, Springer, (2001).

12. Ho, T. K., Hull, J. J., Srihari, S. N.: Decision Combination in Multiple Classifier Systems, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(1), 66-75, (1994).
13. Ihaka, R., and Gentleman, R.: R: A Language for Data Analysis and Graphics, Journal of Computational Graphics and Statistics,Vol. 5, N. 3, pp. 299-314, (1996).
14. Jovanoski, V., Lavrac, N.: Classification rule learning with APRIORI-C, in Proc. of EPIA 2001, P. Brazdil, A. Jorge (Eds.), Springer, LNCS 2258, 44-51, (2001).
15. Jorge, A., Lopes, A.:Iterative Part-of-Speech Tagging, Learning Language in Logic, J. Cussens, S. Dzeroski (Eds), LNAI 1925, Springer-Verlag, (2000).
16. Kononenko, I.: Combining decisions of multiple rules, Artificial Intelligence V: Methodology, Systems, Applications, B. du Boulay, V. Sgurev (Eds.), Elsevier (1992).
17. Li, W., Han, J., Pei, J.: CMAR: Accurate and Efficient Classification Based on MultipleClass-Association Rules, in IEEE International Conference on Data Mining, (2001).
18. Liu, B., Hsu, W. e Ma, Y.: Integrating Classification and Association Rule Mining, Proceedings of the Fourth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 15-18, 1998, New York, USA. ACM, (1998).
19. Liu, B., Hsu, W. , Ma, Y., Pruning and Summarizing the Discovered Associations, Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 15-18, 1999, San Diego, CA, USA. ACM, 125-134, (1999).
20. Meretakis, D., Wüthrich, B.,: Extending Nave Bayes Classifiers Using Long Itemsets, Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 15-18, 1999, San Diego, CA, USA. ACM, pp. 165-174, (1999).
21. Merz, C. J., Murphy, P.:UCI Repository of Machine Learning Database. http://www.cs.uci.edu/∼mlearn, (1996).
22. Neave, H.R., Worthington,P.L.: Distribution-free tests, Unwin Hyman Ltd. , (1988).
23. Quinlan, J. R.: C4.5: Programs for Machine Learning, Morgan Kaufmann, (1993).
24. Witten, I. H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, (1999).
25. Zaki, M.J., Scalable algorithms for association mining, IEEE Transactions on Knowledge and Data Engineering, 12(3):372-390, (2000).