

# Maximizing the Area under the ROC Curve with Decision Lists and Rule Sets

Henrik Boström\*

henrik.bostrom@his.se

School of Humanities and Informatics

University of Skövde

541 28 Skövde

Sweden

## Abstract

Decision lists (or ordered rule sets) have two attractive properties compared to unordered rule sets: they require a simpler classification procedure and they allow for a more compact representation. However, it is an open question what effect these properties have on the area under the ROC curve (AUC). Two ways of forming decision lists are considered in this study: by generating a sequence of rules, with a default rule for one of the classes, and by imposing an order upon rules that have been generated for all classes. An empirical investigation shows that the latter method gives a significantly higher AUC than the former, demonstrating that the compactness obtained by using one of the classes as a default is indeed associated with a cost. Furthermore, by using all applicable rules rather than the first in an ordered set, an even further significant improvement in AUC is obtained, demonstrating that the simple classification procedure is also associated with a cost. The observed gains in AUC for unordered rule sets compared to decision lists can be explained by that learning rules for all classes as well as combining multiple rules allow for examples to be ranked according to a more fine-grained scale compared to when applying rules in a fixed order and providing a default rule for one of the classes.

## 1 Introduction

There has recently been a growing interest in using rule learning methods for maximizing the area under the ROC curve (AUC) [9, 16, 20]. A major reason for using AUC as an alternative to accuracy, which so far has been the most commonly used criterion for comparing rule learning methods, is that it is not sensitive to differences between the class distribution within the training examples and within the examples on which

the model is applied [4, 22]. This means that by using AUC instead of accuracy when comparing models, one is less likely to be misled when choosing a model due to having evaluated the model on a skewed sample. As noted in [4], the AUC can be interpreted as the probability of ranking a true positive example higher than a false positive when ordering examples according to decreasing likelihood of being positive.

In separate-and-conquer rule learning [13], two types of model have traditionally been considered: decision lists (or ordered rule sets) [23], and (unordered) rule sets [1, 6]. Rule sets are typically formed by generating rules for all classes, where the rules are order-independent in the sense that the prediction made by each rule is not dependent on the applicability of other rules (see e.g., [3]). Decision lists can be formed from rule sets by imposing some order on the rules (e.g., by ordering the rules according to decreasing probability of the most probable class of each rule). A more common approach to learning decision lists is to generate rules for all classes, except one, in some specified order, discarding examples that have been covered by preceding rules when generating subsequent rules, and by forming a default rule for the last class in the sequence (see e.g., [7]). With this approach, the rules become order-dependent in the sense that the prediction of a rule is based on the assumption that none of the preceding rules in the sequence apply.

A decision list has the advantage of requiring only a simple inference mechanism for classifying examples (i.e., the first applicable rule is employed), while a rule set requires some method for combining predictions from multiple applicable rules (e.g., using class counts as in [6] or some more sophisticated scheme as in [17, 18]). Furthermore, decision lists consisting of order-dependent rules are normally more compact than rule sets, due to that one rule suffices for the default class. For the same reason, decision lists can be more efficiently generated. However, it is not clear how

---

\*Part of this work was performed while the author was at the Department of Computer and Systems Sciences, Stockholm University and Royal Institute of Technology.

decision lists compare to rule sets when it comes to maximizing AUC. If there is any difference, this could be explained by the effect of choosing one of the classes as a default and by the choice of inference mechanism. The impact of each of these choices has to be clarified in order to understand the reasons for any difference in AUC.

In the next section, we analyze the differences between applying decision lists and rule sets for maximizing the AUC. In Section 3, we perform an empirical investigation to study the impact of these differences. In Section 4, we relate this study to earlier work, and finally, in Section 5, we give concluding remarks.

## 2 Methods

**2.1 Learning Decision Lists and Rule Sets** Incremental reduced error pruning (IREP), which was originally introduced in [15], is a technique that has been extensively used for efficient separate-and-conquer rule learning, e.g., [15, 7, 12, 8, 3]. By pruning each rule immediately after its generation and removing examples covered by the pruned rule, the number of generated rules is kept relatively small compared to keeping each rule unpruned and removing the relatively few examples covered by each, more specific, rule. Since the computational cost grows as the product of the number of generated rules and the number of training examples, IREP normally allows substantially larger training sets to be handled within a given amount of time compared to using separate-and-conquer with no pruning.

In Table 1, two variants of incremental reduced error pruning are shown. The first, called IREP-O, generates order-dependent rules and is a variant of the algorithms presented in [15, 7], while the second, called IREP-U, generates order-independent rules and is taken from [3]. The main difference between the algorithms is that the class probabilities assigned to each rule by the former algorithm are dependent of previously generated rules, while for the latter algorithm these are assigned independently of other rules. This follows from that the prune set is kept constant in the latter algorithm, allowing each rule to be evaluated and pruned independently of previously generated rules, while the former algorithm removes covered examples from the prune set. Another difference between the algorithms is that the former generates a default rule for the last class, while the latter generates rules for all classes in a similar way.

It should be noted that in the original formulation of IREP for order-dependent rules [15], only two-class problems were handled, while this was extended to multi-class problems in [7]. The algorithm for order-dependent rules presented here slightly differs from the

Table 1: Rule learning algorithms.

```

function IREP-O(OrderedClasses, Examples)
  Rules :=  $\emptyset$ 
  Make stratified split of Examples into
  Grow and Prune
  for each Class  $\in$  OrderedClasses do
    if Last(Class, OrderedClasses) then
      Rules := Rules  $\cup$  {DefaultRule(Prune)}
    else
      Pos := {e : e  $\in$  Grow  $\wedge$  Class(e) = Class}
      Neg := Grow  $\setminus$  Pos
      while Pos  $\neq$   $\emptyset$  do
        Rule := GrowRule(Pos, Neg)
        Rule := PruneRule(Rule, Prune)
        if not Exclude(Rule, Prune) then
          Rules := Rules  $\cup$  {Rule}
          Grow :=
            Grow  $\setminus$  Covers(Rule, Grow)
          Prune :=
            Prune  $\setminus$  Covers(Rule, Prune)
        else
          Grow := Grow  $\setminus$  Covers(Rule, Pos)
          Pos := Pos  $\setminus$  Covers(Rule, Pos)
      return Rules

function IREP-U(Classes, Examples)
  Rules :=  $\emptyset$ 
  Make stratified split of Examples into
  Grow and Prune
  for each Class  $\in$  Classes do
    Pos :=
      {e : e  $\in$  Grow  $\wedge$  Class(e) = Class}
    Neg := Grow  $\setminus$  Pos
    while Pos  $\neq$   $\emptyset$  do
      Rule := GrowRule(Pos, Neg)
      Rule :=
        PruneRule(Rule, Prune)
      if not Exclude(Rule, Prune)
      then Rules := Rules  $\cup$  {Rule}
      Pos :=
        Pos  $\setminus$  Covers(Rule, Pos)
    return Rules

```

previous in that a prune set is generated initially, from which examples are removed only if they are covered by a generated rule that should be kept. In the original formulation, the remaining examples to be covered were repeatedly divided into a grow and prune set each time a new rule was to be generated, and the rule generation was terminated whenever a rule was found that should not be included.<sup>1</sup>

Two problems that need to be addressed when applying order-independent rules is how to classify examples that are not covered by any rule and how to classify examples that are covered by multiple, possibly conflicting, rules.

We address the first problem by classifying an uncovered example according to the class distribution

<sup>1</sup>In [7], an alternative stopping condition was introduced, allowing the number of bits required to encode the rules and class labels to grow up to  $d$  when adding a rule compared to the minimum encoding found so far, where  $d$  is a user-specified parameter.

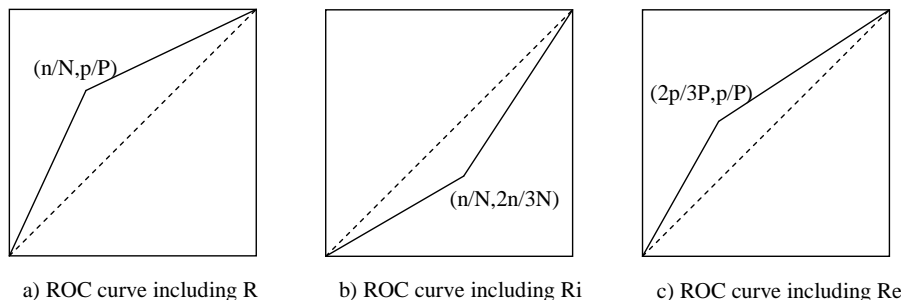


Figure 1: ROC curves after including rules.

of those examples in the prune set that are not covered by any rule.<sup>2</sup>

Two approaches to the latter problem is considered in this work. The first orders rules according to decreasing probability of the most probable class of each rule, and then classifies according to the first applicable rule, hence forming a decision list of the order-independent rules. The second approach does not impose any order on the rules, but combines the class distributions of all applicable rules using naïve Bayes as in [3]. For both order-dependent and order-independent rules, class probability distributions are formed using the covered examples in the prune set together with Laplace correction [5].

**2.2 Pruning and Exclusion Criteria for Maximizing AUC** A number of criteria for deciding how to prune generated rules and whether or not to exclude a generated rule have previously been proposed and evaluated with respect to maximizing accuracy [15, 7, 3].

Several commonly employed pruning criteria for IREP have been shown to be equivalent to maximizing precision, i.e., the fraction  $\frac{p}{p+n}$ , where  $p$  and  $n$  are the number of covered positive and negative examples respectively, and it has been noted that maximizing precision in fact is equivalent to attempting to maximize AUC [14].

To see this, assume we start with a default rule assigning the same probability of being positive to all examples (i.e., the ROC curve is a straight line from  $(0, 0)$  to  $(1, 1)$ , where the x- and y-coordinates give the fraction of covered false and true positives respectively). This corresponds to the dashed lines in Figure 1.

If we add a rule  $R$  that covers  $p$  positive and  $n$  negative examples to this classifier, examples covered by this rule will be given a higher rank than those classified by the default rule alone. The ROC curve will now consist of two segments, passing through  $(0, 0)$ ,

$(n/N, p/P)$  and  $(1, 1)$ , where  $N$  and  $P$  are the total number of negative and positive examples respectively (see Figure 1a).

In order to maximize AUC, we would like to maximize the slope of the first segment<sup>3</sup>, which is given by  $\frac{p/P}{n/N}$ . Since  $P$  and  $N$  are constant for all candidate rules, maximizing the slope of the ROC curve is equivalent to maximizing precision<sup>4</sup>.

A commonly employed exclusion criterion when generating order-dependent rules is  $\frac{p}{p+n} \leq 1/2$  [15, 7], which is natural when maximizing accuracy, since an added rule for the positive class may otherwise be allowed to make more errors than correct classifications. However, when maximizing AUC, this criterion may in fact allow a rule to be added for which the slope of the corresponding first segment of the ROC curve is less than one, i.e., the corresponding ROC curve is concave, since the slope depends on the total number of positive and negative examples as shown above. For example, a rule  $R_i$  for which  $p = 2n$  and  $P = 3N$  would be included using this criterion (since  $2/3 > 1/2$ ), but the slope will only be  $2/3 \leq 1$  (see Figure 1b).

Moreover, this criterion may also exclude rules that result in a slope greater than one. For example, a rule  $R_e$  for which  $n = 2p$  and  $N = 3P$  would be excluded using this criterion (since  $1/3 \leq 1/2$ ), but the slope will be  $3/2 > 1$  (see Figure 1c).

For unordered rule sets, lift (i.e.,  $\frac{p}{P} \frac{p+n}{p+n}$ ) has been the

<sup>2</sup>If this set is empty, the distribution is formed using the original prune set.

<sup>3</sup>This would not necessarily be optimal if we were allowed to add one rule only, but this strategy assumes that an arbitrary number of additional rules may be added.

<sup>4</sup>Maximizing  $\frac{p}{p+n}$  is equivalent to minimizing  $\frac{p+n}{p} = 1 + n/p$  which in turn is equivalent to maximizing  $p/n$ .

basis for both a pruning and an exclusion criterion [3]. It should be noted that using lift as a pruning criterion is equivalent to using precision, since  $\frac{P}{P+N}$  is constant for all rules. However, excluding rules with a lift less than or equal to one turns out to be equivalent to requiring a convex ROC curve for an included rule (i.e., the slope of the first segment must be greater than one), since

$$\begin{aligned} \frac{\frac{p}{p+n}}{\frac{P}{P+N}} \leq 1 &\iff \frac{p}{p+n} \leq \frac{P}{P+N} \iff \\ p(P+N) &\leq P(p+n) \iff pN \leq Pn \iff \\ \frac{p}{P} &\leq \frac{n}{N} \iff \frac{\frac{p}{P}}{\frac{n}{N}} \leq 1 \end{aligned}$$

**2.3 Maximizing AUC with Decision Lists and Rule Sets** Including rules for each class, which is done when generating order-independent rules, as well as allowing for combining all applicable rules, can be advantageous when trying to maximize AUC, as explained below.

Assume that we are facing a two-class learning task, where each class requires two rules if defined separately. Assume further that attached to each rule is a class probability distribution. The generated sequence of order-dependent rules would then typically consist of three rules  $H_O = \{R_1, R_2, R_3\}$ , where the two first rules would assign the same most probable class (positive) to covered examples, while the last would act as a default rule, assigning the other class (negative) to any examples that are not covered by the first two rules. From a ranking perspective, where we want to order a set of examples from the most likely positive to the least likely, the sequence  $H_O$  allows for partitioning the examples into three groups, where all examples in a group are given the same score (i.e., probability of being positive).<sup>5</sup> In particular, all examples that would be classified as negative are placed in the same group and could hence not be differentiated.

On the other hand, the generated set of order-independent rules would typically consist of four rules  $H_U = \{R_1, R_2, R_3, R_4\}$ , for which the class distributions of the two first would give the positive class a higher probability than the negative and vice versa for the last two rules. If a single rule is used for classifying an example, we may now partition all examples in four groups, and the examples can be differentiated independently of the class labels they are given (i.e., examples classified as negative may now be given different scores). Furthermore, if class probabilities are formed from all

<sup>5</sup>There will be fewer possible groups if the same probability distribution is attached to multiple rules.

applicable rules, rather than a single rule, we have up to  $2^4$  possible groups to place an example in. This means that examples can be ranked according to a much more fine-grained scale when multiple rules are combined.

### 3 Empirical Evaluation

#### 3.1 Experimental Setting

**3.1.1 Methods** The methods that are to be compared are variants of the IREP-O and IREP-U algorithms using two different exclusion criteria for IREP-O (accuracy and lift respectively) and with and without post-processing<sup>6</sup> for both algorithms. When classifying examples with order-independent rules, we consider both forming a decision list by ordering the rules according to decreasing probability of the most probable class, and keeping the rule set unordered (using naïve Bayes to combine classifications from multiple rules). All methods use precision as a pruning criterion, and 2/3 of the training examples are used for growing rules, while 1/3 are used for pruning. All methods are given the same grow and prune sets. The employed methods are summarized in Table 2.

Table 2: Employed Methods

Acronym	Output	Algorithm	Excl. crit.	Post-Processing
DL/O	decision list	IREP-O	accuracy	no
DL/OP	decision list	IREP-O	accuracy	yes
DL/OL	decision list	IREP-O	lift	no
DL/OLP	decision list	IREP-O	lift	yes
DL/U	decision list	IREP-U	lift	no
DL/UP	decision list	IREP-U	lift	yes
RS	rule set	IREP-U	lift	no
RS/P	rule set	IREP-U	lift	yes

**3.1.2 Methodology and data sets** We have chosen to compare the methods w.r.t. AUC using ten-fold cross-validation on 34 data sets from the UCI Repository [2]. The names of the data sets together with the number of classes are listed in Table 3. The AUC was calculated for each method on all examples according to [10] and all methods were given exactly the same training and test

<sup>6</sup>It has been observed that significant gains in accuracy can be obtained by post-processing rules generated by IREP through considering replacements of each rule with more general or specific versions followed by eliminating rules that increase the total description length [7]. A similar procedure may be used also for maximizing AUC. In this work, we consider a simplified procedure, in which each rule is either kept or completely eliminated (i.e., replacement rules are not considered), and instead of minimizing the description length, rules that do not contribute positively to the AUC (as estimated on the prune set) are removed.

Table 3: AUC for all 8 methods on the 34 data sets.

Data set	DL/O	DL/OP	DL/OL	DL/OLP	DL/U	DL/UP	RS	RS/P
audiology (24 cl.)	81.37	81.46	83.59	84.36	82.01	80.85	80.82	81.24
balance-scale (3 cl.)	80.19	80.41	78.44	79.77	91.82	92.67	95.83	95.74
breast-cancer (2 cl.)	59.17	59.17	61.22	61.97	66.32	66.58	66.58	65.85
breast-cancer-wisconsin (2 cl.)	95.31	95.07	96.42	96.40	97.20	97.42	99.13	98.76
car (4 cl.)	84.99	85.12	93.27	94.49	97.95	97.98	97.93	98.11
cleveland-heart-disease (5 cl.)	53.06	53.06	68.33	67.71	66.56	65.85	65.50	65.93
crx (2 cl.)	85.70	86.64	85.76	87.39	88.29	88.33	89.95	89.41
cylinder-bands (2 cl.)	73.97	74.17	71.49	70.61	71.46	72.58	72.97	72.94
dermatology (6 cl.)	88.64	88.53	94.88	94.31	96.32	96.26	97.40	97.27
ecoli (8 cl.)	88.12	88.04	92.26	92.27	92.04	92.28	87.83	93.09
glass (6 cl.)	66.80	66.34	67.83	69.25	70.67	69.97	72.25	72.79
hepatitis (2 cl.)	65.98	66.35	73.34	73.42	72.65	71.74	82.96	82.00
house-votes (2 cl.)	97.38	97.61	97.49	97.75	95.47	94.74	97.91	97.19
image-segmentation (7 cl.)	88.32	88.06	89.82	89.85	90.90	89.89	92.11	91.26
ionosphere (2 cl.)	91.80	91.86	91.93	91.92	93.05	92.25	95.11	93.00
iris (3 cl.)	94.75	95.60	95.00	95.87	96.96	96.64	97.91	98.12
kr-vs-kp (2 cl.)	95.92	96.11	97.03	97.25	99.48	99.50	99.54	99.67
lung-cancer (3 cl.)	71.25	71.25	72.68	73.63	69.67	64.94	70.71	70.28
lymphography (4 cl.)	71.82	70.37	75.55	74.35	81.89	82.80	76.32	80.14
mushroom (2 cl.)	99.80	99.80	99.80	99.80	99.96	99.96	99.99	99.98
new-thyroid (3 cl.)	86.98	87.17	90.44	90.55	87.45	85.16	96.43	96.50
pima-indians-diabetes (2 cl.)	65.27	65.27	69.48	69.24	74.83	74.34	76.90	76.66
post-operative-patients (3 cl.)	50.00	50.00	47.83	43.08	40.26	39.73	40.67	39.46
primary-tumor (21 cl.)	57.10	57.19	69.86	70.72	72.10	72.40	67.41	71.32
promoters (2 cl.)	71.64	68.64	72.37	70.10	77.39	77.18	80.78	78.44
sick-euthyroid (2 cl.)	81.75	81.46	90.57	91.11	96.06	96.08	95.67	95.63
soybean-large (19 cl.)	92.45	92.36	92.51	91.96	89.63	89.51	92.22	93.94
spambase (2 cl.)	83.19	83.18	82.75	82.68	93.62	93.76	94.40	94.48
spectf (2 cl.)	57.47	57.47	62.69	62.69	85.28	84.46	87.23	86.00
splice (3 cl.)	95.02	95.36	95.08	95.83	97.69	97.66	98.42	98.32
tae (3 cl.)	49.99	49.99	51.64	51.02	51.87	51.83	52.14	52.14
tic-tac-toe (2 cl.)	96.37	96.40	97.56	97.63	99.38	99.47	99.69	99.65
wine (3 cl.)	89.35	89.89	90.64	90.31	98.46	96.83	99.18	98.84
yeast (10 cl.)	69.25	69.54	73.46	74.43	73.76	73.72	70.60	73.90

examples. For data sets with more than two classes, the total AUC was calculated by summing the AUC for each class weighted by its relative frequency in the data set[9].<sup>7</sup>

**3.1.3 Test hypotheses** The two main null hypotheses can be formulated in the following way:

- forming a decision list from a set of order-independent rules that have been generated for all classes is not more effective w.r.t. AUC than generating a sequence of order-dependent rules with a default rule for one of the classes
- keeping a rule set unordered is not more effective w.r.t. AUC than forming a decision list by ordering the rules according to decreasing probability of the most probable class

In addition, we also test the following null hypotheses:

- lift does not result in a higher AUC than using accuracy as an exclusion criterion for order-dependent rules
- post-processing does not improve AUC

**3.2 Experimental Results** The AUC for all methods on all 34 data sets are shown in Table 3.

In Table 4, the number of wins and losses for each pair of methods is shown, where results for which the p-value (one-sided binomial tail probability) are less than 0.05 are marked with bold-face.

One can see that generating order-independent rules for all classes and ordering them (i.e., DL/U) is more effective than generating a sequence of order-dependent rules with one of the classes as a default (i.e., DL/OL and DL/OLP). The p-value for obtaining 24 wins and 10 losses is  $1.22 \times 10^{-02}$ , allowing the first null hypothesis to be rejected with a low probability of error.

<sup>7</sup>For two-class problems, the total AUC is equivalent to AUC.

Table 4: AUC wins and losses for all 8 methods (row wins/column wins).

	DL/O	DL/OP	DL/OL	DL/OLP	DL/U	DL/UP	RS	RS/P
DL/O	-	12/16	<b>4/29</b>	<b>6/27</b>	<b>5/29</b>	<b>7/27</b>	<b>6/28</b>	<b>5/29</b>
DL/OP	16/12	-	<b>8/25</b>	<b>5/28</b>	<b>5/29</b>	<b>7/27</b>	<b>6/28</b>	<b>5/29</b>
DL/OL	<b>29/4</b>	<b>25/8</b>	-	14/19	<b>10/24</b>	<b>8/26</b>	<b>8/26</b>	<b>5/29</b>
DL/OLP	<b>27/6</b>	<b>28/5</b>	19/14	-	<b>10/24</b>	<b>9/25</b>	<b>7/27</b>	<b>6/28</b>
DL/U	<b>29/5</b>	<b>29/5</b>	<b>24/10</b>	<b>24/10</b>	-	20/14	<b>8/26</b>	<b>8/26</b>
DL/UP	<b>27/7</b>	<b>27/7</b>	<b>26/8</b>	<b>25/9</b>	14/20	-	<b>8/26</b>	<b>5/29</b>
RS	<b>28/6</b>	<b>28/6</b>	<b>26/8</b>	<b>27/7</b>	<b>26/8</b>	<b>26/8</b>	-	20/13
RS/P	<b>29/5</b>	<b>29/5</b>	<b>29/5</b>	<b>28/6</b>	<b>26/8</b>	<b>29/5</b>	13/20	-

Combining all applicable rules in a set of order-independent rules rather than ordering the rules and using the first applicable rule (i.e., RS vs. DL/U) results in a significant increase in AUC. The p-value for obtaining 26 wins and 8 losses is  $1.47 \times 10^{-03}$ , allowing the rejection of the second null hypothesis.

One can see that using lift as an exclusion criterion indeed is clearly more effective than using accuracy for order-dependent rules, independently of whether or not post-processing is employed (i.e., DL/O vs. DL/OL and DL/OP vs. DL/OLP). The p-value of obtaining the observed number of wins and losses, given that the corresponding null hypothesis is true, is  $5.46 \times 10^{-06}$  without post-processing and  $3.31 \times 10^{-05}$  with post-processing, allowing the third null hypothesis to be safely rejected.

It should be noted that in case we consider only the results from the binary classification tasks, the first three null hypotheses can still be rejected. Hence, the conclusions can be drawn independently of the chosen way of calculating total AUC.

When it comes to whether or not post-processing actually is beneficial w.r.t. AUC, the picture is less clear. For order-dependent rules, the use of post-processing appears to be beneficial, with a win/loss ratio for DL/OLP vs. DL/OL of 19/14 and for DL/OP vs. DL/O of 16/12, neither of which however allows the null hypothesis to be rejected (the p-values are 0.243 and 0.286 respectively). For order-independent rules, there is actually a loss, although not significant, w.r.t. AUC from using post-processing. However, this actually supports the above used argument for that the number of ways to partition the examples has an effect on the AUC. By reducing the number of rules that can be combined, the number of ways to partition the examples is reduced and so is the AUC.

From the point of view of interpretability, the rule sets become much smaller with post-processing as shown in Table 5, and one may conclude that the rule sets can be simplified without significantly losing performance.

It can also be observed in Table 5 that the number of rules typically increases when using lift instead of accuracy as exclusion criterion (the number of rules increases 32 times and decreases 1 time without post-processing, while the number of rules increases 29 times and decreases 2 times with post-processing). This indicates that the benefit of using lift compared to accuracy actually comes from including rules that otherwise would have been excluded (due to too low accuracy), rather than from eliminating rules that introduce concavities.

Although not in the direct scope of this study, we also compared the accuracies of all methods, as shown in Table 6 (none of the double-sided binomial tail probabilities are less than 0.05 for the observed number of wins and losses). It can be seen that using lift for order-dependent rules actually performs worse w.r.t accuracy than when using the accuracy-based criterion (i.e., DL/O vs. DL/OL). It can be concluded that the best choice of exclusion criterion for order-dependent rules depends on whether accuracy or AUC is to be maximized.

#### 4 Related Work

In [9], a number of different methods for combining generated rules were evaluated w.r.t. AUC, using two different methods for generating order-independent rules. The results were not conclusive in that study regarding the benefits of combining multiple rules compared to using the single best rule. For one of the induction methods in that study, the former was slightly ahead of the latter (9 wins and 5 losses for weighted voting compared to using the single best rule), while almost the opposite result was obtained for the second method (3 wins and 9 losses). This may be due to that the weighted voting method only takes the highest probability of each rule into account, which contrasts to the use of naïve Bayes in our study that utilizes all class probabilities.

The observation that a higher AUC can be obtained by combining multiple rules has previously been made also for probability estimation trees (PETs). It has been demonstrated that combining rules from PETs obtained

Table 5: Mean no. of rules for all 8 methods on the 34 data sets.

Data set	DL/O	DL/OP	DL/OL	DL/OLP	DL/U	DL/UP	RS	RS/P
audiology (24 cl.)	5.7	5.4	10.5	8.9	19.3	11.4	19.3	14.5
balance-scale (3 cl.)	14.2	11.2	14.5	10.9	58.1	28.3	58.1	37.8
breast-cancer (2 cl.)	2.8	2.8	5.0	4.5	11.3	9.3	11.3	9.8
breast-cancer-wisconsin (2 cl.)	8.8	8.1	9.9	8.1	23.2	12.4	23.2	13.7
car (4 cl.)	16.6	15.2	31.8	24.8	51.6	32.0	51.6	41.5
cleveland-heart-disease (5 cl.)	1.5	1.5	5.7	5.3	13.2	9.6	13.2	10.9
crx (2 cl.)	11.3	8.2	12.7	9.1	27.2	9.8	27.2	19.3
cylinder-bands (2 cl.)	9.4	8.4	10.5	9.1	20.7	17.6	20.7	18.6
dermatology (6 cl.)	8.4	6.9	11.3	8.3	15.8	10.1	15.8	10.6
ecoli (8 cl.)	7.6	6.5	10.9	8.8	28.9	13.2	28.9	16.1
glass (6 cl.)	5.0	4.7	8.5	7.4	10.6	8.2	10.6	9.1
hepatitis (2 cl.)	2.2	2.1	2.8	2.5	12.2	7.9	12.2	8.9
house-votes (2 cl.)	4.1	3.2	4.8	3.6	13.1	6.8	13.1	8.3
image-segmentation (7 cl.)	7.6	6.9	10.3	8.9	15.9	10.7	15.9	13.4
ionosphere (2 cl.)	5.4	5.3	6.3	6.1	16.7	12.1	16.7	12.8
iris (3 cl.)	5.8	4.3	6.2	4.3	11.3	6.5	11.3	7.0
kr-vs-kp (2 cl.)	23.3	16.1	25.0	17.4	54.1	28.0	54.1	35.5
lung-cancer (3 cl.)	2.3	2.3	2.9	2.8	3.8	3.7	3.8	3.7
lymphography (4 cl.)	3.6	3.1	4.8	4.1	11.7	8.2	11.7	8.6
mushroom (2 cl.)	11.2	9.8	11.2	9.8	45.6	24.0	45.6	23.1
new-thyroid (3 cl.)	4.4	4.2	5.4	5.1	9.3	6.1	9.3	7.4
pima-indians-diabetes (2 cl.)	6.7	5.9	9.5	8.2	26.5	16.5	26.5	19.1
post-operative-patients (3 cl.)	1.0	1.0	1.6	1.4	3.7	3.4	3.7	3.4
primary-tumor (21 cl.)	3.0	2.8	9.2	7.9	32.8	16.1	32.8	18.5
promoters (2 cl.)	3.9	3.7	4.7	4.5	8.8	7.4	8.8	7.9
sick-euthyroid (2 cl.)	5.6	5.0	7.2	5.9	31.8	14.8	31.8	21.4
soybean-large (19 cl.)	17.2	16.3	20.9	19.1	33.2	24.2	33.2	26.9
spambase (2 cl.)	32.5	23.2	32.3	22.9	101.5	60.6	101.5	84.4
spectf (2 cl.)	3.0	2.9	3.7	3.6	23.8	15.9	23.8	17.2
splice (3 cl.)	18.0	13.3	25.9	18.1	105.8	69.2	105.8	89.5
tae (3 cl.)	2.9	2.9	6.1	5.8	7.6	6.9	7.6	7.4
tic-tac-toe (2 cl.)	9.6	9.0	11.1	10.1	35.4	20.6	35.4	21.3
wine (3 cl.)	5.9	5.2	6.5	5.7	11.8	7.7	11.8	8.8
yeast (10 cl.)	13.5	10.6	25.0	17.9	39.6	18.6	39.6	19.9

by bagging [21], combining all leaves in a PET based on the deviation of the leaves from the example to be classified [19], and combining all nodes in the path to the leaf [11], all lead to an increase in AUC.

In this study, we have not considered any alternative to the used pruning criterion, since this criterion has earlier been shown to maximize AUC [14]. Nevertheless, work on inducing PETs have demonstrated that a higher AUC can be obtained when pruning is not employed, given that the class probabilities are smoothed using either Laplace correction or the m-estimate [21, 11, 19]. However, as pointed out in [11], it is not clear whether the reason is that current pruning methods for PETs are aiming for maximizing accuracy or if pruning is “intrinsically detrimental”. In contrast to pruning methods for PETs, the pruning criterion used in this study is in fact aiming at maximizing AUC.

## 5 Concluding Remarks

We have considered two central aspects of rule learning when aiming to maximize AUC: i) whether or not to use a default rule for one of the classes and ii) whether or not to impose an order upon the rules.

We explained why learning decision lists by ordering rules that have been generated for all classes could be expected to give a higher AUC than generating order-dependent rules with a default-rule for one of the classes. The main reason for this is that the former method allows examples to be ranked according to a more fine-grained scale compared to the latter, since examples belonging to the default class cannot be differentiated in the latter case. The empirical evaluation did indeed show that the former way of generating a decision list significantly outperforms the latter.

We also explained why a higher AUC could be expected if class probabilities are formed using all applicable rules rather than using the first rule in an ordered rule set. The main reason for this is that the number of partitions according to probability of belonging to a class grows exponentially with the number of rules that may overlap. Again, this allows examples to be ranked according to a more fine-grained

Table 6: Accuracy wins and losses for all 8 methods (row wins/column wins).

	DL/O	DL/OP	DL/OL	DL/OLP	DL/U	DL/UP	RS	RS/P
DL/O	-	5/11	18/10	14/17	12/21	13/19	13/21	14/20
DL/OP	11/5	-	19/11	14/15	12/21	13/19	13/21	13/20
DL/OL	10/18	11/19	-	10/16	11/21	12/19	13/20	13/17
DL/OLP	17/14	15/14	16/10	-	14/19	13/20	14/19	14/18
DL/U	21/12	21/12	21/11	19/14	-	20/11	13/19	12/16
DL/UP	19/13	19/13	19/12	20/13	11/20	-	12/19	13/20
RS	21/13	21/13	20/13	19/14	19/13	19/12	-	16/13
RS/P	20/14	20/13	17/13	18/14	16/12	20/13	13/16	-

scale. This expectation was confirmed by the empirical evaluation that showed a significantly higher AUC for using all applicable rules compared to using the first rule in the ordered set.

In summary, the attractive properties of decision lists compared to rule sets, i.e., to require only a simple inference mechanism and to allow for a more compact representation, actually have a negative effect on the AUC.

## References

- [1] P. C. and T. Niblett. The cn2 induction algorithm. *Machine Learning*, 3, 261–283, 1989.
- [2] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.
- [3] H. Boström. Pruning and exclusion criteria for unordered incremental reduced error pruning. In *Proc. of the ECML/PKDD Workshop on Advances in Inductive Rule Learning*, pages 17–29, 2004.
- [4] A. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(6):1145–1159, 1997.
- [5] B. Cestnik and I. Bratko. On estimating probabilities in tree pruning. In *Proc. of the Fifth European Working Session on Learning*, pages 151–163. Springer-Verlag, 1991.
- [6] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *Proc. Fifth European Working Session on Learning*, pages 151–163, Berlin, 1991. Springer.
- [7] W. Cohen. Fast effective rule induction. In *Proc. of the 12th International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.
- [8] O. Dain, R. Cunningham, and S. Boyer. Irep++, a faster rule learning algorithm. In *Proc. of the Fourth SIAM International Conference on Data Mining*, 2004.
- [9] T. Fawcett. Using rule sets to maximize roc performance. In *Proc. of the IEEE International Conference on Data Mining*, pages 131–138. IEEE Computer Society, 2001.
- [10] T. Fawcett. Roc graphs: Notes and practical considerations for data mining researchers. Technical report, HP Laboratories, Palo Alto, 2003.
- [11] C. Ferri, P. Flach, and J. Hernández-Orallo. Improving the auc of probabilistic estimators trees. In *Proc. of the 14th European Conference on Artificial Intelligence*, volume 2837, pages 121–132. Springer, 2003.
- [12] E. Frank and I. Witten. Generating accurate rule sets without global optimization. In *Proc. of the 15th International Conference on Machine Learning*, pages 144–151. Morgan Kaufmann, San Francisco, CA, 1998.
- [13] J. Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, 1999.
- [14] J. Fürnkranz and P. Flach. Roc ‘n’ rule learning – towards a better understanding of covering algorithms. *Machine Learning*, 58(1):39–77, 2005.
- [15] J. Fürnkranz and G. Widmer. Incremental reduced error pruning. In W. W. Cohen and H. Hirsh, editors, *Proc. of the 11th International Conference on Machine Learning*, pages 70–77. Morgan Kaufmann, 1994.
- [16] N. Lavrac, B. Kavsek, P. Flach, and L. Todorovski. Subgroup discovery with cn2-sd. *Journal of Machine Learning Research*, 5:153–188, 2004.
- [17] T. Lindgren and H. Boström. Classification with intersecting rules. In *Proc. of the 13th International Conference on Algorithmic Learning Theory (ALT’02)*, pages 395–402. Springer-Verlag, 2002.
- [18] T. Lindgren and H. Boström. Resolving rule conflicts with double induction. In *Proc. of the 5th International Symposium on Intelligent Data Analysis*, pages 60–67. Springer, 2003.
- [19] C. X. Ling and R. J. Yan. Decision tree with better ranking. In *Proc. of the Twentieth International Conference on Machine Learning*, pages 480–487, 2003.
- [20] R. Prati and P. Flach. Roccer: A roc convex hull rule learning algorithm. In *Proc. of the ECML/PKDD Workshop on Advances in Inductive Rule Learning*, pages 144–153, 2004.
- [21] F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(3), 2003.
- [22] F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proc. of the 15th International Conference on Machine Learning*, pages 445–453, 1998.
- [23] R. Rivest. Learning decision lists. *Machine Learning*, 2(3), 229–246, 1987.