# Relational Learning Using Constrained Confidence-Rated Boosting (Extended Abstract) *

**Susanne Hoche**and Stefan Wrobel*****

*A long version of this paper appeared in the Proceedings of the 11th International Conference on Inductive Logic Programming, Strasbourg, France, September 2001.

**Magdeburg University, Magdeburg, Germany, Email hoche@iws.cs.uni-magdeburg.de

***Magdeburg University, Magdeburg, Germany, Email wrobel@iws.cs.uni-magdeburg.de

**Abstract.** In propositional learning, boosting has been a very popular technique for increasing the accuracy of classification learners. In first-order learning, on the other hand, surprisingly little attention has been paid to boosting, perhaps due to the fact that simple forms of boosting lead to loss of comprehensibility and are too slow when used with standard ILP learners. In this paper, we show how both concerns can be addressed by using a recently proposed technique of constrained confidence-rated boosting and a fast weak ILP learner. We give a detailed description of our algorithm and show on two standard benchmark problems that indeed such a weak learner can be boosted to perform comparably to state-of-the-art ILP systems while maintaining acceptable comprehensibility and obtaining short run-times.

**Keywords.** First-order learning, ILP, Boosting

## 1 Introduction

In recent years, the field of Machine Learning has seen a very strong growth of interest in a class of methods that have collectively become known as ensemble methods. The general goal and approach of such methods is to increase predictive accuracy by basing the prediction not only on a single hypothesis but on a suitable combination of an entire set of hypotheses. *Boosting* is a particularly attractive class of ensemble methods which construct multiple hypotheses by repeatedly calling a "weak" learner on changing distributions over the given examples. During each round of boosting, a so called weak hypothesis is learned and the weight of examples correctly handled by it is decreased, while the weight of incorrectly handled examples is increased. The total prediction, i.e. the so called strong hypothesis, is obtained by a weighted majority vote of the weak hypotheses.

Given the set of boosting approaches in propositional learning, it is surprising that boosting has not received comparable attention within ILP, with a notable exception of Quinlan's [9] initial experiments. There are two possible reasons for this situation which appear especially relevant. Firstly, understandability of results has always been a central concern of ILP researchers beyond accuracy. Unfortunately, if, as in Quinlan's study, one uses the classic form of confidence-rated boosting (Adaboost.M1) the result will be quite a large set of rules each of which in addition has an attached positive or negative voting weight. To understand the behavior of one rule in this rule set, it is necessary to consider all other rules and their relative weights, making it quite difficult to grasp the results of the learner. Secondly, in propositional learning, boosting is often applied simply by using an unchanged existing propositional learner as a basis. If one carries this over to ILP (e.g. Quinlan simply used FFOIL as a base learner), the runtimes of such a boosted ILP learner clearly would be problematic due to the high effort already expended by a typical ILP system.

In this paper, we show that both of these concerns can be addressed by suitably combining recent advances in boosting algorithms with a fast weak learner. In particular, we show how *constrained confidence-rated boosting* (CCRB), which is our denomination and interpretation of the approach described in [2], can be used to significantly enhance the understandability of boosted learning results by restricting the kinds of rule sets allowed. We combine this with a weak greedy top-down learner based on the concept of *foreign links* introduced in Midos [15] which uses a limited form of look-ahead and optimizes the same heuristic criterion as used in [2]. In an empirical evaluation on two known hard problems of ILP, the well-studied domains of mutagenicity and Qualitative Structure Activity Relationships (QSARs), we show that indeed such a simple weak learner together with CCRB achieves accuracies comparable to much more powerful ILP systems, while maintaining acceptable comprehensibility and obtaining short run-times.

The paper is organized as follows. In section 2, we review boosting, and motivate the basic ideas of CCRB based on [2]. In section 3, we describe our foreign link based weak learner which employs a basic form of look-ahead. The heuristic evaluation functions employed to guide the search in the constrained hypothesis space are described in section 4. Section 5 details how the hypotheses generated by the weak learner are used in the framework of CCRB. Our experimental evaluation of the approach is described and discussed in section 6. For a detailed discussion of related work, we refer to the full version of this paper [3]. Section 7 contains our conclusions and some pointers to future work.

## 2 Boosting

Boosting is a method for improving the predictive accuracy of a learning system by means of combining a set of classifiers constructed by a weak learner into a single, strong hypothesis [11, 9, 8]. It is known to work well with most unstable classifier systems, i.e. systems where small changes to the training data lead to notable changes in the learned classifier. The idea is to "boost" a weak learning algorithm performing only slightly better than random guessing into an arbitrarily accurate learner by repeatedly calling the weak learner on changing distributions over the training instances and combining the weak hypotheses into one strong hypothesis. Each of the resulting hypotheses gets a voting weight corresponding to its prediction confidence, and the total prediction, i.e. the strong hypothesis, is obtained by summing up all these votes.

A probability distribution over the set of training instances is maintained modeling the weights associated with each training instance and indicating the influence of an instance when building a classifier. Initially, all instances have equal influence on the construction of the weak hypotheses. In each iterative call of the learner, a weak hypothesis is learned and a prediction confidence is assigned to it. How this confidence is determined is a design issue of the weak learner and will, for our approach, be detailed in section 4.

On each round of boosting, the distribution over the training instances is modified according to the confidence of the weak hypothesis and the examples covered by it. The weights of misclassified instances are increased and, in analogy, those of correctly classified instances are decreased according to the confidence of the weak hypothesis. Thus, correctly classified instances will have less influence on the construction of the weak hypothesis in the next iteration, and misclassified instances will have a stronger influence, confronting the learner in each new round of boosting with a modified learning task and forcing the focus on the examples not yet correctly classified. Finally, all weak hypotheses are combined into one hypothesis. An instance $x$ is classified by this strong hypothesis by adding up the confidence of each weak hypothesis covering $x$. The class $y$ of $x$ is predicted as positive if this sum is positive, otherwise as negative.

The classic form of (unconstrained) confidence-rated boosting (Adaboost.M1) yields quite a large set of rules each of which in addition has an attached positive or negative voting weight. Moreover, each weak hypothesis may vote with different confidences for different examples. This way, rules inferring the target predicate are learned as well as rules for the negation of the target predicate.

In our ILP setting, we will, in contrast, assume that the weak learner produces on each iteration a hypothesis in form of a single Horn clause $H \leftarrow L_1, L_2, \cdots, L_n$ $[c]$ with an associated real-valued number $c$, where $H$ is the atom $p(X_1, \cdots, X_{a(p)})$ and $p$ the target predicate of arity $a(p)$, the $L_i$ are atoms with background predicates $p_i$, and $c$ represents the prediction confidence of the hypothesis. This confidence is used as the voting weight of the hypothesis on all examples covered by it, where large absolute values indicate high confidence. Moreover, we will restrict the weak hypothesis to vote "0" to abstain on all examples not covered by it.

Thereby, the semantics of a rule is, as opposed to usual ILP practice, determined by the sign of its attached prediction confidence. A hypothesis $H \leftarrow L_1, L_2, \cdots, L_n$ $[c]$ such that $c > 0$ implies that $H$ is true. It is interpreted as classifying all instances covered by it as positive with confidence $c$. $H$ $[c]$ such that $c < 0$ implies that $H$ is false and is interpreted as classifying each instance as negative. Here is an example of a boosting result consisting of 7 weak hypotheses when learning a target predicate $p$.

| | | |
|---|---|---|
| 1. | p(X) ← q(X,a). | [0.2] |
| 2. | p(X) ← q(X,Y), r(Y). | [0.9] |
| 3. | p(X) ← s(X). | [0.1] |
| 4. | p(X) ← q(X,Y), v(Y). | [-0.6] |
| 5. | p(X) ← r(X). | [-0.5] |
| 6. | p(X) ← q(X,b). | [-0.3] |
| 7. | p(X) ← t(X). | [-0.9] |

In order to classify a new instance about which we know q(1,a), v(a), t(1), s(1), we need to check which hypotheses cover this example. Here, 1,3,4,7 cover the example, so we sum up their confidences, yielding $0.2 + 0.1 - 0.6 - 0.9 = -1.2 < 0$, and classify the instance as negative. In other words, to understand the behavior of one rule in this rule set, it is necessary to consider all other rules and their relative weights, making it quite difficult to grasp the results of the learner.

In our approach of *constrained confidence-rated boosting* we will restrict each hypothesis to either predict the positive class with a positive confidence, or to be the default hypothesis $p(X_1, \cdots, X_{a(p)})$ with an assigned negative confidence. This constraint ensures that the resulting set of hypotheses can be more easily interpreted. Namely, in order to appraise the quality of a hypothesis, it suffices to consider its assigned confidence in proportion to just the weight of the default hypothesis, instead of having to consider the entire set of weak hypotheses.

Using the additional restrictions, we see for the above example that with CCRB only results of the following form would be allowed, making learning harder but guaranteeing better understandability:

| | | |
|---|---|---|
| 1. | p(X) ← q(X,a). | [0.2] |
| 2. | p(X) ← q(X,Y), r(Y). | [0.9] |
| 3. | p(X) ← s(X). | [0.1] |
| 4. | p(X). | [-0.3] |

Since the same weak hypothesis might be generated more than once by the weak learner, we can further simplify the set of resulting hypotheses by summarizing hypotheses $H\ [c_1], \cdots, H\ [c_n], 1 \leq i \leq n$, which only differ with regard to their assigned confidences. A set of such identical hypotheses can be replaced by a single hypothesis $H'\ [c], H' = H_i, 1 \leq i \leq n$, with $c = \sum_{1 \leq i \leq n} c_i$.

The constraint on the weak hypotheses requires the weak learner to employ a search strategy guaranteeing that only positively correlated hypotheses, i.e. those predicting the positive class, with a positive prediction confidence are learned, or that the default hypothesis is opted for if no such positive correlated hypothesis can be induced from the training instances. [2] offer a theoretically well founded heuristics for this problem which will be detailed in the following section.

## 3 The Weak Relational Learner

Our weak greedy top-down learner is using a refinement operator based on the concept of *foreign links* introduced in Midos [15], and a basic form of look-ahead. Both will be described in this section. The heuristics guiding the search of the greedy weak learner based on the refinement operator in the constrained hypothesis space will be detailed in section 4. In Table 2, we give a more concise description of the weak greedy learner embedded into the framework of CCRB. In the following, references to steps in Table 2 will be indicated by "T1._".

The hypothesis space consists of non-recursive, function-free Horn clauses $C = H \leftarrow B$, where $H$ is the atom $p(X_1, \cdots, X_{a(p)})$ and $p$ the target predicate of arity $a(p)$. In order to constrain the complexity of the hypothesis space, the refinement operator of our weak greedy top-down learner employs as declarative bias a foreign literal restriction based on the concept of *foreign links* introduced in [15]. When specializing a clause $C$ by adding a new literal $L$, $L$ must share at least one variable with previous literals in $C$. The foreign literal restriction further confines the set of alternative literals that can be added to a clause by only taking into account predicate argument positions which have been a priori explicitly defined.

Furthermore, we employ a limited form of look-ahead in our refinement operator in order to avoid the shortsightedness problem with respect to existential variables in the learned hypotheses. Merely introducing new existential variables in a clause will probably not lead to notable changes, and the greedy learner is apt to rather select a literal that restricts existing variables. Thus, when specializing a clause $C$ into $C' = C, L$ by means of adding a new literal $L$ to $C$, we concurrently add to the set $\rho(C)$ of refinements of $C$ all specializations of the refinement $C'$ which can be obtained by successively instantiating the new variables in $L$.

Given, for example, a target predicate active/1, a predicate atm/3, and a foreign link declaration active[1] → atm[1], applying $\rho$ on $C = \text{active}(X_1)$ would, for a nominal variable $X_2$ with the domain $\{c, cl\}$, and a continuous variable $X_3$ with discretization $\mathcal{D} = [-0.782, 1.002]$, result in the specializations

$$
\begin{aligned}
&\text{active}(X_1) \leftarrow \text{atm}\,(X_1, X_2, X_3),\\
&\text{active}(X_1) \leftarrow \text{atm}\,(X_1, c, X_3),\\
&\text{active}(X_1) \leftarrow \text{atm}\,(X_1, cl, X_3),\\
&\text{active}(X_1) \leftarrow \text{atm}\,(X_1, X_2, X_3), X_3 \leq -0.782,\\
&\text{active}(X_1) \leftarrow \text{atm}\,(X_1, X_2, X_3), X_3 > -0.782,\\
&\text{active}(X_1) \leftarrow \text{atm}\,(X_1, X_2, X_3), X_3 \leq \ \ 1.002,\\
&\text{active}(X_1) \leftarrow \text{atm}\,(X_1, X_2, X_3), X_3 > \ \ 1.002.
\end{aligned}
$$

## 4 Search Strategy

Our weak first-order inductive learner accepts as input instances from a set $E = E^+ \cup E^-$ of training examples along with a probability distribution $D$ over the training instances. The background knowledge is provided in form of a set $B$ of ground facts over background predicates. However, we will sometimes write $E^+$ and $E^-$ somewhat differently than used in ILP, and will say that $E = \{(x, 1) \mid x \in E^+\} \cup \{(x, -1) \mid \neg x \in E^-\}$.

To avoid overfitting in the weak learner, the training instances are randomly split into two sets, $\mathcal{G}, \mathcal{P}$, used to specialize clauses and to prune these refinements later on, respectively. Starting with the target predicate, the weak learner greedily generates specializations which are positively correlated with the training instances and thus have a positive prediction confidence on the training set.

When thinking about strategies to guide the search of a greedy learner, entropy based methods like information gain represent an obvious choice. However, the theoretical framework of boosting provides us with a guiding strategy based on one of the specific features of boosting, namely the probability distribution being modified in each iterative call of the weak learner.

As suggested by [2], the training error can be minimized by searching in each round of boosting for a weak hypothesis maximizing the objective function

$$z(C) =_{def.} \left(\sqrt{w_+(C, \mathcal{G})} - \sqrt{w_-(C, \mathcal{G})}\right)^2 \qquad (1)$$

which is based on the collective weight of all instances in $\mathcal{G}$ covered by clause $C$. For a clause $C$ and a set $\mathcal{S}$, the two weight functions $w_+, w_-$ are defined by

$$w_+(C, \mathcal{S}) \quad =_{def.} \sum_{\substack{(x_i, y_i) \in \mathcal{S}\text{covered} \\ \text{by } C, y_i = 1}} D_i^t,$$

$$w_-(C, \mathcal{S}) \quad =_{def.} \sum_{\substack{(x_i, y_i) \in \mathcal{S}\text{covered} \\ \text{by } C, y_i = -1}} D_i^t. (2)$$

Since clauses $C$ maximizing $z(C)$ may be negatively correlated with the positive class, we restrict, as proposed in [2], the search to positively correlated clauses, i.e. to clauses maximizing the objective function $\tilde{z}$ defined as

$$\tilde{z}(C) =_{def.} \sqrt{w_+(C, \mathcal{G})} - \sqrt{w_-(C, \mathcal{G})}. \qquad (3)$$

The refinement operator $\rho$ of the weak learner iteratively refines the clause $C$ currently maximizing $\tilde{z}$ until either a clause $C'$ is found with hitherto maximal $\tilde{z}(C')$ only covering positive examples, or until $\tilde{z}$ can not be further maximized (T1.2d).

The positively correlated clause $C$ resulting from the refinement process is subject to overfitting on the training set, and is thus immediately examined to see whether it can be pruned. Namely, all generalizations of $C$ resulting from deleting single literals and constants in $C$ from right to left are generated (T1.2e).

The objective function (3) is only maximized on the set $\mathcal{G}$ based on which rules are generated by the weak learner. However, the evaluation of the prediction confidence of a weak hypothesis is based on the entire training set. Thus, it is possible for the weak learner to learn a hypothesis $C'[c], c < 0$, which is, on the entire training set, negatively correlated with the positive class. Such hypotheses are not considered in order to ensure the constraint for a weak hypothesis to be either positively correlated or to be the default hypothesis. Thus, generalizations of $C$ with a non-positive prediction confidence on the whole training set are ruled out (T1.2f). If no generalization of $C$ with a positive confidence exists, the default hypothesis is chosen as current weak hypothesis (T1.2g). The prediction confidence of a clause $C$ on a set $\mathcal{S}$ is defined as

$$c(C, \mathcal{S}) =_{def.} \frac{1}{2} \ln \left( \frac{w_+(C, \mathcal{S}) + \frac{1}{2N}}{w_-(C, \mathcal{S}) + \frac{1}{2N}} \right), \qquad (4)$$

where $N$ is the number of training instances and $\frac{1}{2N}$ is a smoothing constant applied to avoid extreme estimates when $w_-(C, \mathcal{S})$ is small.

All generalizations of $C$ with a positive confidence on the entire training set are then evaluated with respect to their confidence on the set $\mathcal{G}$ and their coverage and accuracy on the set $\mathcal{P}$. This kind of evaluation is proposed by [2] who define, based on the definition of the loss of a clause $C$ with

| | | $C^2$RIB | FOIL | Fors | Progol |
|---|---|---|---|---|---|
| Mutagenicity | Accuracy $\pm$ StdDev | $88.0 \pm 6.0$ | $82.0 \pm 3.0$ [14] | $89.0 \pm 6.0$ [4] | $88.0 \pm 2.0$ [14] |
| | $\oslash$ Run-time (mins.) | 7 | n/a | n/a | 307 |
| | $\oslash$ Number of literals | 64 | 46 [13] | n/a | 28 |
| QSARs | Accuracy $\pm$ StdDev | $83.2 \pm 3.0$ | $82.9 \pm 2.7$ | n/a | $79.8 \pm 3.7$ |
| | $\oslash$ Run-time (mins.) | 57 | 0.7 | n/a | 372 |
| | $\oslash$ Number of literals | 142 | 140 | n/a | 154 |

Table 1 Accuracy, standard deviation, average run-time and number of literals in the final hypotheses on the $188 - \mathcal{B}_4$ mutagenicity dataset [13] and the QSARs dataset [5, 6]

associated confidence $c(C, \mathcal{G})$ of [2], a loss function for a clause $C$ as

$$\begin{aligned} loss(C) \;&=_{def.}\; (1 - (w_+(C,\mathcal{P}) + w_-(C,\mathcal{P}))) \\ &+\; w_+(C,\mathcal{P}) \cdot e^{(-c(C,\mathcal{G}))} \\ &+\; w_-(C,\mathcal{P}) \cdot e^{(c(C,\mathcal{G}))}. \end{aligned} \tag{5}$$

This function is minimized over all generalizations of $C$ with a positive confidence (T1.2(h)i).

In a last step, the positively correlated generalization $C'$ of $C$ with minimal $loss(C')$ and the default hypothesis are compared with respect to their training error (T1.2(h)ii). Since a positively correlated clause is compared to the default hypothesis predicting the negative class, the objective function to be maximized is in this case $z$ as defined in equation (1). Whichever of these two hypotheses maximizes $z$ is chosen as the weak hypothesis of the current iteration.

## 5 Constrained Confidence-Rated Boosting of a Weak Relational Learner

In this section, following [2], we explain how the weak hypotheses generated in each iteration of the greedy learner are used in the framework of CCRB. The weak learner is invoked $T$ times. Let $C_t$ denote the weak hypothesis generated in the $t$-th iteration based on the refinement operator and the heuristic search strategy described in the previous section. $C_t$ is used in function $h_t : X \to \Re$,

$$h_t(x) = \begin{cases} c(C_t, E) & \text{if } C_t \text{ covers e = (x, y)} \\ 0 & \text{else,} \end{cases}$$

mapping each instance to a real-valued number, i.e. to the prediction confidence of $C_t$ on the entire training set if this instance is covered by $C_t$, and to 0 otherwise (T1.2i).

Before starting the next round of boosting, the distribution over the training instances, which is initially uniform, is updated by means of $h_t$, namely by determining $D_i^{t'} = \frac{D_i^t}{e^{(y_i \cdot h_t(x_i))}}$. This way, the weights of all instances not covered by $C_t$ are not modified, whereas the weights of all positive and negative instances covered by $C_t$ are decreased and increased, respectively, in proportion to the prediction confidence of $C_t$. Then, the sum of the resulting weights is normalized, $D_i^{t+1} = \frac{D_i^{t'}}{\sum_i D_i^{t'}}, 1 \le i \le N$, so as to serve as the probability distribution of the next iteration (T1.2j).

After $T$ iterations of the weak learner, the strong hypothesis is defined by means of the weak hypotheses. For each instance, the prediction confidence of all hypotheses covering it are summed up. If this sum is positive, the strong hypothesis classifies the instance as positive, otherwise it is classified as negative:

$$H(x) := sign\left( \sum_{C_t : (x,y) \text{ covered by } C_t} c(C_t, E) \right).$$

## 6 Empirical Evaluation

We conducted an empirical evaluation of our approach to CCRB on the two thoroughly investigated domains of mutagenicity [14] and Quantitative Structure Activity Relationships (QSARs) [5, 6]. For a detailed description of the domains, we refer to [3]. The weak learner is invoked $T = 100$ times. Although the number $T$ of iterations can be automatically determined by cross-validation [2], we treat $T$ as fixed in our experiments. The predictive accuracy is estimated by 10-fold-cross-validation on the data of the mutagenicity domain and by 5-fold-cross-validation on the data of the QSARs domain. The accuracy obtained in our experiment with $C^2$RIB, which stands for **C**onstrained **C**onfidence-**R**ated **ILP** **B**oosting, is

Let $N$ denote the number of training instances $e = (x_i, y_i) \in E = E^+ \cup E^-$, $p$ the target predicate of arity $a(p)$, and let $T$ denote the total number of iterations of the weak learner. Furthermore, let $w_+, w_-$ denote the weight functions defined according to equation (2), $c(C, \mathcal{S})$ the prediction confidence of a clause $C$ on a set $\mathcal{S}$ defined according to equation (4), and $\tilde{z}$ the objective function defined according to equation (3).

1. **Set** $D_i^1 := \frac{1}{N}$ for $1 \leq i \leq N$
2. **For** $t = 1 \ldots T$
   (a) **Split** training set $E$ randomly into $\mathcal{G}$ and $\mathcal{P}$ according to $D_t$ such that $\sum_{(x_i,y_i)\in\mathcal{G}} D_i^t \approx \frac{2}{3}$
   (b) $C := p(X_1, \cdots, X_{a(p)})$
   (c) $\tilde{Z} := 0$
   (d) **While** $w_-(C, \mathcal{G}) > 0$
      i. **Let** $C' := argmax_{C'' \in \rho(C)}\{\tilde{z}(C'')\}$
      ii. **Let** $\tilde{Z}' := \tilde{z}(C')$
      iii. **If** $\tilde{Z}' - \tilde{Z} \leq 0$ exit loop
      iv. **Else** $C := C'$, $\tilde{Z} := \tilde{Z}'$
   (e) $Prunes(C) := \{p(X_1, \cdots, X_{a(p)}) \leftarrow B \mid C = p(X_1, \cdots, X_{a(p)}) \leftarrow BB'\}$
   (f) **Remove** from $Prunes(C)$ all clauses $C'$ where $c(C', E) \leq 0$
   (g) **If** $Prunes(C) = \emptyset$ let $C_t := p(X_1, \cdots, X_{a(p)})$
   (h) **Else**
      i. $C' := argmin_{C'' \in Prunes(C)}\{loss(C'')\}$, where $loss(C'')$ is defined according to equation (5)
      ii. **Let** $C_t := argmax_{C'' \in \{C', p(X_1,\cdots,X_{a(p)})\}}\{\left(\sqrt{w_+(C'', \mathcal{G})} - \sqrt{w_-(C'', \mathcal{G})}\right)^2\}$
   (i) $h_t : X \to \Re$ is the function $h_t(x) = \begin{cases} c(C_t, E) & \text{if e = (x, y) is covered by } C_t \\ 0 & \text{else} \end{cases}$
   (j) **Update** the probability distribution $D_t$ according to
   $D_i^{t'} = \frac{D_i^t}{e^{(y_i \cdot h_t(x_i))}}$ and $D_i^{t+1} = \frac{D_i^{t'}}{\sum_i D_i^{t'}}, 1 \leq i \leq N,$
3. **Construct** the strong hypothesis $H(x) := sign\left(\sum_{C_t:(x,y) \text{ covered by } C_t} c(C_t, E)\right)$

Table 2 Constrained Confidence-Rated Boosting Algorithm

displayed in Table 1 together with reference results on the same data and, in case we did not conduct the experiments ourselves, the sources from which the results are reported. Run-times are referring to results obtained on a sparc SUNW, Ultra-4.

For the mutagenicity domain, several relational descriptions are available [13], ranging from a weakly structured description $\mathcal{B}_2$ to a strongly structured description $\mathcal{B}_4$. We conducted our experiment with C$^2$RIB on the strongly structured description $\mathcal{B}_4$ restricted to a subset of 188 so called regression-friendly compounds 125 of which are classified as having positive levels of mutagenicity. We show only results obtained on this most comprehensive set of background knowledge which we have worked with.[1]

As can be seen from the table, C$^2$RIB performs on par with other ILP learners on the 10-fold-cross-validation data sets of the mutagenicity domain. Moreover, the results are obtained in reasonable time, and the final hypotheses represent fairly comprehensible results. The number of literals in the final hypothesis averages to 64 (32 clauses on average, where the body of each clause averagely comprises two literals), as compared to the result of averagely 46 literals in the hypothe-

[1] Additional results have been obtained by other authors on the $\mathcal{B}_3$ dataset [13], in particular by STILL [12] ($87 \pm 8$) and G-Net [1] ($91 \pm 8$).

DEFAULT RULE:
active(A). [-1.40575]

POSITIVE RULES:
active(A) $\leftarrow$ logp(A,C),C>2.0,logp(A,D),D$\leq$4.0. [0.00082336]
active(A) $\leftarrow$ lumo(A,C),C> -2.0,lumo(A,D),D$\leq$ -1.2. [0.0210132]
active(A) $\leftarrow$ logp(A,C),C>2.0. [0.115733]
active(A) $\leftarrow$ lumo(A,C),C> -2.0,logp(A,D),D$\leq$3.0,atm(A,E,F,29,G). [0.175073]
active(A) $\leftarrow$ atm(A,C,D,35,E). [0.176489]
active(A) $\leftarrow$ atm(A,C,D,1,E). [0.197106]
active(A) $\leftarrow$ ringSize5(A,C). [0.215675]
active(A) $\leftarrow$ atm(A,C,D,27,E). [0.231689]
active(A) $\leftarrow$ lumo(A,C),C$\leq$ -1.2. [0.283592]
active(A) $\leftarrow$ lumo(A,C),C> -2.0,atm(A,D,E,29,F). [0.355777]
active(A) $\leftarrow$ logp(A,C),C>5.0. [0.470995]
active(A) $\leftarrow$ bond(A,C,D,5). [0.582912]
active(A) $\leftarrow$ atm(A,C,D,26,E),atm(A,F,G,1,H),lumo(A,I),I$\leq$ -1.2. [0.584057]
active(A) $\leftarrow$ atm(A,C,cl,D,E),bond(F,C,G,H). [0.763684]
active(A) $\leftarrow$ atm(A,C,D,26,E),logp(A,F),F>3.0. [0.778605]
active(A) $\leftarrow$ atm(A,C,D,27,E),logp(A,F),F>2.0,logp(A,G),G$\leq$3.0. [0.832673]
active(A) $\leftarrow$ atm(A,C,D,27,E),ringSize5(A,F). [0.925553]
active(A) $\leftarrow$ atm(A,C,D,230,E). [0.977438]
active(A) $\leftarrow$ logp(A,C),C>3.0,ringSize5(A,D). [1.00485]
active(A) $\leftarrow$ atm(A,C,D,16,E). [1.01437]
active(A) $\leftarrow$ atm(A,C,D,32,E),bond(F,G,C,2). [1.1001]
active(A) $\leftarrow$ carbon5aromaticRing(A,C). [1.4434]
active(A) $\leftarrow$ bond(A,C,D,3). [1.46341]
active(A) $\leftarrow$ lumo(A,C),C$\leq$ -2.0. [1.64408]
active(A) $\leftarrow$ ringSize5(A,C),logp(A,D),D>4.0. [1.69492]
active(A) $\leftarrow$ atm(A,C,D,28,E). [1.69956]
active(A) $\leftarrow$ anthracene(A,C). [2.21461]
active(A) $\leftarrow$ carbon6Ring(A,C). [3.06628]
active(A) $\leftarrow$ phenanthrene(A,C). [3.55481]

Table 3 A strong hypothesis obtained from $C^2$RIB

ses obtained by FOIL as published in [13], and 28 literals on average in the hypotheses obtained by Progol. A final hypothesis obtained by $C^2$RIB is displayed in Table 3.

The predictive accuracy obtained by $C^2$RIB on the 5-fold-cross-validation data sets of QSARs domain is slightly higher than the ones obtained with the other two systems (however still within the range of the standard deviations). Runtime of $C^2$RIB averages to 57 minutes for 100 iterations, as compared to 372 and 0.7 minutes for Progol and FOIL, respectively. The number of literals in the final hypotheses obtained by $C^2$RIB averages to 142 (71 clauses on average, where the body of each clause averagely comprises two literals), as compared to 140 and 154 literals on average in the hypotheses obtained by FOIL and Progol, respectively. The fact that FOIL yields good results in very short run-times suggests to investigate why FOIL's heuristics are so successful and how elements of FOIL could be incorporated in our weak learner.

## 7 Conclusion

In this paper, we have presented an approach to boosting in first order learning. Our approach, which we have termed *constrained confidence rated boosting* (CCRB), builds on recent advances in the area of propositional boosting; in particular, it adapts the approach of Cohen and Singer [2] to the first order domain. The primary advantage of CCRB is that the resulting rule sets are restricted to a much simpler and more understandable format than the one produced by unconstrained versions, e.g. AdaBoost.M1, as it has been used in the only prior work on boosting in ILP by Quinlan [9]. On two standard benchmark problems, we have shown that by using an appropriate first order weak learner with look-ahead, it is possible to design a learning system that produces results that are comparable to much more powerful ILP-learners both in accuracy and in comprehensibility while achieving short run-times due to the simplicity of the weak learner.

These encouraging results need to be substantiated in future work, in particular in the direction of examining other points in the power/run-time trade-off of the weak learner. The current weak learner has short run-times and already reaches comparable results to other non-boosted systems, but it appears possible to make this weak learner slightly more powerful by adding in more of the standard elements of "full-blown" ILP-learners. While this would certainly slow down the system, it would be an interesting goal of further research to determine exactly the right balance between speed and accuracy of the weak learner.

This work was partially supported by DFG (German Science Foundation), project FOR345/1-1TP6.

## References

1. C. Anglano, A. Giordana, G. Lo Bello, and L. Saitta. An experimental evaluation of coevolutive concept learning. In J. Shavlik, editor, *Proc. of the 15th ICML*, 1998.
2. W. Cohen and Y. Singer. A Simple, Fast, and Effective Rule Learner. *Proc. of 16th National Conference on Artificial Intelligence*, 1999.
3. S. Hoche and S. Wrobel. Relational Learning Using Constrained Confidence-Rated Boosting. *Proc. of 11th Int. Conference on Inductive Logic Programming*, Strasbourg, France, September 2001.
4. A. Karalic. *First Order Regression*. PhD thesis, University of Ljubljana, Faculty of Computer Science, Ljubljana, Slovenia, 1995.
5. R.D. King, S. Muggleton, R.A. Lewis, and M.J.E. Sternberg. Drug design by machine learning: The use of inductive logic programming to model the structure activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. of the National Academy of Sciences of the United States of America* 89(23):11322-11326, 1992.
6. R.D. King, A. Srinivasan, and M. Sternberg. Relating chemical activity to structure: An examination of ILP successes. *New Generation Computing, Special issue on Inductive Logic Programming* 13(3-4):411-434, 1995.
7. S. Muggleton. Inverse Entailment and Progol. *New Generation Computing*, 13:245-286, 1995.
8. J.R. Quinlan. Bagging, boosting, and C4.5. In *Proc. of 14th National Conference on Artificial Intelligence*, 1996.
9. J.R. Quinlan. Boosting First-Order Learning. *Algorithmic Learning Theory*, 1996.
10. J.R. Quinlan and R. M. Cameron-Jones. FOIL: A Midterm Report. In P. Brazdil, editor, *Proc. of the 6th European Conference on Machine Learning*, volume 667, pages 3-20. Springer-Verlag, 1993.
11. R.E. Schapire. Theoretical views of boosting and applications. In *Proc. of the 10th International Conference on Algorithmic Learning Theory*, 1999.
12. M. Sebag and C. Rouveirol. Resource-bounded Relational Reasoning: Induction and Deduction through Stochastic Matching. *Machine Learning* , 38:41-62, 2000.
13. A. Srinivasan, S. Muggleton, and R. King. Comparing the use of background knowledge by inductive logic programming systems. *Proc. of the 5th International Workshop on Inductive Logic Programming*, 1995.
14. A. Srinivasan, S. Muggleton, M.J.E. Sternberg, and R.D. King. Theories for mutagenicity: A study in first-order and feature-based induction. *Artificial Intelligence*, 85:277-299, 1996.
15. S. Wrobel. An algorithm for multi-relational discovery of subgroups. In J. Komrowski and J. Zytkow, editors, *Principles of Data Mining and Knowledge Discovery: First European Symposium - Proc. of the PKDD-97*, pages 78-87, 1997.