

Relational Learning Using Constrained Confidence-Rated Boosting

Susanne Hoche and Stefan Wrobel

Otto-von-Guericke University University, Magdeburg, Germany,
{hoche,wrobel}@iws.cs.uni-magdeburg.de

In Proceedings of the ILP 2001, LNAI, Springer

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.

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 since on the one hand it has originated in theoretical studies of learnability, but on the other hand has also been developed into practical algorithms that have demonstrated superior performance on quite a broad range of application problems. Boosting constructs multiple hypotheses by first calling a “weak” learner on the given examples to produce a first hypothesis. During each subsequent round of boosting, the weight of examples correctly handled by the hypothesis induced in the previous round is decreased, while the weight of examples incorrectly handled is increased. In the resulting set of hypotheses, each hypothesis gets a voting weight corresponding to its prediction confidence, and the total prediction is obtained by summing up all these votes.

Given the set of boosting approaches in propositional learning, it is somewhat surprising that boosting has not received comparable attention within ILP, with a notable exception of Quinlan’s [8] 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 behaviour 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 run-times 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 greedy top-down weak learner based on the concept of *foreign links* introduced in Midos [14] 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 constrained confidence-rated boosting based on [2]. In section 3, we briefly describe our foreign link based weak learner and give a more detailed account of the heuristic evaluation functions employed to guide the search in the constrained hypothesis space. Section 4 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 5. In section 6, we discuss related work in more detail. 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 [10, 8, 7]. 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 set of weak hypotheses into one strong hypothesis. In the resulting set of hypotheses, i.e. the strong

hypothesis, each hypothesis gets a voting weight corresponding to its prediction confidence, and the total prediction is obtained by summing up all these votes.

A probability distribution over the set of training instances is maintained. The probabilities model the weights associated with each training instance and indicate 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, which computes a prediction confidence for each example. How this confidence is determined is a design issue of the weak learning algorithm and will, for our approach, be discussed in detail in section 3.2.

On each round of boosting, the distribution over the training instances is modified in accordance with the learned weak hypothesis, i.e. in dependence of its assigned prediction confidence 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 learned 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. That way, the learner is confronted in each new round of boosting with a modified learning task and forced to focus on the examples in the training set which have not yet been correctly classified. Finally, all weak hypotheses learned are combined into one strong hypothesis. An instance x is classified by the strong hypothesis by adding up the prediction confidence of each weak hypothesis covering x and predicting the class y of x as positive if the sum of confidences of all hypotheses covering x is positive, otherwise predicting y 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, firstly assume that the weak learner produces on each iteration a hypothesis in form of a single Horn clause $H \leftarrow L_1, L_2, \dots, L_n [c]$ with an associated real number c , where H is the atom $p(X_1, \dots, X_{a(p)})$ 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 prediction 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 $H \leftarrow L_1, L_2, \dots, L_n [c]$ such that $c > 0$ implies that H is true. It is interpreted as classifying all instances covered by it as positive with prediction 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. $p(X) \leftarrow q(X,a)$. | [0.2] | 4. $p(X) \leftarrow q(X,Y), v(Y)$. | [-0.6] |
| 2. $p(X) \leftarrow q(X,Y), r(Y)$. | [0.9] | 5. $p(X) \leftarrow r(X)$. | [-0.5] |
| 3. $p(X) \leftarrow s(X)$. | [0.1] | 6. $p(X) \leftarrow q(X,b)$. | [-0.3] |
| | | 7. $p(X) \leftarrow t(X)$. | [-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, we find that 1,3,4,7 cover the example, so we compute the sum of their confidences, yielding $0.2+0.1-0.6-0.9 = -1.2 < 0$, and the instance is classified as negative. In other words, to understand the behaviour 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*, which is our interpretation of the ideas in [2], we will restrict each hypothesis to either of two forms. A hypothesis is either positively correlated, i.e. predicting the positive class, and equipped with a positive prediction confidence, or it is the default hypothesis $p(X_1, \dots, X_{a(p)})$ with an assigned negative confidence. Constraining the hypotheses to either of these two forms 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 prediction confidence in proportion to just the weight of the default hypothesis, instead of having to consider all other hypotheses and their assigned weights.

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

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 $H [c_1], \dots, H [c_n], 1 \leq i \leq n$, 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 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 discussed in more detail in the following sections.

3 The Weak Relational Learner

Our greedy top-down weak learner is using a refinement operator based on the concept of *foreign links* introduced in Midos [14]. This refinement operator is

elucidated in detail in the following section. We will then discuss in section 3.2 the heuristics guiding the search of the greedy weak learner based on this refinement operator in the hypothesis space. In Table 1, we give a more concise description of the weak greedy learner embedded into the framework of constrained confidence-rated boosting. In the following, references to steps in Table 1 will be indicated by “T1.⌊”.

3.1 The Refinement Operator

The refinement operator ρ is based on the concept of *foreign links* introduced in Midon [14]. The hypothesis space consists of non-recursive, function-free Horn clauses $C = H \leftarrow B$, where H is the atom $p(X_1, \dots, X_{a(p)})$ and p the target predicate of arity $a(p)$. In order to constrain the complexity of the hypothesis space, our weak learner employs a foreign literal restriction [14] as declarative bias which is a constrained form of linkedness of clauses. 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 by means of an explicit definition of those literals and variable positions that are to be considered for refinement. Hypotheses are only refined along link paths designated by these definitions, or so called foreign links. For a clause $C = L_1, \dots, L_n$, a foreign link between a variable V first occurring at position p_i in literal L_i with predicate name r , and a different variable U first occurring at position p_j in L_j with predicate name s is defined as $r[p_i] \rightarrow s[p_j]$.

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 hypotheses generated by the greedy weak relational learner. 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 C' 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] \rightarrow atm[1]`, applying ρ on $C = \text{active}(X_1)$ would result in the specializations

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

if X_2 is a nominal variable with the domain $\{c, cl\}$, and X_3 is a continuous variable with discretization $\mathcal{D} = [-0.782, 1.002]$.

More generally, let $L = r(V_1, \dots, V_{a(r)})$ be a literal with predicate name r of arity $a(r)$, and let $Vars(L)$ denote the variables in L not occurring in the clause C to be specialized. Then, adding $C' = C, L$ to $\rho(C)$ results in additionally adding to $\rho(C)$ the following refinements:

1. $C, L\theta_i^j, 1 \leq i \leq a(r), 1 \leq j \leq |Val(V_i)|$, such that $V_i \in Vars(L)$ and V_i is a variable with nominal values, where $Val(V_i)$ denotes the domain of variable V_i and $L\theta_i^j = r(V_1, \dots, V_{i-1}, V_i/c_j, V_{i+1}, \dots, V_{a(r)}), c_j \in Val(V_i)$
2. $C, L, \rho(V_i), 1 \leq i \leq a(r)$ such that $V_i \in Vars(L)$ and V_i is a variable with continuous values. $\rho(V_i)$ is defined for variables V_i with continuous values as follows. If $\mathcal{D} = \{d_1, \dots, d_n\}$ is the discretization of the values of V_i , then for any $d_k \in \mathcal{D}$ $V_i \leq d_k$ and $V_i > d_k$ are in $\rho(V_i)$.

Let $C = L_1, \dots, L_n$ be a clause to be specialized, and let $U_1, \dots, U_{k-1}, U_{k+1}, \dots, U_{a(s)}$ be new variables not occurring in C . Let $a(r)$ denote the arity of a literal with predicate name r , and let $Val(V)$ denote the domain of a variable V . Furthermore, let F be the set of all foreign links defined for the literals at hand. Then the refinement operator ρ can be defined as follows: for any $L_i = r(V_1, \dots, V_{a(r)})$ in C such that $r[m] \rightarrow s[k] \in F$,

1. $L_1, \dots, L_n, s(U_1, \dots, U_{k-1}, V_m, U_{k+1}, \dots, U_{a(s)}) \in \rho(C)$
2. $L_1, \dots, L_n, s(U_1, \dots, U_{k-1}, V_m, U_{k+1}, \dots, U_{a(s)})\theta_l^j \in \rho(C)$
for $1 \leq l \leq a(s), 1 \leq j \leq |Val(U_l)|, U_l$ a variable with nominal values
3. $L_1, \dots, L_n, s(U_1, \dots, U_{k-1}, V_m, U_{k+1}, \dots, U_{a(s)}), \rho(U_l) \in \rho(C)$
for $1 \leq l \leq a(s), U_l$ a variable with continuous values.

3.2 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

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

which is based on the collective weight of all positive and negative 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

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

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

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

The refinement operator ρ of the weak relational learner iteratively refines, as described in detail in section 3.1, the clause C currently maximizing the objective function \tilde{z} until either a clause C' is found with hitherto maximal $\tilde{z}(C')$ that covers only positive examples, or until the objective function \tilde{z} can not be further maximized (T1.2d).

The positively correlated clause C resulting from this greedy refinement process is subject to overfitting on the training instances, 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 set of training examples. 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 which have a non-positive prediction confidence on the whole training set are ruled out (T1.2f). If no generalization of C with a positive prediction 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

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

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

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

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

In a last step, the positively correlated generalization C' of C with minimal $\text{loss}(C')$ is compared to the default hypothesis with respect to the expected 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 of the greedy learner.

4 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 weak greedy learner are used in the framework of CCRB [2]. 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 $h_t : X \rightarrow \mathfrak{R}$,

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

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

Before starting the next round of boosting, the probability distribution over the training instances, which is initially uniform, is updated by means of h_t , namely by determining

$$D_i^{t'} = \frac{D_i^t}{e^{(y_i \cdot h_t(x_i))}}. \quad (6)$$

This way, the weights of all instances x not covered by the weak hypothesis C_t , i.e. such that $h_t(x) = 0$, 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 (T1.2j) by means of h_t .

Then, the sum of the resulting weights is normalized

$$D_i^{t+1} = \frac{D_i^{t'}}{\sum_i D_i^{t'}}, 1 \leq i \leq N, \quad (7)$$

so as to serve as the probability distribution of the next iteration.

Table 1. Constrained Confidence-Rated Boosting Algorithm

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** $t = 1 \dots T$
 - (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) $C := p(X_1, \dots, X_{a(p)})$
 - (c) $\tilde{Z} := 0$
 - (d) **While** $w_-(C, \mathcal{G}) > 0$
 - i. **Let** $C' := \operatorname{argmax}_{C'' \in \rho(C)} \{\tilde{z}(C'')\}$
 - 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) $\operatorname{Prunes}(C) := \{p(X_1, \dots, X_{a(p)}) \leftarrow B \mid C = p(X_1, \dots, X_{a(p)}) \leftarrow BB'\}$
 - (f) **Remove** from $\operatorname{Prunes}(C)$ all clauses C' where $c(C', E) \leq 0$
 - (g) **If** $\operatorname{Prunes}(C) = \emptyset$ let $C_t := p(X_1, \dots, X_{a(p)})$
 - (h) **Else**
 - i. $C' := \operatorname{argmin}_{C'' \in \operatorname{Prunes}(C)} \{\operatorname{loss}(C'')\}$, where $\operatorname{loss}(C'')$ is defined according to equation (5)
 - ii. **Let** $C_t := \operatorname{argmax}_{C'' \in \{C', p(X_1, \dots, X_{a(p)})\}} \left\{ \left(\sqrt{w_+(C'', E)} - \sqrt{w_-(C'', E)} \right)^2 \right\}$
 - (i) $h_t : X \rightarrow \mathfrak{R}$ is the function

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

- (j) **Update** the probability distribution D_t according to

$$D_i^{t'} = \frac{D_i^t}{e^{(y_i \cdot h_t(x_i))}}$$

$$\begin{cases} = D_i^t & \text{if } e = (x_i, y_i) \text{ not covered by } C_t \\ > D_i^t & \text{if } e \text{ covered by } C_t \text{ and } e \in E^- \\ < D_i^t & \text{if } e \text{ covered by } C_t \text{ and } e \in E^+ \end{cases}$$

$$D_i^{t+1} = \frac{D_i^{t'}}{\sum_i D_i^{t'}}, 1 \leq i \leq N$$

3. **Construct** the strong hypothesis

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

After the last iteration of the weak learner, the strong hypothesis is defined by means of all weak hypotheses induced by the training instances. For each instance x the prediction confidences of all hypotheses covering x are summed up. If this sum is positive, the strong hypothesis classifies x as positive, otherwise x is classified as negative:

$$H(x) := \text{sign} \left(\sum_{C_t: (x,y) \text{ covered by } C_t} c(C_t, E) \right) \quad (8)$$

5 Empirical Evaluation

We conducted an empirical evaluation of our approach to CCRB on two domains, namely on the domain of mutagenicity [13], which is a thoroughly investigated benchmark problem for ILP learners, and on the domain of Quantitative Structure Activity Relationships (QSARs), another important test-bed for ILP-systems [4, 5]. 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.

Mutagenicity: The task is to predict the mutagenicity of a set of small, highly structurally heterogeneous molecules (aromatic and heteroaromatic nitro compounds). Mutagenic compounds are often known to be carcinogenic and to cause damage to DNA. Not all compounds can be empirically tested for mutagenesis, and the prediction of mutagenicity is vital to understanding and predicting carcinogenesis. A molecule is described by its atoms, the bonds between them, global properties of and chemical structures present in the molecule.

Several relational descriptions of the domain are available [12], ranging from a weakly structured description \mathcal{B}_2 only involving the atoms and bonds of the molecules, to a strongly structured description \mathcal{B}_4 also involving high level chemical concepts present in the molecules. We conducted our experiment with C²RIB, which stands for **C**onstrained **C**onfidence-**R**ated **I**LP **B**oosting, 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. The predictive accuracy is estimated by 10-fold-cross-validation, where we used the same folds as [12] for their experiments with Progol. The accuracy obtained in our experiment with C²RIB is displayed in Table 2 together with reference results on the 188-dataset using background knowledge \mathcal{B}_4 and the sources from which the results are reported. Runtime¹ of C²RIB averages to 7 minutes for 100 iterations, as compared to 307 minutes for Progol on all 188 compounds (the run-time for Progol was determined in experiments we performed on our sparc SUNW, Ultra-4, in which we obtained the same accuracy as [13]).

In Table 2, we show only results obtained on the most comprehensive set of background knowledge, \mathcal{B}_4 , which we have worked with.² As can be seen

¹ All run-times are referring to results obtained on a sparc SUNW, Ultra-4.

² Additional results have been obtained by other authors on the \mathcal{B}_3 dataset [12], in particular by STILL [11] (87 ± 8) and G-Net [1] (91 ± 8).

from the table, C²RIB 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 hypotheses obtained by FOIL as published in [12]), and 28 literals on average in the hypotheses obtained by Progol. A final hypothesis obtained by C²RIB is displayed in Table 3 in the appendix.

QSARs: The task is to construct a predictive theory relating the activity of chemical compounds to their molecular structure. Often, these so called Qualitative Structure Activity Relationships cannot be derived solely from physical theory, and experimental evidence is needed. Again, not all compounds can be empirically evaluated, and machine learning methods offer a possibility to investigate QSARs. We conducted our experiments on a 5-fold-cross-validation series of 55 pyrimidine compounds as described in [5]. A pyrimidine compound is described by chemical groups that can be added at three possible substitution positions. A chemical group is an atom or a set of structurally connected atoms each of which is described by a set of chemical properties. QSARs problems are in general regression problems, i.e. not a class but real numbers must be predicted. To get around this problem for ILP, the greater activity relationship between pairs of compounds is learned. Rules learned for this relationship can then be employed to rank drugs by their activity. As opposed to [4, 5], we restrict our experiments to the prediction of the greater activity relationship between pairs of the 55 compounds.

We conducted experiments on the same data sets with the systems Progol [6] and FOIL [9] in order to obtain reference results on this domain (Table 2). The predictive accuracy obtained by C²RIB 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 C²RIB 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 C²RIB 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.

6 Related Work

The work described in this paper is based on recent research in the area of propositional boosting, and centrally builds on Cohen and Singer's [2] approach to constrained confidence-rated boosting. However, the properties of the weak learner embedded in the boosting framework, namely the declarative bias em-

Table 2. Accuracy, standard deviation, average run-time and number of literals in the final hypotheses on the 188 – \mathcal{B}_4 mutagenicity dataset and the QSARs dataset

		C ² RIB	FOIL	Fors	Progol
Mutagenicity	Accuracy \pm StdDev	88.0 \pm 6.0	82.0 \pm 3.0 [13]	89.0 \pm 6.0 [3]	88.0 \pm 2.0 [13]
	⊗ Runtime (minutes)	7	n/a	n/a	307
	⊗ Number of literals	64	46	n/a	28
QSARs	Accuracy \pm StdDev	83.2 \pm 3.0	82.9 \pm 2.7	n/a	79.8 \pm 3.7
	⊗ Runtime (minutes)	57	0.7	n/a	372
	⊗ Number of literals	142	140	n/a	154

ployed in form of a foreign literal restriction, the application of look-ahead, and the preclusion of hypotheses negatively correlated with the positive class, distinguish our work from the approach of [2].

The ILP work probably closest related to our approach is that of Quinlan [8]. However, Quinlan uses, in conjunction with Adaboost.M1, a standard ILP learner (FFOIL), so that the boosted ILP learner can be expected to produce fairly large run-times due to the high effort already expended by FFOIL. Moreover, FFOIL itself generates as the first-order learner embedded into the boosting framework weak hypotheses each of which comprises a set of clauses. Thus, the resulting strong hypothesis is apt to be highly complex. Lastly, this approach works, due to the absence of a confidence measure, with equal voting weights for all weak hypotheses, and, instead of a probability distribution over the training instances a re-sampling procedure is used to approximate the weights of the examples.

The weak learner employed in our approach is based on the refinement operator and declarative bias in form of *foreign links* introduced in Midos [14]. Additionally, a limited form of look-ahead has been employed in order to avoid the shortsightedness problem with respect to existential variables in the hypotheses generated by the greedy weak relational 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*, 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 constrained confidence rated boosting 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 [8]. 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 coevolutionary concept learning. In J. Shavlik, editor, *Proceedings 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. A. Karalic. *First Order Regression*. PhD thesis, University of Ljubljana, Faculty of Computer Science, Ljubljana, Slovenia, 1995.
4. 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. *Proceedings of the National Academy of Sciences of the United States of America* 89(23):11322-11326, 1992.
5. 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.
6. S. Muggleton. Inverse Entailment and Progol. *New Generation Computing*, 13:245-286, 1995.
7. J.R. Quinlan. Bagging, boosting, and C4.5. In *Proc. of 14th National Conference on Artificial Intelligence*, 1996.
8. J.R. Quinlan. Boosting First-Order Learning. *Algorithmic Learning Theory*, 1996.
9. J.R. Quinlan and R. M. Cameron-Jones. FOIL: A Midterm Report. In P. Brazdil, editor, *Proceedings of the 6th European Conference on Machine Learning*, volume 667, pages 3-20. Springer-Verlag, 1993.
10. R.E. Schapire. Theoretical views of boosting and applications. In *Proceedings of the 10th International Conference on Algorithmic Learning Theory*, 1999.
11. M. Sebag and C. Rouveirol. Resource-bounded Relational Reasoning: Induction and Deduction through Stochastic Matching. *Machine Learning*, 38:41-62, 2000.
12. A. Srinivasan, S. Muggleton, and R. King. Comparing the use of background knowledge by inductive logic programming systems. *Proceedings of the 5th International Workshop on Inductive Logic Programming*, 1995.

13. 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.
14. 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 - Proceedings of the PKDD-97*, pages 78-87, 1997.

A Sample Output from C²RIB

Table 3. A strong hypothesis obtained from C²RIB

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]
