A Comparative Evaluation of Feature Set Evolution Strategies for Multirelational Boosting [*] [†]

Susanne Hoche[1] and **Stefan Wrobel**[1,2]

[1]Fraunhofer Institute for Autonomous intelligent Systems

[2]Institute of Computer Science III, University of Bonn

{hoche,wrobel}@ais.fraunhofer.de

Abstract

Boosting has established itself as a successful technique for decreasing the generalization error of classification learners by basing predictions on ensembles of hypotheses. While previous research has shown that this technique can be made to work efficiently even in the context of multirelational learning by using simple learners and active feature selection, such approaches have relied on simple and static methods of determining feature selection ordering a priori and adding features only in a forward manner. In this paper, we investigate whether the distributional information present in boosting can usefully be exploited in the course of learning to reweight features and in fact even to dynamically adapt the feature set by adding the currently most relevant features and removing those that are no longer needed. Preliminary results show that these more informed feature set evolution strategies surprisingly have mixed effects on the number of features ultimately used in the ensemble, and on the resulting classification accuracy.

1 Introduction

Boosting is a well established method for decreasing the generalization error of classification learners and has been developed into practical algorithms that have demonstrated superior performance on a broad range of application problems in both propositional and multi-relational domains ([4; 13; 3; 12; 8]). Instead of searching for one highly accurate prediction rule entirely covering a given set of training examples, boosting algorithms construct ensembles of specialized rules by repeatedly calling a base learner on reweighted versions of the training data. Predictions are based on a combination of all members of the learned ensemble.

Previous work showed that this technique can be efficient even in the context of multirelational learning using simple learners and active feature selection [8; 7]. Active feature selection can be embedded into a boosting framework at virtually no extra cost by exploiting the characteristics of boosting itself to actively determine the set of features that is being used in the various iterations of the boosted learner [7]. By monitoring the progress of learning, and incrementally presenting features to the learner only if this appears to be necessary for further learning, we arrive at smaller feature sets and significantly reduced induction times without a deterioration of predictive accuracy.

The abovementioned positive effects on feature set size and induction time were achieved with extremely simple and uninformed selection strategies. In the approach introduced in [7] , feature weights were determined once at the beginning on the initially uniform example distribution, and no attempt was made to remove features that might have become irrelevant during the course of boosting due to the changes in the underlying example distribution.

In this paper, we therefore investigate whether learning results can be further improved by employing the distributional information present in boosting to reweight features and to dynamically adapt the feature set. In addition, we explore the effect of establishing new feature orders based on considering different sorting criteria as well as different subsets of features and examples. A number of different strategies to feature subset evolution are looked at and evaluated on several multirelational domains. Interestingly, the empirical evaluation shows that more informed feature selection strategies have mixed effects on the size of feature sets and classification accuracy, indicating that the increase in the power of the weak learner achieved by making better feature sets available might be offset by the induction of mutually contradictory base hypotheses produced by features that are very specific to extremal distributions towards the end of the boosting process.

This paper is organized as follows. In Section 2, we review constrained con-fidence-rated boosting. Section 3 provides an overview of the simple uninformed approach to active feature selection in the framework of constrained confidence-rated boosting. In section 4, we present our approach to feature set evolution strategies for multirelational boosting. Our experimental evaluation of the approach is described and discussed in Section 5. In Section 6, we conclude with some pointers to future work.

[*] This is a slightly modified version of [6]

[†] Partially supported by the DFG projects (WR 40/1-3) *Nachhaltige Informationsfusion: Aktives Lernen* and (WR 40/2-1) *Hybride Methoden und Systemarchitekturen für heterogene Informationsräume*

2 Constrained Confidence-Rated ILP-Boosting

Boosting has emerged as a successful method for improving the predictive accuracy of a learning system by combining a set of base classifiers constructed by iterative calls to a base learner into one single hypothesis [4; 13; 3; 12; 8]. The idea is to "boost" a weak learner performing only slightly better than random guessing into an arbitrarily accurate learner by constructing an ensemble of base hypotheses and combining them into one final hypothesis.

To this end, a base learner is repeatedly called on reweighted versions of a set E of training instances. In each round t of boosting, a probability distribution D^t over E is maintained which models the weight D_i^t associated with each training example e_i in the t-th iteration. D_i^t indicates the influence of an instance when constructing a base classifier h_t. Initially, all instances have equal influence on the construction of a base hypothesis, i.e. the probability distribution D^1 is uniform. In each iterative call t to the base learner, a base hypothesis h_t is learned based on E weighted according to the current distribution D^t over E, and used to update the distribution for the next iteration. The weights of misclassified instances are increased while the weights of correctly classified instances are decreased, in order to focus on the examples which have not yet been correctly classified. Finally, all base hypotheses learned are combined into one final hypothesis H by a weighted majority vote of the base hypotheses.

In [8], we extended a specific approach to boosting known as constrained confidence-rated boosting, first introduced in [3], to multirelational problems. Combined with an appropriate refinement operator and search heuristics, constrained confidence-rated boosting is an effective approach to producing highly accurate multi-relational models while at the same time ensuring limited complexity of the final ensemble and acceptable induction times, as shown in [8] with the system C^2RIB (Constrained Confidence-Rated ILP Boosting).

Since for the active feature selection strategies developed and evaluated in this paper, we have chosen C^2RIB as the basic algorithm and point of reference, we provide a summary of the algorithm below; for more details, the reader is referred to [8].

C^2RIB accepts as input the total number of iterations of the base learner, and a set $E = \{(x_1, y_1), \cdots, (x_N, y_N)\}$ of positive training examples $(x_i, 1)$ and negative training examples $(x_i, -1)$, where each x_i belongs to an instance space X. Additionally, background knowledge may be provided.

In each iterative call t of the base learner, a base hypothesis h_t is learned on E, based on the current distribution D^t. In the framework of confidence-rated boosting, the prediction of a base hypothesis h_t is confidence-rated. A prediction confidence $c(h_t, E)$ is assigned to each base hypothesis h_t. The sign of $c(h_t, E)$ indicates the label predicted by h_t to be assigned to an instance, whereas the absolute value of $c(h_t, E)$ is interpreted as the confidence in h_t's prediction. This prediction confidence is used to update D^t for the next iteration, and as h_t's vote in the final hypothesis H.

The constrained form of confidence-rated boosting which we apply here is such that the base learner is restricted to induce only hypotheses predicting the positive class with a positive prediction confidence for all examples covered by the hypothesis, and to abstain on all examples not covered by it. Additionally, the so called default hypothesis is admissible, just comprising the target predicate to be learned and satisfying all examples. The confidence assigned to the default hypothesis conforms to the sign of the class of examples with the largest sum of probabilities according to the current distribution D^t. This weighted majority class may change over the course of iterations, depending on the learned base hypotheses and thus the examples on which the learner is currently focusing.

The prediction confidence of the current base hypothesis is used to update the current probability distribution D^t such that misclassified instances will have higher weights in the next iteration of the learner. After the last iteration of the base learner, the final, strong, hypothesis is derived from all base hypotheses induced from the training instances. To classify an instance x the prediction confidences of all hypotheses h_t covering x are summed up. If this sum is positive, the strong hypothesis H classifies x as positive, otherwise x is classified as negative.

3 Simple Baseline Strategy of Active Feature Selection

Efficiency and effectiveness of learning crucially depend on the *representation* of the objects that are used for learning. The inclusion of unnecessary features or, in a multirelational setting, of unnecessary relations, makes learning less efficient and often less effective. Feature selection therefore is a central topic in machine learning research. Using a very simple strategy, it is possible to exploit the characteristics of boosting in order to perform active feature selection, resulting in lower induction times and reduced complexity of the ensemble [7]. Since boosting tries to increase the certainty with which examples are classified by the ensemble of hypotheses — this is expressed in terms of the so-called *margin* —, as shown in [7], one can monitor the development of the margin in order to determine when new features or relations might be needed.

The approach presented in [7] has used quite a simple strategy which we will use as a baseline for comparison of the more advanced and informed strategies we develop and evaluate in this paper. The algorithm C^2RIB^D described in [7] simply orders the available features in the different relations based on a heuristic relevance measure (mutual information). Boosting is started with a minimal set of features, and proceeds until the development of the margin indicates that progress is slowing down at which moment the next feature on the list is added to the representation. In more detail, when integrated into the algorithm of C^2RIB described in section 2, the simple baseline active feature selection strategy works as described in the following (for a complete description see [7]).

C^2RIB^D accepts as input, in addition to the input of C^2RIB, the set $\mathcal{F}$ of features present in the training examples, sorted in descending order according to some criterion, and a subset $\mathcal{F}'$ of the top most features of $\mathcal{F}$.

In order to actively select features depending on the requirements of the problem, and thus accelerate the learning process of the boosted learner C^2RIB without a deterioration of its prediction accuracy, we start the learner with the features in $\mathcal{F}'$ and the relations in which these features occur, monitor the learning progress and include additional features and relations into the learning process only by demand.

The learning progress is monitored in terms of the development of the training examples' mean margins. The margin of an example $e_i = (x_i, y_i)$ under an ensemble H_t of base classifiers $h_1, h_2, \cdots, h_t$ is a real-valued number $margin(H_t, e_i) \in [-1, 1]$ indicating the amount of disagreement of the classifiers in H_t with respect to e_i's class. For the binary case we deal with here, we can define the margin of e_i under H_t as the difference between the sum of the absolute weights of those base classifiers in H_t predicting for e_i its correct class y_i, and the sum of the absolute weights of those base classifiers in H_t predicting for e_i the incorrect class $y \neq y_i$ [7].

Large positive margins indicate a "confident" correct classification. The more negative a margin is, the more confident an incorrect classification is indicated. Boosting is known to be especially effective at increasing the margins of the training examples [15; 5]. By increasing their probabilities, boosting forces the focus on misclassified instances which show small or even negative margins. The learner is forced to search for base hypotheses which correctly classify these hard examples and thus increase their margins. Since the margins are increasing in the course of iterated calls to the base learner, the gradient of the mean margins can be assumed to be positive and be employed to monitor the quality of the learning process.

For monitoring the learning success, we define in each iteration t of boosting the gradient $gradient(t)$ of t as the slope of the line determined by the least square fit to the average margins in each single iteration 1 to t. We then average the gradients over the last T_l iterations as to smooth temporary fluctuations in the margins' development, and compute the ratio of the averaged previous gradients and the current gradient.

The margins' improvement is measured by this ratio which increases from one iteration to the next as long as the margins increase significantly. As soon as the ratio starts to decrease, an estimate for the slowdown in the margins' improvements is determined. This estimate predicts the expected decrease of the ratio and is used to determine when a new feature has to be presented to the learner. Whenever the actual decrease of the ratio exceeds the predicted decrease by a certain threshold α, a new feature is included into the learning process.

4 Feature Set Evolution Strategies for Multirelational Boosting

We investigate whether more complex strategies of feature ordering and selection can further improve the learning results. Instead of relying on the sequence that has been initially determined on the entire training set based on the features' mutual information with the class, a new feature order is established, by reweighting features based on the current distribution over the training examples, every time a feature is requested by the base learner. Such a new feature order can be arrived at by considering:

- the entire training set for feature reweighting, or only the fraction of examples which are misclassified by the current ensemble of base hypotheses;
- the entire original feature set or just the features which have not been presented to the learner yet;
- not only the mutual information of a feature with the class but also the *conditional* mutual information of a feature with the class, given the values of other features.

Moreover, features can be simply presented incrementally to the learner or, alternatively, features that are no longer needed can be substituted by the currently most relevant features.

Based on these considerations, we investigate several approaches to feature ordering and selection. In the following, we discuss the properties with respect to which we categorize the different strategies summarized in Table 1.

- Considered features (column 3 of Table 1): Each ensemble member is specialized on a certain region of the instance space. As the distribution over the examples changes, the required special knowledge might shift. We consider the question whether prediction accuracy can be improved by dismissing features that no longer meet the very current requirements of the learning task.
- Considered examples (column 4 of Table 1): Since one of boosting's basic principle is to focus the base learner on examples which are misclassified by the current ensemble, we consider the question whether predictive accuracy can be improved by presenting to the base learner features which are especially helpful to correctly classify the examples which have been misclassified so far.
- Sorting criterion (column 5 of Table 1): In addition to the mutual information of each single feature with the class we compute the features' conditional mutual information (CMI) with the class, given another feature, to investigate whether learning results can be improved by considering groups of features which optimally separate the given examples. The conditional mutual information between feature F_k and class C, given feature F_j, reflects the amount of information about the class that is obtained when the values of features F_j and F_k are known:

$$CMI(C, F_k|F_j) = E(C) - E(C|F_k, F_j),$$

  where $E(C)$ and $E(C|F_k, F_j)$ denote the entropy and the conditional entropy, respectively, of C.
- Set evolution strategy (column 6 of Table 1): We consider the question whether we can improve prediction accuracy by substituting features that are no longer needed by currently most relevant features instead of augmenting the set of features to be considered for refinement.
- Partitioning continuous features (last column of Table 1): We investigate whether learning results

can be improved by discretizing continuous features in a way that reflects the current distribution over the training examples. To this end, we apply an objective function of the constrained confidence-rated boosting framework (cf. [8]) to partition the continuous range of a feature F such that the training examples are optimally separated by F according to the current distribution.

It is common to all approaches V1 to V16 that, exactly as in the baseline strategy C^2RIB^D, the available features in the different relations are initially ordered based on the heuristic relevance measure of mutual information, and that the learner starts with the top two features according to this order. In contrast to the baseline method, every time a feature is requested by the learner, a new feature order is determined based on the current distribution.

In all versions, except group I (V1 and V2), the value ranges of continuous features are partitioned based on the current distributional information and the boosting objective function (cf. [8]).

The versions in group I to III compute the features' mutual information with the class. V1 to V4 consider only those features which have not been presented to the learner yet, and augment the set of currently active features with the top feature of the new sequence. One version considers all examples for reweighting, the other one considers only those which are misclassified by the current ensemble. V5 and V6 consider the entire feature set for reweighting, substitute the features already presented to the learner with the same number of top features from the new sequence, and activate one additional top feature of the new ranking. Again, one version considers all examples for reweighting, the other one considers the misclassified ones only. V7 and V8 (group III) determine, based on all examples and the misclassified ones only, respectively, a new sequence of all features and substitute the worst active feature with the top feature of the new sequence. Thus, only two features are active at a time.

In V9 to V12 (group IV), the feature with the highest conditional mutual information with the class, given the feature which was last activated, is presented to the learner in an incremental manner. Versions V9 to V12 account for all possible combinations of features and examples to be considered for reweighting.

In V13 to V16 (group V), the set of currently active features is substituted by the feature F with the highest mutual information with the class, and the feature with the highest conditional mutual information with the class, given F, again accounting for all possible combinations of features and examples to be considered for reweighting. Only two features are active at a time.

5 Empirical Evaluation

5.1 Experimental Design

We evaluated the different feature set evolution strategies on a total of six learning problems: two classical ILP domains, Mutagenicity [17] (prediction of mutagenic activity of 188 molecules (description $\mathcal{B}_4$)) and QSARs, Quantitative Structure Activity Relationships, [10; 9] (prediction of a greater-activity relationship between pairs of compounds based on their structure), one artificial problem, the Eastbound Trains[1] proposed by Ryszard Michalski (prediction of trains' directions based on their properties), and three general knowledge and data mining tasks, Task A and AC of the PKDD Discovery Challenge 2000 [1] (classification of loans, where Task AC is based on all loans, and Task A only on the closed loans from Task AC), and Task 2 of the KDD Cup 2001 [2] (prediction of gene functions).

The standard version C^2RIB^D and each of the versions described in Section 4 is run with $T = 50$ iterations of the base learner. In all experiments, the threshold α – which the deviation between the actual decrease of the learning curve and its predicted decrease is not allowed to exceed – is set to 1.01. The value 1.01 has been empirically determined on the domain of Mutagenicity [17], and has not been modified for subsequent experiments on the other domains in order to ensure proper cross validation results.

Since we expected more informed feature set evolution strategies to result in more extreme learning curves, we decided to average the gradients of the examples' mean margin over a smaller number T_l of iterations than in earlier experiments, where we used $T_l = 10$. To ensure a fair comparison of the uninformed approach and the new strategies, we compared the results of the base case C^2RIB^D with $T_l = 10$ against a new value $T_l = 3$. In two thirds of our domains, $T_l = 3$ resulted in a, to some extent rather large, deterioration of classification accuracy, in one third only to very slight improvements. Similarly, we compared for one of the informed feature selection strategies on one of the domains the learning curves with $T_l = 3$ and $T_l = 10$. As expected, the learning curve initially increased significantly stronger and later dropped significantly slower when using a more complex feature ordering and selection strategy. This had the effect that, with $T_l = 10$, the learning curve's estimate was lower than the learning curve itself, and consequently no additional features were requested by the learner. Reducing the number of iterations over which the gradients of the examples' mean margins are averaged to $T_l = 3$ has the effect that the different development of the learning curve can be better estimated and thus features are introduced into the learning process over the course of iterations. Thus, the gradients of the examples' mean margins are averaged for the standard version C^2RIB^D over the last $T_l = 10$, and for the more complex feature set evolution strategies over the last $T_l = 3$ iterations.

5.2 Detailed Results and Discussion

The resulting predictive accuracies and the average number of features required by the learner are depicted in Table 2 together with the standard deviations. The predictive accuracy is estimated by 10-fold-cross validation with the exception of the QSARs domain, where 5-fold-cross validation is used, and the Eastbound Trains, where the data is split into one training and test set partition, and the results are averaged over 10 iterations of the experiment.

[1] The examples were generated with the Random Train Generator available at http://www-users-cs-york.ac.uk/∼stephen/progol.html

Table 1: Strategies to feature ordering and selection based on the distributional information present in boosting. The strategies differ with respect to the features and examples, respectively, considered for establishing a new feature order, the applied sorting criterion, the set evolution strategy, and the discretization strategy for continuous features. The n in Set Evolution Strategy "Substitute n" denotes the number of features already presented to the learner.

Group ♯	Version ♯	Considered Features	Considered Examples	Sorting Criterion	Set Evolution Strategy	Partitioning Continuous Features
I	V1	Remaining	All	MI	Add 1	N
	V2	Remaining	Misclassified	MI	Add 1	N
II	V3	Remaining	All	MI	Add 1	Y
	V4	Remaining	Misclassified	MI	Add 1	Y
	V5	All	All	MI	Substitute n, Add 1	Y
	V6	All	Misclassified	MI	Substitute n, Add 1	Y
III	V7	All	All	MI	Substitute 1	Y
	V8	All	Misclassified	MI	Substitute 1	Y
IV	V9	Remaining	All	CMI	Add 1	Y
	V10	Remaining	Misclassified	CMI	Add 1	Y
	V11	All	All	CMI	Add 1	Y
	V12	All	Misclassified	CMI	Add 1	Y
V	V13	Remaining	All	CMI	Substitute 2	Y
	V14	Remaining	Misclassified	CMI	Substitute 2	Y
	V15	All	All	CMI	Substitute 2	Y
	V16	All	Misclassified	CMI	Substitute 2	Y

The results indicate on the one hand that further improvements can indeed be achieved by using more complex approaches to feature ordering and selection. On the other hand, they clearly show that it has to be considered very carefully which strategy to apply. Combining, for example, substitution of features with reordering all features and applying the CMI criterion (as in V15 and V16), seems to lead to inferior results. The dynamics inherent to boosting already cause the underlying learner to direct its attention on the difficult, or extreme, instances. Strategies overly intensifying the focus in this direction most probably tend to misleadingly lay emphasis on a few extreme examples which leads to inferior results. We will detail on this issue in more depth after a thorough discussion of the single strategies' results summarized in Table 2. We base our discussion of the strategies' performance on their win-loss-tie record in comparison to the baseline strategy C^2RIB^D. Then, possible explanations will be discussed in the context of all results.

The entries of groups I to III in the upper half of Table 1 which all apply the MI criterion show an increased prediction confidence over the base case, C^2RIB^D, in all but one cases. This is especially surprising for the versions V1 and V2 (group I), since they apply a simple reordering of the features not yet presented to the learner based on the examples' current weights, without prior partitioning the continuous features' value ranges. Both strategies add the best feature with respect to the new ranking, and yield a better accuracy than and feature subsets of about the same size as C^2RIB^D. V2 – using only the weights of misclassified examples to establish a new ranking – clearly outperforms V1 – considering all examples for reweighting – with respect to classification accuracy.

Versions V3 and V4 of group II differ from V1 and V2 in terms of continuous features which are discretized based on the current distribution over the training set. Both strategies are superior to C^2RIB^D with respect to classification accuracy, and yield – with one exception – feature subsets of about the same size as or slightly smaller than C^2RIB^D. Again, the version considering the misclassified examples only (V4) is superior to the one considering all examples (V3).

For versions V5 and V6 of group II – considering the entire feature set to establish a new feature ranking, substituting all active features with the same number of currently most relevant features and adding one additional feature – the number of features required by the learner is larger than for the base case in one third, and smaller in one fifth of the cases. Again, considering misclassified examples only for reweighting yields the better results. V5 considers the entire training set, and its prediction confidence is inferior to that of the base case. In contrast, V6 – using only the weights of misclassified examples to establish a new ranking – is clearly superior to C^2RIB^D.

An opposite effect can be observed in group III (versions V7 and V8), where all features are considered for a new ranking, and only two features are active at a time. Every time a new order has been determined, the worst active feature is substituted with the top feature of the new sequence. Here, the version considering the entire training set for reweighting (V7) is superior to the case where only the weights of misclassified examples are used to establish a new feature order (V8). In both cases, the classification accuracy is better, but the number of requested features is predominantly larger than for the base case.

All versions of groups IV and V in Table 1 are based

Table 2: Accuracy ± standard deviation, and number of requested features ± standard deviation after 50 iterations for C^2RIB^D and the feature set evolution strategies V1 to V16 on several multirelational domains

	V ♯	KDD01	Mutagenicity	PKDD-A	PKDD-C	QSARs	Trains
	C^RIB^D	90.15 ±7.92 5.5±3.4	83.50 ±5.80 4.4±2.0	86.70 ±6.64 5.5±1.8	88.57 ±2.95 6.3±1.8	78.76 ±1.76 3.4±1.5	80.00 ±15.32 5.9±1.9
I	V1	91.58 ±6.32 6.6±2.2	80.87 ±9.94 4.2±1.6	86.70 ±6.64 6.9±2.6	88.87 ±3.35 5.7±2.6	81.37 ±2.03 3.3±1.5	78.33 ±8.05 5.8±1.4
	V2	90.33 ±7.73 5.4±3.9	85.60 ±4.45 7.4±1.2	86.70 ±6.64 6.1±2.5	88.87 ±3.35 8.6±3.0	80.77 ±2.93 2.3±0.5	81.67 ±16.57 5.7±1.2
II	V3	90.29 ±7.44 2.5±1.1	85.67 ±6.03 5.4±1.4	86.70 ±6.64 6.6±1.8	88.72 ±3.12 7.0±2.3	81.51 ±2.39 4.0±0.0	86.67 ±10.54 5.8±1.1
	V4	90.57 ±7.32 3.5±2.3	86.19 ±5.01 4.9±1.7	86.70 ±6.64 5.9±2.6	88.87 ±3.35 5.7±4.1	80.96 ±3.45 6.0±2.8	83.33 ±11.11 6.6±1.0
	V5	90.12 ±7.99 3.7±4.0	86.72 ±6.64 5.6±1.8	86.70 ±6.64 7.1±2.6	88.87 ±3.35 8.4±2.0	77.45 ±3.09 5.0±1.4	73.33 ±11.65 6.1±1.2
	V6	89.87 ±8.53 2.4±0.9	87.77 ±6.08 5.2±1.8	86.70 ±6.64 7.3±1.9	88.87 ±3.35 8.9±1.0	80.13 ±3.19 3.5±0.6	88.33 ±13.72 6.0±0.7
III	V7	90.73 ±7.13 2.9±1.9	85.14 ±8.20 5.8±2.2	86.70 ±6.64 6.9±3.0	88.87 ±3.35 7.9±3.3	78.14 ±5.63 6.2±4.7	81.67 ±9.46 7.5±3.4
	V8	89.95 ±8.34 2.2±0.6	85.60 ±9.99 7.4±2.2	86.70 ±6.64 8.5±3.2	88.87 ±3.35 8.0±3.5	76.73 ±4.21 4.4±1.3	85.0 ±14.59 6.3±2.0
IV	V9	90.61 ±6.40 2.4±1.1	82.51 ±7.35 4.2±1.7	86.23 ±6.76 2.9±1.7	87.46 ±2.73 8.7±1.3	78.93 ±4.77 2.8±1.0	81.67 ±9.46 5.8±1.5
	V10	90.65 ±6.26 2.5±1.1	88.55 ±7.07 5.7±2.2	82.88 ±7.25 2.0±0.0	88.65 ±2.78 5.8±2.5	80.43 ±2.61 5.5±0.7	81.67 ±9.46 6.2±0.8
	V11	90.75 ±6.24 2.2±0.4	84.55 ±6.35 3.5±0.7	86.19 ±7.5 2.9±1.2	88.72 ±3.2 4.6±2.0	79.03 ±2.56 2.8±0.8	78.33 ±11.25 4.0±0.0
	V12	90.31 ±6.61 2.2±0.4	82.38 ±8.03 4.5±1.1	85.72 ±7.67 2.0±0.0	88.97 ±3.19 3.3±2.2	78.24 ±2.02 3.0±0.7	78.33 ±8.05 3.4±0.5
V	V13	91.12 ±6.28 3.85±2.9	86.19 ±6.12 9.6± 4.5	82.46 ±6.74 10.8± 7.2	88.43 ± 2.99 11.0±7.7	72.77 ±2.47 4.4±2.2	80.0 ±10.54 12.4±4.2
	V14	90.53 ±6.38 2.8±2.2	86.29 ±5.04 10.0±3.7	87.02 ±7.64 2± 0.0	86.14 ±2.48 5.5±7.0	72.75 ±2.38 4.8±3.4	76.67 ±2.61 9.2±3.9
	V15	89.93 ±6.71 2.6±1.3	82.38 ±8.03 6.4±3.4	85.83 ±7.16 6.6± 5.3	88.13 ±2.66 13.0±6.9	72.64 ±2.33 2.4±0.9	73.33 ±11.65 6.8±2.4
	V16	90.62 ±6.18 2.9±1.8	82.83 ±6.31 8.2±3.7	84.18 ±9.2 2.7±1.2	86.85 ±4.29 2.0±0.0	72.81 ±2.35 6.8±5.6	75.0 ±11.79 10.2±5.1

on the CMI criterion. The results show that some of the strategies (V10 and V11) clearly outperform the baseline strategy C^2RIB^D both in terms of predictive accuracy and reduction of the number of features required by the learner. The remaining versions, however, not only do not yield any improvements but mostly deteriorate the learning results in all respects.

In all versions of group IV (V9 to V12), the features are presented to the learner in a forward manner, and they all arrive at a smaller number of features requested for learning than the base case. V9 and V10 consider only the remaining features to determine a new feature order. Again, the version considering only the weights of the misclassified examples (V10) is superior to the version using the weights of all examples (V9). V10 clearly outperforms C^2RIB^D with respect to accuracy, V9 is on par with the base case. In both versions, the number of features requested by the learner is smaller than or on par with the base case.

In V11 and V12, all features are reordered every time a new feature is requested. As for V7 and V8, considering only the misclassified examples is inferior to considering all examples. V11 clearly outperforms the base case both in terms of accuracy and number of required features. V12 requires smaller feature subsets than C^2RIB^D but yields a lower classification accuracies.

In the versions of group V (V13 to V16), only 2 features are active at at a time, namely the feature F with the currently highest mutual information with the class, and the feature with the currently highest conditional mutual information with the class, given F. None of these versions yields any improvement with respect to the base case. V14 is on par with C^2RIB^D with respect to both accuracy and number of features required for learning. However, the remaining three strategies are clearly inferior to the base case in all respects. Again, the versions only considering the weights of misclassified examples (V14, V16) yield better results than the strategies using the weights of all examples (V13, V15).

These detailed results indicate, that the strategies establishing a new order for those features only which have not been presented to the learner yet, based on the weights of misclassified examples only, and adding the currently best feature to the set of active features (V2, V4, V10) are the most successful strategies, and are clearly superior to the base case. Applying the relevance measure of mutual information of a feature with the class (V2, V4) seems to outperform the use of the CMI measure (V10).

The strategy yielding the – by far – worst results is the combination of considering all features for reordering, all examples for weighting, substitution of the only two active features with the currently best two features, and the relevance measure of CMI (V15). V15 is, with respect to predictive accuracy, inferior to C^2RIB^D on all six domains, and requires more features in two thirds of the domains. V5, the only MI based strategy (groups I to III) which performs worse than the uninformed baseline strategy C^2RIB^D, combines exactly the means which lead to the worst results in the CMI groups (IV and V). Thus, one can conjecture that this combination should be avoided. The comparatively good classification accuracy of V7 seems to contradict this conjecture. However, V7 results in all but one cases in a larger number of features required by the learner which might indicate that the selected features are not optimal.

Version V11 differs from the worst strategy, V15, only with respect to the set evolution strategy but yields both higher classification accuracy and smaller feature subsets than C^2RIB^D. This provides us with an idea about how “extreme” the single strategies are. We can interpret the “all features, all examples, substitute”-strategy (AAS) as the most extreme method. AAS completely deprives the learner of its current equipment and very strongly directs the focus to the present situation. The classification accuracy deteriorates and the number of features required for learning increases as a result to an insufficient equipment of the learner.

When instead the best feature is added to the current equipment, the learner concentrates much less on just the current state, and it becomes less likely that the learner only focuses on a few extreme, or misleading, examples. The “remaining features, misclassified examples, add”-strategy (RMA) which yields the best results in our experiments can be interpreted as the less extreme method. RMA is a fairly cautious strategy and does not further intensify the dynamics inherent to boosting. Not only is the prediction accuracy higher but the number of features requested from the learner is also smaller. The request of a large number of features is most likely due to situations where new features are selected based on misleading information. These features seem to be more promising than they really are. Since the chosen equipment is inadequate, the learner soon tries to level out the insufficiency by requesting yet another feature.

5.3 Summary and Implications

As a bottom line, we can see that positive effects on the classification accuracy and the number of features ultimately used for learning can be achieved by applying more informed feature selection strategies which utilize the distributional information provided by boosting without overly intensifying the dynamics inherent to boosting. The most successful strategies are those which add in a forward manner from the set of features not yet presented to the learner the one that scores best on the misclassified examples with respect to the MI relevance measure. Strategies which further intensify the dynamics of boosting, i.e. which result in a even stronger focus on only a few extreme examples, should be avoided since they lead to a clear deterioration of the results.

One could presume that this deterioration stems from an overfitting effect. However, since C^2RIB^D employs an effective overfitting avoidance strategy, we rather conjecture that learning is inhibited by focusing too much on features that are very specific to extremal distributions over the training data. Preliminary analysis of the base hypotheses' prediction confidences and the training examples' mean margins over the course of iterations rather indicates that the selection of features which are significant only for a very small fraction E' of “extreme” training examples results in the construction of a base hypothesis so unrepresentative for the entire training data, that it is right away leveled out by the regularization mechanisms of the boosting algorithm. In the next iteration, a contrary base hypothesis is induced which in turn, forces the learner to concentrate again on E', and to repeat the same process, or to request that a new feature order be established. Thus, the learner seems to eventually come to a point where further learning is inhibited.

6 Conclusion

In this paper, we have investigated informed approaches to feature ordering and selection in the framework of active feature selection and constrained confidence-rated multirelational boosting. Active feature selection can be embedded into a boosting framework at virtually no extra cost by exploiting the characteristics of boosting itself to actively determine the set of features that is being used in the various iterations of the boosted learner [7]. By monitoring the progress of learning, and incrementally presenting features to the learner only if this appears to be necessary for further learning, one can, even with a simple uninformed approach to feature ordering and selection, arrive at a smaller feature set, and significantly reduced induction times without a deterioration of predictive accuracy.

Here, we investigated whether classification accuracy and the number of features used in the learning process can be further improved by making use of the distributional information present in boosting to reweight features and to dynamically adapt the feature set. In addition, we explored the effect of establishing new feature orders based on considering different sorting criteria as well as different subsets of features and examples.

The empirical evaluation of several different strategies to feature subset evolution on a number of multirelational domains shows that more informed feature selection strategies have mixed effects on the size of feature sets and classification accuracy. Prediction accuracy can be improved by utilizing the distribution over the training examples maintained by boosting, for example in combination with the heuristic relevance measure of mutual information. Positive effects on the classification accuracy and the number of features ultimately used for learning can be achieved with the relevance measure of conditional mutual information, whereby the features and examples used for reordering and reweighting, respectively, have to be carefully considered in order to avoid the selection of features which are only significant for very few examples and misleading for the overall learning process.

As a next step, the dynamics in boosting which lead to the induction of mutually contradictory base hypotheses in the presence of powerful feature subsets have to be thoroughly investigated. Moreover, other relevance measures and approaches to feature selection will be investigated.

We would like to thank Jörg Kaduk for valuable discussions and helpful comments on previous versions of this paper.

References

[1] P. Berka. Guide to the financial Data Set. In: *A. Siebes and P. Berka, editors, PKDD2000 Discovery Challenge*, 2000.

[2] J. Cheng, C. Hatzis, H. Hayashi, M.-A. Krogel, Sh. Morishita, D. Page, and J. Sese. KDD Cup 2001 Report. In: *SIGKDD Explorations, 3(2):47-64*, 2002.

[3] W. Cohen and Y. Singer. A Simple, Fast, and Effective Rule Learner. *Proc. of 16th National Conference on Artificial Intelligence*, 1999.

[4] Y. Freund, and R.E. Schapire. Experiments with a New Boosting Algorithm. *Proc. of 13th International Conference on Machine Learning*, 1996.

[5] A.J. Grove, and D. Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. *Proc. of 15th National Conf. on AI*, 1998.

[6] S. Hoche, and S. Wrobel. A Comparative Evaluation of Feature Set Evolution Strategies for Multirelational Boosting. *Proc. 13th Int. Conf. on ILP*, 2003.

[7] S. Hoche, and S. Wrobel. Scaling Boosting by Margin-Based Inclusion of Features and Relations. *Proc. 13th European Conf. on Machine Learning (ECML'02)*, 2002.

[8] S. Hoche, and S. Wrobel. Relational Learning Using Constrained Confidence-Rated Boosting. *Proc. 11th Int. Conf. on ILP*, 2001.

[9] 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.

[10] 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 USA* 89(23):11322-11326, 1992.

[11] W.J. McGill. Multivariate information transmission. *IRE Trans. Inf. Theory*, 1995.

[12] D. Opitz, and R. Maclin. Popular Ensemble Method: An Empirical Study. *Journal of Artificial Intelligence Research 11, pages 169-198*, 1999.

[13] J.R. Quinlan. Bagging, boosting, and C4.5. *Proc. of 14th Nat. Conf. on AI*, 1996.

[14] R.E. Schapire. Theoretical views of boosting and applications. *Proceedings of the 10th International Conference on Algorithmic Learning Theory*, 1999.

[15] R.E. Schapire, Y. Freund, P.Bartlett, and W.S. Lee. Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. *The Annals of Statistics, 26(5):1651-1686*, 1998.

[16] C.E. Shannon. A mathematical theory of communication. *Bell. Syst. Techn. J.*, 27:379-423, 1948.

[17] 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*, 1996.