

Language Series Revisited: The Complexity of Hypothesis Spaces in ILP (Extended Abstract)

Irene Weber, Birgit Tausend and Irene Stahl

Fakultät Informatik, Universität Stuttgart, Breitwiesenstr. 20-22, D-70565 Stuttgart

Restrictions on the number and depth of existential variables as defined in the language series of CLINT [3] or the *ij*-determinacy constraint of GOLEM [2] are widely used in ILP and expected to produce a considerable reduction in the size of the hypothesis space (see [1] for an empirical comparison). In this paper we will show that this expectation does not hold in general.

In CLINT, language series are introduced in order to allow a dynamic extension of the hypothesis language in case that learning fails in the current hypothesis space. Based on combinatorial investigations, we argue that the language definitions chosen in CLINT are unsuitable for sensible bias shift operations. We propose alternative approaches resulting in the desired reduction of the hypothesis space and allowing for a natural integration of the shift of bias.

Problem Setting. Given a set of examples $E = E^+ \cup E^-$, the background knowledge K and a hypothesis language L_H , the task of Inductive Logic Programming is to find a logic program $H \in L_H$ that is necessary, i.e., $K \not\vdash E^+$, sufficient, i.e., $K \wedge H \vdash E^+$, and consistent, i.e., $K \wedge H \wedge E^- \not\vdash \square$. An important point when comparing and evaluating different systems is the complexity of the hypothesis space that is searched in L_H in worst case.

Bottom-up approaches like CLINT start the learning process by constructing a starting clause that is subsequently generalized. As described in [3], CLINT computes the most specific clause \perp contained in the hypothesis language L_H and then generalizes \perp by the dropping condition rule until a given positive example $e^+ \in E^+$ is covered. The set of generalizations of \perp covering e^+ is called the *set of justifications* $Jus(K, L, e^+)$. The clauses in $Jus(K, L, e^+)$ consistent with the negative examples are used as starting clauses. The complexity of the subsequent generalisation depends on the length and the number of starting clauses for a given positive example. In worst case, when \perp covers e^+ and all body literals of \perp are consistent with the negative examples, \perp has to be chosen as starting clause. The most specific clause \perp depends on the hypothesis language. Since the length of \perp is crucial for the complexity of the hypothesis space, it is a useful measure for comparing the complexity of hypothesis spaces searched by CLINT.

CLINT offers five different language series [3], the basic form of which is

L is the set of clauses c such that
 $head(c) = p(x_1, \dots, x_k)$ and $body(C) \subseteq B(x_1, \dots, x_k)$, where
 $p(x_1, \dots, x_k)$ is the predicate to be learned,
 c is linked and range restricted,
all x_k are different.

The language series mainly differ in the definition of the set of body literals B . For each language, the most specific clause \perp includes all literals in B in its body. Thus, the complexity of \perp equals the number of literals in the set $B(x_1, \dots, x_k)$. In this paper, we investigate the influence of the parameters of the language series on the complexity of the respective sets $B(x_1, \dots, x_k)$. For convenience, the set of all predicates in the theory is denoted by $Pred$. \mathcal{A} is the maximum arity of all predicates in $Pred$.

Language Series 1. The set of body literals $B_i(x_1, \dots, x_k)$ for a language L_i of series 1 is defined as follows.

$$B_i(x_1, \dots, x_k) = \{ q(y_1, \dots, y_n) \mid q \in Pred \text{ and} \\
\begin{aligned}
(1) & \quad |\{y_1, \dots, y_n\} \setminus \{x_1, \dots, x_k\}| \leq i \text{ and} \\
(2) & \quad \{y_1, \dots, y_n\} \cap \{x_1, \dots, x_k\} \neq \emptyset \text{ and} \\
(3) & \quad \text{each variable in } \{y_1, \dots, y_n\} \cap \{x_1, \dots, x_k\} \text{ occurs} \\
& \quad \text{only once} \quad \}
\end{aligned}$$

Table 1 shows the number of literals in $B_i(x_1, \dots, x_k)$ as a function of parameter i . (For equations and combinatorial details see the full length version of this paper or [5].) As background knowledge we assume the following set of predicates $Pred_1 = \{nil/1, atom/1, sort/2, list/3, append/3, partition/4\}$. Since the maximum arity of the predicates in $Pred_1$ is $\mathcal{A} = 4$, the parameter i can be varied from 0 to 3.

	$i = 0$	$i = 1$	$i = 2$	$i = 3$
$B_i(X, Y)$	40	100	136	144

Table 1. The number of literals in \perp in Language Series 1 as a function of i .

As table 1 indicates, incrementing i results in only a small growth of $B_i(x_1, \dots, x_k)$ when i comes close to its maximum value $\mathcal{A} - 1$. Thus, it is not sensible to use this language series in a bias shift environment as described in [3] since the languages differ too little to justify the risk of failing because of the weakness of the hypothesis language.

Language Series 2. In the languages of series 2, the parameter q controls the maximum level of existential quantification of variables [3]. The sets $B_i^q(x_1, \dots, x_k)$ of body literals for a hypothesis language L_i^q of series 2 are defined as follows.

$$B_i^1(x_1, \dots, x_k) = B_i(x_1, \dots, x_k) \\
B_i^q(x_1, \dots, x_k) = B_i(z_1, \dots, z_l), \text{ where } \{z_1, \dots, z_l\} \text{ is the set of} \\
\text{all variables in } B_i^{q-1}(x_1, \dots, x_k).$$

Table 2 shows the number of literals of $B_i^q(x_1, \dots, x_k)$ for $i, q \in \{1, 2, 3\}$ which arise when learning a 2-ary predicate with the set of list predicates $Pred_1$.

$ B $	$q = 1$	$q = 2$	$q = 3$
$i = 1$	100	$1.623 \cdot 10^7$	$9.096 \cdot 10^{23}$
$i = 2$	136	$3.371 \cdot 10^8$	$9.8 \cdot 10^{27}$
$i = 3$	144	$6.472 \cdot 10^8$	$6.94 \cdot 10^{28}$

Table 2. The number of literals in $B_i^q(x_1, \dots, x_k)$ as a function of i and q when learning a 2-ary predicate using background knowledge $Pred$.

Our investigations of language series 2 show that parameter i has only little influence on $|B_i^q(x_1, \dots, x_k)|$, whereas the impact of parameter q is too strong. In fact, $|B_i^q(x_1, \dots, x_k)|$ grows super-exponentially in q [5]. So, we can conclude that the parameters i and q as used here are suitable neither for controlling the complexity of the hypothesis language nor for sensible bias shift operations.

Language Series 3. Language series 3 is a generalisation of language series 2. The set of body literals $B_{i_1, \dots, i_{n-1}, i_n}(x_1, \dots, x_k)$, $i_1, \dots, i_{n-1} \in N_0$, for a language $L_{i_1, \dots, i_{n-1}, i_n}$ of series 3 is defined as follows.

$$B_{i_1, \dots, i_n}(x_1, \dots, x_k) = B_{i_n}(z_1, \dots, z_l) \text{ where } \{z_1, \dots, z_l\} \text{ is the set of all variables in } B_{i_1, \dots, i_{n-1}}(x_1, \dots, x_k).$$

Table 3 shows some values of $B_{i_1, \dots, i_n}(x_1, \dots, x_k)$ for the most restrictive arrangement of languages which can be defined in series 3.

	L_0	$L_{0,1}$	$L_{0,1,0}$	$L_{0,1,0,1}$	$L_{0,1,0,1,0}$	$L_{0,1,0,1,0,1}$
$ B $	40	100	$1.526 \cdot 10^7$	$1.623 \cdot 10^7$	$9.095 \cdot 10^{23}$	$9.095 \cdot 10^{23}$

Table 3. The number of literals in the sets $B_{i_1, \dots, i_n}(X, Y)$ of language series 3.

These results show that there is only little difference between a language $L_{\dots,0}$ and the next complex language $L_{\dots,0,1}$, i.e., $L_0 = L_0^0$, $L_{0,1} = L_1^1 = L_1$, $L_{0,1,0} \approx L_{0,1,0,1} = L_1^2$, and so on. This means that language series 3 does not offer significant improvement compared to series 2.

An alternative language series. The literal sets $B(x_1, \dots, x_k)$ in language series 2 and 3 grow super-exponentially because the number of variables which are newly introduced when shifting the bias depends on the number of variables already available. A method to prevent this growth is to set a fixed limit on the number of variables which are generated on each level of existential quantification as in the following language definition.

$$B_i(x_1, \dots, x_k) = \{ q(y_1, \dots, y_l) \mid q \in Pred, \\ \{y_1, \dots, y_l\} \subseteq \{x_1, \dots, x_k\} \cup \{v_1, \dots, v_i\} \text{ and} \\ \{v_1, \dots, v_i\} \text{ is a set of } i \text{ different new variables,} \\ \text{all variables } y_i, y_j \in pred(y_1, \dots, y_l) \text{ are different,} \\ \{y_1, \dots, y_l\} \cap \{x_1, \dots, x_k\} \neq \emptyset \} \\ \cup \{ equal(v_m, v_n) \mid 1 \leq n \leq i, 1 \leq m \leq i, n \neq m \} \\ B_i^q(x_1, \dots, x_k) = B_i^{q-1}(x_1, \dots, x_k) \cup B_i(z_1, \dots, z_i), \text{ where} \\ \{z_1, \dots, z_l\} = vars(B_i^{q-1}(x_1, \dots, x_k)) \setminus vars(B_i^{q-2}(x_1, \dots, x_n)). \\ vars(B_i^0(x_1, \dots, x_n)) := \{x_1, \dots, x_n\}.$$

As in language series 2 of CLINT, the parameter q in this language defines the maximum level of existential quantification. However, the parameter i does not indicate the maximum number of new variables in each literal, but restricts the total number of variables on each level $1, \dots, q$.

Table 4 shows the number of literals in the sets $B_i^q(X, Y)$ as defined here for the set of list predicates $Pred_1$. In the language series proposed here, $|B_i^q(x_1, \dots, x_k)|$ grows linearly with parameter q .

$ B $	$q = 1$	$q = 2$	$q = 3$
$i = 2$	87	174	261
$i = 3$	249	870	1491
$i = 4$	556	2894	5232
$i = 5$	1056	7386	13716

Table 4. The number of literals in the sets $B_i^q(X, Y)$ of the new language series as a function of i and q .

Predicate Invention. An alternative approach to reduce the complexity of hypothesis languages without sacrificing expressiveness is to use *predicate invention* as bias shift operation. The idea is to produce a starting clause for the target predicate in the more restricted language L_i of series 1 that covers the positive examples but might be inconsistent. If the starting clause is inconsistent, a new auxiliary predicate is introduced that discriminates between positive and negative instantiations of the clause. Then, the process continues with the new predicate as target predicate. The main advantage is that the number k of old variables for the auxiliary predicate can be reduced using methods for finding a minimal set of arguments for a new predicate [4].

Conclusions. Our results on the language series of CLINT are largely negative. The super-exponential growth of the most special clause \perp for language series 2 and 3 makes learning intractable except for toy domains. Furthermore, the chosen parameters are unsuitable for sensible bias shift operations since languages discriminated by the number of existential variables scarcely differ whereas languages discriminated by the depth of existential variables differ too strongly. As a solution for these problems, we proposed a language which allows for tractable learning procedures, and discussed how predicate invention can be used to improve the shift of bias.

References

1. Adé, Hilde, Luc De Raedt und Maurice Bruynooghe. Declarative Bias for Bottom-up ILP Learning Systems, 1994. To appear in *Machine Learning*.
2. Muggleton, S., C. Feng. Efficient Induction of Logic Programs. *Proc. of the 1st Conference on Algorithmic Learning Theory*, Tokyo, OHMSHA, 1990.
3. L. De Raedt. *Interactive Theory Revision: an Inductive Logic Programming Approach*. Academic Press, 1992.
4. I. Stahl and I. Weber. The Arguments of Newly Invented Predicates in ILP. In *Proc. of ILP-94*, 1994.
5. I. Weber. *Komplexität von Hypothesenräumen in der Induktiven Logischen Programmierung*. Diplomarbeit Nr. 1164, Universität Stuttgart, Fakultät Informatik, 1994.