

The Effect of Predicate Order on Curriculum Learning in ILP*

Hank Conn and Stephen H. Muggleton

Imperial College London

Abstract

Development of effective methods for learning large programs is arguably one of the hardest unsolved problems within ILP. The most obvious approach involves learning a sequence of predicate definitions incrementally. This approach is known as Curriculum Learning. However, Quinlan and Cameron-Jones' paper from 1993 indicates difficulties in this approach since the predictive accuracy of ILP systems, such as FOIL, rapidly degrades given a growing set of learned background predicates, even when a reasonable ordering over the predicate sequence is chosen. Limited progress was made in this problem until the recent advent of bias-reformulation methods within Meta-Interpretive Learning. In this paper we show empirically that given a well-ordered predicate sequence, relatively large sets of dyadic predicates can be learned incrementally using a state-of-the-art Meta-Interpretive Learning system which employs a Universal Set of metarules. However, further experiments show how progressive random permutations of the sequence rapidly degrades performance in a fashion comparable to Quinlan and Cameron-Jones's results. On the basis of these results we propose the need for further identification of methods for identifying well-ordered predicate sequences to address this issue.

1 Introduction

In their seminal paper Quinlan and Cameron-Jones [14] demonstrated the difficulties in using FOIL [12] to learn a sequence of inter-related predicate definitions. This approach is now referred to as Curriculum Learning [1]. The experiment involved eighteen list-processing definitions learned incrementally in the sequence order presented in Chapter 3 of Bratko's textbook for learning Prolog [2]. Although predicates in the first half of the sequence were learned accurately and efficiently from extensive tabulated sets of examples, predictive accuracy degraded substantially in the second half of the sequence, leading to FOIL timing-out with error-prone definitions. While some of this behaviour was due to difficulties in effective learning of mutually recursive definitions, the dominant feature was that of progressive expansion of the search space due to the growing vocabulary of learned predicate definitions being added to the background knowledge.

By contrast, more recent work in Meta-Interpretive Learning (MIL) [10] has shown that the use of bias-reformulation methods, such as *dependent learning* [6] allows accurate and efficient learning of extended sequences of string-transformation predicates from small numbers of examples. A key feature of dependent

*The second author acknowledges support from his Royal Academy of Engineering/Syngenta Research Chair at the Department of Computing at Imperial College London.

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: Nicolas Lachiche, Christel Vrain (eds.): Late Breaking Papers of ILP 2017, Orléans, France, September 4-6, 2017, published at <http://ceur-ws.org>

learning is its ability to automatically identify a good sequence ordering for the learning. A second aspect is that MIL, unlike FOIL, is guaranteed to find minimal predicate definitions, avoiding problems with overfitting in the presence of large amounts of background knowledge. Since the minimal representation of the target theory either stays the same or shrinks monotonically with expanding background predicates, this can lead to reductions in search in the case that hypotheses are considered in increasing order of their size.

In this paper we explore the effect that predicate sequence choice has on learning performance. In particular, we show that a) with a set of inter-related family relations, MIL produces efficient and effective learning in the case of a well chosen predicate sequence and b) performance degrades gradually with progressive random permutations of such an ordering. These results re-enforce the need for techniques, such as dependent learning, for addressing the problems of learning large logic programs.

This paper is organised as follows. Section 2 describes related work. The Meta-Interpretive Learning framework is described in Section 3. In Section 4 we describe the implementation of Metagol and the algorithm for running the experiments. The experiments are described in Section 5. We conclude and describe further work in Section 6.

2 Related work

Induction of large programs from data is one of the long term aims of ILP [11]. However, even when learning individual predicate definitions the complexity of admissible search grows exponentially [8]. Although Quinlan’s FOIL [12] provides efficient heuristic search, the lack of admissible search leads to problems with incompleteness associated with zero-gain literals [13] as well as hard issues relating to mutual recursion in multi-predicate learning [14, 5]. One initially promising avenue to avoid Quinlan and Cameron-Jones’ problem with increasing background knowledge was presented by Srinivasan et al [15] who showed that using admissible search Progol [8] can simultaneously increase accuracy while decreasing search time. The reason is that increasing background knowledge reduces the minimal size of a consistent solution, allowing Progol to consequently reduce both the search size and the degree of overfitting for single clause solutions as relevant background knowledge is provided incrementally.

Recent advances in the area of Meta-Interpretive Learning (MIL) [10] have demonstrated a way in which higher-order background knowledge in the form of metarules and abstractions [4] can further constrain admissible hypothesis space search, leading to decreases in search time and increases in predictive accuracy. While Progol guarantees minimal solutions for single clause searches, Metagol [10] achieves minimal and admissible multi-clause predicate definition searches by iterative deepening. However, MIL learns definitions incrementally, which opens the question of effects related to the order in which predicates are learned. Results in our experiments indicate that if the idealised ordering used in experiments is randomly permuted, predictive accuracy degrades rapidly. It is therefore necessary to consider how the order of predicate definition learning should be selected automatically. Initial results in [6] indicate that a technique referred to as *dependent learning* can be effective in selecting such an ordering, though it has still to be clarified what the properties of a target theory are for dependent learning to have guaranteed effectiveness.

3 Framework

MIL [9, 10] is a form of ILP based on an adapted Prolog meta-interpreter. Whereas a standard Prolog meta-interpreter attempts to prove a goal by repeatedly fetching first-order clauses whose heads unify with a given goal, a MIL learner attempts to prove a set of goals by repeatedly fetching higher-order metarules (Fig. 1b) whose heads unify with a given goal. The resulting meta-substitutions are saved in an abduction store, and can be re-used in later proofs. Following the proof of a set of goals, a hypothesis is formed by applying the meta-substitutions onto their corresponding metarules, allowing for a form of ILP which supports predicate invention and the learning of recursive theories.

4 Implementation

Figure 1a shows Metagol_{DF}[6], an implementation of the MIL framework, similar in form to a standard Prolog meta-interpreter. A universal set of metarules (see [3]) (Fig. 1b) is defined separately.

```

prove([], Prog, Prog).
prove([Atom|As], Prog1, Prog2) :-
  metarule(Name, MetaSub, (Atom :- Body), Order),
  Order,
  abduce(metasub(Name, MetaSub), Prog1, Prog3),
  prove(Body, Prog3, Prog4),
  prove(As, Prog4, Prog2).

```

(a) Prolog code for generalised meta-interpreter

Name	Metarule	Order
Inverse	$P(x, y) \leftarrow Q(y, x)$	$P \succ Q$
Chain	$P(x, y) \leftarrow Q(x, z), R(z, y)$	$P \succ Q, P \succ R$
TailRec	$P(x, y) \leftarrow Q(x, z), P(z, y)$	$P \succ Q, x \succ z \succ y$

(b) Metarules with associated ordering constraints, where \succ is a pre-defined ordering over symbols in the signature. The letters P , Q , and R denote existentially quantified higher-order variables. x , y , and z denote universally quantified first-order variables.

Figure 1: Metagol_{DF} meta-interpreter (a) and universal metarules (b)

```

predicate_list = list of predicates in standard order
for each permutation:
  pick two predicates in predicate_list and swap their position
for each predicate in predicate_list:
  generate prolog code
  execute metagol learning task
  check learned definition for accuracy
  save learned definition as background knowledge

```

Figure 2: Trial procedure for experiments

4.1 Metagol Interface

A user interface¹ was created for manually entering data and configurations needed to run a learning task with Metagol. The user can enter predicates and constants, fill in background knowledge, select metarules, enter a predicate to learn, and enter the training set of positive and negative examples. Once the run is assembled, the user can generate the Prolog code for running the task in Metagol, execute the code on the server, and view the results. Finally, the definitions learned in the task can be saved in the database for the given user, and included as background knowledge in subsequent learning tasks. This allows the user to progressively learn large logic programs.

4.2 Algorithm of implementation used for experiments

Figure 2 shows the procedure used for each trial of the experiment. A Metagol learning task is executed for each predicate, and the learned definition is used as additional background knowledge for each successive task.

5 Experiments

Hypotheses

Two hypotheses were tested: (1) *learning a series of predicate definitions using the universal set of metarules does not decrease performance in Metagol* and (2) *learning predicates in a randomized order would lead to lower accuracy*.

Materials

Learning was based on family relationships in Hindi [7]. In Hindi there are a number of terms without specific words in English. For example in Hindi there are different terms for your father’s brother (taao) than for your mother’s brother (maama). Hindi also has specific terms for complex concepts such as the daughter of your mother’s brother (mameri). Family trees of 5000 individuals were randomly generated. The trees were written to a Prolog file containing all of the facts for each individual in terms of the background predicates male/1, female/1, father/2, and mother/2. This file could then be used as background knowledge for learning the definitions of family relationships. In total 43 family relationship concepts were assembled², along with their definitions in Prolog to be learned progressively. The concepts were placed into a reasonable order for learning, where for example the simpler concepts *brother* and *daughter* are learned before the more complex concept *mother’s brother’s daughter*.

¹Available at https://github.com/metagol/metagol_web_interface

²All experimental code and materials available at <https://github.com/metagol/ILP2017>

Methods

In the first experiment the training set was 1% and test set 10% of the total number of positive and negative examples for each predicate, randomly selected with replacement. Predicate accuracies and running times were averaged over 50 trials. In the second experiment the predicates were learned in a randomized order. Predictive accuracies were averaged over 50 trials for each increment of number of swaps. Positive and negative examples were randomly sampled in equal number for each learning task (averaging 56 training examples and 226 test examples) in order to give a default predicate accuracy of 50% for a majority class predictor. The experiments were run on a Windows 7 operating system running YAP 6.2.2, the latest Metagol code from github³ (as of 2017-02-12), and with a test harness running on Java 8 (jre1.8.0.121). The MySQL instance shared by the web interface and test harness was on version 10.1.19-MariaDB.

Results

Figure 3a shows that in experiment 1 the mean accuracy of the learned definitions stayed near 100%, and Figure 3b shows learning times remained consistently below 1s (with some outliers) throughout the series, confirming the first hypothesis.

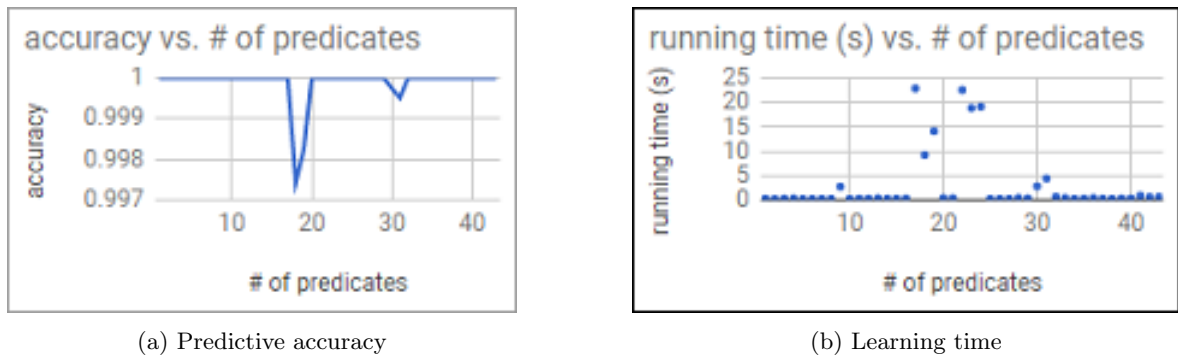


Figure 3: Learning results for experiment 1

Figure 4 shows the mean accuracy gradually decreased as increasing numbers of predicates were swapped out of order, confirming the second hypothesis.

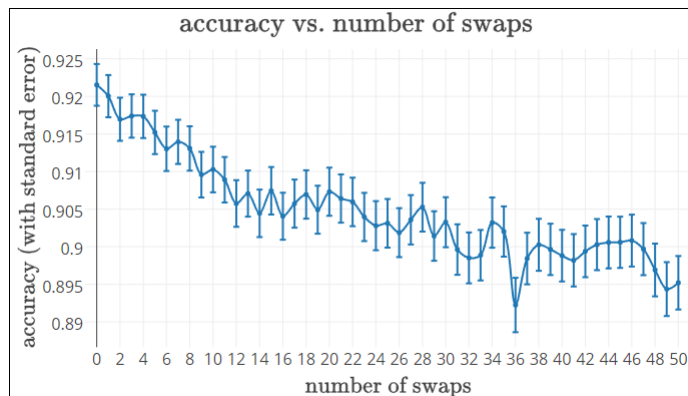


Figure 4: Learning results for experiment 2

6 Conclusions and further work

This paper revisits issues related to Quinlan Cameron-Jones' demonstration that the performance of systems such as FOIL progressively degrades with increasing numbers of predicates. We show that given a reasonable ordering, such as that provided to FOIL, Metagol's performance does not degrade while progressively learning

³Available at <https://github.com/metagol/metagol>

43 Hindi family relationships. However, when the reasonable ordering over predicates learned is randomly perturbed, predictive accuracy also progressively degrades.

In further work we hope to investigate the degree to which *dependent learning* guarantees the discovery of a “reasonable order” for learning a large set of predicate definitions.

References

- [1] Y. Bengio, J. Louradour, R. Collobert, R., and J. Weston. Curriculum learning, 2009.
- [2] I. Bratko. *Prolog Programming for Artificial Intelligence*. Addison-Wesley, London, 1986.
- [3] A. Cropper and S.H. Muggleton. Logical minimisation of meta-rules within meta-interpretive learning. In *Proceedings of the 24th International Conference on Inductive Logic Programming*, pages 65–78. Springer-Verlag, 2015. LNAI 9046.
- [4] A. Cropper and S.H. Muggleton. Learning higher-order logic programs through abstraction and invention. In *Proceedings of the 25th International Joint Conference Artificial Intelligence (IJCAI 2016)*, pages 1418–1424. IJCAI, 2016.
- [5] L. De Raedt and N. Lavrac. Multiple predicate learning in two inductive logic programming settings. *Journal on Pure and Applied Logic*, 4(2):227–254, 1996.
- [6] D. Lin, E. Dechter, K. Ellis, J.B. Tenenbaum, and S.H. Muggleton. Bias reformulation for one-shot function induction. In *Proceedings of the 23rd European Conference on Artificial Intelligence (ECAI 2014)*, pages 525–530, Amsterdam, 2014. IOS Press.
- [7] R.S. McGregor. *The Oxford Hindi-English Dictionary*. Oxford University Press, 1993.
- [8] S.H. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.
- [9] S.H. Muggleton, D. Lin, N. Pahlavi, and A. Tamaddoni-Nezhad. Meta-interpretive learning: application to grammatical inference. *Machine Learning*, 94:25–49, 2014.
- [10] S.H. Muggleton, D. Lin, and A. Tamaddoni-Nezhad. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. *Machine Learning*, 100(1):49–73, 2015.
- [11] S.H. Muggleton, L. De Raedt, D. Poole, I. Bratko, P. Flach, and K. Inoue. ILP turns 20: biography and future challenges. *Machine Learning*, 86(1):3–23, 2011.
- [12] J.R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.
- [13] J.R. Quinlan. Determinate literals in inductive logic programming. In *IJCAI-91: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 746–750, San Mateo, CA., 1991. Morgan-Kaufmann.
- [14] 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 of *Lecture Notes in Artificial Intelligence*, pages 3–20. Springer-Verlag, 1993.
- [15] A. Srinivasan, S.H. Muggleton, and R.D. King. Comparing the use of background knowledge by inductive logic programming systems. In L. De Raedt, editor, *Proceedings of the Fifth International Inductive Logic Programming Workshop*. Katholieke Universiteit Leuven, 1995.