Evolutionary Learning of Existential Rules

Lianlong Wu
Department of Computer Science, University of Oxford, United Kingdom
lianlong.wu@cs.ox.ac.uk

Abstract

Declarative rules such as Prolog and Datalog are common formalisms to express expert knowledge and are used in a number of systems. Since developing such rules is time consuming and requires scarce expert knowledge, it is essential to develop algorithms for *learning* such rules. We address the problem of learning *existential* rules, a richer class of rules which found applications in many use-cases such as Semantic Web and Web Data Extraction. In particular, we concentrate on developing *evolutionary* learning algorithms for rule learning.

1 Introduction

Datalog rules are a common formalism for expressing expert knowledge. For instance, the rule

grandpa(X, Y) :- parent(X, Z), parent(Z, Y), male(X) (1)

expresses the knowledge that if X is a parent of Z who in turn is a parent of Y and, in addition, X is a male, then X is a grandfather of Y. Such rules have found a number of applications in expert and legal systems, medical diagnosing and finance. Most recently, such declarative rules have been used in expressing knowledge and inferring new information in knowledge graph management systems. However, it is known that Datalog rules are not able to express and reason over unknown and incomplete information. For that reason the formalism of Datalog+/- [Cal *et al.*, 2010] has been introduced which is able to express such information. For instance, the rule

managerOf(X, Y) :- employee(Y) (2)

where the variable X is existentially quantified, expresses the fact that each employee must have a manager even though it is unknown who that manager is. Introducing existential quantifiers comes at a cost, however, since reasoning w.r.t. Datalog+/- rules becomes undecidable, unlike in Datalog. Hence, there have been a number of sublanguages of Datalog+/- that have been proposed in which reasoning is not only decidable but also tractable. Warded Datalog+/- [Bellomarini *et al.*, 2018] is one such sublanguage that subsumes Datalog, and allows existential quantifiers, and moreover, reasoning in it is in polynomial time. Furthermore, this language becomes the core of the Vadalog system which enables reasoning over large amount of data. Existential rules have been used in a number of use-cases such as ontological reasoning in Semantic Web, Web Data Extraction (where existentials are used for ID creation) and Recommender Systems (where existentials model liked items of new users).

The vast majority of development time of the rule-based solutions is spent on writing the rules, as they can be quite involved and demand scarce expert knowledge. This hampers rule-based systems and solutions from being widely adopted. However, availability of data and *learning* rules from this data can significantly reduce the development time. Additional advantage of the process of learning rules is that the resulting rules are *interpretable*, as opposed to many black-box machine learning models [Dai and Zhou, 2019].

During my doctoral research I intend to study the problem of learning *existential* rules, in particular Warded Datalog+/- rules, as well as how such rules can help in learning usual (non-existential) rules.

Inductive Logic Programming (ILP) [Nienhuys-Cheng and de Wolf, 1997] is the field that has been studying the problem of learning (non-existential) rules. We can define ILP task as follows: given background knowledge, a set of positive and negative examples expressed as a set of facts, learn a set of rules that entail all the positive and none of the negatives.

For example, assuming we have the following facts constituting the background knowledge: *parent("Bob", "Sharon"). parent("Jane", "Bob"). parent("Victor", "Bob"). male("Victor").* and one positive example *grandpa("Victor", "Sharon").* An ILP system can learn the rule defined in (1).

Classical approaches to ILP perform search over the space of all syntactically correct programs (sets of rules). *Exhaustive search* such as depth-first search is guaranteed to find the high quality rules, but suffers scalability problems as the search space increases. *Heuristic search* strategies like hill-climbing or greedy algorithms are efficient but can easily be trapped in local optima, thus possibly missing the optimal solution [Fürnkranz *et al.*, 2012].

In my thesis I intend to study *evolutionary algorithms* for ILP. These are a family of biology-inspired search algorithms that optimize for most promising solutions, while exploring a wide search space at the same time. In particular, in our setting, atoms, partial or parameterized rules can be treated as chromosomes, from which new population of chromosomes

can be derived via the operations of *mutation*, *crossover* and *selection*. While performing these operations, a quality measure, called *fitness function* is designed to judge whether an obtained new generation of chromosomes is fit for continuing the search. Such evolutionary algorithms typically do not perform exhaustive search and at the same time are less likely to fall into local optima. In addition, they are flexible as they might not require imposed template on shape of rules, as it is typically the case in other approaches to ILP.

2 Research Proposal

2.1 Evolutionary Algorithm for Existential Rules

We start our research by studying a family of evolutionary algorithms for learning *existential* rules. There are a number of parameters that give rise to multiple variations of such algorithms. In particular, they are (i) initialization of initial population, (ii) the operations of mutation, crossover and selection, and (iii) fitness function. In all the aspects (i)-(iii) of evolutionary algorithms, presence of existential quantifiers in the rules gives rise to new challenges. Concerning (i), presence of existentials leads to a much larger set of possible initial populations which leads to the following research question.

RQ1 What is the best way to select the initial population of existential (partial) rules to lead to faster convergence?

Similarly, the new operations of mutation, crossover and selection have to be defined to capture meaningful transformations of existential rules.

RQ2 What are appropriate and meaningful operations of mutation, crossover and selection and what is their impact on the overall performance of the algorithm?

Next, typical fitness functions are based on the various performance measures such as precision or f-score, i.e., that measure how well the inferred facts from the intermediate learned rules match the training examples. However, existential rules can result in inferred facts that contain *marked nulls*, i.e., placeholders for unknown values generated by applying the rules. An example of such fact is *managerOf(z, "Victor")*, where z is a marked null, obtained by applying the rule (2) to the fact *employee("Victor")*. This leads to the question of defining appropriate fitness functions that measure how well such facts match training examples that do not contain marked nulls.

RQ3 How to define meaningful fitness functions and what is their impact on the overall learning performance?

We intend to find inspiration from the database theory literature (such as the notion of certain answer semantics) where the question of query answering in presence of marked nulls was extensively studied.

2.2 Improving Learning of Non-Existential Rules

We plan to study whether existential rules learning can improve learning of Prolog or Datalog (non-existential) rules. In particular, typically during a run of an evolutionary algorithm for learning non-existential rules we may generate a rule of the form *grandpa(X,Y) :- male(X)* which is considered as an invalid Prolog or Datalog rule and thus rejected. But in fact this rule is existential and can be considered as a valid rule by the algorithms resulted from RQ1-RQ3. Adding ability of learning algorithms for non-existential rules also accept existential rules as intermediate chromosomes can lead to faster convergence. For instance, the example rule above can lead to the desired rule (1) faster as it can start exploring most promising chromosomes (with existentials) earlier.

RQ4 Can evolutionary algorithms for non-existential rules be improved with allowing existential rules as intermediate chromosomes?

2.3 Parameterized Rules

Further research questions include extending the developed algorithms to be able to deal with parameterized rules. This can be useful in the scenarios when a structure of a desired rule is known, but the needed names for predicates, variables or constants are unknown. For instance, we might require to learn the rule *grandpa(X, Y) :- ⋆(X, Z), ⋆(Z, Y), male(X)* where only the predicate name for $\star$ is unknown and can match either an existing or a new predicate name.

RQ5 Can the evolutionary algorithms be extended to learn parameters for given parameterized rules?

2.4 Additional Rule Features

Other language constructs besides the existentials are being used in real industry use-cases, such as negation and aggregation. We intend to study the following research question:

RQ6 Can the evolutionary algorithms be extended to learn rules with negation or aggregation?

Acknowledgments

This work is supported by EPSRC iCASE BP Russell Studentship, EPSRC grant EP/M025268/1, the WWTF grant VRG18-013, and the EU grant 809965. Thanks to Georg Gottlob, Emanuel Sallinger, Yavor Nenov and Evgeny Sherkhonov for their valuable advice.

References

[Bellomarini *et al.*, 2018] Luigi Bellomarini, Emanuel Sallinger, and Georg Gottlob. The Vadalog System : Datalog-based Reasoning for Knowledge Graphs. In *VLDB*, 2018.

[Cal *et al.*, 2010] Andrea Cal, Georg Gottlob, Thomas Lukasiewicz, Bruno Marnette, and Andreas Pieris. Datalog + / - : A Family of Logical Knowledge Representation and Query Languages for New Applications. In *LICS*, 2010.

[Dai and Zhou, 2019] Wang Zhou Dai and Zhi Hua Zhou. A Survey on Inductive Logic Programming. *Journal of computer Research and Development*, 56(1):138–154, 2019.

[Fürnkranz *et al.*, 2012] Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of Rule Learning*. Springer-Verlag Berlin Heidelberg, 2012.

[Nienhuys-Cheng and de Wolf, 1997] Shan-Hwei Nienhuys-Cheng and Ronald de Wolf. *Foundations of inductive logic programming*. Springer, 1997.