ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220141214

Inductive Logic Programming (ILP) and Reasoning by Analogy in Context of Embodied Robot Learning

Article *in* International Journal of Agent Technologies and Systems · April 2010

DOI: 10.4018/jats.2010040105 · Source: DBLP

CITATIONS
5

READS
211

3 authors:

Vesna Poprcova
2 PUBLICATIONS 6 CITATIONS
SEE PROFILE

Georgi Stojanov
American University of Paris
68 PUBLICATIONS 294 CITATIONS
SEE PROFILE

Andrea Kulakov
Ss. Cyril and Methodius University in Skopje Macedonia
87 PUBLICATIONS 819 CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project visualisation of abstract concepts View project

Project SOCRATES: Social Cognitive Robotics in The European Society View project

All content following this page was uploaded by Georgi Stojanov on 05 June 2014.

The user has requested enhancement of the downloaded file.

Inductive Logic Programming (ILP) and Reasoning by Analogy in Context of Embodied Robot Learning

Vesna Poprcova[1], Andrea Kulakov[1], Georgi Stojanov[2],

[1] Ss. Cyril and Methodius University of Skopje, Karpos 2, 1000 Skopje, Macedonia
[2] American University of Paris, 147 rue de Grenelle, 75007 Paris, France

VPoprcova@gmail.com, Kulak@feit.ukim.edu.mk, Georgi.Stojanov@aup.fr

Abstract. The ability of reasoning by analogy seems to be essential for many cognitive processes from low and high level perception to categorization. Intuitively, the idea is to use old knowledge in order to explain new observations *similar* in some ways to what is already known. In some sense it is opposite of induction where in order to explain the observations one comes up with new hypotheses/theories. Therefore, a system capable of both ways of reasoning would be superior to either. In this paper we first present an overview of Inductive Logic Programming (ILP) systems that use reasoning by analogy. Then we present the results of applying Analogical Prediction to problem that arise in the context of physically embodied robot which tries to learn regularities in its environment.

Keywords: Inductive Logic Programming, Reasoning by Analogy, Learning in Embodied Systems

1 Introduction

In this paper, we report part of our results that resulted from our research conducted within the research project XPERO: Learning by experimentation (www.xpero.org). Our main goal was to combine reasoning by analogy and Inductive Logic Programming (ILP) as ILP was the main learning mechanism adopted by the XPERO research team.

There is a considerable body of research in analogical reasoning in cognitive sciences (Gentner [4], Hofstadter [5], Kokinov [8], Indurkhya [6]) and Artificial Intelligence (Evans [3], Becker [1], McDermott [12]). Thus, from perception to problem solving, recollection, explanation and case based reasoning, as well as many other cognitive abilities rely (in one way or another) on analogical reasoning. Analogical reasoning is an important research area in AI as a technique to reason from incomplete knowledge.

In the Interim Report of the project XEPRO Deliverable 5.2.1 (2008) we have given an extensive overview of the research on analogy in Artificial Intelligence (AI). We have also suggested a classification of the computational models of reasoning by analogy with respect to the underlying approach to the problem of knowledge representation:

- “symbolic” models, so called because they are largely part of the "symbolic" paradigm in AI, in which symbols, logic, planning, search, means-ends analysis, etc. play a predominant role
- “connectionist” models that adopt, broadly speaking, the framework of the connectionist networks, including nodes, weights, spreading activation, etc. and
- “hybrid” models that lie somewhere in between connectionist and symbolic models, based on the idea that high-level cognition emerges as a result of the continual interaction of relatively simple, low-level processing units, capable of doing only local computations.

Within this research project symbolic paradigm was chosen for the knowledge representation, a fact which guided the work presented in this paper.

Very few attempts try to combine reasoning by analogy and ILP. ILP, as defined in [16], is the intersection of inductive learning and logic programming (LP) which infers new knowledge by inducing hypotheses given some background knowledge and a set of positive and negative examples. The reason for scarcity of research that combines analogy and ILP may be in the fact that ILP techniques have rather well elaborated and formalized theoretical background, whereas this is not the case for analogy based systems. .The remaining part of the paper consists of two main parts a) a review of what we believe to be the most representative research combining reasoning by analogy and ILP and b) results of our own attempt to combine analogy and ILP which relies on IPA, CProgol and HYPER. After reviewing most of the systems we justify our choice of three of them: Inductive Prediction by Analogy (IPA) [2]; ANalogical GEneraLization (ANGEL) [9] and CProgol [15]. We believe that these selected systems are fairly representative with respect to the advantages, limitations, as well as the open issues.

The main idea of our approach in combining analogy and ILP is to generate a hypothesis from the existing knowledge by analogy abd then to give it to HYPER which would verify it against available positive examples as well as eventually give a hint to design new experiments which would generate examples that would confirm or discard the hypothesis generated by analogy.

In the conclusions we summarize our findings and contemplate on future research.

Given the nature of the XPERO project, we present the applicability of ILP system CProgol in a robot discovery task. The scientific goal of XPERO is to make the robot gaining new insights or improving the robot’s theory and current “understanding” about the world. The reason of our interest in analogy is the fact that reasoning by analogy would enable the artificial agent to learn new things faster (given their similarities to existing knowledge), and without the need to run the learning algorithm from scratch.

2 Inductive Prediction by Analogy (IPA)

Inductive Prediction by Analogy (IPA) as proposed by [7] uses analogical reasoning in generating hypotheses using taxonomic information represented by first-order predicate logic.

To solve the problem or to find a hypothetical concept-description of the target concept such that the hypothesis covers all positive examples and no negative examples, IPA utilizes the process of analogical reasoning consisting of four main steps:

1) recognition of a candidate analogous source,
2) elaboration of an analogical mapping between source and target domains,
3) evaluation of mapping and inferences to given examples of the target predicate, and
4) consolidation of the outcome of the analogy.

IPA learns the description of a target predicate similar to a source predicate from a few examples of a target predicate. Similarity here is essentially defined via the taxonomic information (unlike, say, the structural similarity used by the Structure Mapping Engine).

The usefulness of the technique has been validated by real world problems like the function prediction of proteins in molecular biology. IPA technique is applicable to various domains with structurally related predicates.

3 ANalogical GEneraLization (ANGEL)

ANGEL (ANalogical GEneraLization) originally presented in [17] is related to ILP and belongs to the category of learning from examples, in sense that it generates new rules by generalizing given examples. It can also be regarded as a kind of method for learning by analogy.

ANGEL method is capable of generating a new rule which specifies a given target concept from a single example and existing rules. It consists of the following steps:

1) extending a given example,
2) extracting atoms from the example and selecting a base rule out if the set of existing rules,
3) generalizing the extracted atoms by means of the selected rule as a guide,
4) replacing predicates, and
5) generating a rule.

A given example is generalized through mapping a structure of a rule in existing knowledge base.

To evaluate similarity between atoms, ANGEL has to compute deductive closures of each of the atoms, i.e. to create the propositions that are generated as all conclusions of the given set of atoms. The approach evaluating similarities between atoms based on their deductive closures is theoretically interesting but is not practical. For the purpose of practical learning some restrictions on either forms of the background knowledge or the hypothesis language are required.

4 CProgol

CProgol is a state-of-the-art ILP system [15].

When constructing hypothesis clauses consistent with the examples, CProgol conducts a general-to-specific search in the theta-subsumption lattice of a single clause hypothesis. To search the lattice CProgol applies an A*-like algorithm to find the clause with maximum compression. CProgol can learn ranges and functions with numeric data (integer and floating point). The hypothesis language of CProgol is restricted by the means of mode declarations provided by the user. The mode declarations specify the atoms to be used as head literals or body literals in hypothesis clauses. For each atom, the mode declaration indicates the argument types, and whether an argument is to be instantiated with an input variable, an output variable, or a constant. Furthermore, the mode declaration bounds the number of alternative solutions for instantiating the atom. This system uses an approach called mode direction inverse entailment (MDIE) which is based upon model-theory rather than resolution proof-theory. In this way a great deal of clarity and simplicity can be achieved and it is hoped that it is easier to develop completeness and consistency results.

Cprogol4.4 is a version of the Progol family of ILP systems. The version presented in [14] incorporates an implementation of Analogical Prediction (AP) and learning from positive-only data [13]. It means that user is allowed to avoid incorporating negative examples, which are often unnatural to define, and also often unavailable in real-world domains. This is particularly suitable to the nature of problems tackled by XPERO.

AP is implemented as a built-in predicate *aleave*. The notion of AP can be viewed as a midpoint between induction and instance-based learning. We can view analogical reasoning as a special case of AP, in which the example set contains a single base example and the test instance relates to the target. The following analogy issues are handled in the CProgol4.4 AP implementation:

1) a set of base cases is used from the example set based on maximizing compression over the hypothesis space,
2) relevant properties are found by constructing the bottom clause relative to the test instance, and
3) relevant projection properties are decided on the basis of mode declarations.

CProgol has been applied across a wide range of problems. The problems to which CProgol has been applied include program synthesis, knowledge discovery for biological and chemical domains, natural language grammar learning, construction of fault diagnosis models and simple mathematical discovery problems. It has not been applied to the domain of open learning in physically embodied agents.

5 Experimental results: Robot experimental setup

We analyze the applicability of above mentioned ILP systems in robot discovery task in the XPERO project.

The scientific goal of XPERO is to investigate mechanisms of autonomous discovery through experiments in an agent's environment. In XPERO, the experimental domain is the robot's physical world, and the subject of discovery are various, quantitative or qualitative laws in this world. The scientific goals of XPERO are considerably different from the goals of a typical robotics project. The purpose is gaining new insights about the robot's world and the objects therein and to develop and improve its own cognitive skills and overall performance.

Elsewhere, we have presented the results of applying ILP to solving some of these problems. Within a simplified environment some success has been achieved with the ILP system Hyper [2]. In order to enable learning of more complex concepts we improved the learning algorithm as reported in [10] to make it scale better with the size of the learning task. As it is expected that our robot will be able to learn autonomously, the negative examples are not given. Hyper also incorporates the ability to automatically generate negative examples as described in [11].

We present the result obtained with CProgol to problems that arise in the context of physically embodied robot which tries to learn regularities in its environment.

We used the robot traces generated according to the so-called XPERO *movability* scenario presented in [18]. Object *movability* is one of the simplest notions in robot's interaction with the world. The experimental setup consisted of a mobile robot placed inside room with several objects. The robot is able to move some of the objects but some of the objects could not be moved. In two dimensional space robot collected data about predicates: at/3 which means that object Obj states at position P at time T; and move/4 which means that object Obj states at position P2, from position P1 by distance D. The other background predicates are: add/3, different/2, approxEqual/2. The position is a predicate of two values of type float. To induce the theory we used the capability of CProgol to learn from positive-only data which is natural for a real-world domain like this. The induced theory is presented by the following clause:

```
move(Obj,Pos1,Dist,Pos2):- different(Pos1,Pos2)
```

which means that that an object is movable if it has been observed at two different positions in space.

If we compare the results we can say that the induced theory by Hyper is more descriptive in sense that the new concept is invented never mentioned in the data or problem definition:

```
p(Obj):-at(Obj, T1, Pos1), at(Obj, T2, Pos2), different(Pos1, Pos2).

move(Obj, Start, Dist, End):-approxEqual(Start, End), not p(Obj).

move(Obj, Start, Dist, End):- add(Start, Dist, End), p(Obj).
```

We tried to discover this notion using CProgol AP built-in predicate *aleave*. It constructs hypotheses with given: test instance, training set and background knowledge. The constructed hypotheses not only cover some of the training set but also predict a class of the test instance. This can be viewed as a contrast with normal ILP where hypotheses are constructed on the basis of background knowledge and training set alone. The result obtained with this approach was the same except the learning was notably slower. AP performs worst than CProgol in inductive mode considering time. It might be the fact that AP has a tendency to overgeneralise or the advantages for domains in which a large proportion of the examples must be treated as exceptions with respect to the hypothesis vocabulary. Again, negative examples were not part of the learning.

Once we get the hypothesis generated by analogy, we submit it to the HYPER verify predicate to see how many of the examples are and we get an output telling us the percentage of coverage, which can be interpreted as the confidence of the new hypothesis. This is also an original way to measure the goodness or appropriateness of a hypothesis generated by analogy, a very difficult problem in itself.

6 Conclusions

Applying ILP in robotics is characterized with the main difficulties that a) robots are supposed to function in a real world, in a non structured environment; b) their sensors and actuators usually provide high dimensional data that are inherently noisy; and c) they face the problem of scalability of ILP to handle large amounts of data [9]. However, within a simplified environment we report some success. The results presented so far are obtained in real time even with the increased length of the traces. CProgol allows learning from positive-only data which means that user is allowed to avoid incorporating negative examples. This is important because negative examples are not part of the real-world domains and the process of generating negative examples is often unnatural. We apply Analogical Prediction to *movability* scenario and it performs consistently worse than CProgol4.4 in inductive mode.

We are not sure if more complex theories could be learnt with CProgol in inductive mode. We are in the process of testing more complex concepts useful in robotics.

Acknowledgments. The work described in this paper has been partially funded by the European Commission's Sixth Framework Programme under contract no. 029427 as part of the Specific Targeted Research Project XPERO ("Robotic Learning by Experimentation").

References

1. Becker, L. C.: Analogy in Legal Reasoning. Ethics 83 (3):248-255 (1973)

2. Bratko, I.: Prolog Programming for Artificial Intelligence, Addison-Wesley publishers limited (1986)
3. Evans, T.: A program for the solution of a class of geometric analogy intelligence test questions, in M. Minsky (ed.), Semantic information processing, Cambridge MA, MIT Press. (1968)
4. Gentner, D.: Structure-mapping: A Theory and Computer Model of Analogy-Making, Cognitive Science, Vol. 7, No. 2: 155-170 (1983)
5. Hofstadter, D.: Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thoughts, Basic Books, New York. (1995)
6. Indurkhya, B., Metaphor and Cognition, An Interactionist Approach, Kluwer Academic Publishers. (1992)
7. Ishikawa, T., Terano, T.: Inductive Prediction by Analogy Using Taxonomic Information. In: Third International Conference on Multistrategy Learning (1996), AAAI (www.aaai.org)
8. Kokinov, B., A Hybrid Model of Reasoning by Analogy, in Holyoak, K., and Barnden, J., (eds.) “Advances in Connectionist and Neural Computation Theory”, Vol. 2: Analogical Connections: 247-318, Ablex, Norwood, NJ. (1994)
9. Kulakov, A., Laukkanen, J., Mustafa, B., Stojanov, G.: Inductive Logic Programming and Embodied Agents: Possibilities and Limitations, International J. of Agent Technologies and Systems, Vol 1: 34-49 (2009)
10. Laukkanen, J., Stojanov, G., Poprcova, V.: Some Improvements of the ILP System HYPER Used in the Context of Robotic Learning. MLDM Posters 2009: 68-73
11. Leban, G., Zabkar, J., Bratko, I.: An Experiment in Robot Discovery with ILP. ILP 2008: 77-90
12. McDermott, J.: Dialectical Analogy: The Oscillating Center of Rahner's Thought, Gregorianum 75,: 675-703 (1994)
13. Muggleton, S.: Inverse entailment and Progol. New Generation Computing, 13: 245-286 (1995)
14. Muggleton, S., Bain, M.: Analogical Prediction. ILP 1999: 234-244
15. Muggleton, S.H., Firth, J.: CProgol4.4: a tutorial introduction. In S. Dzeroski and N. Lavrac, editors, Relational Data Mining, pages 160-188. Springer-Verlag, (2001)
16. Muggleton, S., De Raedt, L.: ILP: Theory and Methods. Journal of Logic Programming, 19 and 20, 629-679 (1994)
17. Ohkawa, T., Mori, T., Babaguchi, N., Tezuka, Y.: Analogical Generalization. In: International Conference on Fifth Generation Computer Systems, pp. 497-504. Tokyo (1992)
18. Prassler, E., Kahl, B., Henne, T., Juarez, A., Reggianni, M.: Autonomous Design of Experiments for Learning by Experimentation. Forschungsspitzen und Spitzenforschung, pp. 125-135 (2009)

View publication stats