Robotics and Autonomous Systems 83 (2016) 1–14

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Automatic discovery of relational concepts by an incremental graph-based representation

Ana C. Tenorio-González *, Eduardo F. Morales

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro 1, Sta. Ma. Tonantzintla, Puebla, 72840, Mexico

HIGHLIGHTS

- We designed a method to learn relational concepts from a graph-based representation.
- Our method is designed to discover common/useful concepts of an environment.
- Our method can be used by an autonomous agent like a robot.
- Our method learned common concepts in three domains (polygons/furniture/floors).
- Independent human users validated the common concepts learned by our method.

ARTICLE INFO

Article history:
Received 8 December 2014
Received in revised form
11 May 2016
Accepted 27 June 2016
Available online 5 July 2016

Keywords:
Robot learning
Automatic concept learning
Concept discovery
Predicate invention
Inductive logic programming

ABSTRACT

Automatic discovery of concepts has been an elusive area in machine learning. In this paper, we describe a system, called ADC, that automatically discovers concepts in a robotics domain, performing predicate invention. Unlike traditional approaches of concept discovery, our approach automatically finds and collects instances of potential relational concepts. An agent, using ADC, creates an incremental graph-based representation with the information it gathers while exploring its environment, from which common sub-graphs are identified. The subgraphs discovered are instances of potential relational concepts which are induced with Inductive Logic Programming and predicate invention. Several concepts can be induced concurrently and the learned concepts can form arbitrarily hierarchies. The approach was tested for learning concepts of polygons, furniture, and floors of buildings with a simulated robot and compared with concepts suggested by users.

1. Introduction

The increasing capabilities with which robots are provided at present, are allowing them to perform different tasks in a better way. However, before robots can be used to perform simple tasks, it is normally necessary to provide them with data and complex programs designed by users to simplify the reasoning process of the robot. This can be a time consuming process and involves several iterations until the robot is able to achieve the intended goals. An alternative approach is to let the robot discover by itself the required concepts to accomplish its tasks. Robots can obtain useful data, directly from their own experience with the environment, that could be used to induce concepts. Such abstract concepts can be used to simplify robotics tasks, such as navigation, planning or reasoning. In this paper, we describe how concepts can be automatically learned by a robot while it is exploring its environment. Traditionally, concept discovery departs from a given set of examples. In this research, the agent automatically searches and collects instances of potential concepts.

Robot learning has been a very active research area. Most of the research has been based on reinforcement learning and programming by demonstration [1,2] and, to a smaller extend, on concept learning [3]. In reinforcement learning and programming by demonstration the robot learns how to perform a task and normally assumes that the state-space representation is specified beforehand by the user. Other researches have tried to learn concepts that can be used to represent states useful to a robot, however, the agent normally learns a single concept at a time and the user is heavily involved in carefully preparing the learning settings.

Among the most commonly used approaches for concept discovery are those based on Inductive Logic Programming (ILP) with

* Corresponding author.
E-mail addresses: catanace17@inaoep.mx (A.C. Tenorio-González), emorales@inaoep.mx (E.F. Morales).

1 http://www.inaoe.mx.

http://dx.doi.org/10.1016/j.robot.2016.06.012

predicate invention. ILP with predicate invention provides an understandable representation (to humans) and mechanisms to introduce new predicates not available in the initial background knowledge. It is based on a logic programming language with powerful inference mechanisms that cannot be found in other machine learning techniques. ILP systems using predicate invention can be classified into approaches based on reformulation and on demand-driven approaches [4]. Systems using reformulation introduce new predicates produced by combining or restructuring other predicates to produce a more compact and precise theory (e.g., [5–8]). The demand-driven approaches introduce new predicates when the vocabulary is not enough to induce a theory (e.g., [9–18]).

In this paper, we describe a new approach ADC (Automatic Discovery of Concepts) for discovery of concepts with demand-driven predicate invention in a robotic domain. The robot gathers information while exploring its environment, identifies similar instances of potential concepts and learns relational concepts using ILP. Unlike previous systems, the proposed algorithm is designed to learn multiple concepts about objects and their relations, building hierarchies of concepts, based on data incrementally obtained from the direct experience of an agent with an unknown environment. We applied the proposed approach to learn concepts involving polygons, furniture, and floors of buildings that could be used for manipulation and navigation tasks. We evaluated the usefulness of the induced concepts in new environments and compared them to the expected concepts of human users. The rest of the paper is organized as follows. Section 2 describes the close related work compared with ADC. In Section 3, the proposed system is described in detail. Section 4 presents the experimental results and Section 5 gives general conclusions and future research directions.

2. Related work

In most of ILP systems there is a strong dependency on the user, the data is in many cases, collected by the user before the learning process, the learning scenarios must be prepared, and usually, there is no simultaneous concept learning, i.e., only one concept is learned at a time. In this paper, we propose ADC algorithm to learn several concepts (also hierarchical concepts) at the same time with demand-driven predicate invention; what the system learns depends on the information it gathers during the exploration of an environment and it is not known in advance.

In particular, among the ILP approaches with demand-driven predicate invention, *Statistical Predicate Invention* (SPI) [19] addresses the discovery of new concepts and proposes a generalization of predicate invention, known as statistical learning for hidden variable discovery. The algorithm, *Multiple Relational Clustering* (MRC), is presented to cluster objects, attributes and their relations using Markov logic (an extension of FOL). The process of clustering is done automatically, and it is equivalent to the introduction of new predicates (predicate invention), where each cluster represents a unit predicate. In [16] an algorithm based on teleo-reactive programs (TRP) is proposed for learning new concepts and skills of different hierarchies from existing knowledge. TRP represents hierarchical procedural knowledge, based on reactive execution of goal-oriented skills. In the manner of a STRIPS planner, each skill consists of a goal, an initial state or precondition, an action or method to reach the goal, and a final state or post-condition. First a bottom-up inference mechanism is performed to identify the current state using the perceptions of the agent and the background knowledge, then TRP searches for the first high-level goal that is not satisfied, and tries to form a path in the hierarchy of skills from the current state of the agent to that goal. When a sequence of skills to achieve the goal cannot be found, the algorithm introduces new skills. The new skills form their preconditions and goals using preconditions of concepts and skills closest to the goal. SPI and TRP were tested on databases and card games respectively. Learning of multiple concepts with predicate invention has also been addressed in Hyper [20,18]. This system is one of the latest systems using demand-driven predicate invention in robotics domains. Hyper starts with background knowledge and can perform predicate invention adding placeholder predicates (predicates which are being invented) to the target predicates that are being induced. It works automatically once the experimental setup has been properly prepared. The system reuses previously learned concepts for learning new concepts. Positive and negative examples are provided to the system, the positive examples are obtained by the robot while it explores the environment guided by a human (through commands) and the negative examples are synthetically generated. The system was tested for learning the concepts of movable, not movable, and obstacle. In Hyper, multi-predicate and predicate invention is supported, but, predicate invention is performed using templates for new predicates that should be designed before the learning process. Also, examples should be provided to Hyper, by databases or by a manually guided exploration.

ADC introduces new predicates automatically, as SPI, TRP and Hyper, but also automatically discovers examples of potential concepts from the environment. In this sense, ADC does not need databases of examples or using pre-defined templates for the discovery of new predicates. ADC is able to collect positive examples and create negative examples for the induction process. Also as TRP and Hyper, ADC performs incremental hierarchical multi-predicate learning because of its graph-based representation. Although, other approaches have been proposed to also learn hierarchical concepts (e.g., [12,21–27]), they are usually designed to work on databases or in controlled environments, as in SPI, TRP and Hyper.

Recently, Meta-Interpretive Learning (MIL implemented by Metagol$_{D/O}$) [28,29] has been proposed in ILP based on meta-rules, induction and abduction to produce higher-order datalog programs, which takes advantage of predicate invention and recursivity. MIL is based on incremental declarative multi-predicate learning, using meta-rules to conduct the search of the hypothesis from a set of examples. Meta-rules are like templates where the meta-interpreter performs substitutions to introduce new predicates (predicate invention). Predicate invention creates predicates representing relations instead of objects or propositions. Metagol$_D$ has been used to learn high-order concepts in three domains [29]: the East–West train challenge, NELL language learning task and for top-down construction of re-usable robotic strategies in a simplified model of the world (for construction of walls). Metagol$_O$ has also learned general strategies representing a set of plans (as those learned by traditional AI planning) that reduce the use of resources (e.g., battery, distance) [30]. Metagol$_O$ has shown the advantage of using composite objects and actions (formed by other primitive and/or composite objects and actions) to produce resource efficient strategies from examples. Experiments with composite elements were performed with a simulated humanoid robot in one-dimensional space in tasks of delivery and sorting.

ADC is based on inverse entailment induction (Progol [31]) with first-order logic and can perform incremental multi-predicate learning with predicate invention, as Metagol. Metagol has shown the advantages of learning composite objects and actions, meanwhile ADC has been used to discover composite concepts about objects in robotics domains. The use of meta-rules in MIL is a powerful tool for predicate invention, but it depends on the number and design of the meta-rules to produce useful predicates. Also, Metagol, like traditional ILP systems, requires that sets of examples be provided by the user. In this sense, the main advantage of ADC over other ILP systems, including Metagol, is its ability to automatically discover and collect examples of potential concepts directly from the environment. ADC obtains and creates