I-RIM Conference 2019
October 18 - 20, Rome, Italy
ISBN: 9788894580501
DOI: 10.5281/zenodo.4785262

Learning High-Level Planning Symbols from Intrinsically Motivated Experience

Angelo Oddi, Riccardo Rasconi, Emilio Cartoni, Gabriele Sartor, Gianluca Baldassarre, Vieri Giuliano Santucci
Institute of Cognitive Science and Technologies, National Research Council (ISTC-CNR), Rome, Italy
{angelo.oddi, riccardo.rasconi, emilio.cartoni, gabriele.sartor, gianluca.baldassarre, vieri.santucci}@istc.cnr.it

I. Introduction

One of the main challenges in Artificial Intelligence is the problem of abstracting high-level models directly leveraging the interaction between the agent and the environment, where such interaction is typically performed at low-level through the agent's sensing and actuating capabilities. Such information abstraction process indeed reveals invaluable for high-level planning, as it allows to make explicit the causal relations existing at the high-level which would otherwise remain hidden at low-level. In this respect, some interesting work has been done in the recent literature. For instance, in [3] an algorithm is presented for automatically producing symbolic domains based on the Planning Domain Definition Language (PDDL, see [2]), starting from a set of low-level skills represented in the form of *abstract subgoal options*.

The contribution of this work is the following. First, we extend the scope of the information abstraction procedure proposed in [3] by directly linking it to a robotic architecture (GRAIL – Goal-Discovering Robotic Architecture for Intrinsically-Motivated Learning; [6]) able to autonomously discover goals and learn skills based on *intrinsically motivated* learning algorithms [1]. Such skills are then used as input for the subsequent abstraction process, thus creating an automated information processing pipeline from the low-level direct interaction of the agent with the environment, to the corresponding high-level PDDL domain representation of the environment. Second, given a set of low-level domains in which GRAIL operates, we carry out an analysis on the features of the produced abstract PDDL representations depending on the categorization capabilities of the classifiers used for the production of the symbolic vocabulary, thus shedding some light on a number of interesting correlations between low-level generalization capabilities of the abstraction procedure and the quality of the produced PDDL high-level representations. Third, we have tested the overall system within the context of an ESA project called IMPACT (see the acknowledgement footnote for more details), in particular within two different space exploration scenarios, demonstrating the advantages of enhancing the well-known Sense-Plan-Act (SPA) paradigm for controlling robotic systems [4] with autonomous skill-learning capabilities in the general context of space exploration.

This research has been supported by the European Space Agency (ESA) under contract No. 4000124068/18/NL/CRS, project IMPACT - Intrinsically Motivated Planning Architecture for Curiosity-driven roboTs - and the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 713010, Project "GOAL-Robots – Goal-based Open-ended Autonomous Learning Robots". The view expressed in this paper can in no way be taken to reflect the official opinion of the European Space Agency.

II. The GRAIL Skill Learning System

We applied the abstraction procedure on the options found by M-GRAIL [7], an advancement of GRAIL [6]. GRAIL is an open-ended learning system that discovers new interesting events while interacting with the environment and stores them as "goals". GRAIL then automatically learns through *intrinsic motivation* to achieve those goals from different starting conditions. For each goal, GRAIL builds a separate "skill" that achieves that goal. By using competence-based intrinsic motivation, GRAIL focuses its learning to achieve the highest overall competence (i.e. reliability) on all skills as fast as possible. M-GRAIL also keeps a series of predictors that predict the percentage of success of the skill depending on the starting condition, thus enabling M-GRAIL to recognize when the skill can be successfully initiated.

III. The Information Abstraction Procedure

The information abstraction procedure (called *PDDL-Gen* in this work) has the objective of transforming the environmental low-level knowledge learned by M-GRAIL in a PDDL-based representation of the operational domain suitable for high-level planning. The fully detailed description of the domain abstraction procedure can be found in [3].

The procedure accepts in input an *option-based* [8] representation for each skill previously learned by M-GRAIL, expressed in the form of two classifiers for each option (a.k.a. the option's *characterizing set*), namely the *Initiation Set* classifier, $Cl(I)$, and the *Effect Set* classifier, $Cl(E)$.

The generated model is a set-theoretic high-level domain specification using the Planning and Domain Definition Language (PDDL) formalization [2], which is the most widely used input format for most off-the-shelf automated planners. A set-theoretic specification is expressed in terms of a set of propositional symbols $P = \{\sigma_1, ..., \sigma_n\}$, each associated to a grounding classifier $Cl(\sigma_i)$, and a set of operators $A = \{op_1, ..., op_m\}$. Each operator op_i is described by the tuple $op_i = \langle pre_i, \mathit{eff}_i^+, \mathit{eff}_i^- \rangle$, where pre_i contains all the propositional symbols that must be $true$ in a state s for opt_i to be executed from s, while eff_i^+ and eff_i^- contain the propositional symbols that are respectively set to $true$ or $false$ after op_i's execution. All the other propositional symbols remain unaffected by the execution of the operator. In order to produce a correct PDDL representation, it is therefore necessary to populate the three sets (pre_i, eff_i^+ and eff_i^-) *for each option* o_i by properly selecting which symbols, among those contained in P, will fall in any of such sets.

A. Building the datasets for PDDL-Gen

To build the datasets needed for the classifiers and the set representations of the initiation and effects set, we chose to use data from each skill only after that skill becomes fully reliable (i.e. it no longer fails); this assumes a non-stochastic environment where it is possible to learn skills with guaranteed success. To build the $Cl(I)$ classifier training dataset, we considered as positive cases all the low-level variable values before the successful execution of the skill, and as negative cases all the low-level variable values in conditions where GRAIL tried to execute the skill but the predictor was always zero. To build the $Cl(E)$ classifier training dataset, we considered as positive cases all the low-level variable values after the skill was successfully executed. As negative effect cases, we used all the low-level variable values before the execution of that skill, whether it succeeded or not (since we know that GRAIL will not execute the skill if its goal/effect is already achieved). As for the masks dataset, a collection of all successful executions of the skill was used to compare the variables before and after the execution and see which ones were affected by each skill.

B. Choosing a classifier

PDDL-Gen requires that the so-called *projection* operator is applied to the initiation and effect sets. So it is important that the initiation and effect sets are represented with a data structure that lends itself to be "projected". However, PDDL-Gen does not explicitly state the representation method on which this operator can be applied, thus leaving such choice as an implementation decision. Classifiers that build a decision tree, such as C4.5 (used in [3]), can be easily converted into a "projectable" set representation. However, building such a representation from a C4.5 decision tree, does not always yield optimal results. We hence developed a method to derive a projectable set representation that compactly describes the initiation and effect set. We will call this method "*Intersection+Mask*" (*IntM*), and compare it to the simpler representation obtained through C4.5.

IV. Empirical analysis and Conclusion

We have carried on an empirical analysis on a number of interesting correlations between low-level generalization capabilities of the abstraction procedure and the completeness/quality of the produced high-level symbolic domains. In particular, we analyze a number of relevant features in the representations obtained using the *C4.5*, and *IntM* classifiers, testing them in the so-called *bulbs domain* [5] on three different cases: (i) a circular scenario referred to as *Reset* scenario, (ii) a scenario where the addition of some negative effects to the output PDDL representation depends on the kind of classifier used (*Negative* scenario), and (iii) a scenario where some states cannot be reached by the robot actions (*Unreachable* scenario). Lastly, we have tested the overall system within the context of the above cited IMPACT project, demonstrating its capabilities in the following two simulated space scenarios.

1) the *Rover Scenario*, where we demonstrate how the system can discover new ways to reach an already known effect by applying the *PDDL-Gen* procedure. We propose a situation where the orientation mechanism of a planetary rover antenna has been damaged and the rover can no longer use it to point the antenna and establish stable communication. Our technology can be used to demonstrate how the rover is capable of enriching its planning domain with the necessary knowledge to orient the antenna merely using the locomotion capabilities, for example moving around the entire body in order to reach the correct attitude to gain and maintain communication, possibly exploiting terrain slopes and/or small rocks.
2) the *Robot Arm* scenario is designed to demonstrate the ability of the our system to acquire new ways to interact with the environment and integrate them in its planning domain. In this scenario, a robot equipped with a gripper actuator attached to a manoeuvrable arm tries to grasp a *vase-shaped* rock whose diameter exceeds the max opening span of the gripper. The robot is thus not able to pick-up the rock with its basic grasping skill - however, upon failure, the system will automatically trigger the learning of a new skill and the robot will at the end be able to pick-up the *vase-shaped* rock by grasping it from its edge.

Among the possible directions of future work we consider the integration of symbolic planning and open-ended learning to increase the ability on one agent to autonomously acquire new skills.

References

[1] Gianluca Baldassarre, Marco Mirolli (eds.). *Intrinsically motivated learning in natural and artificial systems.* Berlin: Springer. 2013.

[2] M. Ghallab, A. Howe, C. Knoblock, D. Mcdermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL—The Planning Domain Definition Language, 1998.

[3] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018.

[4] Félix Ingrand, Simon Lacroix, Solange Lemai-Chenevier, and Frederic Py. Decisional autonomy of planetary rovers. *Journal of Field Robotics*, 24(7):559–580, 2007.

[5] Angelo Oddi, Riccardo Rasconi, Emilio Cartoni, Gabriele Sartor, Gianluca Baldassarre, and Vieri Giuliano Santucci. Learning high-level planning symbols from intrinsically motivated experience, 2019. *arXiv:1907.08313*.

[6] Vieri Giuliano Santucci, Gianluca Baldassarre, and Marco Mirolli. GRAIL: A goal-discovering robotic architecture for intrinsically-motivated learning. *IEEE Trans. Cognitive and Developmental Systems*, 8(3):214–231, 2016.

[7] Vieri Giuliano Santucci, Gianluca Baldassarre, and Emilio Cartoni. Autonomous reinforcement learning of multiple interrelated tasks. *arXiv preprint arXiv:1906.01374*, 2019.

[8] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1-2):181–211, August 1999.