Learning Sequences of Policies by using an Intrinsically Motivated Learner and a Task Hierarchy

Nicolas Duminy[1,4] Alexandre Manoury[2,4] Sao Mai Nguyen[2,4] Cédric Buche[3,4] Dominique Duhaut[1,4]

***Abstract*— Our goal is to propose an algorithm for robots to learn sequences of actions, also called policies, in order to achieve complex tasks. We consider in this paper multiple and hierarchical tasks of various difficulties. To tackle this highly dimensional learning we propose a new algorithm, named Socially Guided Intrinsic Motivation for Sequence of Actions through Hierarchical Tasks (SGIM-SAHT), based on intrinsic motivation and using different learning strategies. We then present two implementations of this algorithm designed to address this challenge in different ways: through a "procedures" framework for Socially Guided Intrinsic Motivation with Procedure Babbling (SGIM-PB) and owing to planning and a dynamic environment representation learning for Continual Hierarchical Intrinsically Motivated Exploration (CHIME).**

We compare the two implementations and show, through two experiments, how efficiently they learn sequences of actions and dynamically adapt to their environment. We also discuss the benefits of implementing a full unified version of SGIM-SAHT using all the mentioned features of both implementations.

I. Introduction

Using the developmental robotic approach [1], we present a generic algorithmic architecture combining active motor skill learning based on goal-oriented exploration with strategical learning to learn a set of multiple hierarchical and interrelated tasks. This architecture enables a robot to learn a mapping between a continuous space of parametrized tasks (referred to here as outcomes) and a space of parametrized policies (also referred to as actions).

A. Active motor skill learning of multiple tasks

Learning multiple tasks is difficult in classical Reinforcement Learning techniques [2] [3] as they still need a manually designed reward function for each task. The recent introduction of Intrinsic Motivation (IM), which triggers curiosity in humans according to developmental psychology [4], enabled highly-redundant robots to learn a wide range of tasks, using goal-babbling [5] [6].

However, in these studies, the policies were policy primitives of predefined complexity. We would like to consider multiple tasks of various complexity requiring actions of different complexity/duration/length.

[1] Université Bretagne Sud, Lorient, France. nicolas.duminy@telecom-bretagne.eu, dominique.duhaut@univ-ubs.fr

[2] IMT Atlantique, Brest, France. nguyensmai@gmail.com alexandre.manoury@imt-atlantique.fr

[3] Cédric Buche is with ENIB, Brest, France. buche@enib.fr

[4] Lab-STICC, CNRS

The research work presented is partially supported by the European Regional Fund (FEDER) via the VITAAL Contrat Plan Etat Region

B. Learning sequences of motor policies

In this article, we consider the learning of sequences of motor policies (also called complex motor policies).

We wanted to enable the learner to decide autonomously the complexity of the policy necessary to solve a task, so we discarded via-points which often require the setting of a number of via points such as in [3]. Options [7] are temporally abstract policies built to solve one particular task. They have only been proven efficient in the case of a small number of discrete tasks and policies. We get inspired by them and extend the idea for an unlimited number of complex policies and in continuous spaces.

C. Tasks hierarchy

[8] showed that building complex policies made of lower-level policies according to the task hierarchy bootstrap exploration by reaching interesting outcomes more rapidly. Task hierarchy was used in combination with intrinsic motivation in [9] to reuse previously acquired skills to build more complex ones for tool use. We adopted a similar approach, but we do not only use primitive actions and the hierarchy of tasks is not given in advance but learned online.

Our requirements of learning different tasks, potentially infinite sequences of actions, and the task hierarchy, entail an even more high-dimensional space to explore. Unfortunately, the curse of dimensionality makes the efficiency of learning algorithms, even those using IM, plummet when facing higher outcome space dimensionalities [10].

D. Strategic learning

The curse of dimensionality has been tackled by approaches based on strategic learning [11]. They enable a learner to self-organize its learning process by choosing both what [5] (which outcome to focus on) and how [12] (which strategy to use) to learn. A strategic learner was implemented for an infinite set of outcomes and policies in continuous spaces by the SGIM-ACTS (Socially Guided Intrinsically Motivation with Active Choice of Teacher and Strategy) algorithm [13]. It relies on the empirical evaluation of its learning process to actively decide both which strategy to use and which outcome to target. As it showed its potential on a high dimensional robot learning a set of hierarchically organized tasks [14], we extend it for our learner to infer online its learning curriculum when exploring the actions, the task and the hierarchy of tasks spaces.

We adapted SGIM-ACTS to learn complex motor policies of unlimited size by leveraging task hierarchy, and propose a generic algorithmic architecture called SGIM-SAHT, to

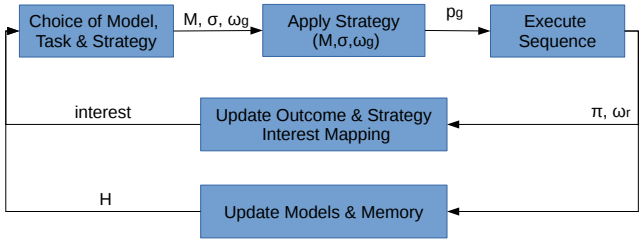

Fig. 1: The SGIM-SAHT algorithmic architecture

actively decide which outcome to focus on, which strategy to use, and how to use the task hierarchy. We describe the approach leading to it and the algorithm. Then we present two different implementations and compare them to finally discuss how they could further be merged together to fully implement SGIM-SAHT.

II. Approach

A. Formalization

Let us consider a robot performing motions described as primitive policies $\pi_\theta \in \Pi \subset \mathbb{R}^{\mathbb{N}}$. It can also perform sequences of primitive policies of any length i, $\pi \in \Pi^i \subset \Pi^{\mathbb{N}}$. The policies performed by the robot, have consequences on its environment, which we call outcomes $\omega \in \Omega$. Those outcomes can be of various types and dimensionalities, and are therefore split in outcome subspaces $\Omega_i \subset \Omega$. The robot has to learn the mapping between the policy space $\Pi^{\mathbb{N}}$ and the outcome space Ω: it learns to predict the outcome ω of each policy π (the forward model L), but it also learns which policy can reach any given outcome (the inverse model L^{-1}).

The policies and outcomes are grouped in the features ensemble $F = \Pi \cup \Omega$. We call feature sequences l_f: *complex policies* if $l_f \in \Pi^{\mathbb{N}}$, or *procedures* if $l_f \in \Omega^{\mathbb{N}}$. However procedures or other sequences composed of outcomes are internally used by the learning agent. Only complex policies can be executed on the environment.

We define a model $M(\Omega_i \to F_j) : \omega \mapsto l_f$ the mapping between an outcome subspace Ω_i and a sequence of features space $F_j \subset F^{\mathbb{N}}$. These models enable a hierarchical representation of the robot environment. We call simple models those which map to a singular feature space $F_j \subset F$. We note H the set of all models. We define a strategy σ any process enabling the building of a sequence of features l_f.

B. Algorithmic Architecture

The SGIM-SAHT algorithm learns by episodes in which a model $M \in H$ to work on, a goal outcome $\omega_g \in \Omega$ and a strategy σ have been selected.

The selected strategy σ applied to the chosen goal outcome ω_g builds a feature sequence l_f to try reaching the goal, under the constraints of the chosen model (details in III-D).

This feature sequence l_f is broken down in a complex motor policy $\pi \in \Pi^{\mathbb{N}}$, to be executed by the robot (section III-E). The outcomes on the environment ω_r are then recorded, along with the policies and built features sequence. This breakdown process is potentially recursive, based on the learned hierarchy between tasks or models (section III-B).

After each episode, the learner stores the executed policies π and feature sequences l_f, along with their reached outcomes in its episodic memory. Then, it computes its competence $competence(\omega_g)$ at reaching the goal ω_g, which depends on the euclidean distance between ω_g and the reached outcome ω_r. Its exact definition depends on the implementation (see section III-C). More importantly, the learner updates its interest map, by computing the interest of the goal outcome for the used strategy $interest(\omega_g, \sigma)$. This interest depends on the progress measure $p(\omega_g)$ which is the derivative of the competence.

The learner then uses these interest measures to partition the outcome space Ω in regions R_i of high and low progress. This process is described in detail in [13]. In the beginning of the next episode, the learner chooses the strategy, model and goal outcome that could bring the most progress, according to the updated interest map.

Algorithm 1 Algorithm

Input: the different strategies $\sigma_1, ..., \sigma_n$
Input: the initial model hierarchy H
Initialization: partition of outcome spaces $R \leftarrow \bigsqcup_i \{\Omega_i\}$
Initialization: episodic memory *Memory* $\leftarrow \emptyset$
loop
  $\omega_g, \sigma, M \leftarrow$ Select Goal Outcome, Strategy and Model(R, H)
  $l_f \leftarrow$ Execute Strategy(σ, ω_g)
  Memory $\leftarrow$ Execute Sequence(l_f)
  Update M with collected data *Memory*
  $R \leftarrow$ Update Outcome and Strategy Interest Mapping(R,*Memory*,ω_g)
  Update models H with *Memory*
end loop

III. Implementations

In this section, we present two different implementations of the algorithmic architecture. Both approaches use a hierarchical representation to help the learning process of an intrinsically motivated learner tackling multiple tasks, but they differ in the specific methods employed, which are the reflection of the different kinds of experimental setups they were built for. The first implementation is Socially-Guided Intrinsic Motivation with Procedure Babbling (SGIM-PB) from [15], the second is Continual Hierarchical Intrinsically Motivated Exploration (CHIME). Both implementations have been tested on their respective setups built to emphasize its specificities and qualities.

A. Setups

SGIM-PB was applied to a Yumi industrial robot arm to interact with a RFID tangible interactive table (Fig. 2) that can sense the position of the robot over the table and play sounds according to the virtual object positions. The overall evaluation of SGIM-PB for the simulation setup are shown on Fig. 6. More details are reported in [15]. The algorithm is being tested on a real version of the setup.

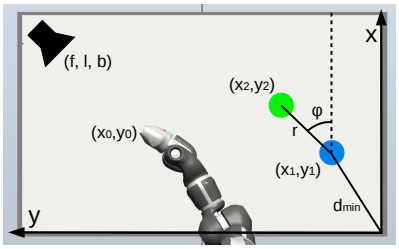

Fig. 2: Representation of the Yumi experimental setup, with the first object in blue, the second in green, and the produced sound represented in the top left corner. It could perform sequences of DMP and produce 5 hierarchically organized types of outcomes.

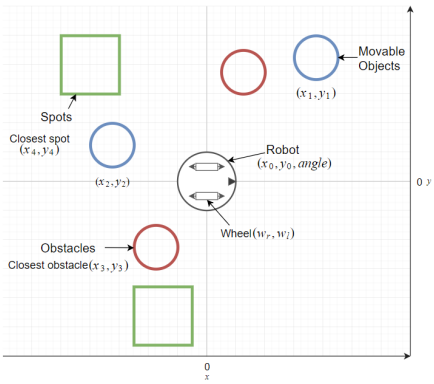

Fig. 3: CHIME experimental setup, (w_r, w_l) are the primitive policy spaces, the outcome spaces are indicated inside parenthesis and a reward outcome value v, depending on whether an object is on a spot or not, is also present.

As CHIME uses planning algorithms, we chose simulated 2D environment to control a mobile robot learning to move itself and other objects while avoiding obstacles (see Fig. 3).

B. Hierarchy representation

The idea of SGIM-PB and CHIME is to use a hierarchical representation of the outcome spaces, to outline the dependencies between tasks to help the reuse of previous knowledge from easy tasks for more complex tasks.

For example in the Yumi setup, the learner can move the first object, then the second object to generate a sound. The hierarchical representation for SGIM-PB is based on a static set of outcome spaces. It corresponds to procedural spaces, which are subsets of $\Omega^{\mathbb{N}}$. Those spaces are used to generate and exploit procedures, which are sequences of previously learned outcomes. They represent combinations of learned skills which are used to learn new, more complex ones. Although this framework theoretically allows the creation of procedures of any length, they have been limited to combinations of two outcomes so as to limit their potential new complexity. This method enabled the learner to effectively discover the outcome hierarchy/dependency [15].

In comparison, for the CHIME algorithm, the hierarchical representation is based on a dynamic set of outcome spaces. It is constructed using simple models $M(\Omega_i \rightarrow O_j)$ which can rely on others: lower models map outcomes to policies while higher models map them to other outcomes that should be reached. For instance in our CHIME setup, the robot can move itself $(x_0, y_0 \in \Omega_0)$ in order to move an object $(x_1, y_1 \in \Omega_1)$ onto a spot to control a reward value $(v \in \Omega_5)$. Moreover, $O_j \subset O$ instead of $O^{\mathbb{N}}$ as in SGIM-PB, thus only primitive policies are learned and not sequences of them. The idea is to then use planning to construct sequences of policies from these learned primitives. The dynamic aspect lets the robot discover and adapt itself the models, considering the environment feedbacks. This dynamic update was inspired by a self adapting SVM, changing its inputs to match the environment, as presented in [16]. At the beginning $H = \emptyset$, no model is present, and the robot chooses itself what model to create or modify: if Ω_i seems to be highly correlated to O_j it may create the model $M(\Omega_i \rightarrow O_j)$. CHIME efficiently discovers the environment hierarchy and constructs adapted models accordingly: on Fig. 4 the 3 models are successively constructed by the robot and corresponds to what a human could interpret from the setup.

C. Goal, strategy and model selection

SGIM-PB uses a static representation of the features and thus only uses one model $M(\Omega \rightarrow O)$, to infer the hierarchy between tasks. Therefore it does not contain a model selection step. The competence measure $Competence(\omega_g, \omega)$ used by SGIM-PB corresponds to the euclidean distance between the reached outcome ω and the goal ω_g, multiplied by a factor γ^n representing the cost of the n-size policy used, so as to limit the size of used policies.

The interest measure $interest(\omega_g, \sigma)$ corresponds to that of the SGIM-ACTS algorithm in [13] divided by the size of the policies built during the episode to guide the learning process towards the less complex outcome spaces first. Strategy σ and goal outcome ω_g are chosen according to the updated interest map. The Yumi experiment showed SGIM-PB could adapt the strategy to each outcome subspace.

In comparison the CHIME algorithm uses a dynamic set of features, via multiple models. Each model has its own interest map. The competence measure used by the main algorithm $Competence(\omega_g, M)$ evaluates the correlation between spaces of M around ω_g. The competence is 1 if the relation is linear, $\simeq 0$ if lowly related and < 1 if noised.

A model M and its interest map are chosen accordingly to the mean interest of each map. The strategy and goal are then chosen as in SGIM-PB among the interest map of M.

In our setup, CHIME manages well the time spent on each model and focuses on newly created and interesting models instead of already known ones. As seen in Fig. 4, the robot selects first simple models: moving itself, then more complicated: moving objects and placing them on spots.

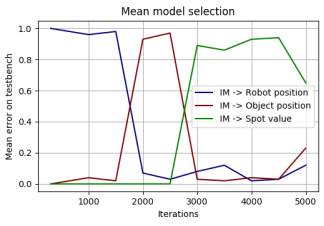

Fig. 4: Mean model selection during episodes all along learning, based on interest

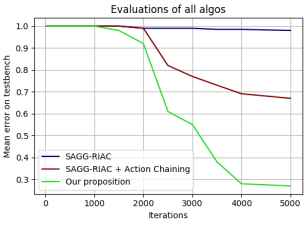

Fig. 5: Learning a hierarchical task: placing objects on the spots, i.e. controlling the observable v

D. Strategies

SGIM-PB uses a combination of autonomous exploration strategies and interactive ones using the expertise from

human teachers to bootstrap the learning process. These types of strategies are used to explore either the policy space $\Pi^{\mathbb{N}}$ or the procedural space $\Omega^{\mathbb{N}}$.

CHIME only uses two strategies: a random policy exploration one, selecting a single primitive policy to execute (this helps the construction of new models), and an autonomous exploration strategy trying to produce w_g using an observable $o \in O$. This strategy is almost identical to SGIM-PB but it can also use planning to create a sequence of observables to be executed in order to reach distant goals or avoid obstacles: $l_f = [o_1, \ldots, o_n]$, each o_i corresponds to a planning step.

E. Sequence execution

In SGIM-PB, although the procedures framework theoretically enables a recursive process to replace each outcome of a complex procedure by simpler ones, we limited it to a one-step process. Complex policies are directly executed and procedures (ω_i, ω_j) are replaced, by the closest ones feasible by the learner according to its current skill set. Then this procedure is broken down to a succession of two policies, possibly complex, which is then executed. During the execution, each step starting from the initial configuration is stored in the learner dataset. Fig 7 shows SGIM-PB could adapt the complexity of its policies to the task at hand.

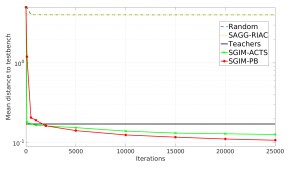

Fig. 6: Mean evaluation of SGIM-PB across learning process

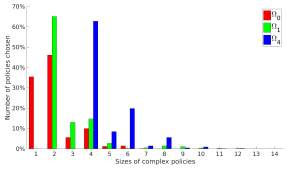

Fig. 7: Size of policies chosen by SGIM-PB for 3 increasingly more complex tasks

In CHIME, the sequence l_f in converted to a sequence of primitive policies thanks to models and planning. For each element o_i of l_f: if o_i is a primitive policy it is then executed by the robot, else a model is found to compute a lower level observable o_i' to reach in order to approach o_i. A planning step is then applied on o_i' to obtain a sequence l_g' and the process continues recursively on each element of l_g'. This hierarchical and planning combination performs well in this experimental environment: indeed, goals blocked by obstacles or distant are reachable. On Fig. 5, we can see a comparison of learning a hierarchical task: placing objects on spots. CHIME manages to reach goals too difficult for SAGG-RIAC (Self-Adaptative Goal Generation - Robust Intelligent Adaptative Curiosity [5]), owing to its planning and its hierarchical representation of the environment.

IV. Conclusion and Future Works

Through this article, we have presented and shown the interest of SGIM-PB and CHIME in learning to perform complex tasks. They both efficiently managed to learn them through procedures and planning. We have also proposed an algorithm unifying both approaches and regrouping similar aspects, as the intrinsically guided strategical learning and the hierarchical representation. As SGIM-PB relies on a given set of outcome features and explores the dependencies between tasks to learn sequences of actions, CHIME builds dynamically its set of features to construct models that are then used to plan sequences of actions.

In future works, we consider developing an implementation of the SGIM-SAHT algorithm using all the described features: learning primitive policies and then planning sequences of them, but also learning and optimizing directly these sequences thanks to the procedure framework. The planning will bootstrap the learning while learning the sequence, and not just the primitive, will result in more optimized policies and will reduce the planning complexity.

We also wish to compare the different performances and features from each algorithm on a common experiment.

References

[1] M. Lungarella, G. Metta, R. Pfeifer, and i. G. Sandin, "Developmental robotics: a survey," *Connection Science*, vol. 15, no. 4, pp. 151–190, 2003.

[2] E. Theodorou, J. Buchli, and S. Schaal, "Reinforcement learning of motor skills in high dimensions: A path integral approach," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2397–2403.

[3] F. Stulp and S. Schaal, "Hierarchical reinforcement learning with movement primitives," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 231–238.

[4] E. Deci and R. M. Ryan, *Intrinsic Motivation and self-determination in human behavior*. New York: Plenum Press, 1985.

[5] A. Baranes and P.-Y. Oudeyer, "Intrinsically motivated goal exploration for active motor learning in robots: A case study," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1766–1773.

[6] M. Rolf, J. Steil, and M. Gienger, "Goal babbling permits direct learning of inverse kinematics," *IEEE Trans. Autonomous Mental Development*, vol. 2, no. 3, pp. 216–229, 09/2010 2010.

[7] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[8] A. G. Barto, G. Konidaris, and C. Vigorito, "Behavioral hierarchy: exploration and representation," in *Computational and Robotic Models of the Hierarchical Organization of Behavior*. Springer, 2013, pp. 13–46.

[9] S. Forestier and P.-Y. Oudeyer, "Curiosity-driven development of tool use precursors: a computational model," in *38th Annual Conference of the Cognitive Science Society (CogSci 2016)*, 2016, pp. 1859–1864.

[10] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 49–73, 2013.

[11] M. Lopes and P.-Y. Oudeyer, "The strategic student approach for life-long exploration and learning," in *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–8.

[12] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *The Journal of Machine Learning Research,*, vol. 5, pp. 255–291, 2004.

[13] S. M. Nguyen and P.-Y. Oudeyer, "Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner," *Paladyn Journal of Behavioural Robotics*, vol. 3, no. 3, pp. 136–146, 2012.

[14] N. Duminy, S. M. Nguyen, and D. Duhaut, "Strategic and interactive learning of a hierarchical set of tasks by the Poppy humanoid robot," in *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Sept. 2016, pp. 204–209.

[15] ——, "Effects of social guidance on a robot learning sequences of policies in hierarchical learning," in *IEEE International Conference on Systems, Man and Cybernetics, 2018. SMC2018*. IEEE, 2018, accepted.

[16] E. Ugur and J. Piater, "Emergent structuring of interdependent affordance learning tasks using intrinsic motivation and empirical feature selection," pp. 1–13, 2016.