# Learning Reusable Skills through Self-Motivation

**Andrea Bonarini**                                    BONARINI@ELET.POLIMI.IT
**Alessandro Lazaric**                                  LAZARIC@ELET.POLIMI.IT
**Marcello Restelli**                                   RESTELLI@ELET.POLIMI.IT
Dep. of Electronics and Information, Politecnico di Milano, piazza Leonardo Da Vinci, 32, 20133 Milan, Italy

## Abstract

This paper presents SMILe (Self-Motivated Incremental Learning), an intrinsically motivated learning framework in which an agent autonomously identifies *interesting* subgoals independently from any task prescription, and, driven by self-motivation, incrementally learns a hierarchy of more and more complex skills. Preliminary experimental activities show that learned skills can be profitably reused when the agent faces different tasks.

## 1. Introduction

One of the key factors that would enable intelligent agents to achieve *autonomy*, is the ability to operate without any external (e.g., human) intervention and to progressively learn optimal solutions for a given task. Up to now, research in the field of Reinforcement Learning (RL) has led to agents that are able to act autonomously in simple and limited environments. To operate in real-world environments, an agent should be able to exploit its knowledge and adapt its skills to situations different from (but similar to) the ones that the agent already faced. The lack of methods for an effective reuse of the knowledge acquired during the learning process to novel situations implies a waste of experience that may be not affordable in many real applications.

Several approaches that involve the transfer of learned knowledge make use of hierarchical representations to structure concepts and solutions. Decomposing the solution of complex problems into simple loosely coupled activities helps to build basic blocks that may

be reused within different contexts. The introduction of higher levels of abstraction reduces the search space, thus making the learning process more effective. The decomposition of a problem into subtasks can be effectively used to improve the performance of RL algorithms, as in *Hierarchical Reinforcement Learning* (HRL) (Barto & Mahadevan, 2003). In general, these approaches exploit particular task decomposition structures and define specific learning processes, that are performed on simple tasks whose solutions can be composed to solve the global problem.

Some works studied how to exploit the hierarchical task decomposition to transfer solutions learned for subtasks to different tasks. (Ravindran & Barto, 2003) define subtask solutions without an absolute frame of reference, so that, after suitable transformations, they can be reused within different contexts. (Mehta et al., 2005) propose a model-based HRL method to transfer the knowledge learned for a specific MDP to other MDPs that share the same dynamics, but that have different reward structures. These approaches need a decomposition given by the designer in advance and, although the learning speed-up obtained by using HRL is relevant, it is often difficult to find an effective task decomposition and long hand-tuning is needed.

Recently, many proposals for the automatic discovery of subgoals have been presented (Simsek et al., 2005; McGovern & Barto, 2001). Most of the automatic *subgoal discovery* techniques identify regions of the environment that are strategic for the solution of a specific problem and learn new skills designed to reach those regions. Although the discovered skills are useful to solve the given problem, they may not always be effectively reused in different tasks, since region identification is task specific. A goal independent approach has been recently proposed by (Mahadevan, 2005). Starting from the analysis of data about the environment

dynamics, proto-value functions are build and are linearly combined to learn any value function that may be defined over the same environment.

On the other hand, *Intrinsically Motivated Learning* (IML) (Singh et al., 2004) enables agents to develop, without any externally imposed goal, new skills that can be reused to solve many different tasks. Since no goal is provided, the agent should be able to autonomously identify situations that may be relevant for the accomplishment of different tasks and, at the same time, an intrinsic motivation should guide the agent to learn new skills to achieve those situations.

In this paper, we propose SMILe (Self-Motivated Incremental Learning), an intrinsically motivated learning framework in which a hierarchy of skills are autonomously learned by iterating a three phase process aimed at exploring the environment, identifying interesting situations, and acquiring skills to reach these situations. The identification of particular configurations in the environment as the goal for the new skills leads to the definition of a hierarchy of goal-independent and general skills. Therefore, unlike hand-coded skills that are often strictly related to a specific task, the knowledge acquired during the development process (i.e. the skills) can be used to solve many different tasks in a short time.

# 2. Self-Motivated Development Process

SMILe implements a self-motivated development process aimed at learning a set of skills that could be used to face different tasks. Each skill is learned through a development process that is divided into three main phases: babbling, motivating and skill acquisition.

## 2.1. Option Framework

The hierarchy of skills developed during the learning process is described using the *option* framework (Sutton et al., 1999) and the environment is formally described as a Semi-Markov Decision Process (SMDP). An option-SMDP is characterized by the tuple $\langle \mathcal{S}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{O}$ is the set of options, $\mathcal{P}(s, o, s')$ is the transition model, $\mathcal{R}(s)$ the reward function and $\gamma$ is the discount factor.

When the development process starts, the skills of the agent are equal to the set of its basic actions $\mathcal{A}$. At the $k$-th iteration of the entire process, the set of admissible options is: $\mathcal{O}^k = \mathcal{O}^{k-1} \cup \{o^k\}$, where $o^k$ is the option learned at the $k$-th iteration and $\mathcal{O}^0 = \mathcal{A}$. Therefore, the initial transition model can be defined as $\mathcal{P}^0(s, o, s') = \mathcal{P}(s, a, s')$, where $a \in \mathcal{A}$. A generic option $o^k \in \mathcal{O}^k$ is defined as the tuple $\langle \pi_{o^k}, \mathcal{I}, \beta \rangle$, where the closed-loop policy of the option $\pi_{o^k} : \mathcal{S} \times \mathcal{O}^{k-1} \to [0, 1]$ is the probability to take an option in state $s$, the initial set $\mathcal{I} \subseteq \mathcal{S}$ is a subset of the state space where the option is defined, $\beta(s)$ is the probability for the option to terminate in state $s$. Since new options are obtained as a composition of the skills available to the agent, each policy $\pi(s, o)$ over options can be flattened upon the initial set of basic actions $\mathcal{A}$, thus obtaining a policy $\overline{\pi} = flat(\pi)$, defined as $\overline{\pi} : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Finally, we define the *state transition* probability, that is the probability to get from state $s$ to $s'$ by following a policy $\pi$ over options as:

$$P_\pi(s, s') = \sum_{o \in \mathcal{O}} \pi(s, o) \mathcal{P}(s, o, s'). \tag{1}$$

This probability may be interpreted as the ease of the agent to get from $s$ to $s'$ with the chosen policy $\pi$ given the transition model $\mathcal{P}$.

## 2.2. Babbling Phase

In the babbling phase, the agent estimates both the transition model and the state transition probabilities using an unbiased exploration policy. Since in this phase no goal guides the agent behavior, at iteration $k$, the explorative policy $\pi_R^k$ is simply a uniform probability distribution over the set of available options:

$$\pi_R^k(s, o) = \frac{1}{|\mathcal{O}^k|}. \tag{2}$$

Since the policy of each option selects basic actions with different probabilities, the flattened policy generally is not a uniform distribution over actions:

$$\overline{\pi}_R^k(s, a) = flat(\pi_R^k) \neq \frac{1}{|\mathcal{A}|}. \tag{3}$$

Thus, while at the first iteration the agent takes actions at random, when new options are added, its behavior is biased by the policy of the options in $\mathcal{O}^k$.

A different exploration policy over basic actions significantly affects also the state transition probabilities and the ability of the agent to move in the environment. At the $k$-th iteration, once the option $o^k$ is learned, a new transition model $\mathcal{P}^k(s, o, s')$, with $o \in \mathcal{O}^k$, is generated. In particular, it is interesting to analyze how the capacity of the agent of moving among states $(P_\pi^k(s, s'))$ consequently changes. Given the random exploration policy $\pi_R^k$, the state transition probability can be written as:

$$\begin{aligned} P_{\pi_R^k}(s, s') &= \sum_{o \in \mathcal{O}^k} \pi_R^k(s, o) \mathcal{P}^k(s, o, s') \\ &= \sum_{a \in \mathcal{A}} \overline{\pi}_R^k(s, a) \mathcal{P}(s, a, s'). \end{aligned} \tag{4}$$

This means that the modification on the state transition probabilities is caused by a different policy over the set of actions. In other words, the introduction of a new option biases the exploration over the state space, making the agent able to reach more (less) frequently regions that were previously less (more) visited.

### 2.3. Motivating Phase

Since the learning process of SMILe is not guided by any extrinsically-defined reward function, the agent identifies relevant states according to a given definition of interest based on the state transition probabilities estimated in the babbling phase $\left(P_{\pi_R^k}(s, s')\right)$. In the motivating phase, we introduce a general definition of an interest function, whose meaning depends on a chosen local measure of interest. Although several characteristics of the state transition model could be used to compute the local interest of a state, in this paper we will focus on the following definition:

$$\rho(s) = (1 - p_{in}(s)) - p_{in}(s)(1 - p_{out}(s)), \quad (5)$$

where $p_{in}(s) = \frac{1}{|\mathcal{S}|} \sum_{s' \in \mathcal{S}} P_{\pi_R}(s', s)$ and $p_{out}(s) = \sum_{s' \neq s} P_{\pi_R}(s, s')$. The intuition behind Equation 5 is that states that, under a random policy, are difficult to reach or that, once reached, can be easily left are relevant as subgoals for many complex tasks whose solution needs the agent to pass through states that cannot be easily reached without a specific skill.

Since the definition of $\rho(s)$ considers only one-step probabilities, it does not take into account the interest of surrounding states. Thus, a global interest function $I^k(s)$ must be computed by propagating the local interest function on the basis of the state transition probabilities estimated in the babbling phase. The global interest function is defined by the Bellman-like equation:

$$I^k(s) = \rho^k(s) + \gamma \sum_{s' \in \mathcal{S}} P_{\pi_R^k}(s, s') I^k(s'). \quad (6)$$

The interest function $I^k(s)$ can be computed using a simple iterative policy evaluation algorithm. Given the global interest function, the agent identifies a new self-motivated goal as the state with the highest interest value, $\overline{s}^k = arg \max_s I^k(s)$.

### 2.4. Skill Acquisition Phase

The goal of the skill acquisition phase is to learn a policy that leads the agent to the identified subgoal. Similarly to (Simsek et al., 2005; McGovern & Barto, 2001) the new option is determined on the basis of a pseudo reward function $\mathcal{R}(s)$ that returns a null reward in all states but the maximum interest state, where it returns a positive reward. The policy $\pi_{o^k}$ is the deterministic option that maximizes the value of the value function $Q^k(s, o)$, learned using the SMDP Q-Learning algorithm, where the reward function is $\mathcal{R}(s)$.

## 3. Experimental Activity

The experiment we discuss is a version of the Playworld proposed in (Stout et al., 2005). The Playworld is characterized by two rooms with a door in between, two panels, and a charger. The panels are in the room at left: the light panel switches the light on and off, while the door panel opens and closes the door. The robot perceives the light intensity, whether the door is open or not, its charge level, and its position (i.e., absolute coordinates and orientation). The robot is initially placed at random in the left room and the light is switched off. The robot can turn clockwise and counter clockwise and move ahead.

The experiment consists of two main stages: intrinsically motivated incremental learning and extrinsically motivated learning. In the first stage the robot explores the environment and develops new skills according to the self-motivated incremental process described in Sec. 2. The salient events we can expect the robot to find are: *light on*, *light off*, *open door*, *close door*, *charge.* The skills developed in early iterations (when the robot simply turns the light on and off) bias the random exploration so that the robot succeeds in activating new and more complex events (e.g., open the door and charge). This shows how SMILe enables the robot to autonomously discover interesting configurations in the environment and to develop self-motivation in learning new skills for achieving them.

In the second stage, the skills developed in the previous stage are tested in order to evaluate their level of reusability on five different goals, imposed by an external designer by providing an extrinsic reward function. In particular, we compare the performance of a robot that exploits the new skills to that of a robot that learns through Q-Learning, on five different tasks:
*Task1*: charge
*Task2*: charge, move to upper left corner of right room
*Task3*: charge, move to upper left corner of left room
*Task4*: charge, move to left room and close the door
*Task5*: charge, move to left room, close the door, switch the light off.

While *Task2* and *Task3* are not strictly related to any salient event, the other tasks require that the robot

achieves configurations relevant for the Playworld environment. Each 1,000 learning episodes, the extrinsic reward function is changed according to the task that must be accomplished and the learning robot should be able to adapt its policy to the new task without restarting the learning from scratch.

Fig. 1 shows the number of steps per learning episode. The first 2,100 episodes, labeled as *Self-Development* in the graph, represent the first stage of the experiment in which the SMILe robot autonomously identifies six different goals for which one new skill is learned at each iteration. On the other hand, in the first stage the Q-Learning robot does nothing, since no extrinsic reward is provided. The second stage starts with the introduction of a positive extrinsic reward for achieving the charger. While Q-Learning robot can only use the basic skills, SMILe robot exploits the skills learned in the first stage and succeeds in finding the optimal policy to reach the charger in less episodes than those needed by Q-Learning. Similarly, SMILe succeeds in exploiting its skills even for changing tasks, while Q-Learning took more time to adapt to new extrinsic reward functions.

Furthermore, in Fig. 2 we compare the total number of steps for both the algorithms and we report their difference. In the first stage, SMILe takes almost 250,000 steps to explore the environment and to learn the new skills, while no steps are taken by the Q-Learning robot. Notwithstanding the initial loss, the total number of steps needed by SMILe after the accomplishment of *Task1* is less than that of Q-Learning. The advantage of SMILe becomes even more relevant at the end of the second stage when Q-Learning took almost twice as much of steps than SMILe. This comparison suggests that SMILe, even though it requires potentially expensive exploration of the environment, leads to the development of useful skills that can be profitably reused in many different tasks. The number of steps saved during the extrinsically motivated learning stage is greater than the number of steps used in the self-development stage already after the first goal.

## 4. Conclusions

This paper introduced SMILe, a new intrinsically motivated learning framework that incrementally builds a hierarchy of skills independently from any given task in a three phase process. The experimental activity suggests that the task-independent skills learned by SMILe are general enough to be effectively reused to face different tasks defined on an environment with the same dynamics but changing reward function, thus significantly speeding up the learning process. Future
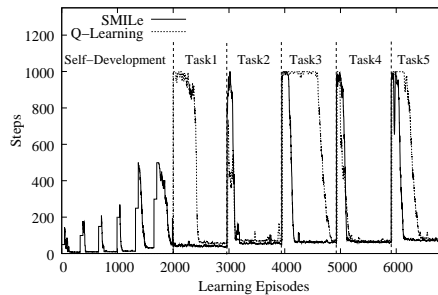


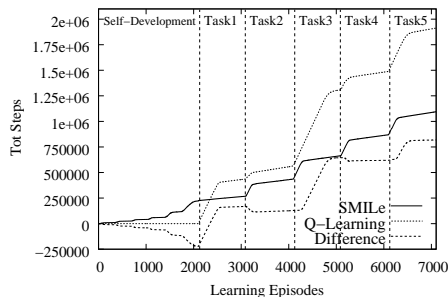*Figure 1.* Comparison of performance between Q-Learning and SMILe (mobile mean over 10 runs.)



*Figure 2.* Comparison of total number of steps between Q-Learning and SMILe.

works will address the problem of the development of skills that can be reused in environments with different dynamics, as in (Ravindran & Barto, 2003).

## References

Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, *13*, 341–379.

Mahadevan, S. (2005). Proto-value functions: Developmental reinforcement learning. *Proceedings of ICML.*

McGovern, A., & Barto, A. G. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density. *Proc. of ICML.*

Mehta, N., Natarajan, S., Tadepalli, P., & Fern, A. (2005). Transfer in variable-reward hierarchical reinforcement learning. *Proceedings of the 2005 NIPS Workshop on Inductive Transfer : 10 Years Later.*

Ravindran, B., & Barto, A. G. (2003). Relativized options: Cho osing the right transformation. *Proceedings of ICML.*

Simsek, O., Wolfe, A. P., & Barto, A. G. (2005). Identifying useful subgoals in reinforcement learning by local graph partitioning. *Proc. of ICML.*

Singh, S., Barto, A., & Chentanez, N. (2004). Intrinsically motivated reinforcement learning. *Advances in Neural Information Processing Systems*.

Stout, A., Konidaris, G., & Barto, A. (2005). Intrinsically motivated reinforcement learning: A promising framework for developmental robot learning. *AAAI Spring Symp. on Developmental Robotics*.

Sutton, R. S., Precup, D., & Singh, S. (1999). Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, *112*, 181–211.