Controlling an Agent by Focusing its Attention on Interactivelly Selected Patterns

Sébastien JODOGNE[+], Justus H. PIATER[*]

UNIVERSITY OF LIEGE, Montefiore Institute (B28), Sart-Tilman, B-4000 LIEGE
[+] *Research fellow of the Belgian National Fund for Scientific Research (FNRS)*
☎*: +32/4/366.26.43,* *: +32/4/366.26.20 &* *: S.Jodogne@ULg.ac.be*
[*] *Same address, but* ☎*: +32/4/366.22.79 &* *: Justus.Piater@ULg.ac.be*

1. INTRODUCTION

Designing robotic controllers can quickly become a challenging problem. Indeed, such controllers have to face a huge number of possible inputs that can be noisy (e.g. think of visual inputs as furnished by cameras), to select actions among a continuous set, and to automatically adapt themselves to evolving or stochastic environmental conditions. In real-world robotic applications, it is difficult to model the environment formally and to specify how to solve a given task directly in a programming language. On the other hand, living beings face such problems everyday, and are able to learn how to solve them efficiently. Some research works have therefore tried to mimic natural strategies for solving robotic tasks.

Neuropsychological evidence shows that the natural learning process is *interactive*: after each interaction with the environment, the agent gets a reward signal, called the *reinforcement*, which measures its performances. The goal of the agent is to maximize its reinforcements over time. As the agent observes the consequences of its reactions on the obtained reinforcements, it gains more and more expertise on its task, which ultimately enables it to act optimally. It is important to note that in such a setup, the agent is never directly told the optimal action to choose: there is no explicit external teacher. As an example, consider the task of a robotic agent escaping from a discrete maze: the reward could be zero until it manages to find the exit of the maze, in which case it obtains a positive reinforcement. Note that in the latter task, the interest of doing some action can appear only a long time after the interaction. This is called the *delayed reward problem*.

Those ideas gave rise to the algorithmic theory of *Reinforcement Learning* (RL) [1], the goal of which is to solve closed-loop adaptive control problems without a model by analyzing the reinforcements earned during a sequence of interactions. The major advantages of the RL protocol are that it is fully automatic, and that it imposes weak requirements on the environment. RL can be applied in any environment that obeys a discrete-time Markovian probabilistic dynamics, and upon which it is possible to define a *reinforcement function r(s,a)* that gives for each state *s* of the environment and each action *a*, a numerical measure of the worthiness of doing this action in this state.

2. INTERACTIVE SELECTION OF DISTINCTIVE PATTERNS

The price to pay for the flexibility of RL is poor performance when the input space of the agent is high dimensional or noisy. Since the rise of embedded CCD sensors, the problem has been exacerbated. Things get even worse when this variable space is continuous, which is for

example the case when haptic sensors are used. In such continuous cases, the traditional solution is to discretize the input domain. Unfortunately, this results in an explosion of the representational space known as the *curse of dimensionality*. This problem comes above all from the fact that traditional RL algorithms use the raw percepts, as furnished by the sensors.

However, to take the right decision, it is often sufficient for an agent to focus its attention on robust and highly informative patterns in its percepts, called *features*. If the agent is able automatically identify which high-level features are pertinent for its task, it can reduce a large, continuous or noisy input domain to a symbolic class in a preprocessing step, thus enhancing the rate of convergence as well as the robustness of RL to noise. For example, in the context of visual inputs, the agent could focus its attention on some patches of pixels located at *points of interest* in the image, i.e. at points at which discontinuities are detected in the visual signal [4]. To the best of our knowledge, the possibility of selecting such context-dependant features has not been studied in the RL literature.

This has motivated the introduction of a novel RL paradigm, *Reinforcement Learning of Classes* (RLC) [2, 3]. Concretely, RLC is built upon a two-level hierarchy: (i) a classifier translates the raw percept to a perceptual class by testing the presence of previously selected features, and (ii) the resulting class number is fed into a standard RL algorithm. Initially, the agent just knows about one class, so that all percepts are mapped to this class. Of course, this introduces a kind of *perceptual aliasing*: the optimal decisions cannot always be made, since percepts requiring different reactions are merged into the same class. The first difficulty behind RLC is therefore to isolate the aliased classes. Since there is no external supervisor, the agent can only rely on statistical analysis of the earned reinforcements. For instance, if the same action leads sometimes to a reward, and other times to a punishment, the agent is certainly "missing something" in the percepts corresponding to the class. Once some aliased class has been detected, the agent has to select a new feature that is *distinctive*, i.e. that best explains the variations in the reinforcements for the perceptual class.

Our approach is inherently incremental: new features are learned until no further aliasing is detected, and a feature is only included in the classifier once it has proven its usefulness for the task. This minimizes the number of selected features, which enhances the interpretability of the results, while reducing the redundancies between the generated perceptual classes. Furthermore, the list of the features that have been extracted by the agent furnishes an interesting feedback on the control problem by emphasizing the important elements in its terms. This is totally different from classical techniques for discretizing perceptual spaces [1], which suppose that the interesting features or their number are known in advance.

In order to turn this learning architecture into a working algorithm, two components are to be defined: (i) a robust *aliasing criterion* that is able to decide when the classification is not fine enough for the task, and (ii) a *class refining algorithm* that chooses the new features to include in the classifier among a possibly infinite set of features.

3. ALGORITHMIC CONTRIBUTIONS

We have defined two working RLC algorithms that can be distinguished by their aliasing criterion. The first algorithm, called *RLC through Sequences of Actions* [2], considers two percepts as aliased when it is possible to build a sequence of actions that gives rise to two different sequences of reinforcements. Since the criterion can only use the interactions that the agent has previously encounteered, it is in practice impossible to consider every possible sequence of actions. Rather, the agent chooses a sequence of actions for each perceptual class. Every time a perceptual class is met, either the choosen sequence of actions is executed, either a random action is taken in order to explore new parts of the input space. Periodically, the

agent chooses new sequences of actions. The considered sequences can either be built by randomization, or by using interesting trajectories in the system according to the information gathered on the task so far.

The main problem with this approach is that it puts strong requirements on the way the interactions are acquired: for instance, it prevents the use of a static database of interactions. It is particularly restrictive in robotic applications, in which collecting a representative set of experiments can be very expensive. Furthermore, it is impossible to know when the classes have been sufficiently refined. This has motivated the introduction of a second algorithm, called *RLC through Q Updates* [3], that considers a class as aliased when there exists some interaction mapped to this class such that its Bellman residual is different from zero. RLC through Q Updates overcomes the main deficiencies of RLC through Sequences of Actions.

Both algorithms share the same class-refining scheme that is borrowed from the algorithms for building decision trees in the field of machine learning. Sketchily, it consists in sorting the interactions according to some evaluation function. Then, for each cutpoint in the sequence of interactions, it creates two buckets of percepts (those relative to the interactions above the cutpoint, and the others), and it extracts the feature that maximizes some information-theoretic score for this partition in two buckets. Finally, the feature that has the maximal score among all the extracted features is used to refine the classifier.

These algorithms have successfully been applied to the problem of escaping from a discrete 2D maze with 54 cells, when the position *(x,y)* in the maze is not directly accessible to the agent, but implicitly encoded in the percepts by an image.

4. CONCLUSION

We have introduced a new class of RL algorithms that are able to focus the attention of the agent on distinctive and robust parts of the inputs, rather than on raw percepts. The abstraction level upon which RL is applied is therefore raised, since it allows the use of context-dependent information. The way pertinent features are selected is incremental and entirely interactive. Importantly, this selection is task-driven: different tasks will not necessarily lead to the selection of the same subset of features. This is similar to human visual learning, for which there is strong evidence that new visual classes are learned when the task requires it. In this sense, the paradigm of RLC can be motivated from a neuropsychological point of view, and not only as an *ad-hoc* RL algorithm. The area of applications is wide, since nowadays autonomous agents are often equipped with noisy or visual sensors. For example, one of our long-term goals is to construct a robotic hand that is able to automatically learn to use optical and haptic sensors to grasp objects.

REFERENCES

[1] D. Bertsekas and J.N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, 1996.

[2] S. Jodogne and J.H. Piater. *Interactive Selection of Visual Features through Reinforcement Learning*. in *24th SGAI Intern. Conference on Innovative Techniques and Applications of Artificial Intelligence*. 2004. Cambridge (UK). To appear.

[3] S. Jodogne and J.H. Piater. *Reinforcement Learning of Perceptual Classes using Q Learning Updates*. in *23rd IASTED Intern. Conference on Artificial Intelligence and Applications*. 2005. Innsbruck (Austria). To appear.

[4] C. Schmid, R. Mohr, and C. Bauckhage, *Evaluation of Interest Point Detectors.* International Journal of Computer Vision, 2000. **37**(2): p. 151-172.