In Berthouze, L., Kozima, H., Prince, C. G., Sandini, G., Stojanov, G., Metta, G., and Balkenius, C. (Eds.)
Proceedings of the Fourth International Workshop on Epigenetic Robotics
Lund University Cognitive Studies, 117, ISBN 91-974741-3-4

Intelligent Adaptive Curiosity: a source of Self-Development

Pierre-Yves Oudeyer
Frederic Kaplan
Sony Computer Science Lab, Paris

Abstract

This paper presents the mechanism of Intelligent Adaptive Curiosity. This is a drive which pushes the robot towards situations in which it maximizes its learning progress. It makes the robot focus on situations which are neither too predictable nor too unpredictable. This mechanism is a source of self-development for the robot: the complexity of its activity autonomously increases. Indeed, we show that it first spends time in situations which are easy to learn, then shifts progressively its attention to situations of increasing difficulty, avoiding situations in which nothing can be learnt.

1. An engine for self-development

Development involves the progressive increase of the complexity of the activities of an agent with an associated increase of its capabilities. This process is now recognized to be crucial for the formation of non-trivial intelligence (Brooks and Steins, 1994; Weng et al., 2001). Thus, one of the goals of developmental robotics is to build robots which develop. Several papers have already been written showing how one could increase the efficiency of the learning of a robot by putting it in situations in which the task to be solved was of increasing difficulty (e.g. Elman, 1993; Nagai et al., 2002). Yet, in all of them to our knowledge, the building of learning situations of progressive complexity was done manually by a human. As a consequence, the robots were not developing autonomously: they were involved in **passive development**. The goal of this paper is to present a mechanism which enables a robot to autonomously develop in a process that we call **self-development** (**active development** may also be used). This means that the robot is put in a complex, continuous, dynamic environment, and that it will be able to figure out by itself without prior knowledge which situations in this environment have a complexity which is suited for efficient learning at a given moment of its development.

This mechanism is called **Intelligent Adaptive Curiosity** (IAC):

- it is a **drive** in the same sense than food level maintenance or heat maintenance are drives, but it is much more complex. Instead of being about the maintenance of a physical variable, the IAC drive is about the maintenance of an abstract dynamic cognitive variable: **the learning progress**, which must be kept maximal.

- it is called **curiosity** because maximizing the learning progress pushes the robot towards novel situations in which things can be learnt.

- it is **adaptive** because the situations that are attractive change over time: indeed, once something is learnt, it will not provide learning progress anymore.

- it is called **intelligent** because it keeps the robot away both from situations which are too predictable and from situations which are too unpredictable (i.e. the edge of order and chaos in the cognitive dynamics). Indeed, naive curiosity algorithms like "go in most novel or in most unpredictable situations" might lead the robot to completely chaotic situations which can be dangerous and where nothing can be learnt (e.g. bumping very fast into walls and bouncing against it).

IAC is related to emotions and **value systems** (Sporns, 2000): indeed, it tags all situations as positive or negative with real numbers and accordingly to their learning progress potential, and makes the robot act so that it finds itself often in positive situations, which bring internal positive rewards.

The next section will explain in detail and in a practical case the concept of Intelligent Adaptive Curiosity. This new system is more general and much more robust than the initial system we proposed in (Kaplan and Oudeyer, 2003), as well as the system of (Schmidhuber, 1991).

2. Implementing Intelligent Adaptive Curiosity

2.1 Operationalizing the concept of learning progress

IAC involves the concept of learning progress, which must be operationalized. Typically, a robot possesses a prediction machine P which allows it to predict what is going to happen (S(t+1)) after taking action A(t) in the sensory-motor context SM(t):

$$P(A(t),SM(t))=S(t+1)$$

After making the prediction S(t+1) and taking the action, the robot can measure the actual consequence Sa(t+1), and compute its error in prediction:

$$E(t) = \text{absoluteValue}(S(t+1) - Sa(t+1))$$

Along with time the robot gets a list of successive errors: LE = E(0), E(1), E(2), E(3), ..., E(t). This list can then be used to measure the evolution of the mean error in prediction, based on a sliding window which computes the mean of the last DELAY errors, which gives a new list: LEm = Em(DELAY), Em(DELAY+1), Em(DELAY+2), ..., Em(t) (with Em(t) = mean of E(t), E(t-1), ..., E(t-DELAY)). We can then propose a first definition of the learning progress at time t as

$$LP(t) = -(Em(t) - Em(t\text{-}DELAY))$$

In the set-up presented in the next section, DELAY=150. In order to be able to evaluate the value of a potential situation at time t+1 in terms of learning progress, the robot must also be able to predict E(t+1), so that it can evaluate Em(t+1) and then LP(t+1). For doing this, it also possess a meta-learning machine MP which learns to predict the errors of P:

$$MP(A(t), SM(t)) = Ep(t+1)$$

(and the learning is done when it gets the actual error E(t+1)).

The algorithm is then quite straightforward: at each time step t, for choosing the action to take, the robots makes a list of possible actions and evaluates each of them in terms of the learning progress it may bring. If the space of action is continuous, then it is sampled randomly. For the evaluation of each action Ai(t), the robot computes MP(Ai(t), SM(t))=Epi(t), then it computes the associated learning progress LPi(t), and chooses the action for which the predicted learning progress is maximal. The robot also can choose a random action with a probability of 0.1 (this allows to discover new niches of learning progress). Indeed, if we view learning progress as an internal reward to be maximized, we are here in a classical problem of reinforcement learning (Kaelbling et al., 1996) where the trade-off between exploitation and exploitation has to be solved. We chose here a simple policy to manage this trade-off, since this aspect is not the topic of the paper, but more complex policies could be used as well.

Now, if one try to use this algorithm right away as in (Kaplan and Oudeyer, 2003), this will not lead the robot towards the desired result (i.e. self-development). Indeed, in most environments, the robot will find a behaviour which will provide high values of LP(t), but will be very inefficient in terms of learning and will not provide the ability to develop. This behaviour is the alternation between very simple and predictable situations in which the error rate is low, and complex unpredictable situations in which the error rate is high. Indeed, passing from the later to the former produces a decrease of the error rate, and is evaluated as learning progress! For example, this would make a mobile robot spends its time bumping randomly against walls (where the error rate of predicting what's going to happen after the bump is high) and then stopping in front of the wall, looking at it (the image and the position are fixed and easy to predict).

So there needs to be a more "intelligent" definition of learning progress. The idea is that instead of comparing the mean error in prediction between situations which are successive in time, we will compare the mean error in prediction between situations which are similar (so with the example above, when the robot is stopped in front of the wall, and has a low error rate, it will compare this error rate with the last time it was stopped in front of the wall, and not with the previous situations in which he was bumping into the walls. Now, the error rate will be in both comparable situations very low, and so no progress will be predicted). We will now present a way to implement this calculation.

The machine P of the robot is now composed by a set of experts which are specialized in particular zones of the sensory-motor space, i.e. in particular kinds of situations. Each expert possesses a set of training examples, and each training example is possessed by only one expert. This set of examples is used to make predictions (here we use the nearest neighbours algorithm). At the beginning, there is only one expert. As new examples are added, a criterion C1 is used to decide whether this expert should be split into two experts. Here, for simplicity and for computational efficiency, the criterion is: split when the number of examples is above a threshold set to NS (here NS=250). Now, there is a second criterion C2 which decides how the expert is split into two experts, i.e. how the set of examples is split in two parts which will be inherited by the new experts. The criterion C2 consists in finding a dimension to cut, and a cut value, to minimize the variance of the examples (in the output space) inherited by the experts while keeping the size of the two example sets relatively

balanced. Each of the new experts are then splitted themselves recursively when the criterion C1 is realized. At each splitting, the experts keep the cut values which were used to create them, as well as the cut values of their parent expert. When a prediction has to be made given A(t) and SM(t), one first determines which is the expert which is a specialist for the zone of the sensory-motor space which contains this point. This is simply done by using the successive cut values stored by the expert.

Additionally, each expert i keeps track of the prediction errors it has made, and computes the mean error in prediction in the last DELAY examples it has evaluated (here, DELAY=150), noted Empi(t). With this, curve, it then computes the variation of this mean over the last DELAY time steps:

$$\text{LPi(t)} = -(\text{Empi(t)} - \text{Empi(t-DELAY)})$$

This value is the way the robot now evaluates its learning progress. As it is local and averaged, the last computed value is used to predict the next learning progress value. This works very well in practice.

2.2 Evaluating the algorithm in a simulated robotic set-up

We use here a robotic simulation implemented with Webots (Webots, 2004). The robot is a box with two wheels (see figure 1). Each wheel can be controlled by setting its speed (real number between -1 and 1). The robot can also emit a sound of a particular frequency. So the action space is 3-dimensional and continuous: (left speed, right speed, frequency). The robot moves in a square room with walls. There is a toy in this room that can also move. This toy moves randomly if the sound emitted by the robot has a frequency belonging to zone f1=[0;0.33]. It stops moving if the sound is in zone f2=[0,34;0,66]. The toy jumps into the robot if the sound is in zone f3=[0,67;1].

The robot perceives the distance to the toy and to the walls with infra-red sensors. Its prediction machine P tries to predict the next distance to the toy, given the current sensory-motor context and an action to take (i.e. a setting of (left speed, right speed, frequency)). Using the IAC algorithm, the robot will thus act in order to maximize its learning progress in terms of predicting the next toy position. The robot has no prior knowledge, and in particular it does not know that there is a difference between setting the speed of the wheels and setting the sounds frequency (for the robot, these are unlabeled wires). It does not know that there exist three zones of the sensory-motor space of different complexities: the zone corresponding to sounds in f1, where the distance to the toy can not be predicted since it is random; the zone with sounds in f3, where the distance to the toy is easy to learn and predict (always around 0); and the zone with sounds in f2, where the distance to the toy is predictable (and learnable) but complex and dependant of the setting of the wheel speeds.

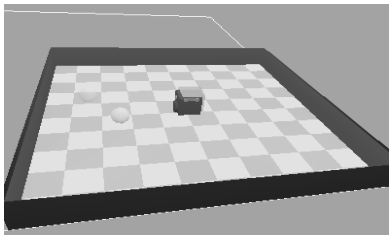

Figure 1: The robotic set-up : a two-wheeled robot moves in a square room and there is also an intelligent toy (represented by a sphere) which moves according to the sounds that the robot produces. The robot has wall sensors and perceives the distance to the toy. The robot tries to predict the position of the toy after performing a given action, which is a setting of (left wheel speed, right wheel speed, sound frequency). He chooses the actions for which it predicts its learning progress will be maximal.

Yet, we will now show that the robot manages to autonomously discover these three zones, evaluate their relative complexity, and exploit this information for organizing its own behaviour.

Figure 2 shows a representation of the set of experts (and their respective error rate) which were built during a simulation. The graph appears as a tree. Each leaf of the tree corresponds to one expert, and each path from the root of the tree to a leaf shows the evolution of the mean error in prediction of the leaf expert. This path is partly shared by experts which have the same expert ancestor, which explains the tree structure (for example, at the root of the tree, there is only one path which corresponds to the error rate of the initial expert, and then when this expert is split, two paths appear, each corresponding to a children expert). This figure shows that there are basically three groups of experts that are formed. After looking in detail at their structure, we discovered that each group of experts was self-coherent: one of them corresponded to situations with sounds in f1, another with sounds in f2, and another with sounds in f3. The robot first made a split between situations with sounds in f3 on the one hand, and situations with sounds in f1 and f2 on the other hand. It discovered that initially the situations with sounds in f3 were the most important sources of learning progress. But as they were also very easy to learn, quickly the learning progress decreased in these situations. The robot then used its splitting of the situations with sounds in f1 and f2 into situations with sounds with f1 on the one hand, and situations with sound f2 on the other hand. This splitting was done thanks to the random action selection that the robot sometimes made while explor-

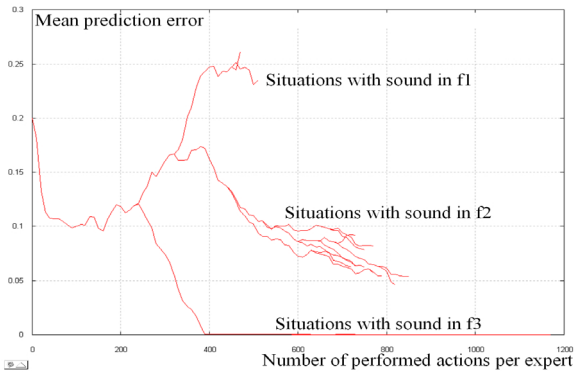

Figure 2: Representation of the tree of experts created in a simulation with the IAC action selection algorithm.

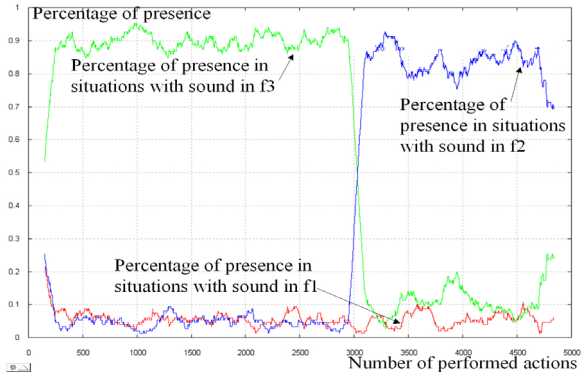

Figure 3: Evolution of the percentage of time spent in each of the three situations.

ing situations with sound f3. The robot then accurately evaluates that the situations with sounds in f2 provide the maximal amount of learning progress, while the situations with sounds in f1 are chaotic and provide no significant learning progress. The robot then begins to focus its actions in the zone of the sensory-motor space where the sound is in f2. This sequencing of behaviours of increasing complexity is represented on figure 3 where each curve represents the percentage of time (at given periods of the simulation) that the robot spends in each of the three kinds of situations.

It then tries to vary its motor speeds within this sub-space with sounds in f2, learning to predict how these speeds affect the distance to the toy. We have obtained also results showing that this exploration of the motor speed subspace is organized, and that the speed settings are themselves done optimally so as to maximize the learning progress. We will present these results in a future paper.

This shows a crucial result from a developmental robotics point of view : the IAC algorithm allows a robot to autonomously scale its behaviour so that it explores sensory-motor situations of increasing complexity and avoids being trapped exploring situations in which there is nothing to learn (here the robot always minimized the time spent emitting sounds in f1, where nothing could be learnt). The algorithm is also a way to implement a kind of attention mechanism (the robot focuses first systematically on one kind of situations, and then focuses systematically on another kind of situations).

3. Conclusion

We showed how the IAC algorithm allows a robot to autonomously scale the complexity of its learning situations by successively and actively focusing its activity on problems of progressively increasing difficulty. This is achieved through the efficient and operational discovery of zones of the sensory-motor space of varying complexity, then through the accurate evaluation of the relative complexity of these zones in terms of potential learning progress, and then through the exploitation of this information by the robot to self-organize its behaviour. To our knowledge, IAC is thus the first method which allows a developmental robot to go through all these steps autonomously and without prior knowledge.

4. References

R.A. Brooks and L.A. Stein. Building brains for bodies. Autonomous Robots, 1(1):7-25, 1994.

Elman, J.L. (1993). Learning and development in neural networks: The importance of starting small. Cognition, 48, 71-99.

Kaelbling, L.P., Littman, M.L., and Moore, A.W. (1996) Reinforcement Learning: A Survey, Journal of Artificial Intelligence Research, Volume 4, pages 237-285.

Kaplan, F., Oudeyer, P-Y (2003) Motivational principles for visual know-how development, In Prince, C.G. and Berthouze, L. and Kozima, H. and Bullock, D. and Stojanov, G. and Balkenius, C., editor, Proceedings of the 3rd international workshop on Epigenetic Robotics : Modelling cognitive development in robotic systems, no. 101, pages 73-80, 2003. Lund University Cognitive Studies.

Y. Nagai, M. Asada, and K. Hosoda. Developmental learning model for joint attention. In Proc. of 15th Int. Conf. on Intelligent Robots and Systems (IROS 2002), pages 932-937, 2002.

Schmidhuber J. (1991), Curious Model-Building Control Systems, in Proc. International Joint Conference on Artificial Neural Networks, Singapore, volume 2, pages 1458-1463, IEEE.

Sporns, O. (2000). Modeling development and learning in autonomous devices. In Workshop on Development and Learning, pages 88-94, E.Lansing, Michigan, USA

Weng J., McClelland J., Pentland A., Sporns O., Stockman I., Sur M., Thelen E. (2001) Artificial Intelligence: Autonomous Mental Development by Robots and Animals, Science 291: 599-600

Webots (2004), www.cyberbotics.com.