



HAL
open science

On bootstrapping sensori-motor patterns for a constructivist learning system in continuous environments

Sébastien Mazac, Frédéric Armetta, Salima Hassas

► **To cite this version:**

Sébastien Mazac, Frédéric Armetta, Salima Hassas. On bootstrapping sensori-motor patterns for a constructivist learning system in continuous environments. *Alife 14: Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, Jul 2014, New York, United States. 10.7551/978-0-262-32621-6-ch028 . hal-01135987

HAL Id: hal-01135987

<https://hal.science/hal-01135987>

Submitted on 26 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On bootstrapping sensori-motor patterns for a constructivist learning system in continuous environments

Sébastien Mazac^{1,2}, Frédéric Armetta² and Salima Hassas²

¹ubiant, Lyon, France

²Université Lyon 1, LIRIS, UMR5205, F-69622, France
sebastien.mazac@liris.cnrs.fr

Abstract

The theory of cognitive development from Jean Piaget (1923) is a constructivist perspective of learning that has substantially influenced cognitive science domain. Indeed it seems that constructivism is a possible trail in order to overcome the limitations of classical techniques stemming from cognitivism or connectionism and create autonomous agents, fitted with strong adaptation ability within their environment, modelled on biological organisms. Potential applications concern intelligent agents in interaction with a complex environment, with objectives that cannot be predefined. There are numerous interesting works in developmental robotics going in this direction. In this work we investigate the application of these principles to a close domain: Ambient intelligence, which is extremely challenging but which also presents interesting aspects to exploit, like the participation of human users. From the perspective of a constructivist theory, the learning agent has to build a representation of the world that relies on the learning of sensori-motor patterns starting from its own experience only. This step is difficult to set up for systems evolving in continuous environments, using raw data from sensors without a priori modelling, primarily because they face a bootstrap problem. In this paper we address this particular issue and propose a decentralized approach based on a multi-agent framework, where the system's representations are constructed through a self-organization process that handles the dynamics between experience discretization and learning.

Introduction

Embodiment is a paradigm in cognitive science and AI considering that *cognition* arises from the *interaction* between an *agent* and its *environment*. Contrary to computational theory of mind, an agent has to be embodied and situated, and does not have a predefined *representation*. On the other hand an agent (for example a robot) must be able to construct its own representation of the regularities through its interaction with the environment, whatever its body's capacities and the environment where it is situated. More generally, embodiment is embraced by constructivist theory of cognition, as explained by Ziemke (2001). Constructivism in cognitive science is bound to constructivist epistemology. Particularly Riegler (2001) explains why the *radical con-*

structivism proposed by Von Glasersfeld (1984) is necessary to go further in the understanding of cognition mechanisms. There is a growing interest in AI for constructivist theories, for example such as *Enaction* (Varela et al. (1991)) stemming from the biology field, or constructivist theory of learning (Piaget (1954)) coming from the psychology field. While it is true that constructivist approaches of learning in AI are usually still theoretical, there are also more practical domains that take inspiration from these concepts such as developmental robotics. This article addresses some fundamental issues related to the application of constructivist learning methods to continuous environments. In the following, we will first introduce constructivist methods for artificial learning and the issues related to real application such as in an ambient intelligent system. In this article we focus on the problem of bootstrapping the learning of sensori-motor regularities starting from raw signals, which is a first step in the more general model of a constructivist learning. Then we will present our approach to this problem, based on a decentralized approach using a self-organizing multi-agents system, and provide some preliminary results.

Research issues and related works

Before going any further, let us consider an illustrative example of the addressed research problem. Unknown sensors $\{s_1; s_2 \dots s_n\}$ and actuators $\{a_1; a_2 \dots a_n\}$ are connected to an ambient intelligent system S . S is considered as an autonomous agent that must be able to learn the effects of, for instance, a_x on s_y and s_z . If S receives an objective from the user that involves one of these perceptions/actions, it should be able to take the knowledge of those interactions into account. For example, let us say that a_x is a wireless actionable plug with an unknown device D connected on it. If D is an electric heating, S should become aware that it can affect both temperature (s_y) and energy consumption (s_z) by switching a_x , whereas it can affect luminosity and energy consumption if D is a lamp. These are simple regularities that may seem trivial to learn. This is true if the system's designer defines an a priori representation, for example the definition of events and of how to manage time

relations. By doing so, a learning agent may directly focus on what is relevant, and find attended regularities out of a reduced search space. Here we consider the case where the agent has no a priori representation and must both discover how to grasp raw signals and how to learn from them at the same time.

Continuous environment and the discretization process

In *real world* environments, if we want to avoid an a priori definition of a model, the system has to deal with the complexity of raw data coming continuously from sensors. The *perceptual aliasing problem* is one of the main issues: the learning agent has to learn precisely what are the boundaries of events both in time and space. But the task of separating an event out of the thread of experience is quite difficult. A human brain seems to be able to focus on the relevant details of experience in order to identify a situation and ignore anything else. In classical AI, particularly with logic based symbolic representation, the exhaustive characterization of a situation is impossible in large environments (*Frame Problem*), most importantly when considering a large buffer memory of past events. More generally, the question of dealing with the complexity of the real world without using a model or symbolic representation is also known as the *Moravec's paradox*. This paradox emphasizes the fact that basic sensorimotor skills of living organisms are more difficult to reproduce artificially than high level cognitive abilities, as argued by Brooks (1991). Handling continuous experience is a particularly significant issue for constructivist learning methods, related to their application to real world problems.

Developmental robotics

The field of developmental robotics proposes interesting works in accordance with the constructivist principles presented above and is confronted with the problem of handling continuous environments (see Lungarella et al. (2003) for a review). This research field stems from the assessment that biological organisms go through a period of development before reaching their final form. One of the main ideas is that learning must be facilitated by the progressive increase of the complexity of the environment perceived by the agent while progressively acquiring highly tuned skills. Indeed, at the beginning of his growth, an infant has limited sensor and motor abilities, as well as limitation in his nervous system. At first sight this may be considered as a defect, but we can also consider that it is an advantage regarding the learning process, since it enables to decrease or prevent any data overload, that would not be useful for the first development steps. If the agent is precociously facing a huge quantity of information, learning can be intractable. Thus the management of the development of an artificial agent during the learning process could be an improvement. In developmental robotics, the programmer does not specify a

targeted task; the robot must freely explore its environment by interaction. For example, an *intrinsic motivation* could be expressed in the robot by a certain curiosity which leads it to focus on interesting things, like situations which are neither too predictable nor not enough, as proposed for example by Oudeyer et al. (2007). The capacity of making predictions is a part of the mechanism of development, from which the cognition is built. Cognitive functions of higher level, such as planning, may be seen as a prediction of series of events based on experience. Thus, in continuous environments the definition of the notion of *event* is crucial. Without pre-given representation, the agent has to learn what an event is, and at the same time it uses events to learn patterns. Before presenting a possible application of such a developmental approach to an ambient intelligent system, it is worth introducing more precisely the constructivist learning paradigm in AI and works related to the problem of bootstrapping regularities from raw experience.

Constructivist learning

In psychology the constructivist theory especially developed by Piaget (1954) postulates that a learning agent constructs its own representation of its world, which it uses to give a meaning to its experience (*assimilation*). Learning, therefore, is simply the process of adjusting its mental models to fit in new experiences (*accommodation*). Following Drescher (1991) who takes inspiration from the work of Piaget to propose a model of constructivist learning, many researchers work on the *schema learning* technique. In this model the pattern of interaction used to build the representation of the agent is called a *schema*. It consists mainly in three elements: a *context*, an *action* and a *result*. The meaning of a schema is: if the agent executes the action *A* in the context *C*, it predicts the result *R*. Schemas are learned incrementally while the agent performs random actions, in the manner of an infant fumbling. A schema is an interesting structure for a representation since it relates an action to its consequences in an environment, based on agents experience. The learning is not goal-oriented, contrary to classical reinforcement learning for example, where the agent learns which action to execute in which context, depending on the purpose of an external learning task. Here, the agent predicts the consequences of its action, and this knowledge may be used for different purposes. The learning model of Drescher (1991) is very interesting because it offers to agents the possibility to learn abstract concepts grounded in experience thanks to the creation of *synthetic items* (see Perroto et al. (2010)). However, this model operates on a discrete environment and is quite resource consuming, so that it cannot be applied directly to continuous environments. Also many other research in constructivist learning methods in AI provide valuable results but are often designed for simplified simulated environments (see Guerin (2011) for a survey).

Related works

There are works that propose solutions to overcome the problem of handling continuous environments. For example we can mention Chaput et al. (2003) who advises to use a hierarchy of self-organizing maps (SOM) to improve the schema mechanism of Drescher. In their experiment the environment is represented by binary vectors and the time is managed in a step-based manner, but they argue that this improvement may enable constructivist learning for realistic environments. Further work of Provost et al. (2006) steps in this direction with a system that couples a SOM based discretization process of environment with reinforcement learning for a robot navigation problem.

Linaker and Jacobsson (2001) propose a system that features an event extraction mechanism (classification system) which converts raw multi-dimensional and time-step based sensor data into series of events of a larger time scale. This discretization of raw data into events enables to learn a policy of action on a higher level with a reinforcement learning mechanism. It is clear that the low level event extraction mechanism will drastically influence the other parts of the learning system. The major problem for such an approach lies in the fact that the classification process is unique and defined a priori. Yet, the possible classes generated by the discretization mechanism may differ according to the different patterns in which the variables are involved. A solution to this problem is to let the discretization process evolve, guided by a feedback from the higher levels of learning, so that the classes are always adapted to the learned pattern.

This is exactly the case in the model of Mugan and Kuipers (2007) who propose an event extraction mechanism coupled with a learning system of predictive rules. Events are extracted from the continuous experience as transitions between states that are not predefined. Then predictive rules are learned in order to associate an event to an other. An interesting aspect of this system comes from the feedback provided by the learning system to the event extraction mechanism that enables to create new states, and thus learn refined correlations. Thus, the agent learns to perceive in a way that enables to learn more efficiently. This work enhances the developmental process that interconnects the discretization of experience with learning. However, each aspects of this process (the discretization algorithm, the predictive rules,...) are particular implementations that suit with the specific problem considered. In the following of this paper, our purpose is to make explicit the fundamental components of such a process, and propose a generic model that allows multiple implementations of each aspect, motivated by handling heterogeneous unknown environments.

Model presentation

Learning following a constructivist approach is a promising trail in order to build autonomous agents strongly adapted to their environment. In particular, since there is no assumption

about the representation and the objectives of the system, it suits well the problem where both the agent's sensori-motor abilities and its objectives are initially unspecified. Ambient intelligent systems for example present these characteristics. Like robots, they are considered as autonomous systems composed of sensors and actuators. Particularly for AmI systems, constructivist learning is a possible solution to face the fact that sensori-motor abilities are unknown and evolve over time (a user may add, move or remove sensors and actuators at anytime). Moreover, the large variety of equipment results in countless combinations that make harder the design of an expert representation. In the previous part we mentioned the difficulty to handle continuous environments for constructivist learning methods, in this section we propose a solution to deal with this problem, with the example of an ambient intelligent system.

Description of the problem

Developmental learning relies on agent-environment interactions, that is to say on *signals* exchanged at the frontier between the agent and its environment. In this model this frontier is represented by a set of numeric continuous variables which represents sensors and actuators $V = \{v_1; v_2; \dots; v_n\}$. As illustrated on Figure 1 (a), these variables evolve in time according to their respective allowed values. Let us denote $v_i(t)$ the value of v_i at time t . This continuous flow of signals is interpreted by the agent as an *event* sequence. In the following we call *experience* $E = \{e_1; e_2; \dots; e_n\}$ the set of events constructed by the agent. We assume that there are some regularities in this experience. A regularity can be represented as a pattern that the agent is able to isolate within its experiential space and identify thanks to its recurring nature.

From the point of view of the learning system, there is no difference between variables connected to actuators and variables connected to sensors. So the patterns learned by the system might indifferently involve actions and perceptions. However, it is clear that some actions have to be performed in order to appear in some patterns. Contrary to robotics, the domain of ambient intelligence does not really allow us to let the agent randomly explore its possible interactions with its environment in the first place. A motor babbling behavior (see Mugan and Kuipers (2007)) might be lengthy and its consequences would definitely constitute a nuisance for users, for obvious reasons. Hopefully, an ambient intelligence system presents an interesting property that we can exploit to face this problem: a user is able to directly perform actions. For safety and ethical reasons, a user must always keep control of its environment. So as a design constraint, we consider that for every action that can automatically be performed by the agent, a user should have the possibility to directly perform it. We make the hypothesis that the agent is always able to perceive such an action performed by a user, as a kind of *proprioception*¹, so that if this action

¹the sense of internally perceiving our own movements

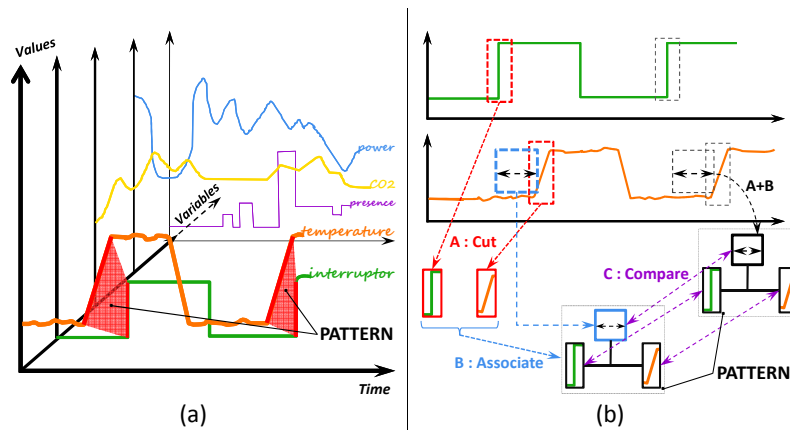


Figure 1: (a) The continuous experience of the system expressed as a set of variables. (b) The 3 types of operations that enable the construction of patterns. *A*: Find a temporal reference in order to perceive a variation. *B*: Find associations between variations. *C*: In order to reinforce existing structures, modify them or create new ones, it is necessary to be able to compare elements of representation.

becomes involved in a pattern, the agent would be able to perform it by itself later. Najjar and Reignier (2013) refer to this as an *observed action*. Thus the initial random motor babbling is replaced by actions driven by the user, which get involved at the very beginning of the learning process as a form of *motivation* for the developing agent.

Constructing patterns and the bootstrapping problem

A pattern is relevant if it contributes to the creation of new patterns, so that to enable the construction of a representation which justifies them. In other words, the agent has to learn how to perceive efficiently in order to be able to learn more. This circular definition raises a bootstrapping problem similar to the one mentioned by Drescher (1991): finding a context to characterize the association between an action and a result requires to know formerly this association, but learning this association is difficult without knowing the context. This kind of *teleological* issue is typical of the study of living systems.

The purpose of the elementary structure of representation is to express the primary kind of regularity that the agent is able to learn in the first place. For example *schemas* (Drescher (1991)) express a logical and temporal relation between three elements. The *result* is likely to be experienced after the *action* if the *context* was experienced initially. But starting with a set of continuous variables as experience, this raises many questions. What are precisely a context, an action and a result? What are the duration of these elements and their relationships? It seems that these notions are already relative to a form of representation. Of course this issue is avoided if there is an a priori abstraction performed by the designer of the system in order to discretize the environment and guide the learning process.

On the other hand one may consider that the schema or any similar kind of structure like the *functional circle* from Von Uexküll (1992), constitutes an undividable elementary structure that is intrinsic to the agent. Thus, Georgeon and Aha (2013) define the *sensori-motor interaction* as such a primitive. This is reminiscent of the *subsumption architecture* from Brooks (1991), but such methods require to formerly implement such functional circles, and thus to know precisely the agent and its environment. Moreover, this preliminary hardcoding of elementary behaviors may be laborious in complex fields such as ambient intelligence systems.

Apart from specifying arbitrary initial actions and perceptions, one way to solve this problem is to consider a developmental approach. At the beginning of the learning process the agent is fitted with rough sensori-motor abilities which enable it to learn primary patterns, then its abilities develop and the learning of a more accurate representation becomes possible. At one point, the already learned representation can guide the development of the abilities. In biological organisms this task is most probably realized both during phylogenesis and ontogenesis, but it needs to be expressed as a single problem for artificial systems. The "bootstrap learning" as presented in Kuipers et al. (2006) corresponds to this critical problem of getting primary patterns from raw data of uninterpreted sensors (see also Guerin (2011) for a review).

In our model, the elementary structure of representation is composed of two generic notions of *events*. First, a *perception* event is a selected moment in the evolution of a variable, for example a variation or a stable state. Second, an *association* event describes a relation between two events, for example a duration between the occurrences of two perceptions. So it is possible to express a *schema-like* regularity with a combination of association and perception events.

More generally, events involved in associations may be associations, therefore enabling the construction of more complex patterns.

A multi-agent system for the construction of patterns

Our proposal is based on a self-organizing multi-agent system (MAS), where three populations of agents² will interact with each other, guided by a feedback from the global system activity in order to construct relevant patterns and provide a grounded representation. The multi-agent architecture suits naturally the view considering the construction of representation as a self-organization process, as well as the heterogeneity of the domain, and the necessity for the system to be resilient. In the MAS, agents perform a function corresponding to their role in the system, that they can change to explore the possibilities to construct a representation. Rather than specifying an a priori model of perception, or using an exhaustive search, this model relies on the elaboration of collaboration, reinforcement, and internal dynamics within the system. Related to the generic notions of perception and association, we identify three primary functions that are necessary to bootstrap the learning of elementary patterns from the continuous experience as illustrated in Figure 1 (b).

- The first operation "cut" corresponds to the creation of a perception event. The set of possible *cut* functions is $F^c = \{f_1^c; f_2^c; \dots; f_n^c\}$. A *cut* function that transforms values into events is defined as

$$f_i^c : v_i \rightarrow \{e_1^c; e_2^c; \dots; e_n^c\}$$

$$V \rightarrow E^c$$

with E^c the set of perception events.

In the multi-agent system, *cut* functions are implemented by *Perception agents* which are connected to a continuous variable. These agents create events that represent a particular time period of the evolution of the variable. For example, a *Temporal Window Agent* may produce perceptions based on sliding windows of a given duration. In this case a parameter that conditions the behavior of the agent could be the duration of the window. If the parameter evolves due to feedback received by the agent, it corresponds to the exploration of a new *cut* function.

- The second function type, *compare* functions ($F^s = \{f_1^s; f_2^s; \dots; f_n^s\}$) which serve to evaluate the similarity between two events, is necessary to classify and manipulate

²In the following we will refer to the above-mentioned autonomous learning agent as "the system" in order to avoid any ambiguity with the MAS's agents.

events. A *similarity* function compares events to determine if they are similar, and is defined as

$$f_i^s : e_x, e_y \rightarrow s$$

$$E \rightarrow [0, 1]$$

where e_x, e_y are created by the same function ($s = 1$ means totally similar, $s = 0$ totally different).

On Figure 2 we present an example of the activity between a *Perception agent* and a *Similarity agent*. A threshold is defined for s in order to decide the similarity and enable classification (e.g. 0.9). For example, the *temporal window agent* creates an event *ec1* which contains the variations compared to the average value of the sliding window. Later another event *ec2* is created by this agent and may be considered as similar or different depending on the behavior of the similarity agent involved. If the similarity agent implements a general function f_1^s (intervals of size 3), the two events look similar (3 occurrences in each interval), whereas if it implements a more specific function f_3^s (intervals of size 1) the events are perceived as different.

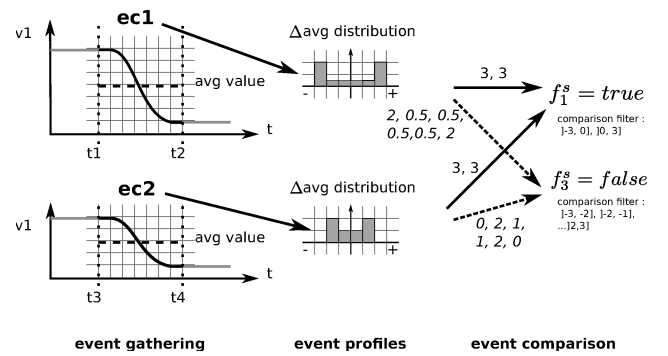


Figure 2: Interaction between a perception agent and a similarity agent.

- Finally the last functions are *associate* functions ($F^a = \{f_1^a; f_2^a; \dots; f_n^a\}$) leading to the creation of association events. An *association* function that creates an event from two other events is defined as

$$f_i^a : e_x, e_y \rightarrow e_z^a$$

$$E \rightarrow E^a$$

($E = E^c \cup E^a$). For example a *Binary Duration Agent* is defined for two *source events* e_1 and e_2 , and creates an association event for every occurrence of e_1 , characterized by the duration with the previous occurrence of e_2 .

A *Similarity Agent* is associated with a perception or an association agent, in order to classify events created by these agents. For instance, on Figure 3, events are created from

the activity of variables v_1 and v_2 by agents which implement functions f_1^c and f_1^c . Different occurrences of e_1^c and e_1^c , (source events) are classified by similarity agents. An association agent creates an event e_1^a based on the occurrences of events e_1^c and e_1^c , and then creates another event e_2^a later when new occurrences e_2^c and e_2^c are perceived as similar to e_1^c and e_1^c . Another similarity agent implementing a function f_2^s compares the association events generated by the association agent f_1^a in order to classify them (as presented on Figure 5).

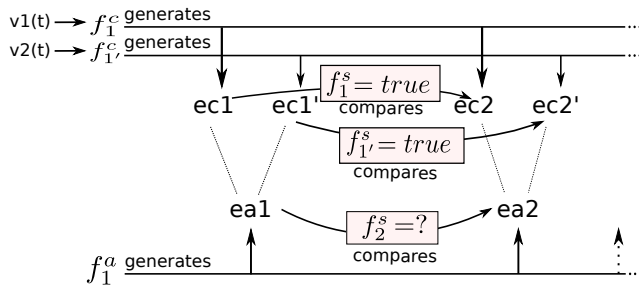


Figure 3: Interactions between the three types of agent.

The evaluation of representation and feedback process

The search space of the system is the set of possible functions $F = F^C \cup C^A \cup F^S$. The relevance of a function can be evaluated according to its participation to the construction of patterns, which is necessarily performed in collaboration with other functions' types. We can represent this feedback by a fitness function $Q : f \in F \rightarrow [0, 1]$ that defines a function's quality. Then the goal of the system is to find a combination of functions that enables to construct relevant patterns. In other words select the optimal set of functions that maximize the overall quality evaluation. This can be defined as $F' \subset F = \{f' \in F' | Q(f') > T_Q\}$ where T_Q is a threshold for the fitness function.

As illustrated on Figure 4, the mechanism is a circular process initiated by the creation of representation elements from the experience discretized by the *cut* operations. Each time a new element is created, it is compared with previously created elements coming from the same source, and it is fused with a similar existing element or constitutes a new distinction, to form classes of events. Then events are used to create new events with the *associate* operations, which in turn are compared and so on. These generic functions and the associated structures enable to model a basis for the mechanism of extracting sensori-motor patterns, but there are many ways to instantiate each of them. A given version of a function may suit well the learning of a specific aspect of the system's experience at a certain level, and fail to describe another one. So in this model these different possibilities are processed in parallel, and the best ones are

reinforced because they enable the construction of a stable representation. Each instance of a function (an agent) has to evaluate if it contributes to a viable representation or not.

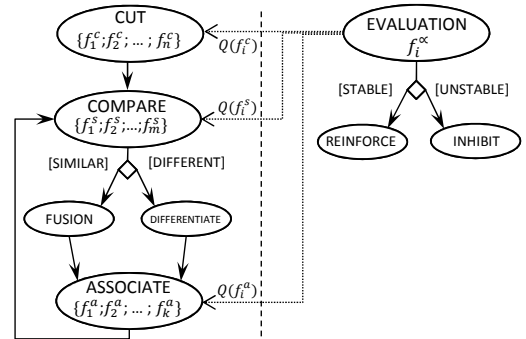


Figure 4: Process of bootstrapping pattern construction according to three types of functions.

The evaluation of patterns may correspond to their ability to make reliable predictions, as used by Mugan and Kuipers (2007). Indeed the adequacy of the representation is tied to the notion of prediction as stated by Von Glasersfeld (1984):

"Quite generally, our knowledge is useful, relevant, viable, or however we want to call the positive end of the scale of evaluation, if it stands up to experience and enables us to make predictions and to bring about or avoid, as the case may be, certain phenomena (i.e., appearances, events, experiences). If knowledge does not serve that purpose, it becomes questionable, unreliable, useless, and is eventually devaluated as superstition."

But in the case of the bootstrapping problem, it is not possible to evaluate the prediction abilities of patterns until we have patterns to consider as a predictive structure. Thus, we introduce a transitional evaluation of *interest*, in order to drive agents to promising areas of the search space and use predictions as a more important feedback in a second step. The interest of primary events is defined according to the notions of specificity and stability. If functions are much too accurate, the classes of events are likely to multiply without being reinforced, and will not constitute regularities. On the contrary, if the operations are much too general, the representation might be quite stable but also unexpressive. Therefore, in order to give feedback to agents, the representation must be evaluated in terms of a tradeoff between accuracy and stability. This intuitive idea is used in order to overcome the bootstrapping problem. Indeed, it enables to guide the activity of the initial rough functions, before first patterns are formed. The specificity of an event can be expressed in relation with a *reference* event, which corresponds to the more general class, that is to say all occurrences of the event. For example for a *Binary duration agent*, the reference is a set

of durations between random events. The *specificity* of an event is: $s(e) = 1 - f^s(e, ref)$ defined between 0 and 1. Let us note $|e|$ the number of occurrences of the event $e \in E^\alpha$, and E^α is the set of event classes generated by the similarity agent among all occurrences. The *weight* of an event is defined according to the repartition of the occurrences of this event in relation with the other specific events:

$$w(e) = \frac{|e|}{\sum_{s(e_i) > \epsilon; e_i \in E^\alpha} |e_i|}$$

The interest of an event is $i(e) = \min(s(e), w(e))$. As illustrated on Figure 5, an association event which has both a high specificity value and weight value, may represent a regularity of the duration between two events, and give a positive feedback to agents that construct these events.

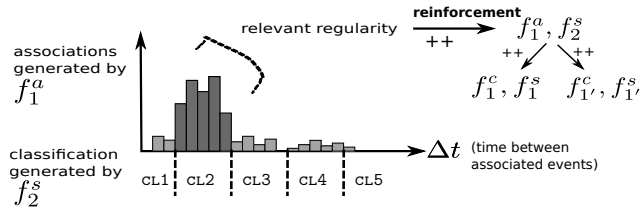


Figure 5: Feedback from an association to participating agents.

The purpose of this architecture is to investigate the dynamics of bootstrapping regularities from continuous signals as a developmental process, without committing to a specific implementation. This is the first step in a more general model of constructivist learning. We let the opportunity to incorporate a variety of agents at different levels into the system for each of the operations involved, in line with the ideas developed by Minsky (1991). In the last section we present preliminary results focusing on the bootstrapping step with a system composed of very simple agents in order to experiment the global mechanism.

Results and conclusion

The experimental setup consists of a simple set of three interacting numeric variables $V_1; V_2; V_3$. V_1 is an interruptor that actions V_2 and V_3 , which vary at different paces and scales. Activations of V_1 are repeated randomly between 2 and 6 seconds after the end of the series of effects on other variables, as represented on Figure 6. The purpose is to verify that structures generated by the system can stabilize on relevant associations, which can enable to bootstrap the pattern learning.

In this experiment, the system is initially composed of a minimal number of elementary agents: one perception agent per variable (a temporal window agent mentioned above which collects variations in a sliding window), and one similarity agent. Each event is represented by a vector (histogram), storing the distribution of the collected values.

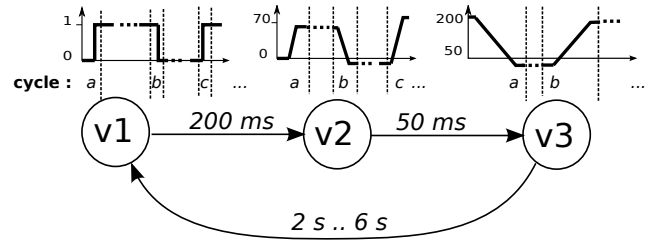


Figure 6: Experimental setup.

Thus the similarity agent compares the intersection percentage between two vectors. At the beginning there is no association agent, since there is no event to connect with. Events that go beyond a certain threshold of interest lead to the creation of an association agent, but only if the pair of agents (perception agent & similarity agent) is stable. A measure of stability takes into account the changes of important criteria for the pair, such as the number of event classes and the respective interest of each events. Stable pairs that do not include any interesting event arouse a negative feedback on their agents, and lead to the creation of new agents for exploration and encourage agents to use other functions.

So when an association agent is connected to two events that match with a regularity, it must appear as an interesting event. For example Figure 7 shows the first interesting association event that is found (after 34 activations of V_1). This association forms a three-events pattern which corresponds to the regularity involving the variation of V_1 followed by the variation of V_2 .

As expected this kind of rough pattern can enable to bootstrap a more elaborated learning mechanism since it contains necessary information about the events and their relation in time to construct a predictive structure. In a second step, there can be a feedback from the prediction mechanism to the discretization process, so that event perception learning is oriented according to its utility for predictions. Further work will be mainly dedicated to the implementation of this mechanism, but also includes: investigate the application of this generic process to higher level in the representation; and also study the possible internal dynamics within the population of agents that can be exploited to improve the performance of the system, and develop self-organization process. In this article, we point out the fundamental problem of bootstrapping learning of sensori-motor patterns from continuous environments in a constructivist perspective of learning. We propose to tackle this problem with an unconventional decentralized approach. The multi-agent architecture is used to model the self-organization process of the construction of representation and the dynamic between learning and the discretization of experience. This model is still in an early stage of development but presents interesting theoretic and applicative perspectives. We show that the proposed

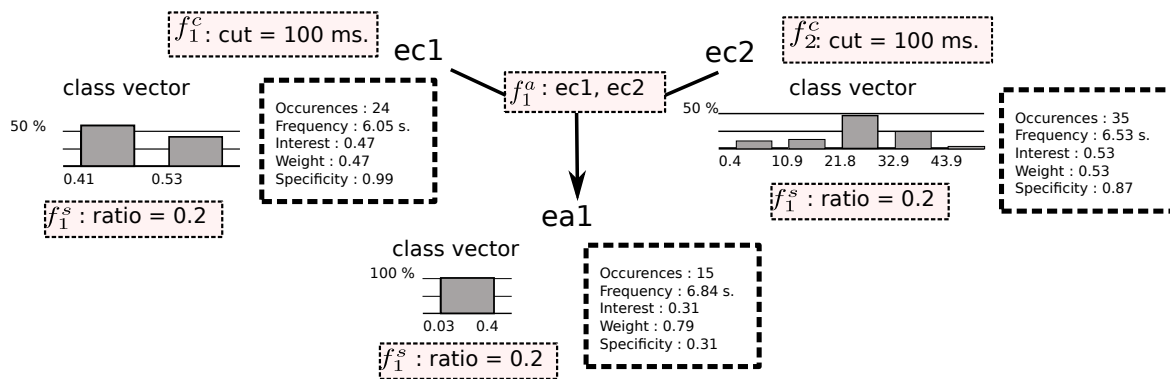


Figure 7: Example of a pattern.

mechanisms lead to the recognition of significant events and regularities without any a priori knowledge. Despite its apparent simplicity, the problem we address is complex if not avoided thanks to an abstraction effort in the design of a preliminary representation. These results are very encouraging and reveal a great potential for such an approach. Going deeper in the understanding of such mechanisms will allow to efficiently address many problems. Applicative fields such as Ambient Intelligence features both learning issues related to the undeterminism of the system, and the practical constraint to handle continuous environments.

References

- Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(13):139 – 159.
- Chaput, H. H., Kuipers, B., and Miikkulainen, R. (2003). Constructivist learning: A neural implementation of the schema mechanism. In *In Proceedings of the Workshop on Self-Organizing Maps (WSOM03)*.
- Drescher, G. (1991). *Made-up minds: a constructivist approach to artificial intelligence*. The MIT Press.
- Georgeon, O. and Aha, D. (2013). The Radical Interactionism Conceptual Commitment. *Journal of Artificial General Intelligence*, 4(2):31–36.
- Guerin, F. (2011). Learning like a baby: a survey of artificial intelligence approaches. *The Knowledge Engineering Review*, 26:209–236.
- Kuipers, B. J., Beeson, P., Modayil, J., and Provost, J. (2006). Bootstrap learning of foundational representations. *Connection Science*, 18(2):145–158.
- Linaker, F. and Jacobsson, H. (2001). Mobile robot learning of delayed response tasks through event extraction: A solution to the road sign problem and beyond. In *International Joint conference on artificial intelligence*, volume 17, pages 777–782. Lawrence erlbaum associates ltd.
- Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*, 15(4):151–190.
- Minsky, M. (1991). Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI magazine*, 12(2):34.
- Mugan, J. and Kuipers, B. (2007). Learning distinctions and rules in a continuous world through active exploration. In *Proceedings of the Seventh International Conference on Epigenetic Robotics (EpiRob-07)*, pages 101–108.
- Najjar, A. and Reignier, P. (2013). Constructivist Ambient Intelligent Agent for Smart Environments. In *PerCom - IEEE International Conference on Pervasive Computing and Communications*, San Diego, États-Unis.
- Oudeyer, P.-Y., Kaplan, F., and Hafner, V. (2007). Intrinsic motivation systems for autonomous mental development. *Evolutionary Computation, IEEE Transactions on*, 11(2):265–286.
- Perroto, F., Alvarez, I., and Buisson, J.-C. (2010). Constructivist Anticipatory Learning Mechanism (CALM): Dealing with Partially Deterministic and Partially Observable Environments. In *International Conference on Epigenetic Robotics (EpiRob)*, pages 110–120. Lund University Cognitive Science.
- Piaget, J. (1954). *The Construction of Reality in the Child*. Basic Books.
- Provost, J., Kuipers, B. J., and Miikkulainen, R. (2006). Developing navigation behavior through self-organizing distinctive-state abstraction. *Connection Science*, 18(2):159–172.
- Riegler, A. (2001). Towards a radical constructivist understanding of science. *Foundations of Science*, 6(1):1–30.
- Varela, F. J., Thompson, E., and Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. The MIT Press.
- Von Glasersfeld, E. (1984). An introduction to radical constructivism. *The invented reality*, pages 17–40.
- Von Uexküll, J. (1992). A stroll through the worlds of animals and men: A picture book of invisible worlds. *Semiotica*, 89(4):319–391.
- Ziemke, T. (2001). The construction of reality in the robot: Constructivist perspectives on situated artificial intelligence and adaptive robotics. *Foundations of Science*, 6(1-3):163–233.