Artif Life Robotics (2000) 4:7–11

INVITED ARTICLE

Dongcheng Hu · Rui Jiang · Yupin Luo

An adaptive classifier system tree for extending genetics-based machine learning in a dynamic environment

Received and accepted: July 22, 1999

Abstract An autonomous agent should possess the ability to adapt its cognition structure to a dynamically changing environment. This ability may be achieved when autonomous agents interact with the environment. In this paper, an adaptive classifier system tree is proposed for extending genetics-based machine learning in a dynamic environment. The architecture has the properties of self-similarity and self-organization. When environmental changes are inspected, the autonomous agent can adapt its cognition structure to the new environment so that cognition can be achieved with great efficiency. After a description of the dynamic structure and the principle of the structure's self-organization, some experiments illustrating how the architecture works are described and discussed.

Key words Autonomous agents · Genetics-based machine learning · Self-organization

Introduction

The traditional knowledge-based approach to artificial intelligence explains the cognitive abilities of the brain by means of symbol manipulation and reasoning. Although this approach is successfully applied in domains such as medical diagnosis and ore exploration,[1] it seems to lack the flexibility and expressiveness of natural cognitive systems. Much of the work done in behavior-based robotics show that this may be a better way to achieve this kind of cognition.[2,3]

Early work in behavior-based robotics focused on the design of appropriate robot behavior and behavior coordination techniques.[4] Recent work by Dorigo et al.[5] develops an architecture of cognition based on both ethnological and evolutionary considerations. Their work shows that the introduction of an evolutionary approach to cognitive processes is a plausible and powerful way to develop intelligent systems.

We point out, however, that an autonomous agent must possess the ability to adapt its own cognition structure to the changing environment. In this paper, we intend to construct an adaptive architecture of cognition based on this consideration. In this architecture, complex environmental input can be inspected and divided into simple items; simple cognition units are designed to achieve the cognition of these simple inputs and pass the cognition result to a higher-level unit. After coordination by a higher-level unit, the agent's final cognition result is obtained. The architecture has the properties of self-similarity and self-organization.

In the next section we briefly review the principles of genetic algorithms, genetics-based machine learning, and classifier systems. We then describe our adaptive architecture and the process of the architecture's self-organization, including principles and algorithms of width and depth extension. Some experiments and their results follow, together with discussion and analysis. Finally, we give a summary of current architecture and a preview of future work.

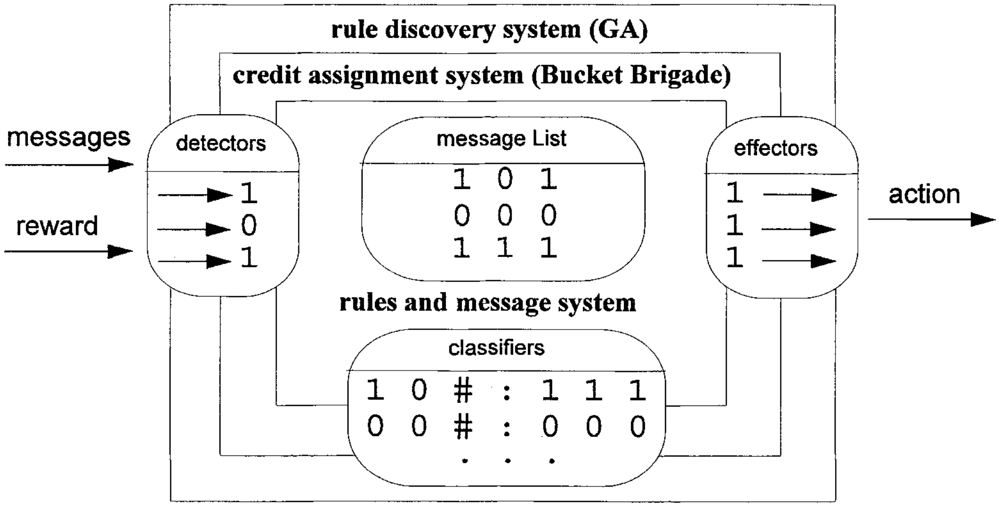
Genetic algorithms, genetics-based machine learning, and classifier systems

Genetic algorithms are intended to get optimum solutions of a given problem by the mechanics of natural selection and natural genetics.[6] Genetics-based machine learning (GBML) uses genetic algorithms to find and recombine new rules based on the hypothesis that new and better rules may be created by a recombination of old ones.[6] A classifier system is a rule-based learning system proposed by Holland.[2,3] Being a common GBML architecture, a classifier system adjusts the strength of each classifier from en-

D. Hu (✉) · R. Jiang · Y. Luo
Department of Automation, Tsinghua University, Beijing 100084, PR China
Tel. +86-10-62773432; Fax +86-10-62786911
e-mail: rjiang@mail.au.tsinghua.edu.cn

This work was presented, in part, at the Fourth International Symposium on Artificial Life and Robotics, Oita, Japan, January 19–22, 1999

Fig. 1. Schematic of the classifier system



vironmental feedback and discovers new rules using genetic algorithms.[6]

A classifier system consists of three subsystems: a rules and message system, a credit assignment system, and a rule discovery system (Fig. 1).

An environmental message, which is recognized by detectors, is sent to the rules and message system, where it is matched with the condition part of each classifier (normalized condition–action rule). The action part of those matched classifiers will be sent to effectors where the corresponding action will be carried out. With the Bucket Brigade algorithm, the credit assignment system evaluates classifiers according to their relative usefulness to the system, i.e., their ability to make the system respond correctly to the environmental messages. In the rule discovery system, those useful rules will be used as "building blocks" to generate new and plausibly better rules under the operations of a genetic algorithm.[6]

In order to increase the adaptability of classifier systems under a dynamic environment, some architectures have been proposed.[4–7] Dorigo et al.[5] developed an architecture of cognition based on both ethnological and evolutionary considerations. Their work shows that a hierarchical and parallel model is a plausible and powerful way to develop adaptive intelligent systems.

Self-organization classifier system tree

The autonomous agent must have the ability to adapt its cognitive structure to the dynamic environment. In order to do this, we have proposed an adaptive architecture which can modify its structure dynamically while interacting with the environment. In this section, we give an overview of our adaptive classifier system tree. A complete model will be given first. Then the dynamic structure of the architecture and the principle of the structure's self-organization as well as two key mechanisms, width extension and depth extension, will be described in detail.

The complete model

The basic unit of the architecture is referred to as a node that consists of a classifier system and a control unit. The control unit can feel the stimulation of the environment and make decisions; the classifier system is the core of the cognition (Fig. 2a). There are many nodes working in parallel in the system. Each node learns a simple unit of knowledge through interacting with the environment or with other node. The goal of the whole system is to implement cognition through the coordination of simple units of knowledge. The whole architecture is a tree-like one that is fractal and has the property of self-similarity (Fig. 2b).

We now give definitions of different types of knowledge.

Definition 1: *(Behavioral knowledge) Knowledge is called behavioral if and only if the input message of the knowledge is directly from the environment.*

Definition 2: *(Coordination knowledge) Knowledge is called coordination knowledge if and only if the input message of the knowledge is from other nodes, not the environment.*

With the above definitions, we can continue our discussion. Typically, only leaf nodes respond to the stimulation of the environment, so learning how to respond to behavioral knowledge is the main duty of a leaf node. By contrast, middle-level nodes and the root node have the responsibility of coordinating the behavior produced by lower-level nodes, so learning coordination knowledge is the main duty of middle-level nodes and the root node. The root node, in particular, plays the most important role in the architecture, for all behaviors will ultimately be coordinated by the root node. The total model is self-organized by the root node by inspecting the changes in the environment, as described below.

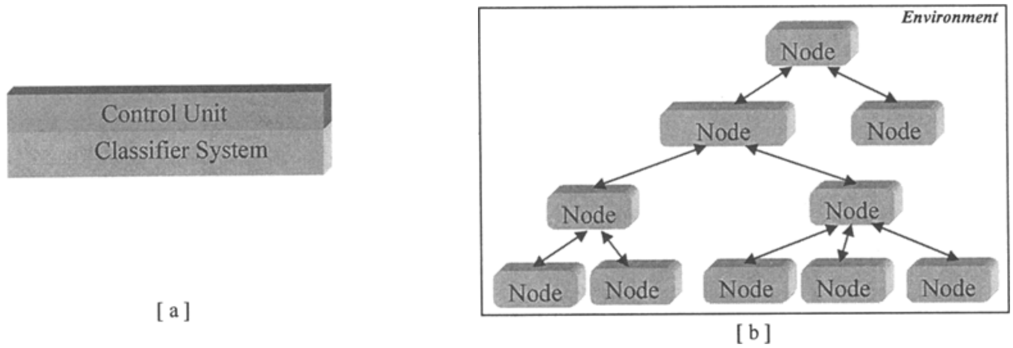


Fig. 2a,b. The complete model

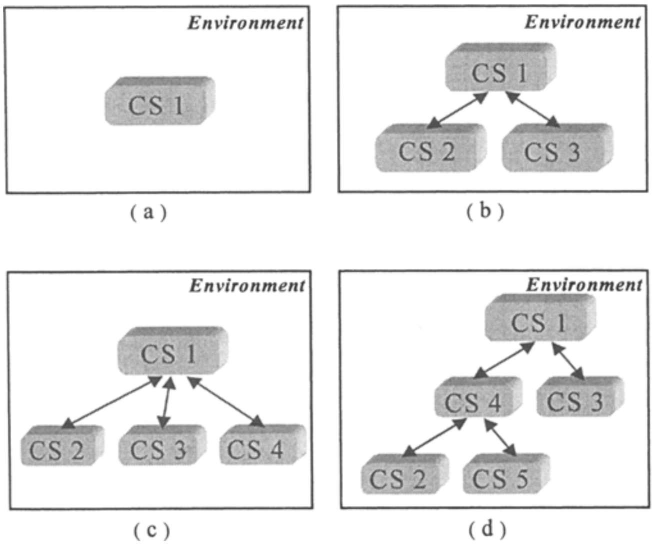


Fig. 3a–d. Self-organization of the architecture

Principle of self-organization

The dynamic structure of the architecture and the principle of the structure's self-organization are shown in Fig. 3, where the nodes are referred to as CS.

We now describe the principle of the architecture.

1. At the beginning (a), there is just one node – the root node (CS1) of the system. The node can realize any stimulation of the environment with its control unit and learn how to respond to it with its classifier system.
2. When the environment changes and a new stimulation is created (b), the root node can inspect the change. It will then create two new leaf nodes (CS2 and CS3) to respond to the stimulation separately, but coordinated by itself.
3. When the environment changes again and a new stimulation is created (c, d), the root node can again inspect the change. According to the system's current status and the type of new stimulation, the system will carry out *width extension* or *depth extension*. Width extension (c) is when an appropriate parent node creates a new leaf node (CS4) in direct response to the stimulation. Depth extension (d) is when the system creates a middle-level node (CS4) and assigns it to create two leaf nodes (CS2 and CS5) to deal with two sub-type stimulations belonging to the same type. After extension, this parent node (CS4) coordinates the child nodes (CS2 and CS5) and transfers the coordination result to its upper levels until it reaches the root node, which will ultimately decide the system's behavior.
4. Nodes related to certain stimulations will be deleted by the system under some special conditions so that infinite increments of the tree can be avoided.
5. Whenever the root node inspects the change in the environment, the structure of the system will adapt to the environment by the principles described in points 3 and 4.

We now summarize the main features of our architecture.

1. The architecture is a tree-like one that is fractal and has the property of self-similarity.
2. All the nodes in the system work in parallel.
3. Leaf nodes learn behavioral knowledge; middle nodes and the root learn coordination knowledge.
4. Once the root node inspects changes in the environment, it will drive the whole system to reconstruct its architecture dynamically.
5. Once the leaf nodes recognize the stimulation of the environment, they will learn with their classifier systems to create related behaviors and transfer them to their parent nodes.
6. Parent nodes at different levels will also learn with their classifier systems to coordinate the behaviors that are passed on by their child nodes. The coordination result will be passed to their upper level until it reaches the root node, which will ultimately decide the system's behavior.

Experiments and discussions

In this section, we explain the experiments conducted with our tree-like architecture. Our purpose in the experiments was to make sure the width and depth extensions can be achieved by the agent itself under various environmental settings. Firstly, we describe the simulation experiment settings, and then we discuss the process of our model learning a series of increasingly defined problems.

Experimental settings

Wilson[8] has proposed a simplified version of Holland's original classifier system. It is called the zeroth-level classifier system (ZCS). Inspired by the fact that ZCS has been successfully used to deal with the animat problem,[9] we use a ZCS as the cognition core in our adaptive architecture. Thus the node in our experiment can be illustrated as in Fig. 4.

Dorigo et al.[5] proposed an experiment in which a simulated robot learns to follow a light source and at the same time avoid a heat source. Our experiment is based on Dorigo's work.

The experiment is about an animat following its food and avoiding its natural enemy. The settings can be described in increasing order of complexity as follows:

Problem 1: *(Simple following) In this problem, the animat should follow its food, which may be moving in a set orbit.*

Problem 2: *(Following and avoiding) In this problem, the animat should follow the moving food; at the same time it should try its best to avoid a moving natural enemy.*

Problem 3: *(Following and avoiding two things) In this problem, the animat should follow the moving food, and at the same time it should try its best to avoid two different moving natural enemies.*

The increasingly complex experimental settings will enable us to test whether our adaptive architecture can modify its structure progressively, because both width and depth extensions are needed in this series of problems.

Results and discussion

We first put the simulated autonomous agent in the environment and let it learn problem 1. Since there is just one behavior – following food, only one node – the root node is needed to achieve cognition. It is shown that the root node can attain this ability in a short time.

The second step is to increase the difficulty of learning. In order to do this, we add a natural enemy in the environment and the problem is changed to problem 2. Since the input messages of food and the natural enemy are different, the root node inspects the environment and does a width extension (see Fig. 3b) to adapt to the change. It is shown that an autonomous agent with a parallel cognition structure can attain the ability to avoid while following quicker than an agent which does not possess this structure.

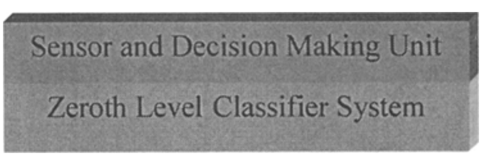


Fig. 4. A node in the current experiment

The third step is to increase the learning difficulty again and make it even more complex. In order to do this, we add another natural enemy in the environment. Thus there are two natural enemies that the agent should avoid while following its food. The environmental setting is thus changed to problem 3. Since the input messages of food and the natural enemies are different, the root node will inspect the change and do a depth extension (see Fig. 3d) to adapt to the environment. It is shown that the mechanism of using different simple cognition units to respond to different kinds of knowledge (following or avoiding) and then coordinating the result with another unit is a better way to achieve complex cognition.

Conclusions

In this paper, an adaptive architecture has been proposed for extending genetics-based machine learning in a dynamic environment. The architecture has the properties of self-similarity and self-organization. There are two key mechanisms when the architecture organizes its structure progressively. One is width extension, which is used when the new input message is exclusive. The other is depth extension, which is used when the new input message is additive. The experimental result shows that our self-organizational architecture can achieve cognition with great efficiency, and this is due to the division of input messages and the parallel running of the nodes.

There are several things worth considering. First, how to determine the type of input message. If message types are predefined, the advantage of a self-organizational architecture will be limited to a large extent. If all the message types are new to the architecture, there is a problem when using the architecture in a real robot because its sensor must be programmed before running. In a real robot, there may be a trade-off between the architecture of cognition and the sensibility to the environment.

Second, although methods of storing and making use of experience knowledge have been introduced when doing width and depth extensions, these are not adequate for a real robot. When constructing a real robot, or in the future when constructing an artificial life entity, its cognition architecture must have the ability to plan, schedule and make decisions as well as respond to the environment. We believe that nature is an inexhaustible source to borrow from. With the coordination of evolution algorithms, a hybrid architecture including an expert system, a neural network, and a petri net, as well as our adaptive classifier system, may be a plausible way to achieve these aims.

Third, our model and experiment only consider one autonomous agent. In the real world, the coordination of multiagents will be a very important research domain. Whether our self-organization architecture can be extended and then used in this domain is a problem worth thinking about.

References

1. Buchanan BG, Shortliffe EH (1984) Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project. Addison-Wesley, Reading
2. Holland JH, Reitman JS (1978) Cognitive systems based on adaptive algorithms. Pattern-directed inference systems. Academic Press, New York, pp 313–329
3. Holland JH (1986) Escaping brittleness: the possibilities of general-purpose learning algorithms applied to parallel rule-based systems. Machine learning: an artificial intelligence approach, vol 2. Morgan Kaufmann, Los Altos
4. Brooks RA (1985) A robust layered control system for a mobile robot. Artificial Intelligence Memo 864, MIT, Cambridge
5. Dorigo M, Schneph U (1993) Genetics-based machine learning and behavior-based robotics: a new synthesis. IEEE Trans Syst Man Cybern 23:141–153
6. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading
7. Song H-H, Lee S-W (1998) A self-organizing neural tree for large-set pattern classification. IEEE Trans Neural Networks 9(3):369–380
8. Wilson SW (1994) ZCS: a zeroth-level classifier system. Evol Comput 2:1–18
9. Wilson SW (1987) Classifier systems and the animat problem. Mach Learn 2:199–228