Organization of the Information Flow
in the Perception-Action Loop of Evolved Agents

Alexander S. Klyubin[*], Daniel Polani[*†], and Chrystopher L. Nehaniv[*†]

Adaptive Systems[*] and Algorithms[†] Research Groups
School of Computer Science, University of Hertfordshire
College Lane, Hatfield Herts AL10 9AB, UK
E-mail: {A.Kljubin,D.Polani,C.L.Nehaniv}@herts.ac.uk

Abstract

Sensor evolution in nature aims at improving the acquisition of information from the environment and is intimately related with selection pressure towards adaptivity and robustness. Recent work in the area aims at studying the perception-action loop in a formalized information-theoretic manner. This paves the way towards a principled and general understanding of the mechanisms guiding the evolution of sensors in nature and provides insights into the design of mechanisms of artificial sensor evolution.

In our paper we study the perception-action loop of agents. We evolve finite-state automata as agent controllers to solve an information acquisition task in a simple virtual world and study how the information flow is organized by evolution. Our analysis of the evolved automata and the information flow provides insight into how evolution organizes sensoric information acquisition, memory, processing and action selection. In addition, the results are compared to ideal information extraction schemes following from the Information Bottleneck principle.

1. Introduction

Consider an agent with sensors and actuators. Sensors capture some information, the information gets processed, and based on the results the actuators act upon the environment. If sensors are seen as taking information in, it seems also reasonable to see actuators as transferring information into the environment. Surprisingly, it seems that little research has been done to quantitatively treat perception-action in terms of information – an observation also made by Touchette and Lloyd in the context of control [8].

In [7, 8] the problem of control is quantitatively treated in terms of Shannon information which is seen as flowing from a controlled system into a controller and then back. An information-theoretic bound is obtained for the usefulness of any sensor for control. [4] introduces an information-theoretic view of perception and actuation and discusses usefulness to an agent as a means to attribute meaning to information. This is further formalized in the context of relevant information in [5]. Relevant information "flows" from sensors to actuators, thus connecting them.

Our interest is in sensory evolution [2, 4] – sensors evolving to capture useful or relevant information. However, it became clear that sensors are often inseparable from what an agent needs to *do* in its environment. It makes more sense to consider perception and action together as a single entity, a loop. The fact that actions via the environment can influence the sensors, creating a loop, is what the quantitative approaches above do not address directly.

The perception-action *loop* enables an agent to use its actuators as a channel to transmit information into the environment. The information can later be acquired from the environment by the agent or by other agents. An agent may also use its actuators as a kind of control channel selecting which information to take in via the sensors.

We approach the study of the perception-action loop using information theory. Our ultimate goal is to find the general principles underlying adaptive behavior. These could be used for studying the perception-action loops of adaptive systems and for constructing them from components interfacing on the level of information. The use of information theory ensures that the principles are indeed general and can be applied irrespective of the implementation used.

2. A Model of the Perception-Action Loop

Here we present a model of the perception-action loop of an agent. The constituents of the loop are modeled as random variables. These are sensors S, actuators A, the memory M of the controller, and R – the rest of the environment. We need R to formally account for actuation and the effects of the environment on the sensors.

IEEE
COMPUTER
SOCIETY

As in [8] we interpret the perception-action loop in terms of a communication channel-like model. In order to model the temporal aspects we unroll the loop in time by introducing a discrete time variable t.

We model the relations between the variables as a Bayesian network which is a directed acyclic graph where any node, given its parents, is conditionally independent from any other node which is not its parent or successor. We show the pattern of relations between variables at two consecutive time steps on Fig. 1. We assume that the pattern of relations is time-invariant. The graph on Fig. 1 is thus just a section of the network.
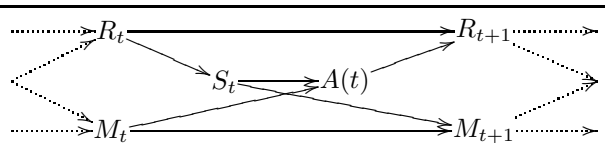

Figure 1. Perception-action loop as a Bayesian network. S **– sensoric input,** A **– action performed,** M **– memory of the controller,** R **– the rest of the environment.**

3. Experimental Approach

3.1. The Model

We base our experiments on a model which despite its simplicity captures important features of systems employing gradient sensors. The environment consists of a two-dimensional grid of infinite size. A source is located at the center of the grid. The source emits a signal, the strength P of which in any cell of the grid is $P(d) = d^{-2}$ $(P(0) = 2)$, where d is the distance to the source.

An agent moves in the world. Its position is denoted with R. The agent has a gradient *sensor*. The gradient points to the cell with highest signal strength among the four adjacent cells. If there are several cells with strongest signal, the gradient randomly points to one of these with equal probability. The agent has an *actuator* – at each time step the agent moves in one of the four adjacent cells. The agent has a controller attached to the sensors and actuators.

According to the model in Sec. 2 the controller is implementing the mapping: $(S_t, M_t) \mapsto (A_t, M_{t+1})$. We assume that the mapping is time-invariant. Then the mapping can be implemented by a finite-state automaton operating on input set $\mathcal{S}$, output set $\mathcal{A}$, and state set $\mathcal{M}$. Although here we use deterministic automata, without any loss of generality our approach can be used with stochastic mappings implemented by nondeterministic finite-state automata.

3.2. Information Flow

Here we use a special case of information flow. We 'inject' information independent of the past and present state of the system into a variable X (e.g., R_0). The *amount of information flow* from X to any variable Y is then the *mutual information*[1] between X and Y, denoted $I(X;Y)$. In general, it is possible to measure information flow without injecting independent information. However, this is out of scope of this paper.

3.3. Method of Evolution

We use evolution as a search in the space of controllers. A minimal setup is used. This is to emphasize that nothing in our general approach is specific to the particular model or the search methods employed.

We evolve the agent's controller. Fitness is evaluated by letting the controller control the agent. We initialize the population with five randomly generated controllers. After each generation five best controllers are kept in the population and produce five offspring each by mutation. To speed up the search we use ideas from simulated annealing and tabu search: (1) the number of mutations is uniformly distributed between 1 and $1 + (G \bmod 20)$, where G is the generation; and (2) we do not add offspring controllers which have been evaluated before or are present in the population.

The controller is represented by a mapping $\mathcal{S} \times \mathcal{M} \rightarrow \mathcal{A} \times \mathcal{M}$. Thus any such deterministic controller can be represented as an array of length $|\mathcal{S}| \cdot |\mathcal{M}|$ with each element containing a value between 1 and $|\mathcal{A}| \cdot |\mathcal{M}|$ corresponding to the action to perform and the next state to go into. A mutation is performed by setting a randomly chosen element of the array to a randomly chosen value in the allowed range.

4. Evolving for Information Capture

For our model it can be shown theoretically that when an agent follows the gradient and ends up near the source most of the information about the agent's initial position passes through the gradient sensor. Can this flow be connected to the agent's memory to capture information about the initial position of the agent? We would like to see how well a controller can perform given limited amount of memory and limited time.

In terms of information flow we want to find a controller which maximizes the information flow from R_0 into $M_{t'}$, where t' is the duration of the run. The controller can be seen as constructing a temporally extended communication channel between the two parts of the system. The channel

---

1   Due to lack of space here we refer the reader to [1] for an introduction to Information Theory.
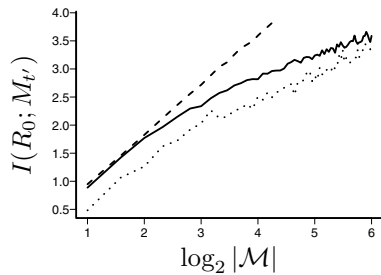
Figure 2. Best controllers. Dotted – gradient-following, solid – unconstrained, dashed – IB. $d = 5$, $t' = 5$, **1000 generations.**

can pass not only via the sensor but also via the memory, the actuator and the environment. In gradient-following controllers the information flow from M to A is blocked. Thus it is interesting to see whether following the gradient is optimal for maximizing the overall flow between R_0 and $M_{t'}$.

To study this question we evolve two populations of controllers. The first one is constrained to follow the gradient, the second one is not. The details of the evolutionary algorithm employed can be found in Sec. 3.3.

To evaluate a controller we distribute the initial position of the agent R_0 uniformly over a square with side d centered at the source. At the beginning of the run the controller is always in state 1. We let the agent run for t' time steps. The fitness function is $I(R_0; M_{t'})$.

Evolution is not guaranteed to find optimal solutions. However, as an upper bound for achievable performance we can use the amount of information about R_0 in principle extractable from the cumulative sensoric input $S_{t'-1}^c = (S_0, S_1, \ldots, S_{t-1})$ into $M_{t'}$. Clearly, no controller can extract more information about R_0 than it obtains via $S_{t'-1}^c$. We can estimate the bound for any gradient-following controller, since $S_{t'-1}^c$ does not depend on the particular controller because they all strictly follow the gradient. To calculate the bound we apply the Information Bottleneck (IB) principle [6] and find a mapping $S_{t'-1}^c \mapsto M_{t'}$ which maximizes mutual information $I(R_0; M_{t'})$.

On fig. 2 performance of evolved controllers is compared to memory size in bits. Controllers manage to extract information temporally "smeared" in sensoric input. Unconstrained controllers clearly outperform gradient-following ones. The results also indicate that the upper-bound obtained using the IB principle for gradient-following controllers may also apply to unconstrained ones.

4.1. Representation of Captured Information

How is the agent's initial position represented in the state of the controllers at the end of the run? To get an answer we
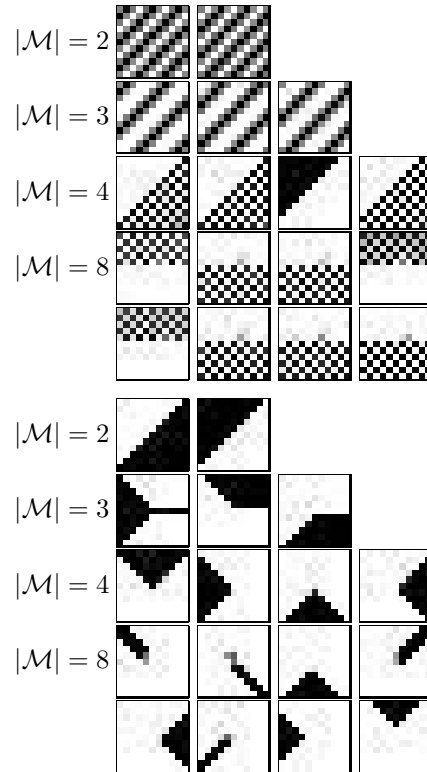

Figure 3. Mapping $M_{15} \mapsto R_0$ **by best controllers. Top – gradient-following, bottom – unconstrained. Each box represents the mapping from a state onto initial position. The intensity of each cell represents the probability of the agent having started in that cell: black – highest, white – zero.**

visualize the mapping of $M_{t'}$ onto R_0. We rerun the evolutionary experiment with $d = 11$ and $t' = 15$ to obtain spatially larger maps.

Mappings implemented by some of the best evolved controllers are shown on Fig. 3. The controllers perform quite efficient compression. In some cases factorial codes are produced: substates of $\mathcal{M}$ code for independent features of position, such as odd/even cell checkerboard pattern, or north/south half (Fig. 3, gradient-following, $|\mathcal{M}| = 8$). This suggests interesting parallels with the results obtained in [3], where factorial codes are produced as a result of maximizing information transmission by a network of neurons.

5. Conclusions & Future Work

We have presented an information-theoretic approach to analysis and construction of the perception-action loop. The approach enables us to measure information flows between

COMPUTER SOCIETY

various parts of the loop through time. To show the approach in action we have presented a two-dimensional grid world with an agent controlled by finite-state automaton and having access only to a gradient sensor. The sensor captures partial information about the agent's position. We have shown that it is possible to capture more of the information by integrating sensoric input over time and by employing an evolved movement strategy.

We have evolved controllers with limited amount of memory to capture as much information as possible about the agent's initial position. To use the memory efficiently the controllers performed lossy compression which resulted in near-hard partitioning of the position with respect to the controller's state.

From the information-theoretic perspective these controllers were evolved to create a temporally extended communication channel of maximum bandwidth between two parts of the system: the agent's position at the beginning of the run and the controller's state at the end of it. The channel is created through interaction of the agent with the environment. Compression, partitioning, the movement strategy employed by the controllers are all induced by maximization of information transfer – a result similar to [3].

The experiments indicate that information-theoretic tools are quite promising. Virtually assumption-free and without being tied to a particular representation or implementation they provide us with bounds, they allow us to measure, predict, and construct. We believe that the field of evolvable hardware can benefit from these ideas.

Firstly, we have shown that it is possible to evolve controllers to extract and process information without assuming much about its representation, the sources it may be acquired from or the ways to acquire it. This may enable agents to deal in a uniform way with changing sensors, actuators, and environment.

By placing only few constraints we let evolution pick from a wider range of solutions, potentially resulting in more efficient solutions. An adaptive agent may choose which information to process internally and which to store in and acquire from the environment. Moreover, the agent may choose the representation of the information. This frees the engineer from committing to a particular representation and or even implementation by enabling the construction of perception-action loops from components which adapt to each other on the level of information.

Secondly, we demonstrate that maximization of information transfer can give rise to intricate behavior. A temporally extended communication channel is created by the evolved controllers through interaction of the agent with its environment. Information transfer maximization can induce the necessary structure in the system. Moreover, the very same principle may be responsible for adaptively reshaping the system when something that can be taken advantage of is discovered in the environment or when the environment simply changes.

Consider an adaptive agent with sensors and actuators connected to a controller made of an adaptive substrate, for example, an evolvable hardware board. The substrate is flexible and is continuously reshaped by the flow of information between sensors, actuators and its own memory. According to which criteria should the substrate adapt? In this paper we have used evolution with an objective function which can only be measured externally. This clearly is not a solution for an autonomous robot, which has to perform calculations on-board and adapt on-line. An objective function is required which can be evaluated based only on the information available to the agent. We are currently researching this question and have grounds to believe that the function may be expressed in simple information-theoretic terms.

Acknowledgments

References

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[2] K. Dautenhahn, D. Polani, and T. Uthmann. Guest editors' introduction: Special issue on sensor evolution. *Artificial Life*, 7(2):95–98, 2001.

[3] J.-P. Nadal and N. Parga. Information transmission by networks of non linear neurons. In *Proc. of the Third Workshop on Neural Networks: from Biology to High Energy Physics, (1994, Elba, Italy)*, International Journal of Neural Systems, Supplement, pages 153–157, 1995.

[4] C. L. Nehaniv. Meaning for observers and agents. In *IEEE International Symposium on Intelligent Control / Intelligent Systems and Semiotics, ISIC/ISAS'99*, pages 435–440. IEEE Press, 1999.

[5] D. Polani, J. T. Kim, and T. Martinetz. An information-theoretic approach for the quantification of relevance. In J. Kelemen and P. Sośik, editors, *Advances in Artificial Life*, LNCS, pages 704–713. Springer, 2001.

[6] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pages 368–377, Sept. 1999.

[7] H. Touchette and S. Lloyd. Information-theoretic limits of control. *Phys. Rev. Lett.*, 84:1156, 2000.

[8] H. Touchette and S. Lloyd. Information-theoretic approach to the study of control systems. *Physica A*, 331(1–2):140–172, Jan. 2004.