On information-optimal scripting of actions

Bente Riegler and Daniel Polani
Adaptive Systems Research Group, School of Computer Science, University of Hertfordshire
b.reichardt@herts.ac.uk d.polani@herts.ac.uk

Abstract

Animals and humans encounter many tasks which permit ritualized behaviours, essentially fixed action sequences or "scripts", similar to options known from Reinforcement Learning, but proceeding without intermediate decisions. While running a script, they proceed in an open-loop fashion. However even when these are already known, an agent needs to decide whether to perform a basic action or to trigger a script regarding the particular task. Here we study if including such scripts (i.e. behaviour rituals) is advantageous from the point of view of the relevant information required to take the decision to start such a script depending on the tasks. To achieve this, we modify the relevant information framework including sequences of basic actions to the possible actions.

Introduction

Many tasks animals or humans encounter are composed of multiple smaller steps. An agent has typically learned such sequences through repeated solution of the task over time. Such "ritualized" behaviour sequences do not require high-level decision-making for every small step, but may permit solutions where fixed action sequences (ritualized behaviours) are triggered. Whenever this is possible, this leads to informationally significantly cheaper control, because fewer decisions need to be made — only ever when a new script is triggered, while it is running, it operates as an open loop controller. In contrast, if only basic actions are available, a decision may be required in every time step. This is a special, but important, case of the more sophisticated option framework (Van Dijk et al., 2009). The exact script to be triggered depends on the specific task and requires information about the current state of the agent. We ask how much *relevant information* (Shannon information about state required to select an action) is required when scripts - sequences of basic actions - can be used in addition to basic actions. Furthermore, we ask whether scripts make some goal states informationally easier to reach than others.

Perception-Action Loop

The perception-action loop setup for our agent is very similar to Reinforcement Learning. In each state, the agent performs actions and as a result, its state changes. This is modelled as a Markov Decision Process (MDP). A set of states $s \in \mathcal{S}$ models the agent's position in the world and in each the agent can choose one action $a \in \mathcal{A}$. For the current experiment we assume the individual transitions $p(s_{t_2}|s_{t_1}, a)$, with t_1 and t_2 being the time before and after the action, to be deterministic. In our work, the agent does not have to freshly decide its next action after every single primitive, but can select an action script instead of a primitive, modelled as an MDP with enhanced action set.

The World

The world consists of the set of all states. Here, we consider a small grid world of 5×5 states, one start-state, at least one goal-state and the set of basic actions $\mathcal{A} = \{$*north, east, south, west*$\}$, with respect to the global directions. The agent carries no internal orientation and is always globally oriented. Every executed basic action incurs a cost of 1. There is no discount over time. We assume there exists a goal which can be any subset of $\mathcal{S}$. Goal states are modelled as absorbing states in the MDP, i.e. all actions taken in a goal state leave the agent where it is, and do not add further cost. On reaching the goal, the currently executed script is effectively interrupted. The grid is finite and has "walls", an action that pushes the agent into the wall leaves the agent unchanged and still incurs the usual cost of 1. Since here we only consider optimal policies, no agent will waste effort walking into walls.

The Action Space Extended by Scripts

The main novelty compared to previous studies (Polani et al., 2006) is the action space . To the set $\mathcal{A}_b$ of basic actions, we add a set of scripts. These scripts are a sequential unconditional (open-loop) combination of the basic actions available in the world. Thus, our new action-space consists of all concatenations of at least one basic action $\mathcal{A}_b^+$; in our setting, we assume a maximum length of scripts, and thus a finite selection of possible (basic or composite) actions. In this work, we assume the agent has already learnt all possible actions. The cost of an action is modelled in two slightly different ways: first, a cost of 1 per every basic action in the script except for actions after reaching the goal; and second, the same, but with an added cost of 1 for taking a decision (note, this MDP cost is *not* informational in the present experiments). The decision cost is a cost only occurring at decision points. The value of 1 is arbitrarily chosen.

Relevant Information

Relevant information for an MDP is defined as the minimal information required about the current state to select an action to achieve a given utility (or, equivalently, in our case, minimal cost, see Polani et al., 2006): $\min_{\pi(A|S) s.t.\ \mathbf{E}^{\pi}[Q(s,a)] \overset{!}{=} Q^{*}(s,a)} I(S;A)$. The relevant information is calculated in two steps. Firstly, precalculate the perfect utility of an agent with respect to the given goal states with a value iteration algorithm. Costs of all actions and scripts are accumulated during a single run, until the goal is reached.

Secondly, the relevant information is computed based on this utility. For this, we use the classic Blahut-Arimoto-algorithm from rate-distortion theory. From these results we calculate the policy and identify which actions or scripts are used in which state.

Experiments

In the experiments, the agent may start in any state. And we examine different classes of goal states. We consider the following: **Northern Border**: This goal is composed of all northernmost states of the world. With only basic actions available, the relevant information is zero. In all states the best action is to go *north*. Expanding the action space with scripts of any length changes the optimal policy so that all scripts containing just the basic action *north* are equally probable in all states. Thus, the relevant information stays zero. When a cost for choosing an action (i.e. decision cost) is added, the longest script possible is preferred because it requires fewer decision points. **Central State:** Only the central state of the grid is a goal state. To reach the goal requires different actions from different states. This results in a high relevant information of 0.1 bit per decision without decision cost and 0.4 bits with. This goal leads to a high relevant information and favours all shorter actions over longer scripts. This does not change after adding a decision cost. **Centre Line:** Here a whole line running through the center is set as goal. This setup falls in between the previous ones. It shares the neighbouring goal-states from the first setup with the centre-character of the second setup. Thus, the result should be in between as well. The relevant information turns out to be roughly 0.08 bits for the centre line setup. Shorter actions are preferred over long ones. When a cost for the decision is added, longer scripts are favoured. **Corner State:** Here, the goal is one state in one of the corners of the world. For this setup, we expect a reduced but nonzero amount of relevant information. Indeed, the relevant information becomes about 0.01 bits. Without a decision cost, this setup favours shorter movements. When a decision cost is added, the scripts modelling diagonal movement are favoured in many states, but the relevant information increases to 0.13.

Discussion and Future Work

We find that the main value of scripts is to avoid re-deciding on what to do while they run, since the scripts are favoured when we assume a decision cost. Note, there is no profound justification for the value of the decision cost for now.

The experiments show a use of scripts for the northern border and the central line goal areas, while the setups with single goal states keep using the basic actions. Thus, scripts are useful for wider goal areas but not when specific states need to be reached. The wider goals represent generic tasks, such as extending the body to reach “as high as possible” for which the northern border goal setup is an abstract model, or “somewhere back there” represented abstractly by both the border and the centre line setup.

Note, that, strictly spoken, despite our present assumptions, behavioural scripts in actual organisms may require low-latency feedback and are not necessarily fully open-loop. So, more strictly, one would have to associate some processing cost also to run the scripts. However, our present paper focuses only on the high-level information required to select and activate the scripts.

In future, it will thus be important to also quantify the trade-off between memorizing, processing the low-level script and and saving high-level relevant information. Ultimately, this relates to the question of how hierarchies should be found and organized (Larsson et al., 2017) and how expensive learning itself is.

Acknowledgements

DP wishes to thank Olaf Witkowski for inspiring conversations that led to this study.

References

Larsson, D. T., Braun, D., and Tsiotras, P. (2017). Hierarchical state abstractions for decision-making problems with computational constraints. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1138–1143. IEEE.

Polani, D., Nehaniv, C. L., Martinetz, T., and Kim, J. T. (2006). Relevant information in optimized persistence vs. progeny strategies. In *In: Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*. Mit Press.

Van Dijk, S. G., Polani, D., and Nehaniv, C. L. (2009). Hierarchical behaviours: getting the most bang for your bit. In *European Conference on Artificial Life*, pages 342–349. Springer.