Hierarchical Behaviours: Getting the Most Bang for your Bit

Sander G. van Dijk, Daniel Polani, and Chrystopher L. Nehaniv

Adaptive Systems Research Group
University of Hertfordshire, Hatfield, UK

Abstract. Hierarchical structuring of behaviour is prevalent in natural and artificial agents and can be shown to be useful for learning and performing tasks. To progress systematic understanding of these benefits we study the effect of hierarchical architectures on the required information processing capability of an optimally acting agent. We show that an information-theoretical approach provides important insights into why factored and layered behaviour structures are beneficial.

1 Introduction

Animals sometimes make performing complex tasks seem almost trivial. For instance, a praying mantis can show a wide arrange of complicated behaviours like feeding, grooming and sexual behaviour, with very limited brain power. Understanding this is only possible when realising that their behaviour is well structured, partially in a hierarchical manner [1]. Nature supplies a large amount of examples of this kind of hierarchical behaviour, e.g. in vocal behaviour in singing birds [2] and ordering tasks in capuchin monkeys [3]. Unsurprisingly, there has for long been a call in the field of ethology to not neglect this structure [4].

Computational models of such structures are well established. Traditionally as static systems [5], but the latest advances in reinforcement learning show that adaptive hierarchical behaviour structures can significantly sped up learning [6][7][8], even when an agent has to autonomously build up its behavioural hierarchy in parallel with learning a new task [9][10]. Recent research has investigated the relationship of biological structures to computational models of layered control [11] and modern adaptive hierarchical architectures [12].

The benefits of hierarchical organisation of behaviour are intuitive: it reduces complexity, eases learning and facilitates skill transfer [13]. What is missing however is a systematic rather than heuristic understanding of the reasons of success of this kind of structures in nature and in artificial agents. Additionally quantitative methods for comparing the growing number computational models of behaviour hierarchies are needed. Currently these are partly, if not wholly, based on assumptions and external knowledge of their designer, of which the necessity is difficult to judge.

Information theory has proven to be a useful tool in understanding fundamental, global properties and limits of agent-environment systems [14][15][16].

Our hypothesis is that the advantages of hierarchical behaviour structures in animals and artificial agents are grounded in their effect on the way an agent processes information. Firstly, they divide the burden of information processing over different layers. Secondly, information about the current context is retained in more abstract behaviours at higher levels, relieving lower levels from performing redundant information processing. In this paper we will first research and quantify these effects seperately. Subsequently we will investigate their combination.

2 Relevant Information

An agent with sensors and actuators combined with its environment forms an action-perception loop, in which at each time step t the agent perceives the state of the environment $s_t \in \mathcal{S}$ and decides on an action $a_t \in \mathcal{A}$. The dynamics of the total system are determined by the *state transition probability distribution* $\mathcal{P}^a_{s_t,s_{t+1}} = p(s_{t+1}|s_t, a_t)$ and the agent's *policy* $\pi(s_t, a_t) = p(a_t|s_t)$[1].

We are interested in agents operating in an environment that rewards certain behaviour, according to an *immediate reward function* $r_{t+1} = \mathcal{R}^{a_t}_{s_t,s_{t+1}} \in \mathbb{R}$. Given this reward we can determine two functions: the *state value function* $V^\pi(s)$, which is defined as the expected future reward an agent will reserve when starting in state s and following policy π, and the *state-action value function* (or *utility function*) $U^\pi(s, a)$ which gives the expected future reward when taking action a when in state s and consequently following policy π [8]:

$$V^\pi(s) = \sum_a p(a|s) \sum_{s'} p(s'|s, a) \left[r(s, a, s') + \gamma V^\pi(s')\right] \tag{1}$$

$$U^\pi(s, a) = \sum_{s'} p(s'|s, a) \left[r(s, a, s') + \gamma V^\pi(s')\right] \tag{2}$$

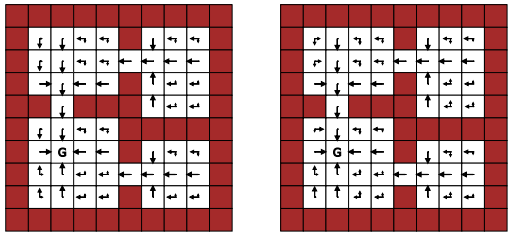
where $\gamma \in (0, 1)$ is an inflation factor to provide an upper bound on the expected future reward and to model preference of short term over long term reward.

An optimal policy for an agent in such an environment is one that maximises the expected utility $E\left[U(s, a)\right] = \sum_{s,a} p(s, a)U(s, a) = \sum_{s,a} p(a|s)p(s)U(s, a)$. To be able to execute this policy, the agent needs to take in and process information from the environment through its sensors. However, not all available information is needed. The concept of *relevant information* provides a concrete minimum of the amount of information an agent needs to acquire to be able to follow an optimal policy, measured by the mutual information between states and optimal actions [14]:

$$I(S; A^*) = \min_{p(a|s):p(a|s)p(s)>0 \Rightarrow a \text{ optimal for } s} I(S; A) \tag{3}$$

This measure is an important, fundamental property of the environment-agent dynamics and a reward function. It does not depend on the actual method of

[1] See appendix for notation



(a) Flat (b) Memory

Fig. 1: Grid-world environment used in experiments with the policy of the best solution found using a flat, memory-less policy (a) and when using memory (b). The size of the vectors is proportional to the probability of choosing the action to move in that direction in a given state ($\pi(s, a) = p(a|s)$).

finding an optimal policy. Mutual information $I(S; A) = H(A) - H(A|S)$ is measured in bits and is defined by the change in entropy (uncertainty) about A when S is being observed.

3 Grid-world Navigation Task

3.1 Environment

The environment used in this paper is a 2-dimensional grid-world as shown in Fig. 1. The set of possible states $\mathcal{S}$ contains all unoccupied cells. The set of available actions $\mathcal{A}$ consists of four actions that move the agent one cell north, east, south or west. The environment is fully deterministic: $\mathcal{P}^{a_t}_{s_t,s_{t+1}} \in \{0, 1\}$. When performing an action that would bring the agent to an occupied grid cell, the agent remains in the same state.

A single cell g is designated as the goal state and a reward is given when the agent arrives in this state: $\mathcal{R}^{a_t}_{s_t,s_{t+1}} = 1$ if $s_t \neq g, s_{t+1} = g$ and 0 otherwise.

3.2 Policy Evolution

We are interested in the minimum amount of state information an agent takes in when following an optimal policy, which we will denote as $\mathfrak{I}$, equipped with different behaviour structures. When using a flat policy, this amount is the relevant information as discussed in section 2: $\mathfrak{I} = I(S; A^*)$, which can be found by generalising the classical Blahut-Arimoto algorithm for rate-distortion[17][14].

However, it is not trivial to adapt this algorithm to the other behaviour structures we will introduce further on in this paper. Therefore, we will use an evolutionary approach to find policies that minimise $\mathfrak{I}$. We start with a population of random individuals, defined by the conditional probability distributions that make up their policies. At each epoch, individuals are iteratively selected from the previous population and combined by crossing their genotypes to create

descendants, which advance to the next generation, possibly undergoing mutation.

The selection is done through tournament selection. A number of individuals is chosen at random and a fitness function $F(\pi)$ is used to rate their policy. The winner is selected for breeding and produces a single offspring, of which a conditional probability distribution that is part of its policy can be crossed with that of the child of another tournament winner. The probability of occurrence of this crossing is determined by a cross-over rate $p_c \in [0, 1]$. Cross-over mixes distributions of the children such that $p_{child1}(A|B) \leftarrow p_{child2}(A|B)$ for $B >= b$ where b is chosen at random. Mutation is applied by permuting a distribution $P(A = a|B = b) \leftarrow P(A = a|B = b) + \epsilon$ for each b with probability p_m, where a and ϵ are chosen at random. After mutation $P(A = a|B = b)$ is renormalised.

In all experiments described in this paper we have used a population size of 100 individuals and a tournament size of 3 individuals. The cross-over and mutation rates have been set to $p_c = p_m = 0.1$. The fitness function is defined as:

$$F = \mathfrak{I} + \beta \sum_{s,a} p(a|s)p(s)U(s, a), \tag{4}$$

where $\mathfrak{I}$ is varied according to the experiments as described in the next sections. Furthermore, we set $\beta = 10^5$ to enforce near-optimal policies. Evolution is run until the change of fitness of the best individual between generations becomes sufficiently small. To establish a baseline we have applied the evolutionary algorithm using memory-less flat policies where $\mathfrak{I} = I(S; A)$. The algorithm consistently converged to a solution with $\mathfrak{I} = 1.26751$ bits, which is the same amount as found with the traditional Blahut-Arimoto algorithm. The policy of this solution is shown in Fig. 1(a).

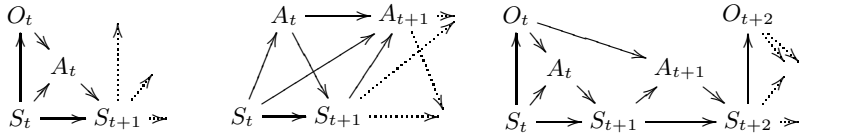
4 Hierarchical Policies

In this section we will investigate the first intuition about the advantage of using a hierarchical policy: it splits up the task into simpler parts. The minimum information intake $\mathfrak{I}$ gives a natural quantitative measure to determine complexity of a (sub)task. Our hypothesis is that using a hierarchy reduces the amount of information required, and thus the necessary complexity, at each layer.

Here we will use the simplest hierarchical model with two levels. At each time step, a general decision is made at the higher level, based on the current state, which we will call an option in accordance to the terminology of Sutton et al [8]. At the lower the actual action that is to be performed is selected based on this option o_t and extra information about the current state. The policies at the higher and lower level are determined by the conditional probability distributions $p(o_t|s_t)$ and $p(a_t|s_t, o_t)$. The corresponding causal Bayesian network of this hierarchical policy is shown in Fig. 2(a).

The total memory intake is the sum of the information intake at each level:

$$\mathfrak{I} = I(S_t; O_t) + I(S_t; A_t|O_t) = I(S_t; A_t) + I(S_t; O_t|A_t) \tag{5}$$



(a) Hierarchical (b) Memory (c) Temporally Extended Options

Fig. 2: Causal Bayesian network of the action-perception loop using a hierarchical policy (b), memory (b) or temporally extended options (c), unrolled in time

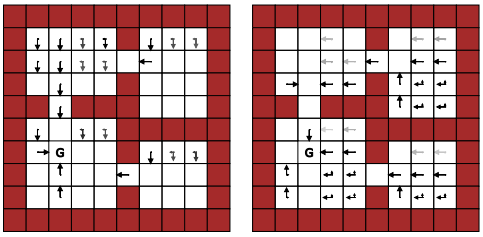


Fig. 3: Policy of the best solution found using a hierarchical, memory-less policy per option. The size of the vectors is proportional to the probability of choosing the action to move in that direction in a given state and option ($p(a|o,s)$). The opacity of the arrows in a cell are proportional to the probability of selecting the respective option in that state ($p(o|s)$).

As can be seen, this sum is greater than or equal to the mutual information between states and actions, with equality if either the current state or the chosen option is completely determined by the selected action. Therefore, the total information intake will never drop below the minimum achieved with a flat policy. It is no surprise then that we find that both the evolutionary approach as an adapted version of the Blahut-Arimoto algorithm find solutions where $I(S_t;O_t)=0$ and $I(S_t;A_t|O_t)=I(S_t;A_t)$, effectively optimising away the hierarchy.

Our goal is to show that using a hierarchy can let one get away with combining simple components. To do this we not only minimise the total sum, but also the information intake of the most complex part by extending the fitness function:

$$F=\Im+\max\left[I(S_t;O_t),I(S_t;A_t|O_t)\right]+\beta\sum_{s,a}p(a|s)p(s)U(s,a) \tag{6}$$

In our experiments we set the number of options to 2. The best solution found by the evolutionary algorithm resulted in average information intakes of $I(S_t;O_t)=0.830826$ bits, $I(S;A|O)=0.975234$ bits, totalling to $\Im=1.8061$ bits. Although the total system needs more information, the most complex level (the lower level) can get away with handling almost 25% less information as compared to a flat policy. The policy of this solution is shown in Fig. 3.

5 Memory

As mentioned in the introduction, the second intuition about hierarchical behaviour structures is that behaviours on higher levels function as a form of memory and retain information about the environment and the current task being performed. In this section we will investigate the effect of memory on necessary information intake of a policy. When looking at trajectories of states and actions, an agent can clearly get away with taking in less information than the sum of the single-step intakes discussed in the previous sections. Knowing about the past already gives you information about the current state without looking, since inherent structure of the world restricts the number of possible state transitions and the selection of an action is fed back into the distribution of states.

To handle channels that show this kind of feedback, the concept of *directed information* $I(S^T \to A^T)$ [18] was developed, which measures the actual information flow from an input sequence (S^T) to an output sequence (A^T) of length T. The directed information intuitively gives an indication of the amount of information an agent has to take in when having access to information about the past. In our experiments with memory we use this measure, scaled by trajectory length, as the average per step intake:

$$\mathfrak{I} = \frac{1}{T} I(S^T \to A^T) = \frac{1}{T} \sum_{t=0}^{T} I(S^t; A_t | A^{t-1}) \tag{7}$$

We use the simplest model, where $T = 2$ and thus memory only goes back a single time step, resulting in the network shown in figure Fig. 2(b). The best solution found in these experiments resulted in an average information intake of $\mathfrak{I} = \frac{1}{2}\left(I(S_t; A_t) + I(S_t, S_{t+1}; A_{t+1}|A_t)\right) = \frac{1}{2}(1.31635 + 1.08365) = 1.2$ bits. The policy of this solution is shown in Fig. 1(b). As expected, this average is lower than in the memory-less case. However, the average intake at the first step of a two-step trajectory $(I(S_t; A_t))$ was consistently higher, increasing the minimum needed complexity of information processing.

6 Temporally Extended Actions

The separate properties of hierarchical behaviour discussed in the previous sections show important effects on necessary complexity, however each require more complexity in some aspects. In this section we study the combination of both properties into a structure that corresponds better with the examples of nature given in the introduction and the notion of Sutton's options.

In these experiments we model a hierarchical policy that uses temporally extended, abstract, actions: at higher levels in the hierarchy such an action (or option) is chosen as in section 4, with the difference that this option is applied for several subsequent time steps. An action is chosen at the lowest level in the same way as in the memory-less case. The Bayesian network corresponding to

the model used in our experiments, with a two-step cycle, is shown in Fig. 2(c). Note that this model is more limited than that of the previous section, since the choice for an option can only retain direct information about the previous state, information about the previous action is only implicit through the lower level policy ($p(a|o,s)$).

In this model the total per-step information intake is the average over the intake during both steps of a cycle:

$$\mathfrak{I} = \frac{1}{2}(I(S_t;O_t) + I(S_t;A_t|O_t) + I(S_{t+1};A_{t+1}|O_t)) \tag{8}$$

Again, we want to minimise both the complexity of the complete system and of its parts, so we use the following fitness function:

$$F = \mathfrak{I} + \max\left[I(S_t;O_t), I(S_t;A_t|O_t), I(S_{t+1};A_{t+1}|O_t)\right] + \beta \sum_{s,a} p(a|s)p(s)U(s,a) \tag{9}$$

Experiments with this structure in the grid-world described in the previous sections is still ongoing work at time of writing. Results obtained with a smaller, 2-room environment, however, show that temporally extended actions combine the advantages quantified in the previous two sections. One solution achieves the minimum necessary information intakes $I(S_t;O_t) = 0.618551$ bits, $I(S_t;A_t|O_t) = 0.972351$ bits, $I(S_{t+1};A_{t+1}|O_t) = 0.35756$ bits and $\mathfrak{I} = 0.97423$ bits, where with a flat, memory-less policy one has $\mathfrak{I} = 1.03253$ bits. This indicates that an agent can achieve an optimal policy with simpler components (as in section 4) and still on average take in less information in total (as in section 5). Preliminary results suggest that these advantages generalise to more complex environments.

7 Discussion and Future Work

The results presented in the previous sections show that hierarchical behaviour structures can reduce the necessary amount of information an agent needs to be able to process to perform a task optimally. We have also shown that simpler parts placed in a hierarchy can perform as well as a more complex, flat system, with the same or even smaller amount of information needed. This offers a quantifiable, environment and architecture-independent argument for the use of these structures.

The methods put forward in this paper promise applications in the study of hierarchical behaviour in ethology, possibly resulting in systematic understanding of the prevalence of this phenomenon in nature. Additionally, they can be applied to computational models and their specific implementations of behaviour structuring to determine the necessity of designer imposed assumptions and heuristics.

Future work will consist including more general notions of behaviour structuring. Most notably, we will investigate other models of memory and information flow [19], study the utility/information processing trade-off under limitations and noise.

References

1. Prete, F.R., Wells, H., Wells, P.H., Hurd, L.E.E.: The Praying Mantids. Johns Hopkins University Press (1999)
2. Yu, A.C., Margoliash, D.: Temporal Hierarchical Control of Singing in Birds. Science **273**(5283) (1996) 1871–1875
3. McGonigle, B.O., Chalmers, M., Dickinson, A.: Concurrent disjoint and reciprocal classification by Cebus apella in serial ordering tasks: evidence for hierarchical organization. Animal cognition **6** (2003) 185–197
4. Dawkins, R.: Hierarchical organisation: A candidate principle for ethology. In: Growing Points in Ethology. Cambridge University Press, Cambridge (1976) 7–54
5. Quinlan, J.R.: Induction of Decision Trees. Machine Learning **1**(1) (2007) 81–106
6. Dietterich, T.G.: Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. Journal of Artificial Intelligence Research **13** (1999) 227–303
7. Parr, .R., Russell, .S.: Reinforcement learning with hierarchies of machines. In: NIPS 97: Proceedings of the 1997 conference on Advances in neural information processing systems 10, Cambridge, MA, USA, MIT Press (1998) 1043–1049
8. Sutton, R.S., Precup, D., Singh, S.: Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. Artif. Intell. **112**(1-2) (1999) 181–211
9. McGovern, A., Barto, A.G.: Automatic Discovery of Subgoals in Reinforcement Learning using Diverse Density. In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 361–368
10. Şimşek, O., Barto, A.G.: Using relative novelty to identify useful temporal abstractions in reinforcement learning. In: ICML 04: Proceedings of the twenty-first international conference on Machine learning, New York, NY, USA, ACM (2004) 95
11. Prescott, T.J., Redgrave, P., Gurney, K.: Layered Control Architectures in Robots and Vertebrates. Adaptive Behavior **7** (1999) 99–127
12. Botvinick, M., Niv, Y., Barto, A.: Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. Cognition (2008)
13. Barto, A.G., Mahadevan, S.: Recent Advances in Hierarchical Reinforcement Learning. Discrete Event Dynamic Systems **13**(4) (2003) 341 – 379
14. Polani, D., Nehaniv, C., Martinetz, T., , Kim, J.T.: Relevant Information in Optimized Persistence vs. Progeny Strategies. Proc. Artificial Life X (2006)
15. Capdepuy, P., Polani, D., Nehaniv, C.L.: Constructing the Basic Umwelt of Artificial Agents: An Information-Theoretic Approach. In Costa, F.A.e., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A., eds.: Proceedings of the Ninth European Conference on Artificial Life. Volume 4648 of LNCS/LNAI., Springer (2007) 375–383
16. Klyubin, A.S., Polani, D., Nehaniv, C.L.: All Else Being Equal Be Empowered. Advances in Artficial Life, European Conference on Artificial Life (ECAL 2005) **3630** (2005) 393–402
17. Blahut, R.E.: Computation of Channel Capacity and Rate-Distortion Functions. (1972)
18. Massey, J.L.: Causality, Feedback and Directed Information. Proceedings of the International Symposium on Information Theory and Its Applications (ISITA '90) (1990) 303–305
19. Ay, N., Polani, D.: Information Flows in Causal Networks. Advances in Complex Systems **11**(1) (2008) 17–41