Using subgoals to reduce the descriptive complexity of probabilistic inference and control programs

Domenico Maisto
Institute for High Performance Computing and Networking
National Research Council
Via Pietro Castellino 111, 80131 Napoli, Italy
domenico.maisto@icar.cnr.it

Francesco Donnarumma
Institute of Cognitive Sciences and Technologies
National Research Council
Via S. Martino della Battaglia, 44, 00185 Rome, Italy
francesco.donnarumma@istc.cnr.it

Giovanni Pezzulo
Institute of Cognitive Sciences and Technologies
National Research Council
Via S. Martino della Battaglia, 44, 00185 Rome, Italy
giovanni.pezzulo@istc.cnr.it

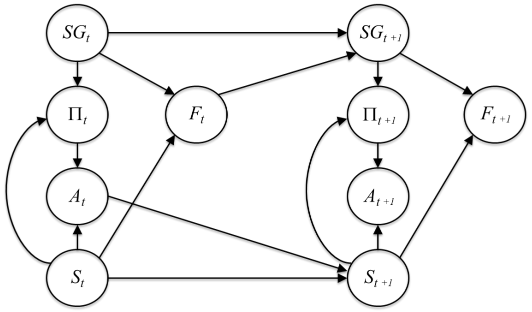
Abstract

Humans and other animals are able to flexibly select among internally generated goals and form plans to achieve them. Still, the neuronal and computational principles governing these abilities are incompletely known. In computational neuroscience, goal-directed decision-making has been linked to model-based methods of reinforcement learning, which use a model of the task to predict the outcome of possible courses of actions, and can select flexibly among them. In principle, this method permits planning optimal action sequences. However, model-based computations are prohibitive for large state spaces and several methods to simplify them have been proposed. In hierarchical reinforcement learning, temporal abstractions methods such as the Options framework permit splitting the search space by learning reusable macro-actions that achieve subgoals. In this article we offer a normative perspective on the role of subgoals and temporal abstractions in model-based computations. We hypothesize that the main role of subgoals is reducing the complexity of learning, inference, and control tasks by guiding the selection of more compact control programs. To explore this idea, we adopt a Bayesian formulation of model-based search: planning-as-inference. In the proposed method, subgoals and associated policies are selected via probabilistic inference using principles of descriptive complexity. We present preliminary results that show the suitability of the proposed method and discuss the links with brain circuits for goal and subgoal processing in prefrontal cortex.

Keywords: goal-directed decision-making; goal; subgoal; temporal abstraction; Bayesian inference; planning-as-inference; descriptive complexity; information compression

Acknowledgements

Research funded by the EU's FP7 under grant agreement no FP7-ICT-270108 (Goal-Leaders).



SG_t	subgoal	$[0, ..., n]$
Π	policy	$[0, ..., m]$
A_t	action	$\{u, d, l, r, \varepsilon\}$
F_t	termination condition	$\{0, 1, 2\}$
S_t	state	$[0, ..., n]$

Table 1: Left: Graphical model (Dynamic Bayesian Network [13]). Right: Stochastic variables

1 Introduction

A widespread idea in neuroscience is that goals and subgoals maintained in prefrontal hierarchies have a key role in exerting cognitive control over behavior [1, 2]. Goals and subgoals are usually assigned several roles: serving as reference states for action selection and monitoring, stabilizing behavior by preventing oscillations between incompatible action patterns, facilitating planning, influencing perception, attention, memory retrieval, and behavior in a top-down way. However, the neuronal and computational mechanisms supporting goal and subgoal processing remain elusive.

In particular, it is only recently that model-based computations and planning have been studied experimentally. A series of monkey studies revealed that representations of goal and subgoal locations are elicited during path planning in lateral prefrontal cortex [3] and that this area guides multistep planning at the level of action goals and not motor actions [4]. A human study revealed the importance of striatum, medial temporal lobe and frontal cortex for navigation planning [5]. In rodents, model-based computations have been linked to a neuronal circuit involving the hippocampus and the ventral striatum [6]. Still, we have incomplete knowledge on (if and) how the brain implements model-based computations.

From a computational perspective, the benefits of goals and subgoals have been studied at three different timescales: learning, inference, and control. During *learning*, subgoals permit learning more efficiently by reducing the search space. This is well exemplified by *temporal abstraction* methods in hierarchical reinforcement learning (HRL) such as the Options framework [7]. Options can be conceptualized as sort of macro-actions whose termination conditions are subgoals. During *planning*, subgoals reduce the search space by permitting planning at a higher level of temporal abstraction (i.e., at the level of macro-actions), see also [8]. During *control*, subgoals permit maintaining the smallest possible information in working memory that is sufficient for task achievement [9] and supports efficient monitoring processes. Despite these progresses, we still lack an integrative theory of the computational role of subgoals in learning, inference, and control.

We argue that the main role of subgoals is reducing the complexity of learning, inference, and control tasks, by guiding the realization and selection of more compact control *programs*. A *program* can be defined as the sequence of actions necessary for the transition from an initial state s to a subgoal state sg. We assume that a program can be determined from a policy π if s and sg are known. Key to our formulation is the conversion of the length of a program (i.e., the number of actions necessary to reach sg from s) into a probability by following principles of descriptive complexity [10]. This approach formalizes the "Occam's razor" principle: a priori, among the strings that represent the procedures returning an output, "simpler" strings (i.e., strings with low descriptive complexity) are more probable. Our formalization uses information-theoretic measures based on Solomonoff's Algorithmic Probability and Kolmogorov Complexity [11, 12].

2 Methods and results

To test the hypothesis, we realized a Dynamic Bayesian Network (DBN) [13] that infers subgoals and policies by considering the descriptive complexity of the resulting programs. The inference uses the graphical model described in Fig. 1. In the model, the transition $P(\pi|SG, S)$ captures the concept of an Option but is expressed in a probabilistic way. Note that we focus on inference, not learning; for this reason the DBN structure and parameters are assumed to be known.

We cast planning and policies selection as *probabilistic inference* problems, see [14, 15, 16, 17, 18, 19, 20]. The inference follows the pseudocode of Algorithm 1. Intuitively, the goal of the inference is finding a policy running from the initial state s_t to a final goal state s_{goal}, which are assumed to be known. Although a policy can be found that covers the whole trajectory from s_t to s_{goal}, the resulting inference would be very costly and often infeasible for even moderately large state spaces. A useful solution in HRL is splitting the search into more manageable subgoals and corresponding Options. In our formulation, the choice of subgoals and policies is driven by considerations of minimum description length.

Algorithm 1 Pseudo-code of the inference procedure

```
$t = 0$
**set** $S_0$ to the starting state $s_0$
**sample** a subgoal state $sg_0$ from the prior probability distribution $p(SG_0)$
**sample** a policy $\pi$ from the conditioned probability distribution $p(\Pi|sg_0, s_0)$
**determine** the action $a_0$ depending on $\pi$ and $s_0$
**set** the termination condition state $f_0$ according to $p(F_0|sg_0, s_0)$
**while** ($F_t \neq 2$) **do**
  $t = t + 1$
  **determine** the state $s_t$ by means of $p(S_t|a_{(t-1)}, s_{(t-1)})$
  **sample** the state $sg_t$ from the conditioned probability distribution $p(SG_t|f_{(t-1)}, sg_{(t-1)})$
  **sample** a policy $\pi$ from the conditioned probability distribution $p(\Pi|sg_t, s_t)$
  **set** the action $a_t$ depending on $\pi$ and $s_t$
  **evaluate** the termination condition variable $F_t$ according to $p(F_t|sg_t, s_t)$
**end while**
```

The inference uses the initial state s_t as a clamped (i.e., observed) state, and the goal state s_{goal} as a prior on the subgoal node SG (this value $P(s_{goal})$ is the only parameter of the model). Setting goals as priors distinguishes our approach from planning-as-inference methods and is similar to the *active inference* scheme of [21].

The inference also uses additional priors on SG that essentially indicate the more likely subgoals in the environment. In HRL the problem of finding useful subgoals for Options is widely debated; most studies have assessed that *bottlenecks* (e.g., a door in a house-like navigation domain) are often useful subgoals [22, 23]. To extract subgoals we used a method inspired by [9] that consides for each state "the amount of Shannon information that the agent needs to maintain about the current goal at a given state to select the appropriate action". We inflect this measure in a probabilistic way by computing the probability that a subgoal sg is the output of some program given each of state s and each of the policy π. Thus, the a priori probability of a generic state to be a subgoal depends on how many programs halt in that state and how long they are (see [11] for a similar method).

Because the number of policies to be searched in all except the most trivial environments is huge, we adopt a *sampling* method: importance sampling [11]. During the sampling, the probability of selecting a specific policy π depends on the length of the *program* that can be generated from π and that permits a transition from the currently examined state s and the currently examined subgoal sg. (Remember that a *program* is the sequence of actions necessary for a transition from s to sg; this length can be exploited to estimate the related a priori probability using the methods devised by Solomonoff and Kolmogorov.) This method returns pairs of subgoals and policies (or in other words, Options) that would (ideally) specify the shortest possible programs from the initial to the goal state. As we noted before, at least in principle this has benefits for both inference and control. During inference, this method permits searching through a smaller search space. During control, it permits achieving goals using the shortest path while at the same time having the smallest cognitive load (e.g., as shorter programs require fewer bits of information to be described, working memory load is alleviated).

The role of the node F is monitoring (sub)goal achievement and guiding the transitions between subgoals (see [20]). When the current state s is the same as the currently selected subgoal sg, a *rest policy* π_ε (i.e., a specific policy associating to every state a "rest" action ε) is selected. The node F thus determines the transition to a new subgoal (selected during inference and incorporated in the transition $SG_t \rightarrow SG_{t+1}$). When the final goal s_{goal} is reached, the transitions end.

We present simulated experiments in a "four-rooms" scenario (similar to [7]) aiming at comparing the performance of our method (which uses subgoals) with an equivalent one that does not use subgoals. In the comparison we consider the number of successfully reached goals, the optimality of behavior (expressed here as the length of the path to achieve a goal), the complexity of the inference (i.e., the number of inferential cycles necessary to infer a control policy) and the complexity of the control (i.e. the cumulative number of bits in working memory necessary to achieve the task, see [9]).

Fig. 1 shows the synthetic discrete world we used. It has 18 states $S = \{s_1 \ldots, s_{18}\}$ and is composed of four "rooms" with a single connection among them (S7 and S12). Even in this simple scenario, the number of potential policies is in around seven millions, making exact inference impracticable. We made 10 simulation runs per number of particles (50, 100 and 1000 particles) with two different modalities. In the first (*without-subgoals*) modality the probability of choosing a subgoal different from the goal state is zero ($P\left(SG \neq s_{goal}\right) = 0$). In the second (*with-subgoals*) modality a discrete probability distribution on the subgoals is used ($P\left(SG \neq s_{goal}\right) > 0$). In the experiments we assumed s_1 as starting state and s_{18} as goal (notice that the probability of choosing the goal state s_{18} is raised to make it the most probable state). The priors on subgoals (calculated using the aforementioned method inspired to [9]) are shown in gray scale in Fig. 1.

Fig. 2 shows the distribution of subgoals found by our inference procedure averaged on the different runs. Results show that in the tested environment, strategies including two subgoals before the actual goal (with a total of three subplans) are more successful than others. Successful examples include for example [S2, S3, S18] and [S16, S17, S18]. Tab. 2

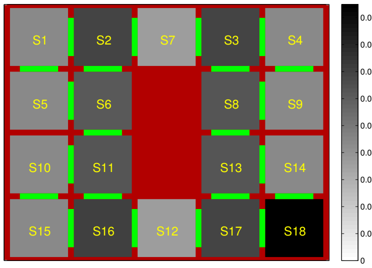


Figure 1: Environment representation with 18 states and subgoal priors depicted in gray scales (S18 is the goal state). Green and red bars represent doors and walls, respectively.

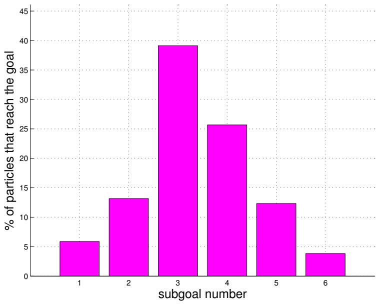


Figure 2: Mean subgoal distribution of the successful strategies. Note that the goal is included so a subgoal of 1 indicates that no additional subgoals were selected.

# of particles	% of success $P(SG \neq s_{goal}) = 0$	% of success $P(SG \neq s_{goal}) > 0$
50	29	45
100	33	47
1000	36	50

Table 2: Percentage of particles that correctly find a plan to the goal, for different number of particles (50, 100 and 1000).

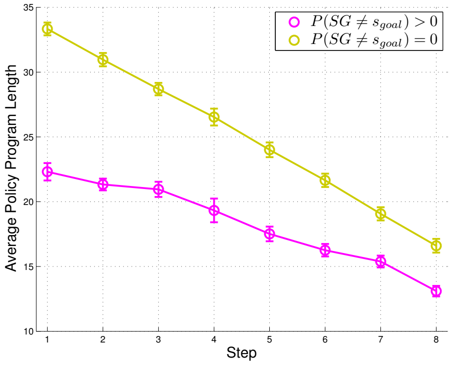


Figure 3: Average Program Length of policies per step for the two modalities of execution (standard deviation shown).

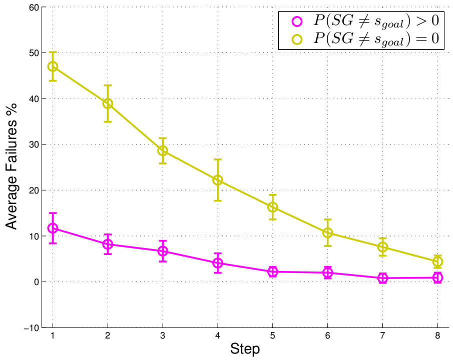


Figure 4: Average Failures Percentage of particles that do not represent a successful strategy (standard deviation shown).

shows the differences between the percentage of successful strategies carried out in the *without-subgoals* and *with-subgoals* modalities. The latter strategy achieves a better performance by using subgoals to split the search space.

Fig. 3 shows the average program length of the policies per step in the two modalities of execution (*with-subgoals* is pink, *without-subgoals* is yellow) for $N = 100$ particles and averaged on 10 runs (we obtained similar results with 50 and 1000 particles). This measure is related to the working memory necessary for inference and control, suggesting that the method using subgoals requires less memory resources. Fig. 4 shows the average percentage of particles that fail to find a suitable strategy (with $N = 100$). The results show that the inference method using subgoals is more efficacious, especially in the first steps. The percentage of failures is stable in all the steps, suggesting robustness of the method.

3 Conclusions

We proposed that goals and subgoals help lowering the description complexity of task-relevant information during learning, inference, and control. We presented preliminary evidence suggesting that model-based decision-making can use subgoals to lower the descriptive complexity of the planned policies and programs.

From the computational perspective, our proposal links to *planning-as-inference* [14, 15], which uses probabilistic inference to reach desired rewards [18] or goals [19, 20] and to *active inference* where goals are used as Bayesian *priors* in a variational probabilistic scheme that minimizes free energy [21]. Similarly, we use goal states as priors but we also consider subgoals and use descriptive complexity to evaluate candidate policies. Still, in large environments searching through all the possible policies is inefficient. This problem can be alleviated by assigning priors to policies (depending e.g., on past searches or the average length of their associated programs) or "caching" them. This could permit at least in principle to form libraries of Options or skills that can be reused across families of problems, as in *transfer learning* [24]. A further implication of this method is that goals and subgoals guide information compression in the cortical hierarchies by biasing which control programs are stored. The idea that information compression is a key organizing principle brain hierarchies has received some attention in neuroscience [25] but its empirical validity remains to be tested.

Besides, our study can offer a normative perspective on planning and subgoal processing in living organisms. The monkey PFC encodes a sequence of activation of goals and subgoals during a delay period prior to action [4, 3] and monitors goals at feedback time [26]. The proposed model suggests a possible computational principle for the encoding and monitoring of subgoal sequences. Further evidence is necessary to assess the biological plausibility of the model.

References

[1] E. K. Miller and J. D. Cohen. An integrative theory of prefrontal cortex function. *Annu Rev Neurosci*, 24:167–202, 2001.

[2] Richard E Passingham and Steven P Wise. *The neurobiology of the prefrontal cortex: anatomy, evolution, and the origin of Insight*, volume 50. Oxford University Press, 2012.

[3] Naohiro Saito, Hajime Mushiake, Kazuhiro Sakamoto, Yasuto Itoyama, and Jun Tanji. Representation of immediate and final behavioral goals in the monkey prefrontal cortex during an instructed delay period. *Cereb Cortex*, 15(10):1535–1546, Oct 2005.

[4] Hajime Mushiake, Naohiro Saito, Kazuhiro Sakamoto, Yasuto Itoyama, and Jun Tanji. Activity in the lateral prefrontal cortex reflects multiple steps of future events in action plans. *Neuron*, 50(4):631–641, May 2006.

[5] Dylan Alexander Simon and Nathaniel D Daw. Neural correlates of forward planning in a spatial decision task in humans. *J Neurosci*, 31(14):5526–5539, Apr 2011.

[6] Matthijs A A. van der Meer and A.D. Redish. Expectancies in decision making, reinforcement learning, and ventral striatum. *Frontiers in Neuroscience*, 4:6, 2010.

[7] R.S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[8] Milos Hauskrecht, Nicolas Meuleau, Leslie Pack Kaelbling, Thomas Dean, and Craig Boutilier. Hierarchical solution of markov decision processes using macro-actions. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 220–229. Morgan Kaufmann Publishers Inc., 1998.

[9] Sander G van Dijk and Daniel Polani. Grounding subgoals in information transitions. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on*, pages 105–111. IEEE, 2011.

[10] M. Li and P. Vitànyi. *An introduction to Kolmogorov complexity and its applications*. Springer, 2nd edition, 1997.

[11] David J. C. Mackay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 1st edition, June 2002.

[12] Juergen Schmidhuber. Discovering neural nets with low kolmogorov complexity and high generalization capability. *Neural Networks*, 10:10–5, 1997.

[13] Kevin P. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, UC Berkeley, Computer Science Division, 2002.

[14] H. Attias. Planning by probabilistic inference. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[15] Matthew Botvinick and Marc Toussaint. Planning as inference. *Trends Cogn Sci*, 16(10):485–488, Oct 2012.

[16] Giovanni Pezzulo and Francesco Rigoli. The value of foresight: how prospection affects decision-making. *Front. Neurosci.*, 5(79), 2011.

[17] Giovanni Pezzulo, Francesco Rigoli, and Fabian Chersi. The mixed instrumental controller: using value of information to combine habitual choice and mental simulation. *Front Psychol*, 4:92, 2013.

[18] Alec Solway and Matthew M. Botvinick. Goal-directed decision making as probabilistic inference: A computational framework and potential neural correlates. *Psychol Rev*, 119(1):120–154, Jan 2012.

[19] Marc Toussaint and Amos Storkey. Probabilistic inference for solving discrete and continuous state markov decision processes. In *Proceedings of the 23rd international conference on Machine learning*, pages 945–952. ACM, 2006.

[20] Deepak Verma and Rajesh P. N. Rao. Planning and acting in uncertain environments using probabilistic inference. In *IROS*, pages 2382–2387. IEEE, 2006.

[21] Karl J Friston, Jean Daunizeau, and Stefan J Kiebel. Reinforcement learning or active inference? *PLoS One*, 4(7):e6421, 2009.

[22] M. Botvinick, Y. Niv, and A. Barto. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, 119(3):262–280, 2009.

[23] Nir Lipovetzky and Hector Geffner. Width and serialization of classical planning problems. In *ECAI*, pages 540–545, 2012.

[24] G.D. Konidaris and A.G. Barto. Building portable options: Skill transfer in reinforcement learning. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2007.

[25] S J. Kiebel, J. Daunizeau, and K J. Friston. A hierarchy of time-scales and the brain. *PLoS Comput Biol*, 4(11):e1000209, Nov 2008.

[26] Satoshi Tsujimoto, Aldo Genovesio, and Steven P. Wise. Frontal pole cortex: encoding ends at the end of the endbrain. *Trends Cogn Sci*, 15(4):169–176, Apr 2011.