Distinct policy identification via model-based belief update

Panos Alefantis[1], **Zhe Li**[3], **Noushin Quazi**[2], **Dora Angelaki**[1], **Xaq Pitkow**[2]

1.New York University, 2.Carnegie Mellon University, 3.Baylor Collage of Medicine

Abstract

Both artificial and biological agent face the challenge of interacting with the world based on incomplete and noisy observations. In general the agent relies on a latent representation that integrates past experience, and makes decisions based on it. When the agent has an internal model of the environment, its latent representation can be a distribution about the unknown state, *i.e.* belief. A major challenge in using this framework is that the model's belief dynamics are generally difficult to derive by hand, and this approach is certainly not scalable to complex environments. We developed an algorithm that automates this process, requiring only a simulator of the environment dynamics that the agent assumes. Our approach uses this simulator to estimate dynamics of the internal model beliefs. We applied our algorithm to explain behavioral data of a monkey foraging experiment, where the monkey needs to navigate between three food boxes and decide when to open the box based on noisy color cues. We found the beliefs span on a low dimensional manifold in the belief space, and they are well organized by behaviorally relevant variables. We also developed an algorithm to identify distinct modes of behavior, accommodating potentially non-stationary policies. We verified the algorithm on simulated behavior data, and show an initial application to real behavior, revealing changes in strategy over time. In summary, our tools allow us to infer latent representation of model-based agents and the policy defined over this representation, which will enable a search for neural evidence of possible internal model properties that are not directly measurable.

Automated model-based belief update

In a partially observable Markov decision process (POMDP), an agent can act upon the belief state following Markovian dynamics. The belief state b is a distribution over environment states $s \in \mathcal{S}$, and is updated by agent action a and the new observation o according to Bayes' rule:

$$b'(s') \equiv P(s'|a, o, b) = \frac{\mathcal{O}(s', a, o) \sum_{s_i \in \mathcal{S}} \mathcal{T}(s_i, a, s') b(s_i)}{\sum_{s_j \in \mathcal{S}} \mathcal{O}(s_j, a, o) \sum_{s_i \in \mathcal{S}} \mathcal{T}(s_i, a, s_j) b(s_i)}. \quad (1)$$

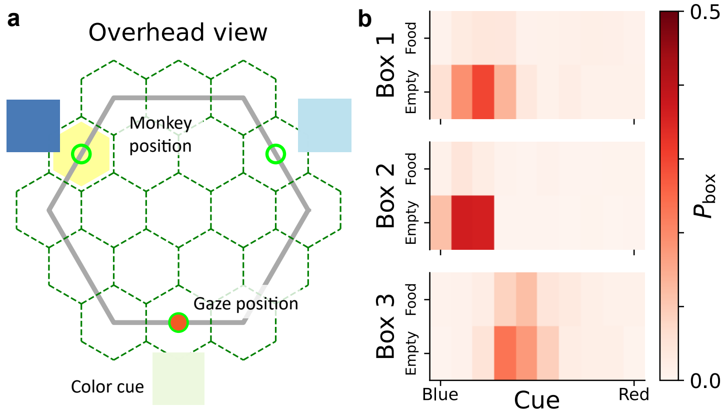

Figure 1: Model-based belief in an foraging task. (a) Overhead view of an arena in which three food boxes (green circles) are set up. The space is discretized to hexagonal tiles, with the monkey position marked as yellow and where it is looking at marked by an orange dot. Color cues on each food box is shown by the nearby patches. (b) Beliefs about food boxes computed based on the assumed internal model. The belief is an distribution over possible box states, and is updated by the agent's action and new observation from the environment.

However, the full observation model $\mathcal{O}(s', a, o) = P(o|a, s')$ and transition function $\mathcal{T}(s, a, s') = P(s'|s, a)$ is difficult to specify. Even when the full model is available, it is challenging to efficiently perform the marginalization in Eq. 1. What is easier to construct is a simulator that can generate (s, a, s', o) tuples, so we developed a sample-based algorithm to perform approximate belief updates suitable for a broader range of applications.

The algorithm first draws a large number of states $\{s_i\}$ from current belief $b(s)$, and applies the action a on each state using the provided simulator to get a set of candidate new states $\{s'_i\}$. Importance weights are assigned to each s'_i by a separately learned observation model $\mathcal{O}(o|s')$. Finally the new belief $b'(s')$ is estimated from the weighted set of new states $\{s'_i\}$. We estimated b' stochastically by drawing from the posterior distribution $P(b'|\{s'_i\})$; the more sample states are used, the less noisy b' will be.

We applied the algorithm to a monkey foraging task in which the animal needs to estimate when food will become available at any of several boxes. Assuming the animal has already learned a correct internal model, we observed that its belief basically follows a timer whose speed is determined by box quality (Fig. 1).

Low-dimensional belief manifold

Since each belief b represents a distribution over $\mathcal{S}$, the belief space is high-dimensional even when the state space is moderately discretized. We used a variational autoencoder (VAE) to compress the belief traces, and found the reconstruction loss hardly decreases when the hidden layer size D_z is larger than a certain value. In our specific example, the loss is minimal for around 5 dimensions, but $D_z = 2$ (Fig. 2b) or $D_z = 3$ (Fig. 2c) gives reasonably faithful reconstructions as well. We also found the embedded belief is organized by several behaviorally relevant variables, such as the current position of the monkey, and the time since each box was last opened.

Hidden Markov model of policy dynamics

We further investigate whether the animal is using a single stationary policy or switching between multiple policies. We assume this strategy changes following a hidden Markov model (HMM). Computationally, we jointly optimize a set of policies $\{\pi_i(a|b)\}$ based on the beliefs that our algorithm infers, choosing policy i at each time to maximize the overall likelihood of the agent's actions and the transitions between the policies according to the HMM. We require each individual policy to have low entropy and to be dissimilar from each other in terms of KL divergence.

We first validated our algorithm on a simulated episode, in which the agent either behaves like a real monkey, or follows a random policy that moves much less coherently. The algorithm successfully identified the two distinct policies (Fig. 2a). We visualize both policies on a 2D belief space (Fig. 2b). The same method was applied to real monkey data, allowing three candidate policies. In future analyses, we will compare different assumptions about the number of policies, and find the optimal segmentation of monkey data. This will help us interpret the behavior pattern and possibly find the neural signals associated with them.

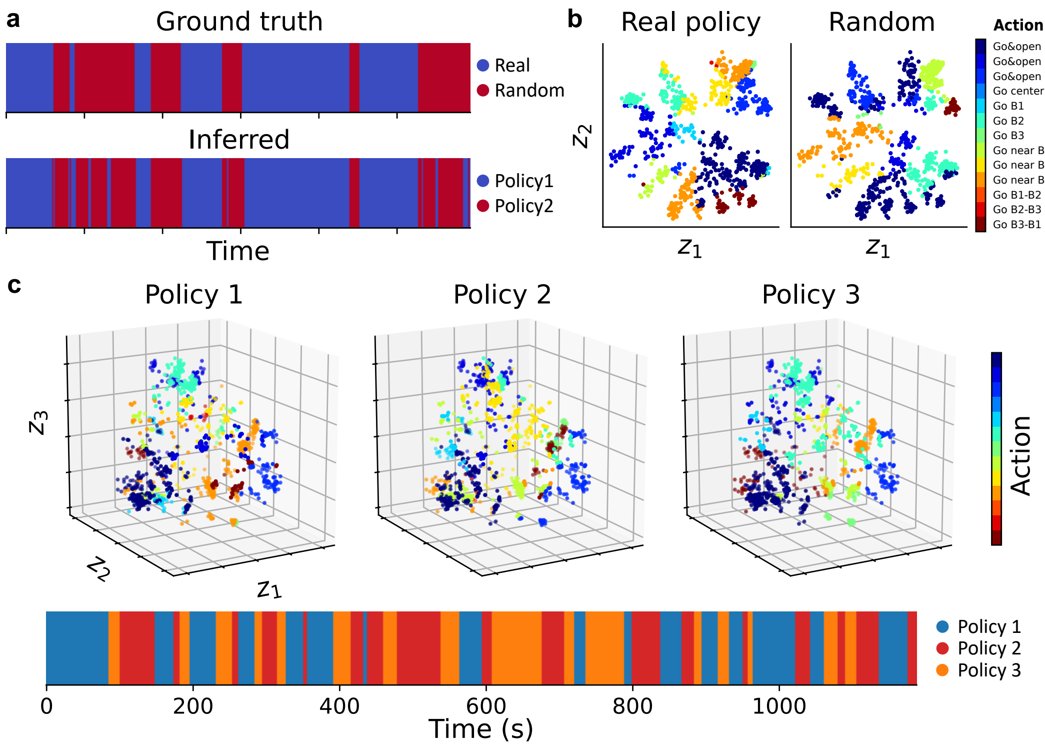

Figure 2: Identification of discrete policies from agent behavior. (a) A simulated agent follows either the monkey behavior or a random policy alternatively. The inferred policy identity is compared against ground truth. (b) Visualization of inferred policies in 2-dimensional belief space. Each dot is the internal belief at one time step, colored by the most probable action for it. (c) 3 discrete policies identified from real monkey foraging experiment. Animal's internal belief is embedded to 3-dimensional latent space in this example.