COPerMix
European ITN project

Finite-State Controllers for Olfactory Navigation

Kyrell Vann B. Verano, Emanuele Panizon, Antonio Celani
The Abdus Salam International Center for Theoretical Physics - ICTP

1 Concept

The olfactory navigation problem consists in a searcher (agent) that has to reach in the minimal time an odor source (target). The searcher does not have any a priori knowledge of the spatial position of the source but can only use odor detections as cues to reach the target. How are these intermittent cues translated in an efficient strategy to locate an odor source? Because of the turbulent mixing of air and odor particles (signal) in the environment, odors arrive to the searcher sparsely and randomly. Hence, building an effective strategy requires a memory that efficiently encodes past observations and actions. Here, we approach the problem as a memory-augmented partially observable Markov decision process with limited memory. These memory states are the finite-state controllers that govern the search process.

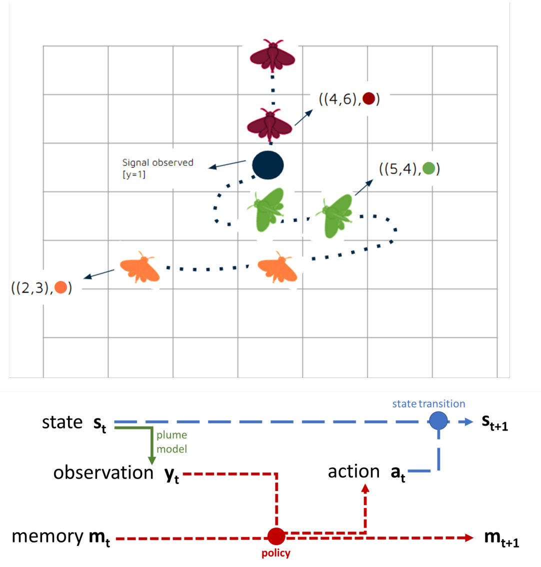

Figure 1: Sample of state representation and an illustration of the memory augmented POMDP

2 Algorithm

We frame the problem in a 2-dimensional grid-world. The state of the agent is the combination of its position (known) and the position of the source (unknown). The actions available to the agent are: `left`, `right`, `up`, `down` (optionally, also the additional action `stay`). From either a model of odor detection or from data, we derive a probability distribution of signals (observations) received by the agent. At each step, the agent incurs a fixed cost until it reaches a target range centered around the source. In this way, minimizing the costs also minimizes the time it takes to reach the target. The policy is how the agent acts given a memory state and observation. At each step the agent takes an action and updates its memory state depending on observation received, the action taken, and the current memory state. This policy is parameterized and we implement a policy gradient to maximize the average rewards given a distribution of signals or observations.

3 Code Availability

We have created a github repository that contains the implementation of the algorithm: https://github.com/kyrellverano/Finite-Memory-Controller. The code allows to compute the optimal policy, evaluate its performance in terms of search time and illustrate trajectories of the agent. Demos of optimal policies for given choices of the environment are given.

4 Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement N°956457.