Deep Bayesian Nonparametric Learning of Rules and Plans from Demonstrations with a Learned Automaton Prior

Brandon Araki\textsuperscript{1}, Kiran Vodrahalli\textsuperscript{2}, Thomas Leech\textsuperscript{1,3}, Cristian-Ioan Vasile\textsuperscript{1}, Mark Donahue\textsuperscript{3}, and Daniela Rus\textsuperscript{1}

\textsuperscript{1}CSAIL, MIT \hspace{1cm} \textsuperscript{2}Columbia University \hspace{1cm} \textsuperscript{3}MIT Lincoln Laboratory

**Goal:** Learn rules and policies for rule-based environments in an **interpretable** and **manipulable** way

**Interpretability**

*Pick up a sandwich or hamburger and pack it, then pick up a banana and pack it*

Represent rules as a finite state automaton (FSA)

---

**Manipulability**

FSA is an input to a recursive planner, so changes to the FSA result in changes to the policy

---

**Data**

- A low-level environment
- A dataset of trajectories

**Learn**

- An FSA representation of the rules
- A reward function

---

**Summary of our Approach**

1. Model the environment as a POMDP
   \[
   (S \times \mathcal{P} \times \mathcal{E}, A, T \times M \times TM, \mathcal{R}, S \times \mathcal{P}, O, \gamma(d))
   \]
2. Parameterize the policy as a function of an FSA, a reward function, and a low-level environment
   \[
   Q^{t+1}(s, f, a) \leftarrow R(s, f, a) + \gamma \sum_{s', o} \tau(s'|s, o)V^{t}(s', f) \\
   \bar{V}^{t+1}(s, f) \leftarrow \max_{a} Q^{t+1}(s, f, a) \\
   V^{t+1}(s, f) \leftarrow \sum_{F \in F} TM(f|f, M(a))\bar{V}^{t}(s, f')
   \]
3. Model the policy rollout on the POMDP as an HMM
4. Use variational inference to infer the latent variables of the HMM including FSA and reward function

---

**Experiments and Results**

<table>
<thead>
<tr>
<th>Gridworld</th>
<th>Lunchbox Packing</th>
<th>Dungeon</th>
<th>Driving Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Gridworld Diagram" /></td>
<td><img src="image" alt="Lunchbox Packing Diagram" /></td>
<td><img src="image" alt="Dungeon Diagram" /></td>
<td><img src="image" alt="Driving Domain Diagram" /></td>
</tr>
</tbody>
</table>

- **Gridworld**: Go to a, then b, then c, then d
- **Lunchbox Packing**: Pick up sandwich (a) or hamburger (b), put it in the lunchbox (d), then pick up banana (c) and put it in the lunchbox (d)
- **Dungeon**: Go to the goal (g), can’t pass through Door x (dx) until Key X is obtained (ke)
- **Driving Domain**: Go to the goal and avoid work zones and obstacles
  - Stop if there’s a red light in front of you and go if there’s a green light
  - Prefer the right lane to the left lane