Meta-Reinforcement Learning

- Leverage prior experience to quickly learn new tasks
- Meta-training: Extract fast RL algorithm
- Meta-testing: Quickly adapt to new tasks

Challenge: Meta-RL requires well defined reward functions

Problem with Reward

- Hard to design
- Hard to provide
- Hard to learn from

Human in the loop RL

- Replace reward with human feedback
- Language provides natural form of supervision
- Contains more bits of info than scalar reward

Framework

Problem: Solve new tasks quickly via interactive language corrections given prior experience on related tasks.

Problem Setup

- A human guides the agent with language corrections
- Agent incorporates correction to move closer to the solution
- Ground language using multi-task, meta-learning framework

Algorithm

Overview:

- Ground language corrections during training using expert policies
- Solve test tasks with only a few corrections

Model:

- Map corrections to changes in agent’s behavior
- Incorporate previous trajectories and corrections of them
- Process language instruction

Training Procedure:

- Use DAgger like procedure conditioned on corrections
- Assume access to expert policies and human labeler during training

Tasks

- Multi-Room Object Manipulation
  - Tests underspecified instruction (partially observed env)
  - Instructions are move specific object to specific goal.
  - Corrections guide agent to room locations of object and goal.
- Robotic Object Relocation
  - Tests ambiguous instruction (human has imprecise goal)
  - Corrections guide the object to correct location

Experimental Results

Example rollouts

Try our demo at https://lgplserver.com

Results

<table>
<thead>
<tr>
<th>Env</th>
<th>Instruction Full Info</th>
<th>MIVOA (Instr.)</th>
<th>MIVOA (Full Info)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Room</td>
<td>0.055</td>
<td>0.64</td>
<td>0.82</td>
</tr>
<tr>
<td>Obj Relocation</td>
<td>0.64</td>
<td>0.96</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Table: Mean completion rates on test tasks. $c_i$ denotes agent has received $i$ corrections

Meta-test complexity

- GPL (ours) achieves high test task completion with just 5 trajectories and corrections without using reward
- RL takes many more test trajectories and requires test reward
- GPR replaces language with reward, demonstrating language conveys more information

Ablations

<table>
<thead>
<tr>
<th>Ablations</th>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$c_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>0.055</td>
<td>0.64</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No instruction</td>
<td>0.055</td>
<td>0.45</td>
<td>0.62</td>
<td>0.73</td>
<td>0.78</td>
<td>0.93</td>
</tr>
<tr>
<td>No trajectory</td>
<td>0.057</td>
<td>0.44</td>
<td>0.62</td>
<td>0.70</td>
<td>0.76</td>
<td>0.77</td>
</tr>
<tr>
<td>Only immediate correction</td>
<td>0.067</td>
<td>0.49</td>
<td>0.44</td>
<td>0.58</td>
<td>0.59</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Table: Ablation Experiments analyzing the importance of various components of the model.