Self-Paced Context Evaluation for Contextual Reinforcement Learning

Theresa Eimer¹, André Biedenkapp², Frank Hutter² ³, Marius Lindauer¹

¹Leibniz University Hannover | ²Albert-Ludwigs University Freiburg | ³Bosch Center for Artificial Intelligence

Contextual Reinforcement Learning

- Extending RL with task instances
- Each instance is defined by a context, e.g. the pole length in CartPole
- Requires generalization to solve

SPaCE in a Nutshell

- Creating instance curricula using the agent’s value function \( V \)
- Change in \( V \) as proxy for agent capability (PIC)
- Difficulty rating: difference in \( V \) between training steps
- Start training on few instances and increase over time
- New instances are used whenever \( V \) converges

Experimental Results

**Contextual CartPole Test Performance**

**Contextual CartPole Curriculum**

**PointMass Test Performance with SPDRL [Klink et al. 2020]**

<table>
<thead>
<tr>
<th>Environment</th>
<th>Domain knowledge independent</th>
<th>Improved sample efficiency</th>
<th>Better overall generalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evolution</td>
<td>X</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Curricula through Self-Play [Shahbabaar et al. 2018]</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Student-Teacher approaches [Madisetti et al. 2009]</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Difficulty appropriate instance sampling [Klink et al. 2020]</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SPaCE (ours)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Main Take-Aways:

- Better generalization performance as well as better sample efficiency during training
- Difficulty progression in curricula is not always linear, but successfully goes from easy to difficult

Why use SPaCE?

- Creating instance curricula using the agent’s value function \( V \)
- Change in \( V \) as proxy for agent capability (PIC)
- Difficulty rating: difference in \( V \) between training steps
- Start training on few instances and increase over time
- New instances are used whenever \( V \) converges