Numeric Reward Machines

Kristina Levina[1,2], Nikolaos Pappas[1], Athanasios Karapantelakis[2], Aneta Vulgarakis Feljan[2], Jendrik Seipp[1]

[1] Linköping University, Linköping, Sweden

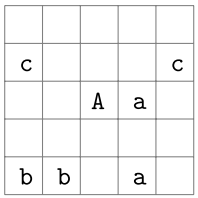
[2] Ericsson Research, Stockholm, Sweden

{kristina.levina, nikolaos.pappas, jendrik.seipp}@liu.se {aneta.vulgarakis, athanasios.karapantelakis}@ericsson.com

Motivation

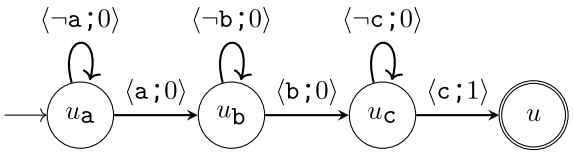
- Reward machines (RMs) can handle **non-Markovian rewards**. This is useful for partially observable or sparse environments.
- Agents with access to the RM structure can simulate experiences, speeding up learning. Thus, learning is **sample-efficient**.
- However, RMs only accept Boolean features. Hence, we aim to extent RMs with **numeric features** so that RMs can excel in inherently numeric tasks like energy optimisation.

Environment: Craft Domain



Picture depicts an example map in the Craft domain, a simple grid world with four-connected cells. Agent A and objects of types a, b, and c are located on the map. Agent A can visit any object of the instructed type.

Original Boolean RM

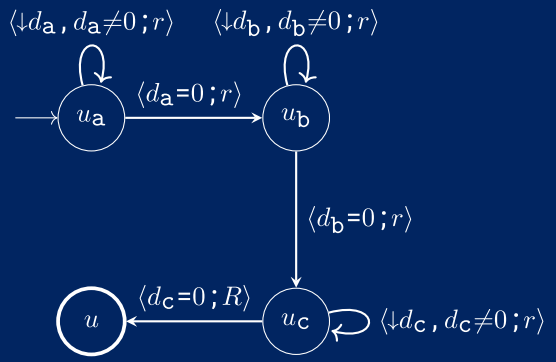


Boolean RM was introduced by Icarte et al. In the example task, the agent should visit a, b, and c in order. **Boolean features** a, b, and c become `True` upon the agent's arrival at the respective cell. The rewards are sparse. **Automatic reward shaping** can be applied.

RM Structure Exploitation

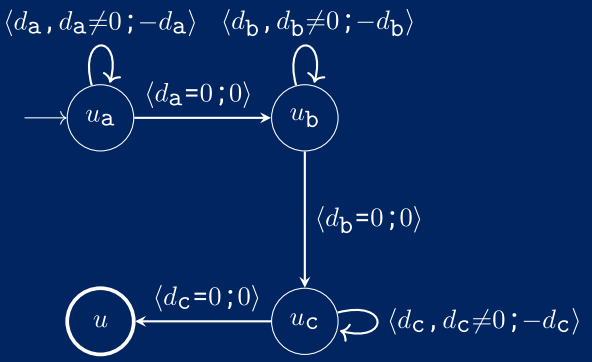
- (BASELINE) **QRM: cross-product *Q*-learning** over the environment and RM states. QRM handles non-Markovian rewards but doesn't offer sample efficiency.
- (RM EXPLOITED) **CRM: *Q*-learning with RM counterfactual experiences**. The agent simulates experiences for each RM state per single interaction with the environment!
- (RM EXPLOITED) **HRM: hierarchical *Q*-learning with RMs**. The agent learns high- and low-level policies to transition between RM states greedily.

Numeric–Boolean RM



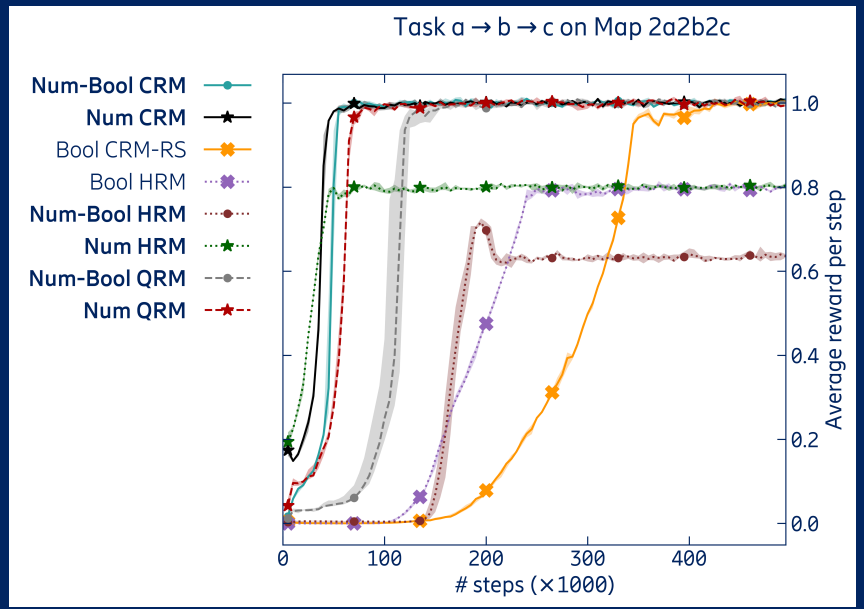
Agent–target Manhattan distance d (numeric feature) is translated to two Boolean features: $\downarrow d$ (d decreases) and $d{=}0$ (target is reached). If d decreases, the agent is rewarded with fixed $r > 0$. If the target is reached, transition to the next RM state occurs. Thus, the agent is positively reinforced to approach the target.

Numeric RM



Numeric feature d is used along with Boolean feature $d{=}0$. The latter governs the transitions between the RM states as previously, and the former is used for rewarding the agent with $-d$. In this way, the agent is rewarded for each experience by the negative distance to the target.

Results



Conclusion

Numeric–Boolean and numeric RMs speed up learning in comparison with Boolean RMs with automatic reward shaping, while offering **interpretable reward acquisition**.

However, for complete realisation of the potential of numeric features, we need to allow numeric features govern both RM transitions and rewards.

Future Work

- Let **numeric features guide RM transitions** as well.
- Include change in numeric features into rewards.
- Test tabular domains with obstacles, continuous-space domains, and continuous-control tasks.
- Apply numeric RMs for energy optimisation of radio units in radio base stations.

Reference

LINKÖPING UNIVERSITY

ERICSSON

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM