Learning Compact Regular Decision Processes using Priors and Cascades

Ahana Deb[1] Anders Jonsson[1] Alessandro Ronca[2] Sadegh Talebi[3]

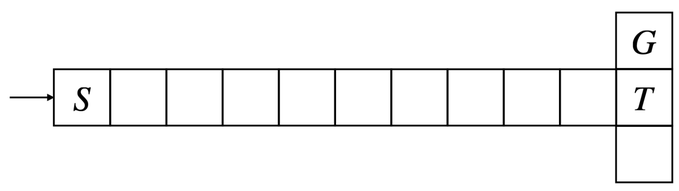
[1]*Universitat Pompeu Fabra* [2]*IRIS-AI* [3]*University of Copenhagen*

Regular Decision Processes

Episodic Non-Markov Decision Process $\langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}, \bar{R}, H \rangle$, where $\bar{T} : (\mathcal{A}\mathcal{O})^* \times \mathcal{A} \to \Delta(\mathcal{O})$ and $\bar{R} : (\mathcal{A}\mathcal{O})^* \times \mathcal{A} \to \mathbb{R}$ are functions of the entire interaction history $(\mathcal{A}\mathcal{O})^*$.

In a Regular Decision Process (RDP), the functions $\bar{T}$ and $\bar{R}$ depend *regularly* on the interaction history, i.e., $\bar{T}$ and $\bar{R}$ can be represented by a Probabilistic-Deterministic Finite Automaton (PDFA).

Example. T-maze [1] with corridor length $N = 10$ and noisy observations.



The observation at the initial position S indicates the position of the goal G at the end of the corridor for the current episode.

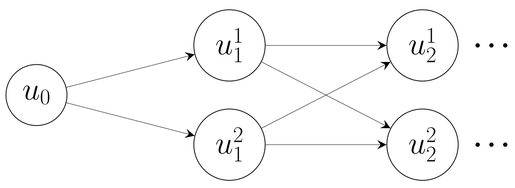
Objective

Given a dataset $\mathcal{D}$ of episodes, collected from an unknown RDP $\mathbf{R}$ and unknown behavior policy π^b, compute a near-optimal policy for $\mathbf{R}$, using the smallest $\mathcal{D}$ possible.

Goal: To learn a compact domain–specific cyclic (if applicable) automata, by incorporating (domain) prior knowledge.

AdaCT-H | State-of-the-art for PDFA Learning

AdaCT-H [3] (using language metric) computes the PDFA of the underlying RDP: builds the *graph of transitions* by comparing candidate states against the states already discovered.



AdaCT-H employs the *language metric*, defined below.

Language Metrics

Language metrics [3] for distributions over strings

For a class $\mathcal{X}$ of languages, the corresponding *language metric* is

$$L_{\mathcal{X}}(p_1, p_2) \coloneqq \max_{X \in \mathcal{X}} |p_1(X) - p_2(X)| \ , \ \text{where} \ \ p_i(X) \coloneqq \sum_{x \in X} p_i(x).$$

Language metrics for RL

Let $\mathcal{X}_{i,j}$ be a language family parameterized on two integers i and j

– $i \in [\![3]\!]$: the number of elements in $\mathcal{A}$, $\mathcal{O}$ and $\mathcal{R}$ considered jointly

– $j \in [\![t]\!]$: the length of subsequences considered for comparison

Examples:

$\mathcal{X}_{1,1}$: single instance of one action, one observation or one reward

$\mathcal{X}_{3,2}$: subsequence of two instances of triplets in $\mathcal{A}\mathcal{O}\mathcal{R}$

Contributions

- We introduce a novel notion of **priors** for automaton learning that allows for exploiting domain-related prior knowledge.
- We utilise the language metric introduced in [3] and introduce a novel algorithm that allows for learning **cycles** in the domain-specific automaton. This enables us to learn a significantly more compact RDP with cycles, which are crucial for scaling to larger, more complex environments.

Automata Cascades

Cascades offer a richer way to learn compact domain-specific automata by incorporating prior knowledge of the structure of states and transition function.

A *cascade* is an automaton $\mathbf{C} = \langle \Sigma, \mathcal{U}, \tau, u_0, \Omega, \theta \rangle$ given by the composition $\mathbf{A}_1 \ltimes \cdots \ltimes \mathbf{A}_d$ where every $\mathbf{A}_i = \langle \Sigma_i, \mathcal{U}_i, \tau_i, u_0^i \rangle$ is a *semiautomaton*.
A *component* of the cascade, $\mathbf{A}_i$, has input alphabet $\Sigma_i \coloneqq \mathcal{U}_1 \cdots \mathcal{U}_{i-1}\Sigma$, allowing it to read the states of the preceding components in addition to inputs from Σ. The output function $\theta : \mathcal{U}_1 \times \cdots \times \mathcal{U}_d \to \Omega$ of a cascade is over a factored state space, which makes it explicit how the function depends on the single state components.

Prior for RDPs

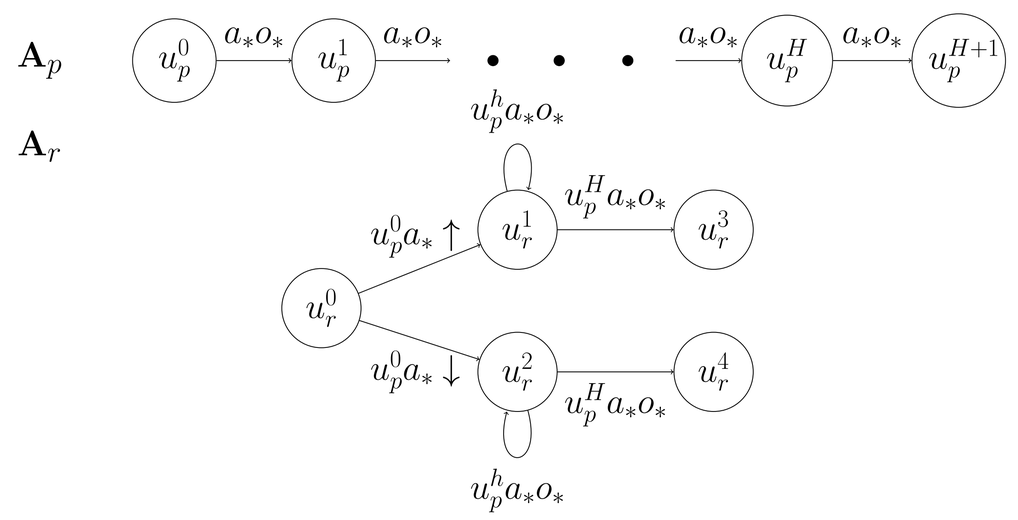
Priors allow for shaping the state space of an RDP with fundamental structures known a priori. Examples:

Markov Priors: Simple semiautomata that store the previous observation.

Timestep Priors: Allow for specifying that the current timestep in an episode may be a relevant feature in determining distributions over episode suffixes.

Spatial Priors: Allow for describing the physical space (its geometry) of a domain, and specify that the current position in such space.

Example:



Sample Complexity Analysis

Theorem 1. AdaCT–L$(\mathcal{D}, \delta)$ returns a minimal automaton $\mathbf{A}$ with probability at least $1 - 2AOUU_{\mathrm{p}}\,\delta$ when using a language set $\mathcal{X}$ that distinguishes $\mathbf{A} \ltimes \mathbf{A}$ under the behavior policy π^{b} with associated distinguishability $\mu_{\mathcal{X}}$ and the size of the dataset $\mathcal{D}$ is at least

$$|\mathcal{D}| \geq \widetilde{\mathcal{O}}\left(\frac{C^*_{\mathbf{R}} \log(1/\delta) \log|\mathcal{X}|}{d^*_{\mathsf{m}} \cdot \mu^2_{\mathcal{X}}}\right),$$

where $d^*_{\mathsf{m}} = \min_{u,ao} d^*(u, ao)$ is the minimum occupancy of π^*.

Experimental Evaluation

		FlexFringe			ADACT-H			AdaCT–L		
Name	H	U	r	time	U	r	time	U	r	time
Corridor	5	11	**1.0**	0.03	11	**1.0**	**0.01**	**3**	**1.0**	**0.01**
T-maze(c)	5	29	0.0	0.11	18	**1.0**	0.26	**5**	**1.0**	**0.15**
Cookie	9	220	**1.0**	0.36	91	**1.0**	**0.08**	**11**	**1.0**	**0.08**
Cheese	6	669	0.69 ± .04	19.28	1178	0.87 ± .03	12.11	**85**	**0.89 ± .04**	**7.27**
Mini-hall	15	897	0.33 ± .04	25.79	6098	0.86 ± .03	29.90	**65**	**0.87 ± .04**	**25.18**

Table: H, U are the horizon and the number of states in the learned automaton respectively, r is the average normalised reward (over 100 episodes) of the derived policy, and 'time' is the running time in seconds of automaton learning.

References

[1] B. Bakker. Reinforcement learning with long short-term memory. In *NeurIPS*, 2001.

[2] R. Baumgartner and S. Verwer. Learning state machines from data streams. In *ICGI*, 2023.

[3] A. Deb, R. Cipollone, A. Jonsson, A. Ronca, and M. S. Talebi. Offline RL in regular decision processes: Sample efficiency via language metrics. In *ICLR*, 2025.

Ahana Deb ahana.deb@upf.edu

Anders Jonsson anders.jonsson@upf.edu

Alessandro Ronca alessandro@ronca.me

Sadegh Talebi sadegh.talebi@di.ku.dk