State Abstraction Discovery in Model-Based Reinforcement Learning

PGMODays

EDF Lab Paris-Saclay, Palaiseau, France

November 20. 2024

Orso Forghieri (École polytechnique)

orso.forghieri@polytechnique.edu

Under supervision of

Hind Castel (Télécom SudParis)
Emmanuel Hyon (LIP6, Université Paris-Nanterre)
Erwan Le Pennec (École polytechnique)

Markov Decision Processes

- Observable State s_t, Action a_t, Reward r_t, Next state s_{t+1}
- Optimization problem : $\max_{\pi \in \mathcal{A}^S} \sum_{t \geq 0} \gamma^t r_t$, $\gamma = 0.99$

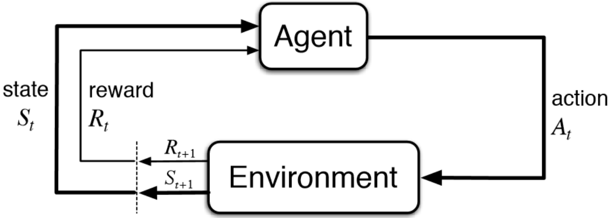
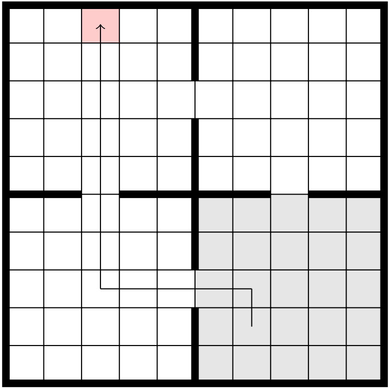


Figure 1: Principle of Reinforcement Learning [Sutton and Barto, 2018].

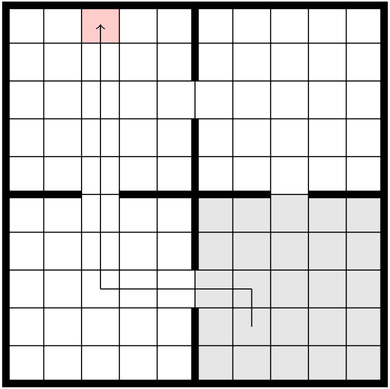
Four Rooms instance



Four rooms:

- $\mathcal{S} = [\![1, 100]\!]$, $\mathcal{A} = \{N, S, E, W\}$
- Reward: -1 until exit is reached, 0 otherwise.
- Step forward success with probability .8 (if step is doable)

Four Rooms instance



Naturally hierarchical!

- Spatial: each room as a single state
- Temporal: learn to exit a room

Hierarchical Reinforcement Learning

Why use HRL in MDPs?

- Solving large MDPs
- Enhancing explainability and interpretability of MDPs
- Ensuring solution quality

How to implement it?

- **Temporal Abstraction**: Subgoal discovery, meta-action learning
- **Spatial Abstraction**: Building MDPs approximation

Related works

Handling Large MDP Spaces:

- With factorization hypothesis (Siddiqi2010)
- Hierarchical approach (Hengst2012)
- State information abstraction (Coulom2006)

State aggregation and abstraction discovery:

- Abstraction discovery (Abel2016)
- Abstraction and MDP approximation (Ferrer2020, Abel2020)
- MDP solving acceleration (Jothimurugan2021)

Context and work

In this talk, we present:

- MDP solving context
- The HRL approach
- Our contribution[1]

We study a method in which we:

(i) Link MDP abstraction and Approximate Dynamic Programming
(ii) Estimate error induced by abstraction
(iii) Present a practical way to abstract MDP and solve them exactly
(iv) Conduct a numerical comparison using real-world models

[1] [Forghieri et al., 2024]

Table of Contents

Value-based Dynamic Programming

$$
\begin{aligned}
\text{Solving the MDP} &\iff \text{Maximizing upcoming rewards relatively to } \pi \\
&\iff \max_{\text{policy}} \mathop{\mathbb{E}}_{a_t=\pi(s_t)} \left[\sum_{t \geq 0} \gamma^t r_t \middle| s_0 = s \right] \\
&\iff \max_{\pi \in \mathcal{A}^{\mathcal{S}}} V^\pi(s) := V^*(s)
\end{aligned}
$$

Value Iteration approach: iterate the contraction

$$\mathcal{T}^* : V \to \max_{a \in \mathcal{A}} (R_a + \gamma \cdot T_a \cdot V)$$

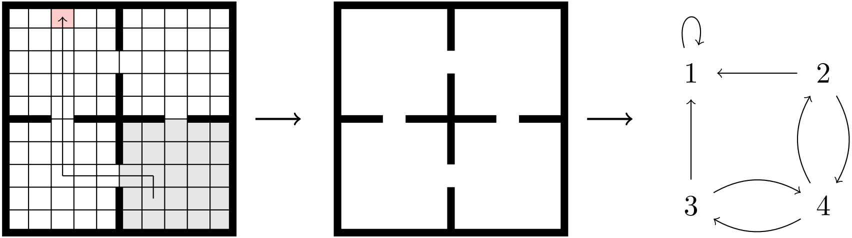
to find V^*, solution of

$$V = \mathcal{T}^* V$$

Table of Contents

Hierarchical RL — State Abstraction

MDP approximation from state space partition:



Less states, same action space:

$$\mathcal{S} = \bigsqcup_k S_k \to \mathcal{K} = \{s_1, \ldots, s_k\}$$

Averaged transition and reward:

$$T \to \omega \cdot T \cdot \phi$$

$$R \to \omega \cdot R$$

State Abstraction Definition

Definition (Abstract MDP [Li et al., 2006])

Given $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, R)$ s.t. $\mathcal{S} = \bigsqcup_k S_k$, the abstract MDP

$$(\mathcal{K}, \mathcal{A}, \tilde{T}, \tilde{R})$$

is defined using

- Averaged transition $\tilde{T} = \omega \cdot T \cdot \phi$
- Averaged reward $\tilde{R} = \omega \cdot R$
- Abstract state space $\mathcal{K} = \{s_k\}$

where

- $\omega \in [0,1]^{K \times \mathcal{S}}$ weights with sum 1 on each region
- $\phi = (\mathbb{1}_{s \in S_k})_{s,k}$

Approximate Bellman Operator

Theorem (Tsitsiklis and Van Roy, 1996)

An L2 approximation of the optimal Bellman operator

$$\mathcal{T}^* = V \to \max_{a \in \mathcal{A}} (R_a + \gamma \cdot T_a \cdot V)$$

is its averaged version

$$\Pi \mathcal{T}^*(V) = \phi \cdot \omega \cdot \mathcal{T}^*(V) = \phi \cdot \arg\min_{V_A \in \mathbb{R}^K} \|\phi \cdot V_A - \mathcal{T}^* V\|_2$$

Moreover, $\tilde{V}^*$ *solution of* $V = \Pi \mathcal{T}^* V$ *checks*

$$\|\tilde{V}^* - V^*\|_\infty \leq \max_{1 \leq k \leq K} \frac{\max_{S_k} V - \min_{S_k} V}{1 - \gamma}.$$

Notes on State Abstraction

We note that:

- Abstractions that groups similar states have bounded approximations errors[2]:

$$\forall s, s' \in S_k,\ |Q^*(s,a) - Q^*(s',a)| \leq \varepsilon \implies \|V^* - V^{\tilde{\pi}}\|_\infty \leq K \cdot \varepsilon$$

- Efficient partition criterion often depends on V^*, Q^*, π^*...

We propose to:

- Estimate the quality of any valuation of state abstraction
- Refine abstraction along VI steps

[2] [Abel et al., 2016]

Table of Contents

Approximate Bellman Operator and Abstraction

Approximate Bellman $\Pi \mathcal{T}_Q^* = \phi \cdot \omega \cdot \mathcal{T}_Q^*$ is the exact Bellman of an abstract MDP:[3]

Lemma (O.F.)

For any state abstraction $(\mathcal{K}, \omega, \phi)$ *and its value function* $Q_A = \omega \cdot \tilde{Q} \in \mathbb{R}^K$,

$$\phi \cdot \mathcal{T}_{Q,A}^* Q_A = \Pi \mathcal{T}_Q^* (\phi \cdot Q_A)$$

[3] [Forghieri et al., 2024]

Quality of a piecewise constant value function (1/2)

Theorem (Quality of a piecewise constant value function, O.F.)

Given $\mathcal{M}$, its abstraction $(\mathcal{K}, \omega, \phi)$, and the piecewise constant value function $\tilde{V}$,

$$\|\tilde{V} - V^*\|_\infty \leq \frac{1}{1-\gamma}\left(\max_{1 \leq k \leq K} \operatorname{Span}_{S_k} \mathcal{T}^*\tilde{V} + \|\tilde{V} - \Pi\mathcal{T}^*\tilde{V}\|_\infty\right)$$

where $\operatorname{Span}_{S_k} V := \max_{s \in S_k} V(s) - \min_{s \in S_k} V(s)$.

→ Dependence on the $\tilde{V}$ ($\equiv$ value function on the abstract MDP) and on the aggregation structure !

→ True for $\mathcal{T}_Q^*$, $\mathcal{T}^\pi$

Quality of a piecewise constant value function (2/2)

$$\|\tilde{V} - V^*\|_\infty \leq \frac{1}{1-\gamma}\left(\max_{1\leq k\leq K} \operatorname{Span}_{S_k} \mathcal{T}^*\tilde{V} + \|\tilde{V} - \Pi\mathcal{T}^*\tilde{V}\|_\infty\right)$$

Two terms:

- $\max_{1\leq k\leq K} \operatorname{Span}_{S_k} \mathcal{T}^*\tilde{V}$: do we lose information aggregating ?
- $\|\tilde{V} - \Pi\mathcal{T}^*\tilde{V}\|_\infty$: is $\tilde{V}$ close to optimal value of abstract MDP ?

Quality of a piecewise constant value function

$$\|\tilde{V} - V^*\|_\infty \leq \frac{1}{1-\gamma} \left(\max_{1 \leq k \leq K} \operatorname{Span}_{S_k} \mathcal{T}^* \tilde{V} + \|\tilde{V} - \Pi \mathcal{T}^* \tilde{V}\|_\infty \right)$$

Fortunately,

- $\max_{1 \leq k \leq K} \operatorname{Span}_{S_k} \mathcal{T}^* \tilde{V}$ can decrease refining aggregation $(S_k)_k$
- $\|\tilde{V} - \Pi \mathcal{T}^* \tilde{V}\|_\infty$ can decrease iterating $\Pi \mathcal{T}^*$ over $\tilde{V}$
- $\Pi \mathcal{T}^*$ is simple to compute

Approximate VI is cheaper to compute

Operator	Complexity	Approximation	Complexity
$\mathcal{T}^*$	$\mathcal{S}^2\mathcal{A}$	$\Pi\mathcal{T}^*$	$\mathcal{S}K\mathcal{A}$
$\mathcal{T}^\pi$	$\mathcal{S}^2$	$\Pi\mathcal{T}^\pi$	K^2
$\mathcal{T}_Q^*$	$\mathcal{S}^2\mathcal{A}$	$\Pi\mathcal{T}_Q^*$	$K^2\mathcal{A}$

Table 1: Number of operations to update a value function. $K \ll S$ generally.

$\rightarrow$ Simpler to compute, contract space with factor γ, but converge to $\tilde{V} \neq V^*$... Need to refine aggregation lowering the span !

Progressive State Space Disaggregation Process

We propose starting with a trivial abstraction:

$$K = 1,\ S_1 = \mathcal{S},\ \tilde{V}_0 = (0)_{s \in \mathcal{S}}$$

Then iterate as follows:

1. Apply $\Pi\mathcal{T}^*$ until $\|\tilde{V} - \Pi\mathcal{T}^*\tilde{V}\|_\infty \leq \epsilon$
2. Refine each region by splitting until

$$\max_{S_k} \mathcal{T}^* V_t - \min_{S_k} \mathcal{T}^* V_t \leq \varepsilon$$

   holds for each region k.

→ This process converges to V^* with arbitrary accuracy.

Disaggregation process

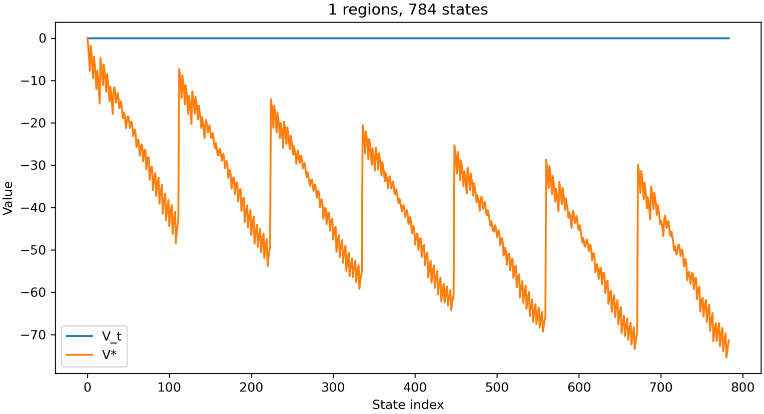


Figure 2: Disaggregation process applied to Tandem Queues model [Tournaire et al., 2022]

First disaggregation step

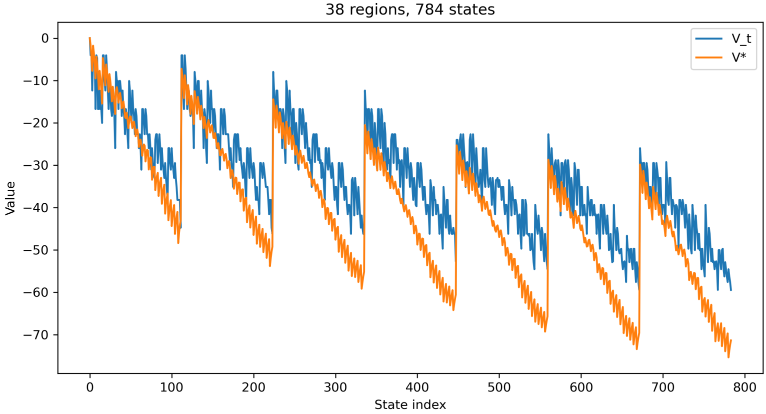


Figure 3: Disaggregation process applied to Tandem Queues model [Tournaire et al., 2022]

Second disaggregation step

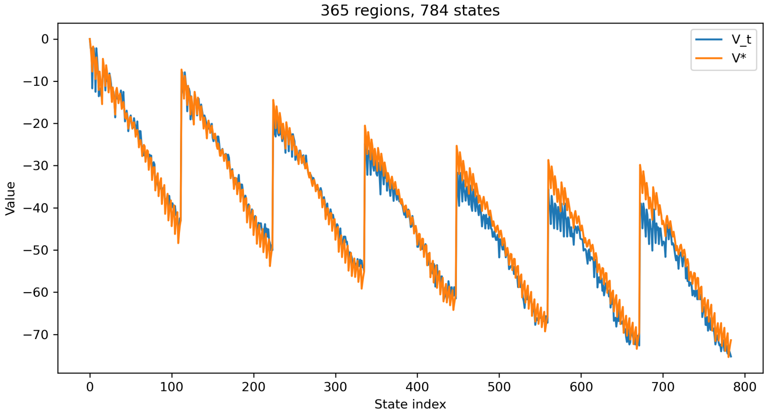


Figure 4: Disaggregation process applied to Tandem Queues model [Tournaire et al., 2022]

→ Problem when $|\mathcal{K}| \to |\mathcal{S}|$, no more gain...

Progressive Disaggregation Convergence

Theorem (Final Abstraction Quality, O.F.)

Considering the final value and abstraction $(V_A, (S_k)_k)$,

1. *the process finishes in a finite number of steps.*
2. *the distance to optimal value function checks:*

$$\|\phi \cdot V_A - V^*\|_\infty \leq \frac{2\epsilon}{1-\gamma}$$

Moreover, for any region k,

$$\forall s, s' \in S_k,\ |V^*(s) - V^*(s')| \leq \frac{4\epsilon}{1-\gamma}.$$

→ Abstraction quality is ensured!

Toys models

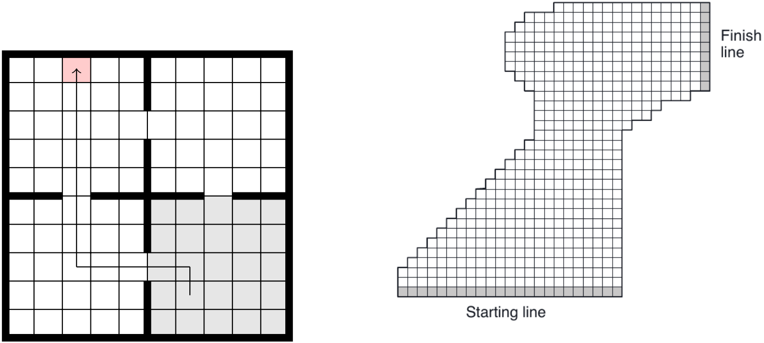


Figure 5: Four Rooms and Sutton's racectrack models

[3] [Hengst, 2012, Sutton and Barto, 2018]

Real-world models

- Servers in tandem and stochastic arrivals [Tournaire et al., 2022]
- Inventory Control [Winston, 2004]
- Hydro-valley electricity production management [Carpentier et al., 2018]

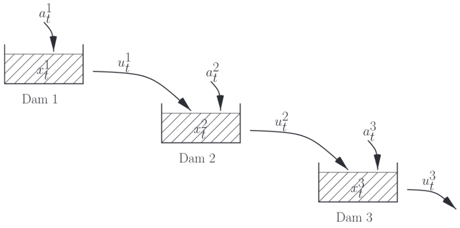


Figure 6: Hydro-valley management [Carpentier et al., 2018]

Solving methods

Traditional Dynamic Programming

- Value Iteration
- Modified Policy Iteration

Our (Progressive Disaggregation) adapted to

- Value Iteration, Q-Value Iteration
- Modified Policy Iteration

Alternative Aggregation approach

- Modified Policy Iteration with adaptive aggregation boosting [Bertsekas et al., 1988]
- Aggregation-disaggregation for Temporal-Difference learning [Chen et al., 2022]

Runtime comparison

$\|\mathcal{S}\|$	Rooms 193.6k	Barto 1M	Tandem 19.6k	Hydro-valley 118k	Inventory 250k
PDVI	11712.5	73501.4	979.4	>24h	2877.4
PDQVI	**160.3**	**46.8**	657.4	>24h	**1458.5**
VI	2520.2	46524.7	553.2	>24h	4032.7
Chen	12844.1	33238.3	>24h	>24h	>24h
PDPIM	>24h	18044.8	**254.2**	**2051.0**	>24h
PIM	>24h	20164.2	348.8	2273.9	>24h
Bertsekas	>24h	33238.3	>24h	>24h	>24h

Table 2: Runtimes for solving different models. Discount $\gamma = 0.9999$, precision $\varepsilon = 10^{-3}$.

$\rightarrow$ Gain of time for real-world models for high discount (γ close to 1)
$\rightarrow$ Higher discounts (γ closer to 1) lead to larger gains

Remarks

Experimental Results:

- For toy models, a suitable decomposition generally exists $\rightarrow$ runtime reduced by a factor of 15
- For real-world models: runtime reduced by a factor of 1.3

Table of Contents

Conclusion

In those slides:

- Hierarchical RL context
- Link between **Approximate DP and State Abstraction**
- Practical **State Abstraction discovery** criterion
- MDP **solving benchmark**

Upcoming work :

- Total reward convergence proof
- Larger benchmark
- How to transpose it to model free ?

Thank you for listening!

Abel, D., Hershkowitz, D., and Littman, M. (2016).
Near optimal behavior via approximate state abstraction.
In *International Conference on Machine Learning*, pages 2915–2923. PMLR.

Abel, D., Umbanhowar, N., Khetarpal, K., Arumugam, D., Precup, D., and Littman, M. (2020).
Value preserving state-action abstractions.
In *International Conference on Artificial Intelligence and Statistics*, pages 1639–1650. PMLR.

Bertsekas, D. P., Castanon, D. A., et al. (1988).
Adaptive aggregation methods for infinite horizon dynamic programming.
IEEE Transactions on Automatic Control.

Carpentier, P., Chancelier, J.-P., Leclère, V., and Pacaud, F. (2018).
Stochastic decomposition applied to large-scale hydro valleys management.

European Journal of Operational Research, 270(3):1086–1098.

Chen, G., Gaebler, J. D., Peng, M., Sun, C., and Ye, Y. (2022).
An adaptive state aggregation algorithm for markov decision processes.
In *AAAI 2022 Workshop on Reinforcement Learning in Games.*

Coulom, R. (2006).
Efficient selectivity and backup operators in monte-carlo tree search.
In *International conference on computers and games*, pages 72–83. Springer.

Ferrer-Mestres, J., Dietterich, T. G., Buffet, O., and Chades, I. (2020).
Solving k-mdps.
In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 110–118.

Forghieri, O., Le Pennec, E., Castel, H., and Hyon, E. (2024).

Progressive state space disaggregation for infinite horizon dynamic programming.
In *34th International Conference on Automated Planning and Scheduling*.

Hengst, B. (2012).
Hierarchical approaches.
In *Reinforcement learning*, pages 293–323. Springer.

Jothimurugan, K., Bastani, O., and Alur, R. (2021).
Abstract value iteration for hierarchical reinforcement learning.
In *International Conference on Artificial Intelligence and Statistics*, pages 1162–1170. PMLR.

Li, L., Walsh, T. J., and Littman, M. L. (2006).
Towards a unified theory of state abstraction for mdps.
In *AI&M*.

Siddiqi, S., Boots, B., and Gordon, G. (2010).
Reduced-rank hidden markov models.

In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 741–748. JMLR Workshop and Conference Proceedings.

Sutton, R. S. and Barto, A. G. (2018).
Reinforcement learning: An introduction.
MIT press.

Tournaire, T., Jin, Y., Aghasaryan, A., Castel-Taleb, H., and Hyon, E. (2022).
Factored reinforcement learning for auto-scaling in tandem queues.
In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7. IEEE.

Winston, W. L. (2004).
Operations research: applications and algorithm.
Thomson Learning, Inc.