Theory and Methods for Reinforcement Learning

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 11: Deep Model-based RL

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-618 (Spring 2020)

lions@epfl

License Information for Reinforcement Learning Slides

- Last class:
  - Model-Based RL

- This class:
  - Model-based RL
    1. Recap: Model Free vs. Model Based
    2. State Abstraction
    3. DeepMDP
    4. Model-based deep reinforcement learning with theoretical guarantees

- Next class:
  - Inverse reinforcement learning

Recommended reading

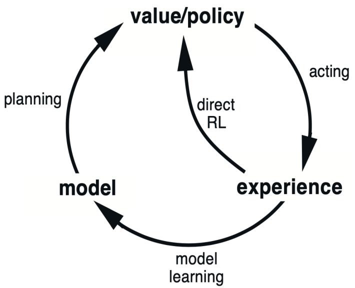
- Chapter 8,9 in S. Sutton, and G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.
- Gelada, Carles, et al. "Deepmdp: Learning continuous latent space models for representation learning." arXiv preprint arXiv:1906.02736 (2019).
- Luo, Yuping, et al. "Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees." arXiv preprint arXiv:1807.03858 (2018).

Motivation

Motivation

Can We use neural networks to learn our model in Dyna-style RL?

Recap:Model Free vs. Model based



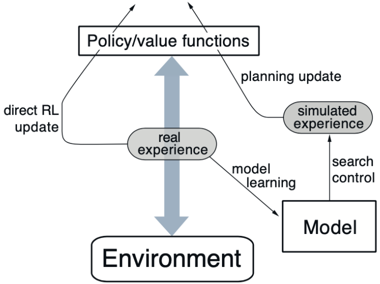


Figure: Dyna Architecture

How does learning a model help learning?

1. When does using a model help? [6]
2. if state space is easily compressible, then doing the policy on an abstract model space helps (remember DQN)
3. if dynamics are "easy" to learn, then we can "learn" a simulator and then learn our policy on that with much less real samples
4. even if not, if horizon is small, *small* model errors might not hurt too much

So what makes a "good" model?

1. State abstraction => what can we ignore without losing information moment to moment?
2. Bisimulation metrics => what abstractions lead to the same behaviour in the long run? [2]

State Abstraction

Figure: Atari

State Abstraction

- A state abstraction is a mapping ϕ that maps the original (or primitive/raw) state space $\mathcal{S}$ to some finite abstract state space; for brevity we use $\phi(\mathcal{S})$ to denote the codomain of the mapping. Intuitively, if $s^{(1)}$ and $s^{(2)}$ are mapped to the same element, that is $\phi\left(s^{(1)}\right) = \phi\left(s^{(2)}\right)$, they are treated as the same state.

  1. Policy irrelevant: ϕ is an ϵ_{π^*}-approximate π^*-irrelevant abstraction, if there exists an abstract policy $\pi : \phi(\mathcal{S}) \to \mathcal{A}$ such that $\left\| V_M^* - V_M^{\pi_M} \right\|_\infty \leq \epsilon_{\pi^*}$
  2. Q irrelevant: ϕ is an ϵ_{Q^*}-approximate Q^*-irrelevant abstraction if there exists an abstract Q-value function $f : \phi(\mathcal{S}) \times \mathcal{A} \to \mathbb{R}$, such that $\left\| [f]_M - Q_M^\star \right\|_\infty \leq \epsilon_{Q^*}$.
  3. Model irrelevant: ϕ is an (ϵ_R, ϵ_P)-approximate model-irrelevant abstraction if for any $s^{(1)}$ and $s^{(2)}$ where $\phi\left(s^{(1)}\right) = \phi\left(s^{(2)}\right), \forall a \in \mathcal{A}$

$$\left| R\left(s^{(1)}, a\right) - R\left(s^{(2)}, a\right) \right| \leq \epsilon_R, \quad \left\| \Phi P\left(s^{(1)}, a\right) - \Phi P\left(s^{(2)}, a\right) \right\|_1 \leq \epsilon_P$$

- When $\epsilon_\pi, \epsilon_{Q^*}, \epsilon_R, \epsilon_P = 0$, it is exact abstraction without losing anthying.

State Abstraction

- For the given (ϵ_R, ϵ_P) of abstraction ϕ, we can bound the loss.

$$\left\| V_M^* - V_M^{\pi^*_{M_\phi}} \right\|_\infty \leq \frac{2\epsilon_R}{1-\gamma} + \frac{\gamma \epsilon_P R_{\max}}{(1-\gamma)^2}$$

- Abstractions for Model-Based RL: Our goal is to choose an abstraction h from a candidate set $\mathcal{H}$ so as to minimize the loss of the optimal policy for M_D^h. And [4] shows this loss can be bounded.

$$\mathrm{Loss}(h, D) = \left\| V_M^* - V_M^{\pi^*_{M_D^h}} \right\|_\infty \leq \frac{2}{(1-\gamma)^2} (\mathrm{Appr}(h) + \mathrm{Estm}(h, D, \delta))$$

State Abstraction

- The bounding the loss of state abstraction should follow the Rate–distortion theory.

$$R(D) = \min_{p(\mathbf{z}|\mathbf{x}) = \|d(x,\vec{x})| \leq D} I(X; \tilde{X})$$

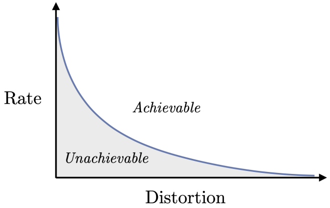
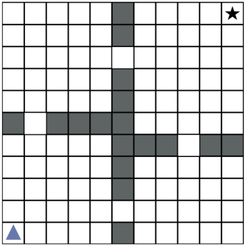


Figure: RD lower Bound

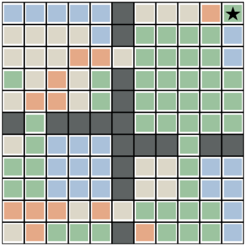
- The information bottleneck method extends RD theor to prediction. The IB defines relevant information according to how well a random variable Y can be predicted from each $\tilde{x} \in \tilde{\mathcal{X}}$, which implies the optimal trade off between compression and performance.[1]

$$\mathcal{L}[p(\tilde{x}|x)] = I(\tilde{X}; X) - \beta I(\tilde{X}; Y)$$

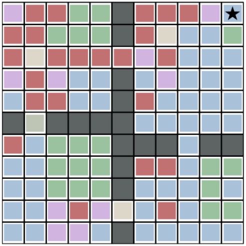
State Abstraction[1]



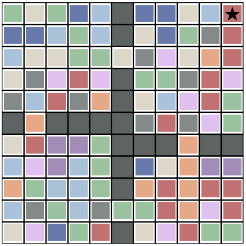
(a) The Four Rooms Domain



(b) ϕ with $\beta = 1$, $|\mathcal{S}_\phi| = 4$

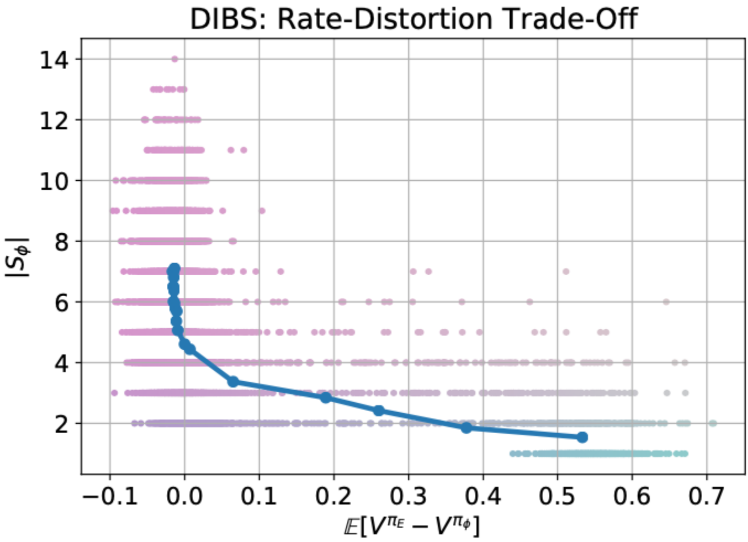


(c) ϕ with $\beta = 2$, $|\mathcal{S}_\phi| = 5$



(d) ϕ with $\beta = 20$, $|\mathcal{S}_\phi| = 9$

State Abstraction



So how can we actually learn these using neural networks?[3]

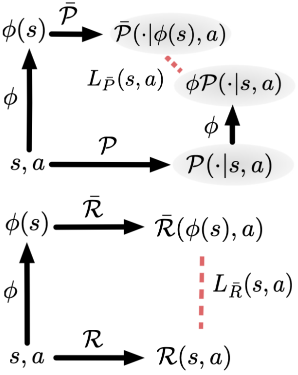


Figure: Diagram of the latent space losses

DIBS is one example, DeepMDP another. $L_{\bar{R}}, L_{\bar{P}}$ attempt to induce representations which allow learning of approximately-bisimilar transition and reward dynamics in the latent space w.r.t. the true MDP,i.e. "learning what matters".

Model-based Deep Reinforcement Learning

Q:Now that we can learn good models, are we guaranteed a good policy?
A: Difficult question when using deep RL with deep representations!

Model-based Deep Reinforcement Learning[5]

1. Iterative lower bound:
$$V^{\pi,M^*} \geq V^{\pi,\widehat{M}} - D(\widehat{M},\pi)$$
2. Neighborhood of a reference policy π_{ref}
$$V^{\pi,M^\star} \geq V^{\pi,\widehat{M}} - D_{\pi_{\mathrm{ref}},\delta}(\widehat{M},\pi), \quad \forall \pi \text{ s.t. } d\left(\pi,\pi_{\mathrm{ref}}\right) \leq \delta \tag{R1}$$
3. Vanished discrepancy bound
$$\widehat{M} = M^\star \Longrightarrow D_{\pi_{\mathrm{ref}}}(\widehat{M},\pi) = 0, \quad \forall \pi, \pi_{\mathrm{ref}} \tag{R2}$$
$$D_{\pi_{\mathrm{ref}}}(\widehat{M},\pi) \text{ is of the form } \underset{\tau\sim\pi_{\mathrm{ref}},M^\star}{\mathbb{E}}[f(\widehat{M},\pi,\tau)] \tag{R3}$$

where f is a known differentiable function.

Model-based Deep Reinforcement Learning

Algorithm 1 Meta-Algorithm for Model-based RL

Inputs: Initial policy π_0. Discrepancy bound D and distance function d that satisfy equation (R1) and (R2).
For $k = 0$ to T:

$$\pi_{k+1}, M_{k+1} = \underset{\pi \in \Pi,\ M \in \mathcal{M}}{\text{argmax}} \quad V^{\pi,M} - D_{\pi_k,\delta}(M, \pi) \tag{3.3}$$

$$\text{s.t.} \;\; d(\pi, \pi_k) \leq \delta \tag{3.4}$$

Figure: Model-based iterative algorithm

Theorem 3.1. *Suppose that $M^\star \in \mathcal{M}$, that D and d satisfy equation (R1) and (R2), and the optimization problem in equation (3.3) is solvable at each iteration. Then, Algorithm 1 produces a sequence of policies $\pi_0, \ldots, \pi_T$ with monotonically increasing values:*

$$V^{\pi_0,M^\star} \leq V^{\pi_1,M^\star} \leq \cdots \leq V^{\pi_T,M^\star} \tag{3.5}$$

Moreover, as $k \to \infty$, the value $V^{\pi_k,M^\star}$ converges to some $V^{\bar{\pi},M^\star}$, where $\bar{\pi}$ is a local maximum of $V^{\pi,M^\star}$ in domain Π.

Figure: Monotonical Iteration

References

[1] David Abel, Dilip Arumugam, Kavosh Asadi, Yuu Jinnai, Michael L Littman, and Lawson LS Wong.
State abstraction as compression in apprenticeship learning.
In *Proceedings of the AAAI Conference on Artificial Intelligence. AAAI Press*, 2019.

[2] Norm Ferns, Prakash Panangaden, and Doina Precup.
Metrics for finite markov decision processes.
In *UAI*, volume 4, pages 162–169, 2004.

[3] Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare.
Deepmdp: Learning continuous latent space models for representation learning.
arXiv preprint arXiv:1906.02736, 2019.

[4] Nan Jiang, Alex Kulesza, and Satinder Singh.
Abstraction selection in model-based reinforcement learning.
In *International Conference on Machine Learning*, pages 179–188, 2015.

[5] Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma.
Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees.
arXiv preprint arXiv:1807.03858, 2018.

[6] Hado P van Hasselt, Matteo Hessel, and John Aslanides.
When to use parametric models in reinforcement learning?
In *Advances in Neural Information Processing Systems*, pages 14322–14333, 2019.