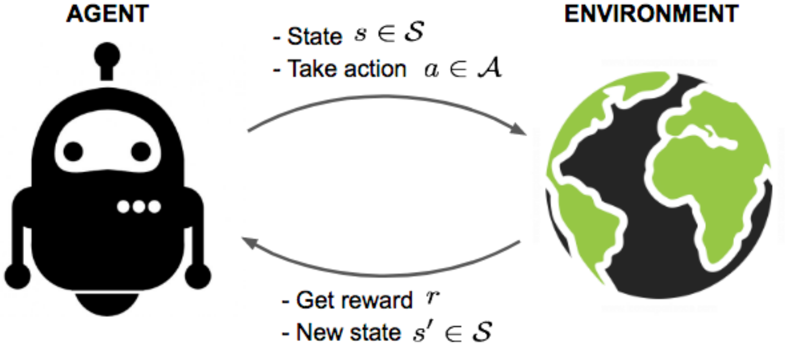
Practical Distributionally Robust Markov Decision Processes with Kullback-Leibler Divergences

William Greenall
Petros Dellaportas

March 4, 2021

Introduction: Reinforcement Learning Concepts



Recent advancements

Figure: AlphaGo: Beat the world Go champion Lee Sedol

Figure: Able to beat top human players at StarCraft II

Both systems based heavily on the use of Reinforcement Learning

Reinforcement Learning Paradigms:

- \- model-based : build a model of the environment and use it to acquire a good policy
- \- model-free : learn good policies based entirely on observed actions, transitions and rewards

Markov Decision Process

Definition (Markov Decision Process)

A Markov Decision process is a tuple: $\langle \mathcal{S}, \mathcal{A}, \mathrm{p}, r(s, a, s'), \gamma \rangle$ where:

$\mathcal{S}$, the state space;

$\mathcal{A}$, the set of actions;

p, a transition tensor of size $|\mathcal{S}| \times |\mathcal{S}| \times |\mathcal{A}|$

$r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathrm{R}$, the reward function;

$\gamma \in (0, 1)$, a discount factor.

A policy π is a function $\pi : \mathcal{S} \to [0, 1]^{\mathcal{A}}$ That represents a rule describing the probability of taking an action - so $\pi(s)$ is the distribution over actions to be taken.

Markov Decision Process: Examples

Figure: Monopoly is an MDP!

Figure: Practical Example: Airline Pricing

Bellman Equation

'Value' of a state:

$$v^{\pi}(s_0) = \mathbb{E}_p \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) \right] \quad (1)$$

where a_t is chosen according to the policy π.
We use a recursion known as the Bellman equation:

$$v(s) = \max_{\{a\}} \mathbb{E}_p \left[r(s, a, s') + \gamma v(s') \right] \quad (2)$$

This recursion can be iterated to get the optimal value function and policy!

Problem

The method described requires knowledge of p.
We can use some data to estimate it: Assume we have n episodes of T transitions each.

$$\mathcal{D} = \{\{s_t, a_t, r_t, s_{t+1}\}_{t=0}\}_{i=1}^{n}$$

Problem: poor estimates of the transition tensor can lead to bad performance! (Mannor et al., 2007)

Dealing with poor estimates

Introduce two extensions:

Robust Markov Decision Process: $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r(s,a,s'), \gamma \rangle$

where:

$\mathcal{S}$, the state space;

$\mathcal{A}$, the set of actions;

$\mathcal{P}$, a set of potential transition models;

$r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathrm{R}$, the reward function.

$\gamma \in (0,1)$, a discount factor.

Distributionally Robust Markov Decision Process: $\langle \mathcal{S}, \mathcal{A}, \mathcal{F}, r(s,a,s'), \gamma \rangle$

where:

$\mathcal{S}$, the state space;

$\mathcal{A}$, the set of actions;

$\mathcal{F}$, a set of distributions over transition models;

$r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathrm{R}$, the reward function.

$\gamma \in (0,1)$, a discount factor.

Robust MDP

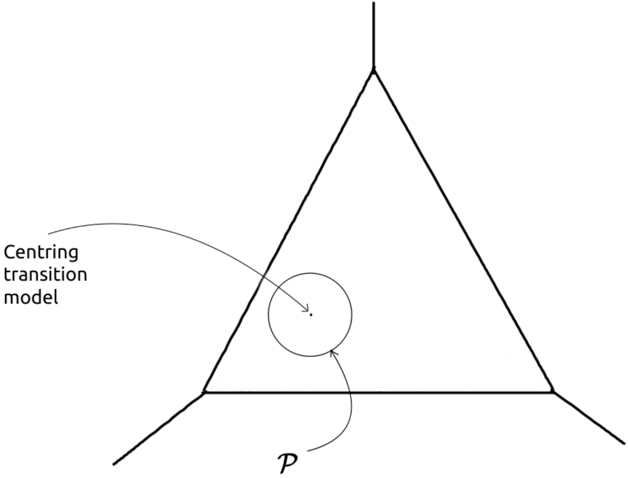


Figure: Representation of Robust Ambiguity Set

Distributionally Robust MDP

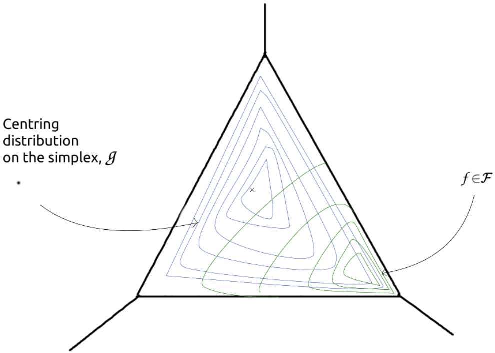


Figure: Representation of Distributionally Robust Ambiguity Set

Classic example of distribution on the simplex: Dirichlet

Bellman Equations

The Bellman equations become:

Robust MDP

$$v(s) = \min_{p \in \mathcal{P}} \max_a \mathbb{E}_p \left[r(s, a, s') + \gamma v(s') \right] \quad (3)$$

Distributionally Robust MDP

$$v(s) = \min_{f \in \mathcal{F}} \max_a \mathbb{E}_{p \sim f} \left[r(s, a, s') + \gamma v(s') \right] \quad (4)$$

$$(5)$$

Distributionally Robust MDP

Two ways to describe $\mathcal{F}$:

- moment-matching: choose distributions whose moments have some useful property
- statistical distance: $\mathcal{F}$ is a set of distributions a given statistical distance from a centring distribution - usually the *empirical* distribution

The latter commonly based on the use of the Wasserstein distance in the literature - this is not usually available analytically

Contribution

Our Bayesian setup allows for use of KL-Divergence to describe $\mathcal{F}$
We define the ambiguity set we use:

$$\mathcal{F} = \bigotimes_{s,a} \{f : D_{KL}[f || \hat{g}_{s,a}] \leq \beta\}$$

where:

$$\hat{g} = f(q|\mathcal{D}) \propto \mathcal{L}(\mathcal{D}|q)\, g(q) \tag{6}$$

with

$$q : \text{a potential transition tensor} \tag{7}$$

$$g : \text{a prior distribution over transition models} \tag{8}$$

$$\hat{g} : \text{the Bayesian posterior} \tag{9}$$

Can we be sure that there is an optimal policy?

There are theorems for Robust MDPs that ensure that there exists a robust policy, i.e. one that maximises the robust value function.
Two steps to show there is an optimal policy for our setup:

- Show that the value function only depends on the expected value of the distribution over transition models
- Show that there is a Robust MDP with the ambiguity set consisting of the expected values of the distributions in $\mathcal{F}$.

Can we be sure that there is an optimal policy?

First step: comes from the linearity of the expectation operator in the Bellman equation

Second step: we show that, for a given ambiguity set of a Distributionally Robust MDP:

$$\mathcal{F} = \{f : D_{KL}[f||\hat{g}] \leq \beta\} \tag{10}$$

there is an ambiguity set of a Robust MDP:

$$\mathcal{P} = \{p : D_{KL}[p||q] \leq \beta'\}$$

where:
- q: expectation of posterior $\hat{g}$,
- p: expectation of functions $f \in \mathcal{F}$,
- $\beta' < \beta$.

Can we be sure that there is an optimal policy?

Let $f(v), g(v)$ represent the densities of distributions on the simplex ($v \in \Delta^S$), with $p = \mathbb{E}_f[v]$, and $q = \mathbb{E}_g[v]$. Then:

$$D_{KL}[p||q] \leq D_{KL}[f||g] \tag{11}$$

We show this using calculus of variations and the Bhatia-Davis inequality.

Combining these two steps we see that there is a corresponding Robust MDP with the same optimal policy, built from an ambiguity set $\mathcal{P}$ made up of the expectations of the elements of the ambiguity set $\mathcal{F}$.

Practical Implementation

We can implement the setup by having $\mathcal{F}$ made up of Dirichlet distributions. Then we have:

$$\mathcal{F} = \bigotimes_{s,a} \left\{ f \;\middle|\; \frac{\ln B(\alpha)}{\ln B(\tilde{\alpha})} + \psi(\alpha_0)(\alpha_0 - \tilde{\alpha}_0) + \sum_{i=0} \psi(\alpha_i)\left(\tilde{\alpha}_i - \alpha_i\right) \leq \beta \right\}$$

With:

α_0: $\sum_k \alpha_k$, sum of parameters α of the given distribution $f_{s,a}$

$\tilde{\alpha}_0$: $\sum_k \tilde{\alpha}_k$, sum of parameters $\tilde{\alpha}$ of the posterior $\hat{g}_{s,a}$

ψ: the Digamma function

B: the Beta function

Practical Implementation

We can also extend to Dirichlet mixtures to make $\mathcal{F}$ richer:

$$\mathcal{F} = \bigotimes_{s,a} \left\{ f \mid D_{KL}\left[f || \hat{g}\right] \leq \beta, f = \sum_{i=1} w_i h_i \right\}$$

With:

h_i:	a Dirichlet distribution
w_i:	Mixing probability for mixture component i

Practical Implementation: Extension to mixtures

However: Mixture KL-divergence not usually available - so we can use an upper bound.

$$\mathrm{D}_{KL}\left[f||\hat{g}\right] \leq \sum_i w_i \left[D_{KL}(h_i||\hat{g}) + \ln\left[\sum_j w_j \exp\left\{-D(h_i||h_j)\right\}\right]\right]$$

This estimate is based on the work in (Kolchinsky and Tracey, 2017)

Continuous State Model-Based RL (WIP)

Can we extend to continuous environments?

- value iteration for each state is not viable
- need way to represent continuous state transition model

Continuous State Model-Based RL (WIP)

Continuous state, Model-based RL techniques are usually based on Gaussian Processes as transition models

A Gaussian Process is a stochastic process $\mathcal{GP} = \{X_t\}$ so that any finite set of values of the process are joint-normally distributed.

With appropriate choice of covariance function K, we can use them to model prior belief over functions.

Best Examples: PILCO (Deisenroth and Rasmussen, 2011), PDDP (Pan and Theodorou, 2014)

Continuous State Model-Based RL (WIP)

Stage 1:

Assume a function describing dynamics:

$$\begin{aligned} s_{t+1} &= f(s_t, a_t) \\ \Rightarrow \Delta_t &\equiv f(s_t, a_t) - s_t \end{aligned}$$

and then describe prior belief over Δ:

$$p(s_{t+1} - s_t | s_t, a_t) = \mathbb{N}(0, \Sigma_t) \tag{12}$$

$$\text{or } p(s_{t+1} | s_t, a_t) = \mathbb{N}(\mu_t, \Sigma_t) \tag{13}$$

where:

$\mu_t = s_t + \mathbb{E}_f [\Delta_t]$

$\Sigma_t = Var_f [\Delta_t]$ (the variance implied by the Gaussian process)

Continuous State Model-Based RL (WIP)

This is a Gaussian process prior over Δ_t. We train it using the transitions $\{s_{t+1} - s_t, a_t\}_t$ from $\mathcal{D}$ as before (e.g., data from a sequence of airline's pricing decisions and observables).

Stage 2:

Use the learnt dynamics model to build a local value function estimate around a nominal trajectory - follows the technique in (Pan and Theodorou, 2014)

Continuous State Model-Based RL (WIP)

Stage 3:

Minimise this value function estimate w.r.t the dynamics model within a given KL-divergence of our learnt dynamics model.

Problem: we want to evaluate

$$v(s_0) = \mathbb{E}_{f \sim \mathcal{GP}} \left[\sum_{t=0}^{T} r(s_t, a_t, s_{t+1}) \right] \tag{14}$$

Continuous State Model-Based RL (WIP)

With Gaussian Process dynamics model,

$$p(s_{t+1}|s_t, a_t) = \mathbb{N}(\mu_t, \Sigma_t) \tag{15}$$

But

$$p(s_{t+i}|s_t, a_t) \neq \mathbb{N}(\mu_t, \Sigma_t) \tag{16}$$

where $i = 2, 3, 4, ...$

Continuous State Model-Based RL (WIP)

To see why, note the following diagram (from (Deisenroth and Rasmussen, 2011))

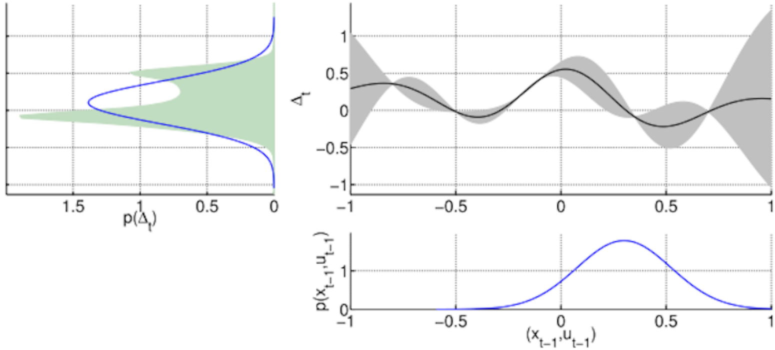


Figure: Gaussian process propagation

Continuous State Model-Based RL (WIP)

How to solve this?

- Assume a moment-matched Gaussian (technique used by (Pan and Theodorou, 2014; Deisenroth and Rasmussen, 2011))
- OR perhaps estimate this density another way?

Current working idea: use Hermite functions to estimate the density to get good value function estimates

Continuous State Model-Based RL (WIP)

Thank you very much for your time!
Happy to discuss any of the ideas herein with you - you may email me at william.greenall.19@ucl.ac.uk

Deisenroth, M. P. and C. E. Rasmussen (2011). PILCO: A model-based and data-efficient approach to policy search. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 465–472.

Kolchinsky, A. and B. D. Tracey (2017, jul). Estimating mixture entropy with pairwise distances. *Entropy 19*(7).

Mannor, S., D. Simester, P. Sun, and J. N. Tsitsiklis (2007, feb). Bias and variance approximation in value function estimates. *Management Science 53*(2), 308–322.

Pan, Y. and E. Theodorou (2014). Probabilistic Differential Dynamic Programming. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, Volume 27, pp. 1907–1915. Curran Associates, Inc.