ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Epistemic Risk-Sensitive Reinforcement Learning

Hannes Eriksson[1,2] and Christos Dimitrakakis[2,3] *

1- Chalmers University of Technology, Gothenburg - Sweden
2- Zenuity AB, Gothenburg - Sweden
3- University of Oslo, Oslo - Norway

Abstract. We develop a framework for risk-sensitive behaviour in reinforcement learning (RL) due to uncertainty about the environment dynamics by leveraging utility-based definitions of risk sensitivity. In this framework, the preference for risk can be tuned by varying the utility function, for which we develop dynamic programming (DP) and policy gradient-based algorithms. The risk-averse behavior is compared with the behavior of risk-neutral policy in environments with epistemic risk.

1 Introduction

We consider the problem of reinforcement learning (RL) with policies risk-sensitive policies due to *epistemic* uncertainty, i.e. due to the agent not knowing how the environment works. Previous work in risk-sensitive RL, focused on aleatory uncertainty, i.e. due to environmental stochasticity. Epistemic risk-sensitivity makes more sense in RL, where most uncertainty is due to the lack of information about the environment, especially for applications such as autonomous driving, where systems are nearly deterministic. Within a Bayesian utilitarian framework, we develop novel algorithms for policy optimisation, and compare their performance quantitatively with *risk-neutral* and aleatory-risk-sensitive policies.

RL is a sequential decision-making problem under uncertainty. The classical goal is to maximise expected return $R \triangleq \sum_{t=1}^{T} r_t$ to some horizon T, with the agent acting in a Markov decision process (MDP). For any given discrete MDP $\mu \in \mathcal{M}$, the optimal risk-neutral policy $\pi^*(\mu) \in \arg\max_\pi \mathbb{E}_\mu^\pi[R]$ can be found via dynamic programming. Because in RL the μ is unknown, the optimal policy must take into account expected information gain. In the Bayesian setting, we maintain a subjective belief in the form of a probability distribution ξ over MDPs $\mathcal{M}$ and the optimal risk-neutral policy solution is given by: $\max_\pi \mathbb{E}_\xi^\pi[R] = \max_\pi \int_{\mathcal{M}} \mathbb{E}_\mu^\pi[R]\, d\xi(\mu)$. This problem is generally intractable, as the optimisation is performed over *adaptive* policies, which is exponentially large in the problem horizon. In our paper, instead of maximising expected return, we will instead maximise a non-linear function of the expected return to induce risk-sensitive behaviour with respect to uncertainty about μ.

*This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation and the computations were performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC).

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

1.1 Related work

Defining risk sensitivity with respect to the return R can be done in several ways. We focus on the expected utility formulation, where the utility is defined as a function of the return: concave functions lead to risk aversion and convex to risk-seeking. Within RL, this approach was first proposed by [1], who derived efficient temporal-difference algorithms for exponential utility functions. However, the authors only considered aleatory risk, i.e. with respect to MDP stochasticity. Much work has focused on conditional value-at-risk (CVaR) [2]. Compared to more traditional risk measures such as Mean-Variance trade-off [3] or expected utility framework [4], CVaR allows us to control for tail risk. CVaR has been used for risk-sensitive MDPs in [5].

Epistemic risk has been considered in Robust MDPs. For example [6], obtains optimal pessimistic and optimistic policies within a set of possible MDPs. A Bayesian setting is also considered in [7], which decomposes the risk in aleatory and epistemic components. However, the authors are essentially considering risk due to variance in individual rewards. Our paper instead considers the risk with respect to the total return, which we believe is a more interesting setting for long-term planning problems under uncertainty.

1.2 Contribution

A lot of risk-aware work in RL used different mechanisms for aleatory and epistemic uncertainty. We consider both under a coherent utility maximising framework, where the convexity of the utility with respect to the return R determines risk-seeking or aversion. Applying this utility on the actual or expected return, we can be risk-sensitive with respect to either aleatory or epistemic uncertainty respectively. We also introduce two novel algorithms to handle risk-sensitiveness in MDPs. The first is based on dynamic programming and we apply it to tabular domains. The latter is based on Bayesian Policy gradient and we apply it in continuous state spaces. We evaluate our algorithms on familiar domains with epistemic uncertainty, as well as multi-tasks extensions of them. These are created by introducing a distribution over the parameters of the single-task environment, with the agent being in a different sampled environment at the beginning of each episode. In the multi-task setting, the agent never knows what task they are solving at the beginning of each episode, and so inherent epistemic uncertainty remains.

2 Optimal policies for epistemic risk

Under the expected utility hypothesis, risk sensitivity can be modelled [4] through a concave or convex utility function $U : \mathbb{R} \to \mathbb{R}$ of the return R. Then, for a given μ, the optimal U-sensitive policy with respect to *aleatory* risk is the solution to $\max_\pi \mathbb{E}^\pi_\mu[U(R)]$. In the case where we are uncertain about what is the true MDP, we can express it through belief ξ over μ. Then the optimal policy is the

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

solution to

$$\pi^A(U,\xi) \triangleq \arg\max_\pi \int_{\mathcal{M}} \mathbb{E}^\pi_\mu[U(R)]\, d\xi(\mu). \tag{1}$$

However, this is only risk-sensitive with respect to the stochasticity of the underlying MDPs. We believe that *epistemic* risk, i.e. the risk due to our uncertainty about the model is more pertinent for RL. The optimal *epistemic risk sensitive* policy maximises:

$$\pi^E(U,\xi) \triangleq \arg\max_\pi \int_{\mathcal{M}} U\left(\mathbb{E}^\pi_\mu(R)\right)\, d\xi(\mu). \tag{2}$$

When U is the identity function, both solutions are risk-neutral. In this paper, we shall consider functions of the form $U(x) = \beta^{-1}e^{\beta x}$, so that $\beta > 0$ is risk-seeking and $\beta < 0$ risk-averse. When the policy is risk-seeking optimality is lost and the agent focuses more on exploration than a risk-neutral agent. On the other hand, a risk-averse agent prefers exploitation over a risk-neutral agent. In environments with no risk, the optimal risk-neutral solution can be found by all of the three.

We consider two algorithms for this problem. The first, based on an approximate dynamic programming algorithm for Bayesian RL introduced in [8], is introduced in Section 2.1. The second, based on the Bayesian policy gradient (BPG) framework [9], allows us to extend the previous algorithm to larger MDPs and allows for the learning of stochastic policies. It is detailed in Section 2.2.

2.1 Risk sensitive backward induction

Algorithm 1 is an Approximate Dynamic Programming (ADP) algorithm for optimising policies in our setting. The algorithm is given for a belief over a finite set of MDPs, but can be easily extended to arbitrary ξ through Monte-Carlo sampling, as in [8].

The algorithm maintains a separate Q_μ-value function for every MDP μ. At every step, it finds the best overall policy π with respect to the utility function U. Then the V_μ of each MDP reflects the value of π within that MDP.

2.2 Bayesian policy gradient

Policy gradient [10] is commonly used in model-free RL, but is also very useful in model-based settings, and specifically for the Bayesian RL problem, where sampling from the posterior allows us to construct efficient stochastic gradient algorithms. Even though BPG [11] is a risk-neutral algorithm, we can extend it to the risk sensitive setting, by maximising Eq. 2, where we our utility function models risk sensitivity:

$$\nabla_\theta \frac{1}{\beta} \log \int_{\mathcal{M}} \exp\left(\beta \mathbb{E}^{\pi_\theta}_\mu[R]\right) d\xi(\mu) = \frac{\int_{\mathcal{M}} \exp\left(\beta \mathbb{E}^{\pi_\theta}_\mu[R]\right) \nabla_\theta \mathbb{E}^{\pi_\theta}_\mu[R] d\xi(\mu)}{\int_{\mathcal{M}} \exp\left(\beta \mathbb{E}^{\pi_\theta}_\mu[R]\right) d\xi(\mu)} \tag{3}$$

Our choice of policy parametrisation is a softmax policy with non-linear features. The probability of selecting action a in state s, given current parameters

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Algorithm 1 Epistemic Risk Sensitive Backwards Induction (ERSBI)

Input: $\mathcal{M}$ (set of MDPs), ξ (current posterior)
repeat
  for $\mu \in \mathcal{M}$ $s \in \mathcal{S}$, $a \in \mathcal{A}$ **do**
    $Q_\mu(s,a) = \mathcal{R}_\mu(s,a) + \gamma \sum_{s'} \mathcal{T}_\mu^{ss'} V_\mu(s')$
  end for
  for $s \in \mathcal{S}, a \in \mathcal{A}$ **do**
    $\mathcal{Q}_\xi(s,a) = \sum_\mu \xi(\mu) U[(Q_\mu(s,a)]$
  end for
  $\pi(s) = \arg\max_a \mathcal{Q}_\xi(s,a)$.
  for $\mu \in \mathcal{M}$ **do**
    $V_\mu(s) = Q_\mu(s, \pi(s))$.
  end for
until *convergence*
return π

θ, is $\pi_\theta(a|s) = \frac{e^{\phi(s,a,\theta)}}{\sum_{a' \in \mathcal{A}} e^{\phi(s,a',\theta)}}$, where the features $\phi(s,a,\theta)$ are calculated by a feedforward neural network with one hidden layer.

3 Experimental setup

We conduct experiments in two episodic environments, in both single-task and multi-task settings. In the **single-task** setting, we expect the epistemic risk-sensitive algorithms to perform the same as risk-neutral algorithms after convergence, but the aleatory risk policies to be worse. In the **multi-task** setting, the agent begins each episode at a different environment, which may have been seen before, but does not know which environment it is in. Consequently, there is inherent epistemic uncertainty at the beginning of every episode. Thus, we expect our methods to outperform both risk-neutral and aleatory-risk methods with respect to epistemic risk measures in this setting. In both cases, we evaluate our methods in both discrete and continuous environments and compare with the aleatoric risk-sensitive algorithm of [1] (ARSBI), as well as the risk-neutral algorithms: PSRL (c.f. [12]), MMBI [8] and BPG [11].

The **discrete environment** is *Chain* [8]. In this environment an agent has to travel through a chain of states, with the state with the highest reward at the end of the chain. Traversing to the end is hard as each the agent can "fall" back to the start with some probability. Our model in this setting uses NormalGamma priors on rewards and Dirichlet priors on transitions and over tasks.

The **continuous state-space** environments are based on *Stopping problems*, which are common testbeds for experiments dealing with risk-sensitivity [3, 5, 13]. For this problem, we maintain function priors in the form of Gaussian Processes on the models, together with a hyperprior on the tasks in the multi-task case. Details of the experimental results and conclusions are presented in Section 4.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Algorithm 2 Epistemic Risk Sensitive Policy Gradient (ERSPG)

Input: Policy parametrisation θ_t, ξ_t (current posterior).
repeat
  `Simulate to get` θ_{t+1}
  for $i = 1$ **to** N **do**
    $\mu_{(1)}, \mu_{(2)} \sim (\mathcal{M}_t, \mathcal{R}_t)$
    for $j = 1$ **to** M **do**
      $\tau^{(1)}_{\mu_{(1)}}, \tau^{(2)}_{\mu_{(1)}} \sim \pi_\theta, \mu_{(1)}$
      $\tau^{(3)}_{\mu_{(2)}} \sim \pi_\theta, \mu_{(2)}$
    end for
  end for
  $\theta_{t+1} \leftarrow \theta_t - [\sum_{i=0}^{N} \exp\left(\beta {\tau_{\mu_i}}^{(1)}\right) {\tau_{\mu_i}}^{(2)} \nabla_\theta \log \pi_\theta(a|s)] / [\sum_{i=0}^{N} \exp\left(\beta {\tau_{\mu_i}}^{(3)}\right)]$
  Deploy $\pi_{\theta_{t+1}}$ and obtain $\tau \sim \mu, \pi_{\theta_{t+1}}$
  $\xi_{t+1} \leftarrow \xi_t, \tau$
until *convergence*

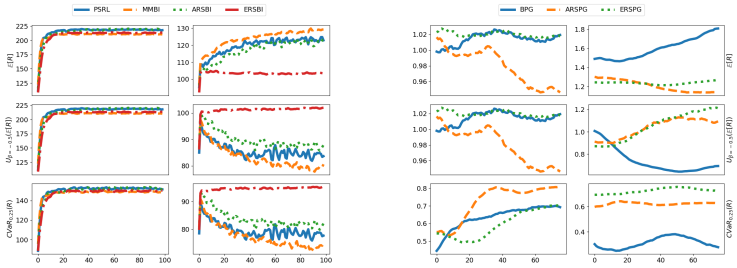
4 Discussion and conclusion

Figure 1 shows the results of running the algorithms across the four different domains. The comparison metrics shown are $\mathbb{E}[R]$, the standard maximisation objective, $U_\beta(\mathbb{E}[R])$, which measures the epistemic risk for a given β, as well $\text{CVaR}_\alpha(R)$, which is the expected performance of the α-quantile. In short, the first measures our actual performance, the second our epistemic risk-sensitive performance and the third our aleatoric risk-sensitive performance. We can see that in the single-task experiment in Figure 1 (a, left), all methods can identify the same optimal behavior. Indeed, in this experiment, there is minimal stochasticity so you would expect their behavior to be similar. In Figure 1 (a, right) we can see the results of the runs in a multi-task setting. In this setting, there is inherent epistemic uncertainty and only our proposed method (ERSBI) is able to find a good risk-averse policy.

In Figure 1 (b, left) we can see the algorithms evaluated in a single-task continuous state-space environment. As expected, in an environment with no inherent epistemic uncertainty (ERSPG) will perform similarly to that of the risk-neutral baseline (BPG). This environment carries significant stochasticity which can explain why the aleatory risk-sensitive agent behaves differently. Finally, in Figure 1 (b, right) we evaluated the agents in a multi-task stopping problem. In this environment there is epistemic uncertainty so we expect the risk-averse agents to be more cautious.

Overall, we believe that our unified framework for epistemic risk in RL is useful as it allows us to obtain risk-averse or risk-seeking behavior through a risk parameter β. We proposed two novel algorithms for controlling epistemic risk in tabular and continuous state-space domains and have shown why they are useful to consider in settings with inherent epistemic uncertainty. In general we see that being sensitive with respect to epistemic risk leads both to good exploration

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.



(a) Chain (b) Stopping problem

Fig. 1: Expected return ($\mathbb{E}[R]$), Risk-sensitive utility (U_β) and CVaR for (a) Chain (left, single-task), (right, multi-task) (b) Stopping (left, single-task), (right multi-task) for PSRL (posterior sampling), Risk-neutral policies (MMBI/BPG), Aleatory-Risk policies (ARSBI), and Epistemic-Risk policies (ERSBI/ERSPG)

behaviour in the single task case, as well as excellent risk aversion behaviour overall. Especially in the multi-task case, we can see that epistemic risk aversion also leads to excellent performance in terms of the CVaR metric.

References

[1] O. Mihatsch and R. Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49(2-3):267–290, 2002.

[2] K. Sivaramakrishnan and R. Stamicar. A cvar scenario-based framework: Minimizing downside risk of multi-asset class portfolios. *The Journal of Portfolio Management*, 44:114–129, 12 2017.

[3] A. Tamar, D. Di Castro, and S. Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the twenty-ninth international conference on machine learning*, pages 387–396, 2012.

[4] M. Friedman and L. J. Savage. The Utility Analysis of Choices Involving Risk. *The Journal of Political Economy*, 56(4):279, 1948.

[5] Y. Chow and M. Ghavamzadeh. Algorithms for cvar optimization in mdps. In *Advances in neural information processing systems*, pages 3509–3517, 2014.

[6] R. Givan, S. Leach, and T. Dean. Bounded-parameter markov decision processes. *Artificial Intelligence*, 122(1-2):71–109, 2000.

[7] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pages 1192–1201, 2018.

[8] C. Dimitrakakis. Robust bayesian reinforcement learning through tight lower bounds. In *European Workshop on Reinforcement Learning (EWRL 2011)*, pages 177–188, 2011.

[9] M. Ghavamzadeh and Y. Engel. Bayesian actor-critic algorithms. In *Proceedings of the 24th international conference on Machine learning*, pages 297–304. ACM, 2007.

[10] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[11] M. Ghavamzadeh and Y. Engel. Bayesian policy gradient algorithms. In *NIPS 2006*, 2006.

[12] I.. Osband, D. Russo, and B. Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011, 2013.

[13] Aviv Tamar, Yonatan Glassner, and Shie Mannor. Optimizing the cvar via sampling. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.