Thesis for The Degree of Licentiate of Engineering

Priors and uncertainty in reinforcement learning

Emilio Jorge

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
Gothenburg, Sweden, 2023

Priors and uncertainty in reinforcement learning

Emilio Jorge

© Emilio Jorge, 2023
except where otherwise stated.
All rights reserved.

ISSN 1652-876X

Department of Computer Science and Engineering
Division of Data Science and AI
Chalmers University of Technology | University of Gothenburg
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2023.

To my family.

Priors and uncertainty in reinforcement learning

Emilio Jorge

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg

Abstract

Handling uncertainty is an important part of decision-making. Leveraging uncertainty for guiding exploration to discover higher rewards has been a standard approach for a long time, using both ad hoc and more principled approaches. Additionally, in the last decades, more work has been done with treating uncertainty as something to be avoided and creating risk-sensitive decision makers that wish to avoid risky behaviour. In this licentiate thesis, we study different approaches for managing uncertainty by presenting two papers. In the first paper, we look at how to model value function distributions in a way that captures the dependence between models and future values. We use the observation that the probability of a particular model depends on the value function to create a Monte Carlo algorithm that takes this into account. In the second paper, we study how a zero-sum minimax game between nature that selects a task distribution and an agent that selects a policy can be used to find minimax priors. We show some properties of this game and propose methods for finding its solution. Additionally, we show experimentally that the agents that optimize for this prior are robust to prior misspecification.

Keywords

Uncertainty, Bayesian decision making, Minimax Bayes, Reinforcement learning, Bayesian reinforcement learning

List of Publications

Appended publications

This thesis is based on the following publications:

[**Paper I**] **E. Jorge**, H. Eriksson, C. Dimitrakakis, D. Basu, D. Grover, *Inferential Induction: A Novel Framework for Bayesian Reinforcement Learning*
Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops, PMLR 137:43-52, 2020.

[**Paper II**] T.K Buening, C. Dimitrakakis, H. Eriksson, D. Grover, **E. Jorge**, *Minimax-Bayes Reinforcement Learning*
In submission to AISTATS 2023.

Other publications

The following publications were published previously, or are currently in submission/under revision. However, they are not appended to this thesis, due to contents overlapping that of appended publications or contents not sufficiently related to the thesis.

[**a**] **E. Jorge**, M. Kågebäck, FD. Johansson, E. Gustavsson, *Learning to play Guess Who? and inventing a grounded language as a consequence. Deep Reinforcement Learning NIPS 2016 Workshop.*

[**b**] **E. Jorge**, L. Brynte, C. Cronrath, O. Wigström, K. Bengtsson, E. Gustavsson, B. Lennartson, M. Jirstrand, *Reinforcement learning in real-time geometry assurance.*
Procedia CIRP 72 (2018), 1073-1078.

[**c**] C. Cronrath, **E. Jorge**, J. Moberg, M. Jirstrand, B. Lennartson, *BAgger: A Bayesian Algorithm for Safe and Query-efficient Imitation Learning.*
IROS 2018 Workshop on Machine Learning in Robot Motion Planning.

[**d**] A. Rahbar, **E. Jorge**, D. Dubhashi, M.Haghir Chehreghani, *Do Kernel and Neural Embeddings Help in Training and Generalization?*
Neural Processing Letters (2022): 1-15.

[**e**] T. Kleine Büning; C. Dimitrakakis; H. Eriksson, D. Grover, **E. Jorge**, *Minimax-Bayes Reinforcement Learning.*
The 15th European Workshop on Reinforcement Learning (2022).

[**f**] **E. Jorge***, H. Eriksson*, C. Dimitrakakis*, D. Basu, D. Grover, *On Bayesian Value Function Distributions.*
The 15th European Workshop on Reinforcement Learning (2022).

Acknowledgment

I would like to start by thanking my supervisor Christos Dimitrakakis, his support, knowledge, and guidance have been a very important part of this process. I also want to thank the rest of the group, especially Divya and Hannes who have kept me company in Gothenburg as well as Thomas, Meirav and Marie with whom I have shared many pleasant discussions over Zoom. I would like to give a special thank you to Debabrota, who has been incredibly helpful both as a friend and also as a researcher and whose wisdom I hope will brush off on me, even just a little bit. I would also like to thank our project collaborators, Kevin and Emir at KTH for the friendly discussions.

I would like to thank my discussion leader Brendan, who has taken himself time to read my work and to make the seminar possible.

My work is funded by the Wallenberg AI, Autonomous Systems and Software Program (WASP) which in turn is funded by the Knut and Alice Wallenberg Foundation and I am very grateful for their support. I am also grateful to the Chalmers Centre for Computational Science and Engineering (C3SE) and the Swedish National Infrastructure for Computing (SNIC), which enabled many of the experiments needed for my work.

I would like to thank my regular lunch-buddies during the years, Niklas, Tobias, and Emil for the great company, encouragement, and lots of fun. I am also grateful for all the other PhD students in the corridor, whose presence make the visits to the office more fun. I would also like to thank the other staff at our division, the faculty, and the administrative staff, especially Morteza, Devdatt, Alexander, and Fredrik who have shared their guidance and funny stories.

My friends outside the department should not be forgotten, especially those whom I have pestered with my latest research problem; Helga, Erik, Edvin, and Jacob.

Finally, I would like to thank my family. To my mother, brother, and sister that are always there for me if I need them. To my father who is no longer with us, but always with me. To my partner Linnea for being supportive and to Hjalmar whose little laugh can brighten the darkest of days.

Contents

Abstract iii

List of Publications v

Acknowledgement vii

I Summary 1

1 Introduction 3

2 Background 5
2.1 Reinforcement learning 5
2.1.1 Q-learning 7
2.1.2 Bayesian reinforcement learning 7
2.1.2.1 Model-based Bayesian reinforcement learning . 8
2.1.2.2 Model-free Bayesian reinforcement learning . . 9
2.1.2.3 Bayes-adaptive policies 9
2.1.3 Regret 10
2.1.3.1 Bayes optimal regret 10
2.1.3.2 Bayesian regret 10
2.2 Minimax problems and zero-sum games 10

3 Summary of Included Papers 13
3.1 Inferential Induction: A Novel Framework for Bayesian Reinforcement Learning 13
3.2 Minimax-Bayes Reinforcement Learning 16

4 Discussion and Future Work 19

Bibliography 21

II Appended Papers 27

Paper I - Inferential Induction: A Novel Framework for Bayesian Reinforcement Learning

Paper II - Minimax-Bayes Reinforcement Learning

Part I

Summary

Chapter 1

Introduction

Almost everything in life is uncertain. Everything from the weather, election results, and outcomes of an experiment or a football match is unknown until it happens. Even many of the things we often think of as certain, such as "The train leaves in 5 minutes" are uncertain, it just states the intended, and hopefully most likely, outcome. In many cases, we are aware of the uncertainty so we bring an umbrella to avoid getting wet if it rains, while in other cases we are taken totally by surprise.

In this thesis, we discuss how understanding our uncertainties can help us take good decisions. In many real-world cases, we try to avoid uncertainty and try to handle all possible scenarios, slowing down when reaching a corner, just in case something is coming from the other side, or bringing an umbrella if there is a very low risk of rain. In other situations we might wish to seek uncertainty, trying out a new restaurant or meeting new acquaintances, on the chance that maybe will it be rewarding in the long run.

Uncertainty can be separated into two variants. One kind is the uncertainty that comes from probabilistic sampling, such as the outcome of rolling a fair die, and is sometimes referred to as aleatoric uncertainty. Uncertainty due to lack of information, sometimes known as epistemic uncertainty, is the kind of uncertainty that can be reduced by obtaining more information, such as rolling more dice to obtain information about the probability of each outcome.

Reinforcement learning is a framework for sequential decision-making that has made great strides in the last few years (Mnih et al. 2015; Vinyals et al. 2019; Schulman et al. 2017) and is solving more and more complicated tasks. An issue with many approaches is the large amount of data needed. Leveraging uncertainty can be a good way of reducing the sample complexity (Auer, Jaksch and Ortner 2008; Azar, Osband and Munos 2017; Osband et al. 2016). One way of reducing sample complexity is to try to estimate the underlying model that governs the state transitions, often combined with the uncertainty of the estimate (Dearden, Friedman and Andre 1999; Deisenroth and Rasmussen 2011; Osband, Russo and Van Roy 2013; Chua et al. 2018; Lin, Jin and Jordan 2020; Moerland, Broekens and Jonker 2021). In reinforcement learning, it is primarily the epistemic variant of uncertainty that is of interest as the aleatoric

uncertainty disappears when it is the average performance that is of interest, as it often is in reinforcement learning. Risk-sensitive reinforcement learning is an exception to this (Chow et al. 2017; Clements et al. 2019; Eriksson and Dimitrakakis 2019), as unfavorable aleatoric risk can cause bad outcomes, such as a car crash.

One way of representing uncertainty is through a Bayesian approach, having probability distributions over the properties of interest. Generally, this is done by having suitable prior and posterior such that they are a conjugate distribution for easier calculations. The setting of priors is often important for the performance of the algorithms, yet not very well studied in the literature.

This thesis is structured as follows. In Chapter 2 we introduce the necessary overview of the related topics to help readers that are not familiar with all of the topics. In Chapter 3 the two papers, Jorge et al. (2020) (Paper I) and Buening et al. (2022) (Paper II), are summarized with the most important results. Thereafter, in Chapter 4, a discussion about possible extensions and future work can be found. Finally, the second half of the thesis contains the included papers. Some papers by the author, Jorge et al. (2016), Jorge et al. (2018), Cronrath et al. (2018), Rahbar et al. (2022) and Jorge et al. (2022), are not included in this thesis as they are either older work or not sufficiently relevant to the topic.

Chapter 2

Background

2.1 Reinforcement learning

Reinforcement learning is a framework for interacting with sequential decision-making tasks. This is usually formulated as a Markov decision process (MDP) μ which is defined by its tuple $(\mathcal{P}, \rho, \gamma, A, S)$. In the MDP, an agent interacts with the environment, visiting different states and taking actions that yield rewards and new states. The sets S and A define the states the agent can visit and the actions the agent can take. In this thesis, we will focus on the setting with finite sets of states and actions. Then, the transitions of the system, $\mathcal{P} : S \times A \to S$, is a transition matrix that defines $P(s'|s, a)$ and the reward is given by $\rho(r|s, a)$. The discount factor γ controls how much the agent cares about immediate rewards compared to rewards occurring later on in the future.

The agent interacts with the MDP in rounds denoted t. The agent observes a state $s_t \in S$, interacts with the environment using an action a_t, receives a reward $r_t \sim \rho(s_t, a_t)$ and observes a new state $s_{t+1} \sim \mathcal{P}(s_t, a_t)$. The agent uses a strategy for selecting the action which is known as the policy, denoted $\pi : S \to A$. The agent's policy can be either stochastic or deterministic but is generally assumed to be a stationary function. It is also possible to have a policy that adapts as it observes information about the MDP, such policies are known as adaptive policies, more on them in Section 2.1.2.3.

The goal of reinforcement learning is to find the policy

$$\pi^*(s) = \arg\max_{\pi \in \mathit{\Pi}} V_\mu^\pi(s) \tag{2.1}$$

where $\mathit{\Pi}$ is the set of available policies and $V_\mu^\pi : S \to \mathbb{R}$ is the discounted expected total reward from a certain state

$$V_\mu^\pi(s_0) = \mathbb{E} \sum_{t=0}^{\infty} \gamma^t r_t,\ a_t \sim \pi(s_t),\ s_{t+1} \sim \mathcal{P}(s_t, a_t),\ r_t \sim \rho(s_t, a_t). \tag{2.2}$$

for policy π in MDP μ.

Alternatively, the horizon of interest can be a finite value H which instead gives the value function

$$V_\mu^\pi(s_0) = \mathbb{E}\sum_{t=0}^{H}\gamma^t r_t,\ a_t \sim \pi(s_t),\ s_{t+1} \sim \mathcal{P}(s_t, a_t),\ r_t \sim \rho(s_t, a_t). \tag{2.3}$$

Often γ is set to 1 in this case, removing the discounting. Similarly to the value function V, we can create a value function $Q_\mu^\pi : S \times A \to \mathbb{R}$ that gives the expected sum of rewards when taking an action in the state and then following π thereafter.

In the infinite horizon case, we can write these properties using the Bellman equations, which are defined recursively

$$V_\mu^\pi(s_t) = \sum_{a_t \in A}\pi(a_t|s)\left(\mathbb{E}_{r\sim\rho(s_t,a_t)}[r] + \gamma\sum_{s_{t+1}\in S}P(s_{t+1}|s_t,a_t)V_\mu^\pi(s_{t+1})\right) \tag{2.4}$$

$$\begin{aligned} Q_\mu^\pi(s_t, a_t) =& \mathbb{E}_{r\sim\rho(s_t,a_t)}[r] + \\ &\gamma\sum_{s_{t+1}\in S}P(s_{t+1}|s_t,a_t)\sum_{a_{t+1}\in A}\pi(a_{t+1}|s_t)Q_\mu^\pi(s_{t+1},a_{t+1}) \end{aligned} \tag{2.5}$$

Sometimes the model μ is known[1] and S and A are finite. In these cases, an optimal policy can be found efficiently using dynamic programming approaches such as value iteration or policy iteration (cf. (Sutton and Barto 2018)). These approaches are generally based on the Bellman optimality equations given by

$$V_\mu^*(s_t) = \max_{a_t\in A}\left(\mathbb{E}_{r\sim\rho(s_t,a_t)}[r] + \gamma\sum_{s_{t+1}\in S}P(s_{t+1}|s_t,a_t)V_\mu^*(s_{t+1})\right) \tag{2.6}$$

$$Q_\mu^*(s_t, a_t) = \mathbb{E}_{r\sim\rho(s_t,a_t)}[r] + \gamma\sum_{s_{t+1}\in S}P(s_{t+1}|s_t,a_t)\max_{a_{t+1}\in A}Q_\mu^*(s_{t+1},a_{t+1}) \tag{2.7}$$

If the model is unknown, which is often the case of interest, samples from interaction with the environment must be used to estimate the value functions. This gives rise to an important part of solving the problem, making sure that the agent's policy explores the state and action space enough to obtain necessary information about all parts of the state-action space. This has to be balanced against taking actions that seem to give high rewards. This dilemma is known as the exploration-exploitation trade-off.

In the case where there is no state (or when $P(s_{t+1}|s_t, a_t) = P(s_{t+1})$, the reinforcement learning problem is known as the multi-armed bandit (MAB) problem and is a well-studied research area and these problems are generally easier to work with. In MABs, no transition kernel is needed and the sequential aspect of reinforcement learning, where a sequence of good actions may be needed to obtain a large reward, can be ignored.

[1] In practice this means that $\mathcal{P}$ and ρ are known, whereas γ, A and S are always assumed to be known.

2.1.1 Q-learning

A frequently used algorithm for solving reinforcement learning tasks is Q-learning (Watkins and Dayan 1992). In its original form, Q-learning handles settings with discrete S and A and works by leveraging the Bellman optimality equation for the state-action value function $Q(s, a)$ found in Equation (2.7). The algorithm uses observed data obtained by the agent to update the Q-function using

$$Q[s, a] \leftarrow (1 - \alpha)Q[s, a] + \alpha \left(r + \gamma \max_{a'} Q[s', a'] \right). \tag{2.8}$$

Here α is a learning rate, s' is the new state and r is the observed reward. Often this update is combined with an ϵ-greedy policy[2] to guide the exploration, but other policies are also possible. The algorithm is shown in Algorithm 1.

Algorithm 1 Q-learning

Input: Initialize Q-table $Q(s, a)$ arbitrarily.
for Episode i **do**
  Starting state $s = s_0$
  for step t until s terminal state **do**
    Select action $a = a_t$ using $Q(s, \cdot)$, often ϵ-greedy.
    Obtain $r = r_t, s' = s_{t+1}$ from environment.
    if s' is terminal state **then**
      $Q[s, a] \leftarrow (1 - \alpha)Q[s, a] + \alpha r$
    else
      $Q[s, a] \leftarrow (1 - \alpha)Q[s, a] + \alpha \left(r + \max_{a'} \gamma Q[s', a']\right)1$
    end if
    s = s'
  end for
end for

Under some technical conditions[3] the Q-table is guaranteed to converge to Q^*, from which the optimal policy can be extracted.

There are many extensions of Q-learning such as extensions to non-discrete states, adding function approximation, and reusing of data, making it applicable to most MDP settings (Dearden, Friedman and Russell 1998; Mnih et al. 2015; Hasselt, Guez and Silver 2015; Bellemare, Dabney and Munos 2017).

2.1.2 Bayesian reinforcement learning

The goal of this section is to give a brief overview of the Bayesian reinforcement learning setting. To read more on the topic, a more thorough exposition can be found in Ghavamzadeh et al. (2015) and more recently in Wang et al. (2022).

[2]The policy that with probability $(1 - \epsilon)$ selects $a_t = \arg\max_a Q(s_t, a)$ and uniformly random from the set of actions otherwise.

[3]Decay of α_t, either $\gamma < 1$ or a finite episode length, and sufficient exploration.

Bayesian reinforcement learning (BRL) integrates the ideas of Bayesian probabilities with reinforcement learning. Generally, just as with reinforcement learning in general, we can separate them into two variants, model-free and model-based.

A common trait for both the model-based and model-free algorithms is that they try to leverage the uncertainty to improve the policy. Generally, the algorithms use the uncertainty to guide exploration towards the uncertain regions of the state-action space that have yet to be sufficiently mapped out (Osband, Russo and Roy 2013; Jorge et al. 2020; Hao and Lattimore 2022). The model uncertainty can also be used in the context of risk-sensitive RL to guide safe exploration, in a way that attempts to avoid the agent from taking actions that can give undesirable events. Examples of this are Clements et al. (2019), Cronrath et al. (2018), Derman et al. (2020) and Eriksson et al. (2022).

2.1.2.1 Model-based Bayesian reinforcement learning

With model-based BRL, we are interested in leveraging an explicit distribution over models. Often, the desired property is given by an integral over the posterior over models

$$V^{\pi}_{\beta_D}(s) = \int_{\mathcal{M}} V^{\pi}_{\mu}(s)\beta(\mu|D)d\mu \tag{2.9}$$

where $\beta_D = \beta(\mu|D)$ is the posterior over MDPs. The posterior $\beta(\mu|D)$ is given by Bayes' theorem

$$\beta(\mu|D) = \frac{P(\mu|D)P(\mu)}{\int_{\mathcal{M}} P(\mu|D)P(\mu)d\mu} \tag{2.10}$$

where D is the observed data, and $\beta(\mu) = P(\mu)$ is the prior distribution over MDPs.

One way to use this is to just take the gradient with respect to the policy, which is the approach used in Ghavamzadeh and Engel (2006).

Another popular approach is based on the ideas of Thompson sampling (Thompson 1933), originating in the MAB setting which samples the actions proportionally to how likely they are to be the action with the highest value under the current posterior over the rewards of each arm. In the RL setting this is referred to as posterior sampling for reinforcement learning (PSRL) where at timestep t a model μ_t is sampled from $\beta(\mu)$ and the policy then acts optimally with respect to the sampled model in the real environment for a certain amount of time, before updating the posterior with the new data and sampling a new model. Examples of this approach and its theoretical analysis can be found in Strens (2000), Osband, Russo and Roy (2013), Agrawal and Jia (2017) and Ouyang et al. (2017). This generally requires a tractable posterior over the models, something that is only simple in a few cases, such as in discrete MDPs with uncorrelated states or linear models. In some cases, Gaussian processes or other methods can be used, but they come with their own demands or assumptions (Tziortziotis, Dimitrakakis and Blekas 2014; Deisenroth and Rasmussen 2011; Fan and Ming 2021).

When more complicated models are required, such as neural networks, it is harder to maintain a posterior. Different approaches, such as bootstrapped models or other forms of Bayesian neural networks, can be used in these cases (Gal, McAllister and Rasmussen 2016; Chua et al. 2018; Zhou, Li and Wang 2020).

2.1.2.2 Model-free Bayesian reinforcement learning

Model-free approaches to BRL avoid, at least explicitly, using a model. Instead, they compute the uncertainty in the value functions directly, using variants of the Bellman equations, and use this to guide the agent's decisions. Examples of this are Dearden, Friedman and Russell (1998), Osband et al. (2016), Bellemare, Dabney and Munos (2017), O'Donoghue et al. (2018), Tang and Agrawal (2018), Osband, Aslanides and Cassirer (2018) and O'Donoghue (2021) which incorporate uncertainty into the Q-function in Q-learning in a variety of ways. The approach used in O'Donoghue et al. (2018) is to use an estimate, and upper bound, of the uncertainty of the Q-function and to use Thompson sampling of the Q-functions to guide exploration. In Osband et al. (2016) and Osband, Aslanides and Cassirer (2018) it is instead bootstrapped Q-functions that are used to model the uncertainty.

Some other examples of model-free BRL can be found in Engel, Mannor and Meir (2003) and Fellows, Hartikainen and Whiteson (2021).

2.1.2.3 Bayes-adaptive policies

Often when talking about policies, the policies are assumed to be stationary, such that $a_t \sim \pi(a|s_t)$. It is also possible to have policies that are history-dependent, $a_t \sim \pi(a|s_t, a_{t-1}, s_{t-1}, ..., a_0, s_0)$. These policies are known as adaptive policies and can be seen as more of an algorithm, that incorporates new information to adapt and change its behaviour. One way of doing this in the BRL context is using a Bayes-adaptive policy. Bayes-adaptive policies utilize that an MDP can be augmented to a history-dependent MDP, where the new MDP has states S^+ that contain both the original state and the history of all states and actions. The dynamics of this MDP incorporate both the starting belief $\beta(\mu)$ as well as the fact that the posterior will be updated for each new state that is visited. The dynamics of this augmented MDP are known, as the posterior update is deterministic for a given sample, and as such, this MDP can be solved optimally. A thorough exposition of Bayes-adaptive policies can be found in Duff (2002).

An issue with Bayes-adaptive policies is that the posterior update dynamics create a branching tree for the different possible histories, making computation of these policies intractable in many cases that do not have a small horizon. In the case of multi-armed bandits, an efficient algorithm can be found as described in Gittins, Glazebrook and Weber (2011). Approaches for finding such policies approximately in the RL case also exist, such as Guez, Silver and Dayan (2012). Another approach is to approximately solve the problem repeatedly, creating an embedding of beliefs that the policy can act upon, as is done in Boutilier et al. (2020) and Zintgraf et al. (2021).

2.1.3 Regret

When evaluating policies, it is common to use the concept of regret. Regret can be seen as the answer to the question: "How much worse is my policy than the optimal policy?" and can be written as

$$R(\pi, \mu) = V_\mu^{\pi^*}(s_0) - V_\mu^{\pi}(s_0). \tag{2.11}$$

Generally, regret is talked about in terms of $\mathcal{O}$ complexity in the limit of the horizon T. The goal is to have algorithms that are sub-linear in T without too large a dependence on the other parameters. Often it is also written as $\widetilde{\mathcal{O}}$ such that polylogarithmic terms are discarded.

As an example, for general discrete MDPs Azar, Osband and Munos (2017) show that their algorithm, UCBVI, has a complexity of $\widetilde{\mathcal{O}}(\sqrt{HSAT})$ with high probability.

2.1.3.1 Bayes optimal regret

In the Bayesian setting, it makes sense to define the Bayes optimal regret. This is given by comparing a policy with the Bayes optimal policy, which is the policy $\pi^*(\beta) = \arg\max_\pi \int_{\mathcal{M}} \beta(\mu) V_\beta^\pi \, d\mu$, for a given belief. This is then defined

$$\begin{aligned} R(\pi, \beta) &= V_\beta^{\pi^*(\beta)} - V_\beta^\pi && (2.12) \\ &= \int_{\mathcal{M}} \beta(\mu) \left(V_\beta^{\pi^*(\beta)} - V_\mu^\pi \right) d\mu && (2.13) \end{aligned}$$

2.1.3.2 Bayesian regret

Another alternative definition of regret is to compare it with a policy that knows what MDP it is interacting with. This gives the Bayesian regret

$$\begin{aligned} L(\pi, \beta) &= \mathbb{E}_{\mu \sim \beta} R(\pi, \mu) && (2.14) \\ &= \int_{\mathcal{M}} \beta(\mu) \left(V_\mu^{\pi^*(\mu)} - V_\mu^\pi \right) d\mu. && (2.15) \end{aligned}$$

This definition can be easier to work with, as it does not require knowledge about the Bayes optimal policy.

2.2 Minimax problems and zero-sum games

In the field of game theory, the concept of games is a construction where multiple agents interact and where each agent receives a payoff based on the actions of all the agents. In this thesis, we are interested in simultaneous zero-sum games which are games where there are two agents that take actions simultaneously, and the sum of the agents' payoffs is always zero. One example of such a game is rock-paper-scissors, seen in Table 2.1 where the agent has a payoff of 1 if it wins, and -1 if it loses.

Table 2.1: The payoff matrix for rock-paper-scissors.

		Agent 2		
		Rock	Paper	Scissors
Agent 1	Rock	0,0	-1,1	1,-1
	Paper	1,-1	0,0	-1,1
	Scissors	-1,1	1,-1	0,0

The value for each agent is its expected payoff under the two agents' policies, here for agent 1,

$$V^{(1)}(\pi_1, \pi_2) = \mathbb{E}_{a^{(1)} \sim \pi_1, a^{(2)} \sim \pi_2} \text{Payoff}^{(1)}(a^{(1)}, a^{(2)}). \tag{2.16}$$

A property that can be of interest is finding the Nash equilibrium. A Nash equilibrium is the equilibrium where neither agent benefits from deviating from their policy unless the other agent also does it.

Under certain conditions, the existence of such equilibrium can be guaranteed by Sion's minimax theorem, stated below. As a small note, quasi-convexity is a weaker requirement than convexity since all convex functions are quasi-convex (but not vice-versa), and the same holds for quasi-concavity.

Theorem 1. *Sion's minimax theorem (Sion 1958). Let M and N be convex spaces of which one is compact, and f a real-valued function on $M \times N$ with the following properties: f is upper semicontinuous and quasi-concave in M for all $y \in N$ and quasi-convex and lower semicontinuous in N for all $x \in M$. Then* $\sup_{x \in M} \inf_{y \in N} f(x, y) = \inf_{y \in N} \sup_{x \in M} f(x, y)$. *If the spaces are compact,* sup *and* inf *can be replaced by* max *and* min.

When the conditions of the theorem hold and $V = \sup_{x \in M} \inf_{y \in N} f(x, y) = \inf_{y \in N} \sup_{x \in M} f(x, y)$, the game is said to have value V.

For the interested reader, more background on minimax games can be found in Berger (1985).

Chapter 3

Summary of Included Papers

3.1 Inferential Induction: A Novel Framework for Bayesian Reinforcement Learning

In model-free Bayesian reinforcement learning, there is no explicit model over the environment. Nevertheless, the sampling of data to update the value functions with the Bellman equations makes implicit use of the empirical MDP, i.e. the MDP you get when you bootstrap transitions from the data.

In this work, we take a backward induction approach, iteratively calculating the value function $P^\pi_\beta(V_i|D_t)$ from $P^\pi_\beta(V_{i+1}|D_t)$. This can be written

$$P^\pi_\beta(V_i \mid V_{i+1}, D_t) = \int_{\mathcal{M}} P^\pi_\mu(V_i \mid V_{i+1}) \, \mathrm{d}P^\pi_\beta(\mu \mid V_{i+1}, D_t). \tag{3.1}$$

where D_t is the observed data at time step t and β is the prior over MDPs.

Here, we need to note that $P^\pi_\beta(\mu \mid V_{i+1}, D_t) \neq P_\beta(\mu \mid D_t)$ to avoid the incorrect modeling choice from methods that implicitly use the empirical MDP.

Using the idea formulated in Equation (3.1), we create a Bayesian algorithm that jointly takes the model and the value function into account which can be found in Algorithm 2. To make this more practical, it relies on Monte Carlo sampling and uses an approximate representation ψ_i to represent $P(V_i|D)$.

Generally, we can write

$$\psi_i(B) \triangleq P^\pi_\beta(V_i \in B|D_t) \tag{3.2}$$

$$= \int_{\mathcal{V}} \int_{\mathcal{M}} \mathbb{I}\left\{\mathscr{B}^\pi_\mu V_{i+1} \in B\right\} \, \mathrm{d}P^\pi_\beta(\mu|V_{i+1}, D_t) \, \mathrm{d}\psi_{i+1}(V_{i+1}), \tag{3.3}$$

where $\mathscr{B}$ is the Bellman operator

$$\mathscr{B}^\pi_\mu V(s) \triangleq \rho(s) + \gamma \sum_{s' \in S} \sum_{a \in A} \pi(a|s) P_\mu(s' \mid s, a) V(s'). \tag{3.4}$$

In practice, we use a multivariate Gaussian distribution for ψ_i which we fit using the samples we obtain.

Algorithm 2 Bayesian Backwards Induction (BBI) with Method 1

Input: Prior β, data D, lookahead H, discount γ, N_μ, N_V.
Initialise $\psi_H = P(V)$.
Sample $\hat{M} \triangleq \left\{ \mu^{(j)} \;\middle|\; j \in [N_\mu] \right\}$ from $\beta(\mu \mid D)$.
for $i = H-1, \dots, 1$ **do**
    Sample $V_{i+1}^{(k)} \sim \psi_{i+1}(\boldsymbol{v})$ for $k \in [N_V]$.
    Generate n utility samples $u_m^{(j)}$ using rollouts in μ_j.
    Calculate weight w_{jk} from $P(V_{i+1}^{(k)} | u_m)$.
    Calculate Bayesian Q-value $\mathcal{Q}_i$ with weighted Q-learning using w_{jk}.
    Update policy π_i.
    Fit ψ_i using weighted Bellman update.
end for
return $\pi = (\pi_1, \dots, \pi_H)$.

Using Bayes rule we can reformulate the inner term in Equation (3.1)

$$P_\beta^\pi(\mu \mid V_{i+1}, D_t) = \frac{P_\mu^\pi(V_{i+1})\,\mathrm{d}\beta(\mu \mid D_t)}{\int_{\mathcal{M}} P_\mu^\pi(V_{i+1})\,\mathrm{d}\beta(\mu \mid D_t)}, \tag{3.5}$$

the issue still remains on how to compute $P_\mu^\pi(V_{i+1})$.

We chose to model this using the sum of discounted rewards u_m of sampled trajectories from state s_m in μ and use these samples with a Gaussian likelihood. Sampling N_μ MDPs $\mu^{(j)} \sim \beta(\mu \mid D_t)$ and N_V value samples $V_{i+1}^{(k)} \sim \psi_{i+1}$ we get weights

$$w_{jk} \triangleq \frac{\sum_{m=1}^{n} e^{-\frac{\left|V_{i+1}^{(k)}(s_m) - u_m^j\right|^2}{2\sigma^2}}}{\sum_{j'=1}^{N_\mu}\sum_{m=1}^{n} e^{-\frac{\left|V_{i+1}^{(k)}(s_m) - u_m^{j'}\right|^2}{2\sigma^2}}}, \tag{3.6}$$

on each of these samples. The weight can be seen as an estimate of how well the value function sample matches the MDP sample.

For each sample, we can do a one-step update of the value function

$$V_i^{(j,k)} \triangleq \mathscr{B}_{\mu^{(j)}}^\pi V_{i+1}^{(k)}. \tag{3.7}$$

With these new values $V_i^{(k)}$, together with the weights, we fit the approximation

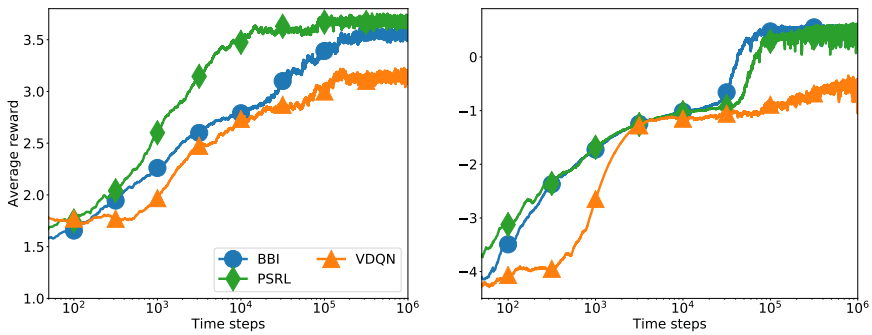
$$\psi_i(B) = \frac{1}{N_V N_\mu} \sum_{k=1}^{N_V}\sum_{j=1}^{N_\mu} \mathbb{I}\left\{ V_i^{(j,k)} \in B \right\} w_{jk}. \tag{3.8}$$

The policy is updated in each backward induction step using Bayesian Q-values with the same weights and samples,

$$\mathcal{Q}_i(s,a) \approx \sum_{j,k} \left[\rho_{\mu^{(j)}}(s,a) + \sum_{s'} P_\mu^{(j)}(s'|s,a) V_{i+1}^{(k)}(s') \right] \frac{w_{jk}}{N_\mu N_V} \tag{3.9}$$

and taking the action with the highest value.

We evaluate this algorithm on some benchmarks, a subset of which can be found in Figure 3.1. We find that the algorithm is competitive with PSRL (Osband, Russo and Roy 2013), but not performing noticeably better. Additionally, BBI scales significantly worse due to the expensive backward induction step with respect to multiple MDPs, compared to the single MDP for PSRL.



(a) NChain (b) LavaLake 10×10

Figure 3.1: Evolution of the average reward for NChain, LavaLake 10×10.

While the algorithm does not outperform all state-of-the-art algorithms, we believe that this new perspective on conditional value function distributions can open up new and more efficient algorithms, when Bayesian value functions are calculated more correctly.

3.2 Minimax-Bayes Reinforcement Learning

Bayesian reinforcement learning is a well-studied framework for analyzing reinforcement learning tasks. Although a lot of research has studied the development of algorithms, much less work has been done on principled methods for selecting the priors needed for these algorithms. In this paper, we take a minimax approach to find worst-case priors. Inspired by Grünwald and Dawid (2004) we formulate a zero-sum simultaneous game between nature and an agent. In our game, nature attempts to find a prior over the set of reinforcement learning tasks (MDPs) that will maximize the agent's Bayesian regret, while the agent attempts to find a robust policy that minimizes the Bayesian regret.

The task is easier to work with in the case where there is a finite set of tasks that nature sets its prior over. In that case, the Bayesian regret for a given policy becomes a hyperplane over the prior beliefs, as the Bayes regret expectation is linear in the prior. The Bayesian regret of the optimal (and prior-aware) policy becomes a convex function, lower bounded by planes of all the possible policies. We visualize this in Figure 3.2.

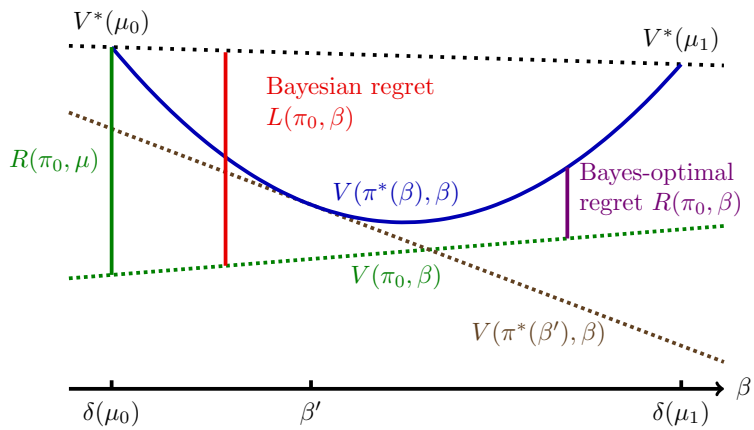


Figure 3.2: Illustration of the notions of regret for different policies with a belief over two MDPs. The figure also shows how the Bayesian regret of the optimal, prior-aware policy is lower bounded by the different policies.

Finding these minimax solutions is often not easy, but in the case of finite sets of MDPs, we outline two methods for locating the solutions.

Firstly, we have a stochastic gradient descent algorithm found in Algorithm 3 which finds a $\epsilon-$stationary point with an iteration complexity of

$$\mathcal{O}\left(|A|^3 T^6 \left(\frac{\left(T^4+\sigma^2\right)\widehat{\Delta}_\Phi}{\epsilon^6} + \frac{\widehat{\Delta}_0}{\epsilon^4}\right)\max\left\{1, \frac{\sigma^2}{\epsilon^2}\right\}\right). \tag{3.10}$$

Here T is the horizon of the task, σ^2 is the variance of the gradient estimator G,

$\mathscr{P}$ is a projection operator, and $\widehat{\Delta}_0, \widehat{\Delta}_\Phi$ are constants relating to the Moreau envelope of the initial policy π_0.

Algorithm 3 Stochastic GDA

Input: policy π_0, belief β_0, learning rates (η_π, η_β) and stochastic gradient estimators G_π, G_β for $\nabla_\pi \mathscr{L}, \nabla_\beta \mathscr{L}$.
for $t = 1, \ldots, T$ **do**
  Get directions $g_\beta = \frac{1}{M} \sum_i G_\beta^{(i)}(\pi_{t-1}, \beta_{t-1})$
  and $g_\pi = \frac{1}{M} \sum_i G_\pi^{(i)}(\pi_{t-1}, \beta_{t-1})$ using M i.i.d samples
  $\pi_t \leftarrow \pi_{t-1} - \eta_\pi g_\pi$
  $\beta_t \leftarrow \mathscr{P}_{\mathcal{B}}\Big(\beta_{t-1} + \eta_\beta g_\beta\Big)$
end for
Output β^*, π^* uniformly at random from $\{(\beta_1, \pi_1), \ldots, (\beta_T, \pi_T)\}$

Secondly, we have a cutting plane algorithm found in Algorithm 4. First, it takes a belief β_t close to the middle of the solution space K_t. Then it finds the optimal policy $\pi^*_{\beta_t}$ for this belief and calculates the regret of the policy for each of the MDPs. The regret $\pi^*_{\beta_t}$ for each of these MDPs gives us a plane C_t. This plane can then be used to cut away a part of K_t from the space of plausible solutions, forming K_{t+1}. This can be repeated until the solution space has an arbitrarily small volume.

Algorithm 4 Cutting plane algorithm for finding the minimax belief

Input: Initial belief set of constraints K_0, Optimal Policy oracle, Policy evaluation oracle, $t = 0$;
for t=0:T-1 **do**
  Obtain $\beta_t \approx \mathbb{E}_{K_t}[x]$
  Obtain optimal policy $\pi^*_{\beta_t}$ and $C_t^{(i)} = R(\pi^*(\beta_t), \beta = \delta_{\mu_i})$.
  $K_{t+1} = K_t \cap \{\beta : C_t^T(\beta - \beta_t) > 0\}$
end for
Return $\beta^* \in K_T$ that has $\frac{\text{VOL}(K_T)}{\text{VOL}(K_0)} < \left(\frac{2}{3}\right)^T$ with high probability and corresponding $\pi^*(\beta^*)$.

Experimentally, we study how the minimax policies can be more robust to change in prior compared to other policies. Figure 3.3 shows the Bayesian regret of the policy that is minimax optimal compared with the policy that is optimal for the uniform prior, as we interpolate between the minimax prior and the uniform prior in the space of priors. We can see that the regret of the minimax policy has lower Bayesian regret in all but the very closest priors to the uniform prior, which makes sense as the competing policy is optimized for the uniform prior.

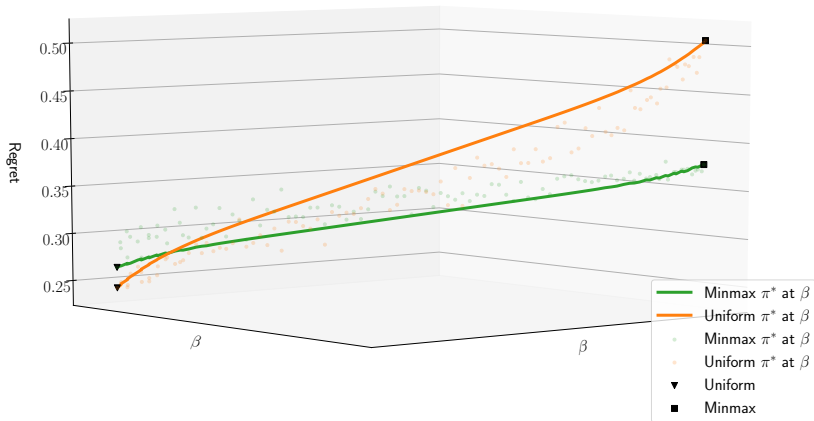


Figure 3.3: Comparison of regret of the minimax policy and the policy that is optimal for the uniform belief β_1. The policies are evaluated over priors that interpolate between the minimax prior and the uniform prior, embedded with a t-SNE projection.

Chapter 4

Discussion and Future Work

In this licentiate thesis, we have discussed a few approaches to Bayesian reinforcement learning. In the first paper, we have illustrated issues with how current Bayesian value functions are calculated in some cases and introduced some ways of computing it more correctly. In the second paper, we discuss the idea of minimax priors in reinforcement learning. We have illustrated some of their properties and proposed methods for finding such priors. Additionally, we have some results indicating that using such priors helps with robustness, potentially opening up possibilities in terms of safer reinforcement learning policies.

A problem with Bayesian algorithms in general is that they tend to be more computationally expensive. An interesting direction for future work is to focus on making fast model-based algorithms, that often are more efficient in terms of how much data is needed, without reducing the distribution to the empirical model.

Another interesting direction is to move in the direction of approximate posteriors that allow for higher-dimensional and more powerful representations, compared to the discrete conjugate posteriors that were primarily covered in this thesis. Interesting approaches would be those of neural networks with Langevin dynamics (Welling and Teh 2011; Mazumdar et al. 2020; Xu et al. 2022) that are still relatively slow yet give good posteriors, or Epistemic neural networks (Osband et al. 2021) which provide a heuristic approximation of the posterior but are very fast.

In the direction of minimax-Bayes reinforcement learning, it would be very interesting to find theoretical results for parametric distributions, such as Dirichlet priors, which are limited to experimental results in our current work.

Bibliography

Agrawal, Shipra and Randy Jia (2017). "Posterior sampling for reinforcement learning: worst-case regret bounds". In: *Advances in Neural Information Processing Systems*, pp. 1184–1194 (cit. on p. 8).

Auer, Peter, Thomas Jaksch and Ronald Ortner (2008). "Near-optimal regret bounds for reinforcement learning". In: *Advances in neural information processing systems* 21 (cit. on p. 3).

Azar, Mohammad Gheshlaghi, Ian Osband and Rémi Munos (2017). "Minimax regret bounds for reinforcement learning". In: *International Conference on Machine Learning*. PMLR, pp. 263–272 (cit. on pp. 3, 10).

Bellemare, Marc G, Will Dabney and Rémi Munos (2017). "A distributional perspective on reinforcement learning". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 449–458 (cit. on pp. 7, 9).

Berger, James O. (1985). *Statistical decision theory and Bayesian analysis*. Springer (cit. on p. 11).

Boutilier, Craig, Chih-Wei Hsu, Branislav Kveton, Martin Mladenov, Csaba Szepesvári and Manzil Zaheer (2020). "Differentiable meta-learning of bandit policies". In: *Advances in Neural Information Processing Systems* 33, pp. 2122–2134 (cit. on p. 9).

Buening, Thomas Kleine, Christos Dimitrakakis, Hannes Eriksson, Divya Grover and Emilio Jorge (2022). "Minimax-Bayes Reinforcement Learning". In: (cit. on p. 4).

Chow, Yinlam, Mohammad Ghavamzadeh, Lucas Janson and Marco Pavone (2017). "Risk-constrained reinforcement learning with percentile risk criteria". In: *The Journal of Machine Learning Research* 18.1, pp. 6070–6120 (cit. on p. 4).

Chua, Kurtland, Roberto Calandra, Rowan McAllister and Sergey Levine (2018). "Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett. Vol. 31. Curran Associates, Inc. (cit. on pp. 3, 9).

Clements, William R., Benoît-Marie Robaglia, Bastien Van Delft, Reda Bahi Slaoui and Sébastien Toth (2019). "Estimating Risk and Uncertainty in Deep Reinforcement Learning". In: *CoRR* abs/1905.09638. arXiv: 1905.09638. URL: http://arxiv.org/abs/1905.09638 (cit. on pp. 4, 8).

Cronrath, Constantin, Emilio Jorge, John Moberg, Mats Jirstrand and Bengt Lennartson (2018). "BAgger: A Bayesian algorithm for safe and query-efficient imitation learning". In: *Machine Learning in Robot Motion Planning–IROS 2018 Workshop* (cit. on pp. 4, 8).

Dearden, Richard, Nir Friedman and David Andre (1999). "Model based Bayesian exploration". In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 150–159 (cit. on p. 3).

Dearden, Richard, Nir Friedman and Stuart Russell (1998). "Bayesian Q-learning". In: *Aaai/iaai*, pp. 761–768 (cit. on pp. 7, 9).

Deisenroth, M. P. and C. E. Rasmussen (2011). "PILCO: A Model-Based and Data-Efficient Approach to Policy Search". In: *International conference on Machine Learning (ICML)*. Bellevue, WA, USA (cit. on pp. 3, 8).

Derman, Esther, Daniel Mankowitz, Timothy Mann and Shie Mannor (2020). "A bayesian approach to robust reinforcement learning". In: *Uncertainty in Artificial Intelligence*. PMLR, pp. 648–658 (cit. on p. 8).

Duff, Michael O'Gordon (2002). "Optimal Learning Computational Procedures for Bayes-adaptive Markov Decision Processes". PhD thesis. University of Massachusetts at Amherst (cit. on p. 9).

Engel, Yaakov, Shie Mannor and Ron Meir (2003). "Bayes meets Bellman: The Gaussian process approach to temporal difference learning". In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 154–161 (cit. on p. 9).

Eriksson, Hannes, Debabrota Basu, Mina Alibeigi and Christos Dimitrakakis (2022). "SENTINEL: taming uncertainty with ensemble based distributional reinforcement learning". In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Ed. by James Cussens and Kun Zhang. Vol. 180. Proceedings of Machine Learning Research. PMLR, pp. 631–640. URL: https://proceedings.mlr.press/v180/eriksson22a.html (cit. on p. 8).

Eriksson, Hannes and Christos Dimitrakakis (2019). "Epistemic risk-sensitive reinforcement learning". In: *arXiv preprint arXiv:1906.06273* (cit. on p. 4).

Fan, Ying and Yifei Ming (2021). "Model-based Reinforcement Learning for Continuous Control with Posterior Sampling". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 3078–3087. URL: http://proceedings.mlr.press/v139/fan21b.html (cit. on p. 8).

Fellows, Mattie, Kristian Hartikainen and Shimon Whiteson (2021). "Bayesian Bellman Operators". In: *Advances in Neural Information Processing Systems* 34, pp. 13641–13656 (cit. on p. 9).

Gal, Yarin, Rowan McAllister and Carl Edward Rasmussen (2016). "Improving PILCO with Bayesian neural network dynamics models". In: *Data-Efficient Machine Learning workshop, ICML*. Vol. 4. 34, p. 25 (cit. on p. 9).

Ghavamzadeh, M. and Y. Engel (2006). "Bayesian Policy Gradient Algorithms". In: *NIPS 2006* (cit. on p. 8).

Ghavamzadeh, Mohammad, Shie Mannor, Joelle Pineau and Aviv Tamar (2015). "Bayesian Reinforcement Learning: A Survey". In: *Foundations and Trends in Machine Learning* 8.5-6, pp. 359–483 (cit. on p. 7).

Gittins, John, Kevin Glazebrook and Richard Weber (2011). *Multi-armed bandit allocation indices*. John Wiley & Sons (cit. on p. 9).

Grünwald, Peter D. and A. Philip Dawid (2004). "Game theory, Maximum Entropy, Minimum Discrepancy and Robust Bayesian decision Theory". In: *Annals of Statistics* (cit. on p. 16).

Guez, Arthur, David Silver and Peter Dayan (2012). "Efficient Bayes-adaptive reinforcement learning using sample-based search". In: *Advances in neural information processing systems* 25 (cit. on p. 9).

Hao, Botao and Tor Lattimore (2022). "Regret Bounds for Information-Directed Reinforcement Learning". In: *arXiv preprint arXiv:2206.04640* (cit. on p. 8).

Hasselt, Hado van, Arthur Guez and David Silver (2015). *Deep Reinforcement Learning with Double Q-learning*. arXiv: `1509.06461 [cs.LG]` (cit. on p. 7).

Jorge, Emilio, Lucas Brynte, Constantin Cronrath, Oskar Wigström, Kristofer Bengtsson, Emil Gustavsson, Bengt Lennartson and Mats Jirstrand (2018). "Reinforcement learning in real-time geometry assurance". In: *Procedia CIRP* 72, pp. 1073–1078 (cit. on p. 4).

Jorge, Emilio, Hannes Eriksson, Christos Dimitrakakis, Debabrota Basu and Divya Grover (Dec. 2020). "Inferential Induction: A Novel Framework for Bayesian Reinforcement Learning". In: *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*. Vol. 137. Proceedings of Machine Learning Research. PMLR, pp. 43–52. URL: `http://proceedings.mlr.press/v137/jorge20a.html` (cit. on pp. 4, 8).

— (2022). "On Bayesian Value Function Distributions." In: *15th European Workshop on Reinforcement Learning* (cit. on p. 4).

Jorge, Emilio, Mikael Kågebäck, Fredrik D Johansson and Emil Gustavsson (2016). "Learning to play guess who? and inventing a grounded language as a consequence". In: *Deep Reinforcement Learning NIPS 2016 Workshop* (cit. on p. 4).

Lin, Tianyi, Chi Jin and Michael Jordan (2020). "On gradient descent ascent for nonconvex-concave minimax problems". In: *International Conference on Machine Learning*. PMLR, pp. 6083–6093 (cit. on p. 3).

Mazumdar, Eric, Aldo Pacchiano, Yian Ma, Michael Jordan and Peter Bartlett (2020). "On Approximate Thompson Sampling with Langevin Algorithms". In: *International Conference on Machine Learning*. PMLR, pp. 6797–6807 (cit. on p. 19).

Mnih, Volodymyr et al. (2015). "Human-level control through deep reinforcement learning". In: *nature* 518.7540, pp. 529–533 (cit. on pp. 3, 7).

Moerland, Thomas M., Joost Broekens and Catholijn M. Jonker (2021). *Model-based Reinforcement Learning: A Survey*. arXiv: `2006.16712 [cs.LG]` (cit. on p. 3).

O'Donoghue, Brendan (2021). "Variational bayesian reinforcement learning with regret bounds". In: *Advances in Neural Information Processing Systems* 34, pp. 28208–28221 (cit. on p. 9).

O'Donoghue, Brendan, Ian Osband, Remi Munos and Volodymyr Mnih (2018). "The uncertainty bellman equation and exploration". In: *International Conference on Machine Learning*, pp. 3836–3845 (cit. on p. 9).

Osband, I., D. Russo and B. Van Roy (2013). "(More) Efficient Reinforcement Learning via Posterior Sampling". In: *NIPS* (cit. on pp. 8, 15).

Osband, Ian, John Aslanides and Albin Cassirer (2018). "Randomized prior functions for deep reinforcement learning". In: *Advances in Neural Information Processing Systems* 31 (cit. on p. 9).

Osband, Ian, Charles Blundell, Alexander Pritzel and Benjamin Van Roy (2016). "Deep exploration via bootstrapped DQN". In: *Advances in neural information processing systems* 29 (cit. on pp. 3, 9).

Osband, Ian, Daniel Russo and Benjamin Van Roy (2013). "(More) efficient reinforcement learning via posterior sampling". In: *Advances in Neural Information Processing Systems* 26 (cit. on p. 3).

Osband, Ian, Zheng Wen, Mohammad Asghari, Morteza Ibrahimi, Xiyuan Lu and Benjamin Van Roy (2021). "Epistemic Neural Networks". In: *arXiv preprint arXiv:2107.08924* (cit. on p. 19).

Ouyang, Yi, Mukul Gagrani, Ashutosh Nayyar and Rahul Jain (2017). *Learning Unknown Markov Decision Processes: A Thompson Sampling Approach*. arXiv: `1709.04570 [cs.LG]` (cit. on p. 8).

Rahbar, Arman, Emilio Jorge, Devdatt Dubhashi and Morteza Haghir Chehreghani (2022). "Do Kernel and Neural Embeddings Help in Training and Generalization?" In: *Neural Processing Letters*, pp. 1–15 (cit. on p. 4).

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford and Oleg Klimov (2017). "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (cit. on p. 3).

Sion, Maurice (1958). "On general minimax theorems." In: *Pacific Journal of mathematics* 8.1, pp. 171–176 (cit. on p. 11).

Strens, Malcolm (2000). "A Bayesian framework for reinforcement learning". In: *ICML 2000*, pp. 943–950 (cit. on p. 8).

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press (cit. on p. 6).

Tang, Yunhao and Shipra Agrawal (2018). "Exploration by distributional reinforcement learning". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, pp. 2710–2716 (cit. on p. 9).

Thompson, W.R. (1933). "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of two Samples". In: *Biometrika* 25.3-4, pp. 285–294 (cit. on p. 8).

Tziortziotis, Nikolaos, Christos Dimitrakakis and Konstantinos Blekas (2014). "Cover tree bayesian reinforcement learning". In: *Journal of Machine Learning Research* 15, pp. 2313–2335 (cit. on p. 8).

Vinyals, Oriol et al. (2019). "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature* 575.7782, pp. 350–354 (cit. on p. 3).

Wang, Zicheng, Hua Meng, Zhengchun Zhou, Yanghe Feng, Yang Gao and Chao Yu (2022). "Towards Uncertainty in Decision: A Survey on Recent

Advances and Challenges in Bayesian Reinforcement Learning". In: (cit. on p. 7).

Watkins, Christopher JCH and Peter Dayan (1992). "Q-learning". In: *Machine learning* 8.3, pp. 279–292 (cit. on p. 7).

Welling, Max and Yee W Teh (2011). "Bayesian learning via stochastic gradient Langevin dynamics". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688 (cit. on p. 19).

Xu, Pan, Hongkai Zheng, Eric V Mazumdar, Kamyar Azizzadenesheli and Animashree Anandkumar (2022). "Langevin monte carlo for contextual bandits". In: *International Conference on Machine Learning*. PMLR, pp. 24830–24850 (cit. on p. 19).

Zhou, Qi, Houqiang Li and Jie Wang (2020). "Deep model-based reinforcement learning via estimated uncertainty and conservative policy optimization". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04, pp. 6941–6948 (cit. on p. 9).

Zintgraf, Luisa et al. (2021). "VariBAD: variational Bayes-adaptive deep RL via meta-learning". In: *The Journal of Machine Learning Research* 22.1, pp. 13198–13236 (cit. on p. 9).