Nonparametric General Reinforcement Learning

Jan Leike

A thesis submitted for the degree of
Doctor of Philosophy
at the
Australian National University

November 2016

No reinforcement learners were harmed in the making of this thesis.

Except where otherwise indicated, this thesis is my own original work.

Jan Leike
28 November 2016

Acknowledgements

There are many without whom this thesis would not have been possible. I sincerely hope that this page is not the way they learn how grateful I am to them. I thank in particular ...

... first and foremost, Marcus Hutter: he is an amazing supervisor; always very supportive of my (unusual) endeavors, spent countless hours reading my drafts with a impressive attention to detail. I am also grateful to him for forcing me to be absolutely rigorous in my mathematical arguments, and, of course, for developing the theory of universal AI without which this thesis would not have existed. I could not have picked a better supervisor.

... the Australian National University for granting me scholarships that let me pursue my academic interests unrestricted and without any financial worries.

... Csaba Szepesvári and the University of Alberta for hosting me for three months.

... Matthias Heizmann and the University of Freiburg for hosting me while I was traveling in Europe.

... the Machine Intelligence Research Institute for enabling me to run MIRIx research workshops.

... CCR, UAI, Google DeepMind, ARC, MIRI, and FHI for supporting my travel.

... Tor Lattimore for numerous explanations, discussions, and pointers that left me with a much deeper understanding of the theory of reinforcement learning.

... Laurent Orseau for interesting discussions, encouragement, and for sharing so many intriguing ideas.

... my fellow students: Mayank Daswani, Tom Everitt, Daniel Filan, Roshan Shariff, Tian Kruger, Emily Cutts Worthington, Buck Shlegeris, Jarryd Martin, John Aslanides, Alexander Mascolo, and Sultan Javed for so many interesting discussions and for being awesome friends. I especially thank Daniel, Emily, Mayank, and Buck for encouraging me to read more of Less Wrong and Slate Star Codex.

... Tosca Lechner for studying statistics with me despite so many scheduling difficulties across all these time zones.

... Tom Sterkenburg, Christian Kamm, Alexandra Surdina, Freya Fleckenstein, Peter Sunehag, Tosca Lechner, Ines Nikolaus, Laurent Orseau, John Aslanides, and especially Daniel Filan for proofreading parts of this thesis.

... the CSSA for being a lovely bunch that made my stay in Australia feel less isolated.

... my family for lots of love and support, and for tolerating my long absences from Europe.

Abstract

Reinforcement learning problems are often phrased in terms of Markov decision processes (MDPs). In this thesis we go beyond MDPs and consider reinforcement learning in environments that are non-Markovian, non-ergodic and only partially observable. Our focus is not on practical algorithms, but rather on the fundamental underlying problems: How do we balance exploration and exploitation? How do we explore optimally? When is an agent optimal? We follow the nonparametric realizable paradigm: we assume the data is drawn from an unknown source that belongs to a known countable class of candidates.

First, we consider the passive (sequence prediction) setting, learning from data that is not independent and identically distributed. We collect results from artificial intelligence, algorithmic information theory, and game theory and put them in a reinforcement learning context: they demonstrate how an agent can learn the value of its own policy.

Next, we establish negative results on Bayesian reinforcement learning agents, in particular AIXI. We show that unlucky or adversarial choices of the prior cause the agent to misbehave drastically. Therefore Legg-Hutter intelligence and balanced Pareto optimality, which depend crucially on the choice of the prior, are entirely subjective. Moreover, in the class of all computable environments every policy is Pareto optimal. This undermines all existing optimality properties for AIXI.

However, there are Bayesian approaches to general reinforcement learning that satisfy objective optimality guarantees: We prove that Thompson sampling is asymptotically optimal in stochastic environments in the sense that its value converges to the value of the optimal policy. We connect asymptotic optimality to regret given a recoverability assumption on the environment that allows the agent to recover from mistakes. Hence Thompson sampling achieves sublinear regret in these environments.

AIXI is known to be incomputable. We quantify this using the arithmetical hierarchy, and establish upper and corresponding lower bounds for incomputability. Further, we show that AIXI is not limit computable, thus cannot be approximated using finite computation. However there are limit computable ε-optimal approximations to AIXI. We also derive computability bounds for knowledge-seeking agents, and give a limit computable weakly asymptotically optimal reinforcement learning agent.

Finally, our results culminate in a formal solution to the grain of truth problem: A Bayesian agent acting in a multi-agent environment learns to predict the other agents' policies if its prior assigns positive probability to them (the prior contains a grain of truth). We construct a large but limit computable class containing a grain of truth and show that agents based on Thompson sampling over this class converge to play ε-Nash equilibria in arbitrary unknown computable multi-agent environments.

Keywords. Bayesian methods, sequence prediction, merging, general reinforcement learning, universal artificial intelligence, AIXI, Thompson sampling, knowledge-seeking agents, Pareto optimality, intelligence, asymptotic optimality, computability, reflective oracle, grain of truth problem, Nash equilibrium.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Everything I did was for the glamor, the money, and the sex. — Albert Einstein

After the early enthusiastic decades, research in artificial intelligence (AI) now mainly aims at specific domains: playing games, mining data, processing natural language, recognizing objects in images, piloting robots, filtering email, and many others (Russell and Norvig, 2010). Progress on particular domains has been remarkable, with several high-profile breakthroughs: The chess world champion Garry Kasparov was defeated by the computer program Deep Blue in 1997 (IBM, 2012a). In 2011 the world's best Jeopardy! players were defeated by the computer program Watson (IBM, 2012b). As of 2014 Google's self-driving cars completed over a million kilometers autonomously on public roads (Google, 2014). Finally, in 2016 Google DeepMind's AlphaGo beat Lee Sedol, one of the world's best players, at the board game Go (Google, 2016).

While these advancements are very impressive, they are highly-specialized algorithms tailored to their domain of expertise. Outside that domain these algorithms perform very poorly: AlphaGo cannot play chess, Watson cannot drive a car, and DeepBlue cannot answer natural language queries. Solutions in one domain typically do not generalize to other domains and no single algorithm performs well in more than one of them. We classify these kinds of algorithms as *narrow AI*.

This thesis is not about narrow AI. We expect progress on narrow AI to continue and even accelerate, taking the crown of human superiority in domain after domain. But this is not the ultimate goal of artificial intelligence research. The ultimate goal is to engineer a mind—to build a machine that can learn to do all tasks that humans can do, at least as well as humans do them. We call such a machine *human-level AI* (HLAI) if it performs at human level and *strong AI* if it surpasses human level. This thesis is about strong AI.

The goal of developing HLAI has a long tradition in AI research and was explicitly part of the 1956 Dartmouth conference that gave birth to the field of AI (McCarthy et al., 1955):

> We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An

> attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

In hindsight this proposal reads vastly overconfident, and disappointment was inevitable. Making progress on these problems turned out to be a lot harder than promised, and over the last decades any discussion of research targeting HLAI has been avoided by serious researchers in the field. This void was filled mostly by crackpots, which tainted the reputation of HLAI research even further. However, this trend has recently been reverted: Chalmers (2010), Hutter (2012a), Schmidhuber (2012), Bostrom (2014), Hawking, Tegmark, Russell, and Wilczek (2014), Shanahan (2015), and Walsh (2016) are well-known scientists discussing the prospect of HLAI seriously. Even more: the explicit motto of Google DeepMind, one of today's leading AI research centers, is to "solve intelligence."

1.1 Reinforcement Learning

The best formal model for strong AI we currently have is reinforcement learning (RL). Reinforcement learning studies algorithms that learn to act in an unknown environment through trial and error (Sutton and Barto, 1998; Szepesvári, 2010; Wiering and van Otterlo, 2012). Without knowing the structure of the environments or the goal, an agent has to learn what to do through the carrot-and-stick approach: it receives a reward in form of a numeric feedback signifying how well it is currently doing; from this signal the agent has to figure out autonomously what to do. More specifically, in a *general reinforcement learning problem* an *agent* interacts sequentially with an unknown *environment*: in every time step the agent chooses an *action* and receives a *percept* consisting of an *observation* and a real-valued *reward*. The sequence of past actions and percepts is the *history*. The goal in reinforcement learning is to maximize cumulative (discounted) rewards (this setup is described formally in Section 4.1).

A central problem in reinforcement learning is the balance between *exploration* and *exploitation*: should the agent harvest rewards in the regions of the environment that it currently knows (exploitation) or try discovering more profitable regions (exploration)? Exploration is costly and dangerous: it forfeits rewards that could be had right now, and it might lead into traps from which the agent cannot recover. However, exploration may pay off in the long run. Generally, it is not clear how to make this tradeoff (see Section 5.6).

Reinforcement learning algorithms can be categorized by whether they learn *on-policy* or *off-policy*. Learning on-policy means learning the value of the policy that the agent currently follows. Typically, the policy is slowly improved while learning, like *SARSA* (Sutton and Barto, 1998). In contrast, learning off-policy means following one policy but learning the value of another policy (typically the optimal policy), like *Q-learning* (Watkins and Dayan, 1992). Off-policy methods are more difficult to handle

in practice (see the discussion on function approximation below) but tend to be more data-efficient since samples from an old policy do not have to be discarded.

Reinforcement learning has to be distinguished from *planning*. In a planning problem we are provided with the true environment and are tasked with finding an optimal policy. Mathematically it is clear what the optimal policy is, the difficulty stems from finding a reasonable solution with limited computation. Reinforcement learning is fundamentally more difficult because the true environment is unknown and has to be learned from observation. This enables two approaches: we could learn a model of the true environment and then use planning techniques within that model; this is the *model-based* approach. Alternatively, we could learn an optimal policy directly or through an intermediate quantity (typically the value function); this is the *model-free* approach. Model-based methods tend to be more data-efficient but also computationally more expensive. Therefore most algorithms used in practice (Q-learning and SARSA) are model-free.

1.1.1 Narrow Reinforcement Learning

In the reinforcement learning literature it is typically assumed that the environment is a *Markov decision process* (MDP), i.e., the next percept only depends on the last percept and action and is independent of the rest of the history (see Section 4.1.3). In an MDP, percepts are usually called *states*. This setting is well-analyzed (Puterman, 2014; Bertsekas and Tsitsiklis, 1995; Sutton and Barto, 1998), and there is a variety of algorithms that are known to learn the MDP asymptotically, such as *TD learning* (Sutton, 1988) and Q-learning (Watkins and Dayan, 1992).

Moreover, for MDPs various learning guarantees have been proved in the literature. First, there are bounds on the agent's *regret*, the difference between the obtained rewards and the rewards of the optimal policy. Auer et al. (2009) derive the regret bound $\tilde{O}(dS\sqrt{At})$ for ergodic MDPs where d is the *diameter* of the MDP (how many steps a policy needs on average to get from one state of the MDP to any other), S is the number of states, A is the number of actions, and t is the number of time steps the algorithm runs. Second, given ε and δ, a reinforcement learning algorithm is said to have *sample complexity* $C(\varepsilon, \delta)$ iff it is ε-suboptimal for at most $C(\varepsilon, \delta)$ time steps with probability at least $1 - \delta$ (probably approximately correct, PAC). For MDPs the first sample complexity bounds were due to Kakade (2003). Lattimore and Hutter (2012) use the algorithm UCRLγ (Auer et al., 2009) with geometric discounting with discount rate γ and derive the currently best-known PAC bound of $\tilde{O}(-T/(\varepsilon^2(1-\gamma)^3)\log\delta)$ where T is the number of non-zero transitions in the MDP.

Typically, algorithms for MDPs rely on visiting every state multiple times (or even infinitely often), which becomes infeasible for large state spaces (e.g. a video game screen consisting of millions of pixels). In these cases, *function approximation* can be used to learn an approximation to the value function (Sutton and Barto, 1998). Linear function approximation is known to converge for several on-policy algorithms (Tsitsiklis and Roy, 1997; Sutton, 1988; Gordon, 2001), but proved tricky for off-policy algorithms (Baird, 1995). A recent breakthrough was made by Mahmood et al. (2015) and Yu (2015)

with their emphatic TD algorithm that converges off-policy. For nonlinear function approximation no convergence guarantee is known.

Among the historical successes of reinforcement learning is autonomous helicopter piloting (Kim et al., 2003) and TD-Gammon, a backgammon algorithm that learned through self-play (Tesauro, 1995), similar to AlphaGo (Silver et al., 2016).

1.1.2 Deep Q-Networks

The current state of the art in reinforcement learning challenges itself to playing simple video games. Video games are an excellent benchmark because they come readily with the reward structure provided: the agent's rewards are the change in the game score. Without prior knowledge of any aspect of the game, the agent needs to learn to score as many points in the game as possible from looking only at raw pixel data (sometimes after some preprocessing).

This approach to general AI is in accordance with the definition of intelligence given by Legg and Hutter (2007b):

> Intelligence measures an agent's ability to achieve goals in a wide range of environments.

In reinforcement learning the definition of the goal is very flexible, and provided by the rewards. Moreover, a diverse selection of video games arguably constitutes a 'wide range of environments.'

A popular such selection is the Atari 2600 video game console (Bellemare et al., 2013). There are hundreds of games released for this platform, with very diverse challenges: top-down shooting games such as Space Invaders, ball games such as Pong, agility-based games such as Boxing or Gopher, tactical games such as Ms. Pac-Man, and maze games such as Montezuma's Revenge. An overview over some of the games is given in Figure 1.1 on page 13.

Mnih et al. (2013, 2015) introduce the deep Q-network (DQN) algorithm, combining Q-learning with nonlinear function approximation through convolutional neural networks. DQN achieves 75% of the performance of a human game tester on 29 of 49 Atari games. The two innovations that made this breakthrough possible are (1) using a not so recent target Q-function in the TD update and (2) experience replay. For experience replay, a set of recent state transitions is retained and the network is regularly retrained on random samples from these old transitions.[1]

DQN rides the wave of success of *deep learning* (LeCun et al., 2015; Schmidhuber, 2015; Goodfellow et al., 2016). Deep learning refers to the training of artificial neural networks with several layers. This allows them to automatically learn higher-level abstractions from data. Deep neural networks are conceptionally simple and have been studied since the inception of AI; only recently has computation power become cheap enough to train them effectively. Recently deep neural networks have taken the top of the machine learning benchmarks by storm (LeCun et al., 2015, and references therein):

[1] The slogan for experience replay should be 'regularly retrained randomly on retained rewards'.

These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics.

Since the introduction of DQN there have been numerous improvements on this algorithm: increasing the gap on the Q-values of different actions (Bellemare et al., 2016), training in parallel (Nair et al., 2015; Mnih et al., 2016), improvements to the experience replay mechanism (Schaul et al., 2016), generalization to continuous action spaces (Lillicrap et al., 2016), solve the overestimation problem (van Hasselt et al., 2016), and improvements to the neural network architecture (Wang et al., 2016). The Q-values learned by DQN's neural networks are intransparent to inspection; Zahavy et al. (2016) use visualization techniques on the Q-value networks. Finally, Liang et al. (2016) managed to reproduce DQN's success using only *linear* function approximation (no neural networks). The key is a selection of features similar to the ones produced by DQN's convolutional neural networks.

Regardless of its success, the DQN algorithm fundamentally falls short of the requirements for strong AI: Q-learning with function approximation is targeted at solving large-state (fully observable) Markov decision processes. In particular, it does not address the following challenges.

- *Partial observability.* All games in the ATARI framework are fully observable (except for Montezuma's revenge): all information relevant to the state of the game is visible on the screen at all times (when using the four most recent frames).

  However, the real world is only partially observable. For example, when going to the supermarket you have to remember what you wanted to buy because you currently cannot observe which items you are missing at home. A strong AI needs to have memory and be able to remember things that happened in the past (rather than only learning from it).

  An obvious approach to equip DQN with memory is to use recurrent neural networks instead of simple feedforward neural networks (Heess et al., 2015). Hausknecht and Stone (2015) show that this enables the agent to play the games when using only a single frame as input. However, it is currently unclear whether recurrent neural networks are powerful enough to learn long-term dependencies in the data (Bengio et al., 1994).

- *Directed exploration.* DQN fails in games with delayed rewards. For example, in Montezuma's Revenge the agent needs to avoid several obstacles to get to a key before receiving the first reward. DQN fails to score any rewards in this environment. This is not surprising: the typical approach for reinforcement learning, to use ε-exploration for which the agent chooses actions at random with a certain probability, is insufficient for exploring complex environments; the probability of random walking into the first reward is just too low.

  Instead we need a more targeted exploration approach that aims at understanding the environment in a structured manner. Theoretical foundations are provided

by *knowledge-seeking agents* (Orseau, 2011, 2014a; Orseau et al., 2013). Kulkarni et al. (2016) introduce a hierarchical approach based on intrinsic motivation to improve DQN's exploration and manage to score points in Montezuma's Revenge. However, their approach relies on quite a bit of visual preprocessing and domain knowledge.

- *Non-ergodicity.* When losing in an Atari game, the agent always gets to play the same game again. From the agent's perspective, it has not actually failed, it just gets transported back to the starting state. Because of this, there are no strong incentives to be careful when exploring the environment: there can be no bad mistakes that make recovery impossible.

  However, in the real world some actions are irreversibly bad. If the robot drives off a cliff it can be fatally damaged and cannot learn from the mistake. The real world is full of potentially fatal mistakes (e.g. crossing the street at the wrong time) and for humans, natural reflexes and training by society make sure that we are very confident of what situations to avert. This is crucial, as some mistakes must be avoided without any training examples. Current reinforcement learning algorithms only learn about bad states by visiting them.

- *Wireheading.* The goal of reinforcement learning is to maximize rewards. When playing a video game the most efficient way to get rewards is to increase the game score. However, when a reinforcement learning algorithm is acting in the real world, theoretically it can change its own hard- and software. In this setting, the most efficient way to get rewards is to modify the reward mechanism to always provide the maximal reward (Omohundro, 2008; Ring and Orseau, 2011; Bostrom, 2014). Consequently the agent no longer pursues the designers' originally intended goals and instead only attempts to protect its own existence. The name *wireheading* was established by analogy to a biology experiment by Olds and Milner (1954) in which rats had a wire embedded into the reward center of their brain that they could then stimulate by the push of a button.

  Today's reinforcement learning algorithms usually do not have access to their own internal workings, but more importantly they are not smart enough to understand their own architecture. They simply lack the capability to wirehead. But as we increase their capability, wireheading will increasingly become a challenge for reinforcement learning.

1.1.3 General Reinforcement Learning

A theory of strong AI cannot make some of the typical assumptions. Environments are partially observable, so we are dealing with *partially observable Markov decision processes* (POMDPs). The POMDP's state space does not need to be finite. Moreover, the environment may not allow recovery from mistakes: we do not assume ergodicity or weak communication (not every POMDP state has to be reachable from every other state). So in general, our environments are infinite-state non-ergodic POMDPs. Table 1.1 lists the assumptions that are typical but we do not make.

Assumption	Description
Full observability	the agent needs no memory to act optimally
Finite state	the environment has only finitely many states
Ergodicity	the agent can recover from any mistakes
Computability	the environment is computable

Table 1.1: List of assumptions from the reinforcement learning literature. In this thesis, we only make the computability assumption which is important for Chapter 6 and Chapter 7.

Learning POMDPs is a lot harder, and only partially successful attempts have been made: through predictive state representations (Singh et al., 2003, 2004), and Bayesian methods (Doshi-Velez, 2012). A general approach is *feature reinforcement learning* (Hutter, 2009c,d), which aims to reduce the general reinforcement learning problem to an MDP by aggregating histories into states. The quest for a good cost function for feature maps remains unsuccessful thus far (Sunehag and Hutter, 2010; Daswani, 2015). However, Hutter (2014) managed to derive strong bounds relating the optimal value function of the aggregated MDP to the value function of the original process even if the latter violates the Markov condition.

A full theoretical approach to the general reinforcement learning problem is given by Hutter (2000, 2001a, 2002a, 2003, 2005, 2007a, 2012b). He introduces the Bayesian RL agent *AIXI* building on the theory of sequence prediction by Solomonoff (1964, 1978). Based in algorithmic information theory, Solomonoff's prior draws from famous insights by William of Ockham, Sextus Epicurus, Alan Turing, and Andrey Kolmogorov (Rathmanner and Hutter, 2011). AIXI uses Solomonoff's prior over the class of all computable environments and acts to maximize Bayes-expected rewards. We formally introduce Solomonoff's theory of induction in Chapter 3 and AIXI in Section 4.3.1. See also Legg (2008) for an accessible introduction to AIXI.

A typical optimality property in general reinforcement learning is *asymptotic optimality* (Lattimore and Hutter, 2011): as time progresses the agent converges to achieve the same rewards as the optimal policy. Asymptotic optimality is usually what is meant by "Q-learning converges" (Watkins and Dayan, 1992) or "TD learning converges" (Sutton, 1988). Orseau (2010, 2013) showed that AIXI is not asymptotically optimal. Yet asymptotic optimality in the general setting can be achieved through optimism (Sunehag and Hutter, 2012a,b, 2015), Thompson sampling (Section 5.4.3), or an extra exploration component on top of AIXI (Lattimore, 2013, Ch. 5).

In our setting, learning the environment does not just involve learning a fixed finite set of parameters; the real world is too complicated to fit into a template. Therefore we fall back on the *nonparametric* approach where we start with an infinite but countable class of candidate environments. Our only assumption is that the true environment is contained in this class (the *realizable case*). As long as this class of environments is large enough (such as for the class of all computable environments), this assumption is

rather weak.

1.2 Contribution

The goal of this thesis is not to increase AI capability. As such, we are not trying to improve on the state of the art, and we are not trying to derive practical algorithms. Instead, the emphasis of this thesis is to further our *understanding* of general reinforcement learning and thus strong AI. How a future implementation of strong AI will actually work is in the realm of speculation at this time. Therefore we should make as few and as weak assumptions as possible.

We disregard computational constraints in order to focus on the fundamental underlying problems. This is unrealistic, of course. With unlimited computation power many traditional AI problems become trivial: playing chess, Go, or backgammon can be solved by exhaustive expansion of the game tree. But the general RL problem does not become trivial: the agent has to learn the environment and balance between exploration and exploitation. That being said, the algorithms that we study do have a relationship with algorithms being used in practice and our results can and should educate implementation.

On a high level, our insights can be viewed from three different perspectives.

- *Philosophically.* Concisely, our understanding of strong AI can be summarized as follows.

  $$\text{intelligence} \;=\; \text{learning} \;+\; \text{acting} \tag{1.1}$$

  Here, *intelligence* refers to an agent that optimizes towards some goal in accordance with the definition by Legg and Hutter (2007b). For *learning* we distinguish two (very related) aspects: (1) arriving at accurate beliefs about the future and (2) making accurate predictions about the future. Of course, the former implies the latter: if you have accurate beliefs, then you can also make good predictions. For RL accurate beliefs is what we care about because they enable us to plan for the future. Learning is a passive process that only observes the data and does not interfere with its generation. In particular, learning does not require a goal. With *acting* we mean the selection of actions in pursuit of some goal. This goal can be reward maximization as in reinforcement learning, understanding the environment as for knowledge-seeking agents, or something else entirely. Together they enable an agent to learn the environment's behavior in response to itself (on-policy learning) and to choose a policy that furthers its goal. We discuss the formal aspects of learning in Chapter 3 and some approaches to acting in Chapter 4.

  Given infinite computational resources, learning is easy and Solomonoff induction provides a complete theoretical solution. However, acting is not straightforward. We show that in contrast to popular belief, AIXI, the natural extension of Solomonoff induction to reinforcement learning, does not provide the objectively best answer to this question. We discuss some alternatives and their problems in

Chapter 5. Unfortunately, the general question of how to act optimally remains open.

AIXI*tl* (Hutter, 2005, Ch. 7.2) is often mentioned as a computable approximation to AIXI. But AIXI*tl* does not converge to AIXI in the limit. Inspired by Hutter search (Hutter, 2002b), it relies on an automated theorem prover to find the provably best policy computable in time t with a program of length $\leq l$. In contrast to AIXI, which only requires the choice of universal Turing machine, proof search requires an axiom system that must not be too weak or too strong. In Section 5.2.3 we discuss some of the problems with AIXI*tl*. Moreover, in Corollary 6.13 we show that ε-optimal AIXI is limit computable, which shows that AIXI can be computably approximated by running this algorithm for a fixed number of time steps or until a timeout is reached. While neither AIXI*tl* nor this AIXI approximation algorithm are practically feasible, the latter is a better example for a computable strong AI.

In our view, AIXI should be taken as a descriptive rather than prescriptive model. It is descriptive as an abstraction from an actual implementation of strong AI where we ignore all the details of the learning algorithm and the computational approximations of choosing how to act. It should not be viewed as a prescription of how strong AI should be built and AIXI approximations (Veness et al., 2011, 2015) are easily outperformed by neural-network-based approaches (Mnih et al., 2015).

- *Mathematically.* Some of the proof techniques we employ are novel and could be used to analyze other algorithms. Examples include the proofs for the lower bounds on the computability results (Section 6.3.2) and to a lesser extent the upper bounds (Section 6.3.1), which should work analogously for a wide range of algorithms. Furthermore, the proof of the asymptotic optimality of Thompson sampling (Theorem 5.25) brings together a variety of mathematical tools from measure theory, probability theory, and stochastic processes.

  Next, the *recoverability assumption* (Definition 5.31) is a novel technical assumption on the environment akin to ergodicity and weak communication in finite-state environments. It is more general, yet mathematically simple and works for arbitrary environments. This assumption turns out to be what we need to prove the connection from asymptotic optimality to sublinear regret in Section 5.5.

  Moreover, we introduce the use of the recursive instead of the iterative value function (Section 6.4). The iterative value function is the natural extension of expectimax search to the sequential setting and was originally used by Hutter (2005, Sec. 5.5). Yet it turned out to be an incorrect and inconvenient definition: it does not correctly maximize expected rewards (Proposition 6.19) and it is not limit computable (Theorem 6.22 and Theorem 6.23). However, this is only a minor technical correction.

  Finally, this work raises new mathematically intriguing questions about the properties of reflective oracles (Section 7.1).

- *Practically.* One insight from this thesis is regarding the effective horizon. In practice geometric discounting is ubiquitous which has a constant effective horizon. However, when facing a finite horizon problem or an episodic task, sometimes the effective horizon changes. One lesson from our result on Thompson sampling (Section 5.4.3 and Section 5.5) is that you should explore for an effective horizon instead of using ε-greedy. While the latter exploration method is often used in practice, it has proved ineffective in environments with delayed rewards (see Section 1.1.2).

  Furthermore, our application of reinforcement learning results to game theory in Chapter 7 reinforces this trend to solve game theory problems (Tesauro, 1995; Bowling and Veloso, 2001; Busoniu et al., 2008; Silver et al., 2016; Heinrich and Silver, 2016; Foerster et al., 2016, and many more). In particular, the approximation algorithm for reflective oracles (Section 7.1.3) could guide future applications for computing Nash equilibria (see also Fallenstein et al., 2015b).

On a technical level, we advance the theory of general reinforcement learning. In its center is the Bayesian reinforcement learning agent AIXI. AIXI is meant as an answer to the question of how to do general RL disregarding computational constraints. We analyze the computational complexity of AIXI and related agents in Chapter 6 and show that even with an infinite horizon AIXI can be computationally approximated with a regular Turing machine (Section 6.3.1). We also derive corresponding lower bounds for most of our upper bounds (Section 6.3.2).

Chapter 5 is about notions of optimality in general reinforcement learning. We dispel AIXI's status as the gold standard for reinforcement learning. Hutter (2002a) showed that AIXI is Pareto optimal, balanced Pareto optimal, and self-optimizing. Orseau (2013) established that AIXI does not achieve asymptotic optimality in all computable environments (making the self-optimizing result inapplicable to this general environment class). In Section 5.1 we show that every policy is Pareto optimal and in Section 5.3 we show that balanced Pareto optimality is highly subjective, depending on the choice of the prior; bad choices for priors are discussed in Section 5.2. Notable is the *dogmatic prior* that locks a Bayesian reinforcement learning agent into a particular (bad) policy as long as this policy yields some rewards. Our results imply that there are no known nontrivial and non-subjective optimality results for AIXI. We have to regard AIXI as a *relative* theory of intelligence. More generally, our results imply that general reinforcement learning is difficult *even when disregarding computational costs.*

But this is not the end to Bayesian methods in general RL. We show in Section 5.4 that a Bayes-inspired algorithm called *Thompson sampling* achieves asymptotic optimality. Thompson sampling, also known as *posterior sampling* or the *Bayesian control rule* repeatedly draws one environment from the posterior distribution and then acts as if this was the true environment for a certain period of time (depending on the discount function). Moreover, given a recoverability assumption on the environment and some mild assumptions on the discount function, we show in Section 5.5 that Thompson sampling achieves sublinear regret.

Finally, we tie these results together to solve an open problem in game theory:

Chapter	Publication(s)
Chapter 1	-
Chapter 2	-
Chapter 3	with links to Leike and Hutter (2014a, 2015d); Filan et al. (2016)
Chapter 4	-
Chapter 5	Leike and Hutter (2015c); Leike et al. (2016a)
Chapter 6	Leike and Hutter (2015b,a, 2016)
Chapter 7	Leike et al. (2016b)
Chapter 8	-
Appendix A	Leike and Hutter (2014b)

Table 1.2: List of publications by chapter.

When acting in a multi-agent environment with other Bayesian agents, each agent needs to assign positive prior probability to the other agents' actual policies (they need to have a *grain of truth*). Finding a reasonably large class of policies that contains the Bayes optimal policies with respect to this class is known as the *grain of truth problem* (Hutter, 2009b, Q. 5j). Only small classes are known to have a grain of truth and the literature contains several related impossibility results (Nachbar, 1997, 2005; Foster and Young, 2001). Moreover, while AIXI assumes the environment to be computable, our computability results on AIXI confirm that it is incomputable (Theorem 6.15 and Theorem 6.17). This asymmetry elevates AIXI above its environment computationally, and prevents the environment from containing other AIXIs.

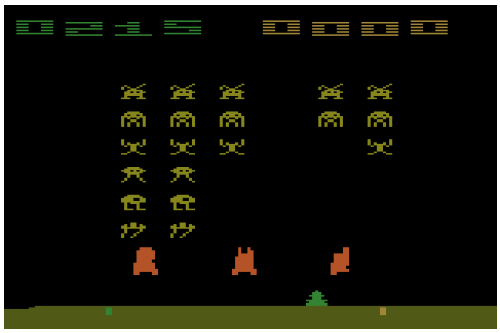
In Chapter 7 we give a formal and general solution to the grain of truth problem: we construct a class of policies that avoid this asymmetry. This class contains all computable policies as well as Bayes optimal policies for every lower semicomputable prior over the class. When the environment is unknown, our dogmatic prior from Section 5.2 makes Bayes optimal agents fail to act optimally even asymptotically. However, our convergence results on Thompson sampling (Section 5.4.3) imply that Thompson samplers converge to play ε-Nash equilibria in arbitrary unknown computable multi-agent environments. While these results are purely theoretical, we use techniques from Chapter 6 to show that they can be computationally approximated arbitrarily closely.

1.3 Thesis Outline

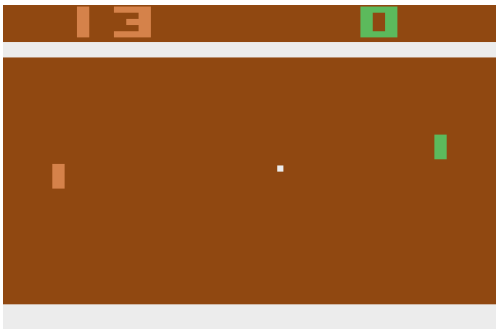
This thesis is based on the papers Leike and Hutter (2014a,b, 2015a,b,c, 2016); Leike et al. (2016a,b). During my PhD, I was also involved in the publications Leike and Heizmann (2014a,b, 2015); Heizmann et al. (2015, 2016) based on my research in termination analysis (in collaboration with Matthias Heizmann), Daswani and Leike (2015) (co-authored with Mayank Daswani in equal parts), Everitt et al. (2015) (co-authored with Tom Everitt in equal parts), Filan et al. (2016) (written by Daniel Filan as part of his honour's thesis supervised by Marcus Hutter and me). Leike and Hutter (2016) is

still under review. Leike and Hutter (2014a, 2015d) are tangential to this thesis' main thrust, so the results are mentioned only in passing. A list of papers written during my PhD is given in Table 1.3 on page 14, with a corresponding chapter outline in Table 1.2. The core of our contribution is found in chapters 5, 6, and 7.

Every thesis chapter starts with a quote. In case this is not blatantly obvious: *these are false quotes*, a desperate attempt to make the thesis less dry and humorless. None of the quotes were actually stated by the person they are attributed to (according to our knowledge).

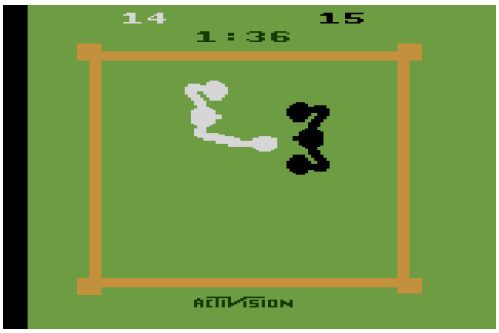


(a) Space Invaders: the player controls the green cannon on the bottom of the screen and fires projectiles at the yellow ships at the top. The red blobs can be used as cover, but also fired through.



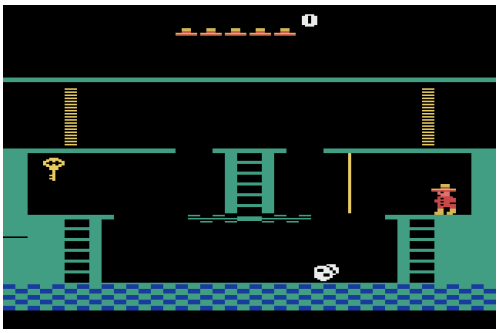
(b) Pong: the player controls the green paddle on the right of the screen and needs to hit the white ball such that the computer opponent controlling the red paddle on the left fails to hit the ball back.

(c) Ms. Pac-Man: the player controls the yellow mouth and needs to eat all the red pallets in the maze. The maze is roamed by ghosts that occasionally hunt the player and kill her on contact unless a 'power pill' was consumed recently.



(d) Boxing: the player controls the white figure on the screen and extends their arms to throw a punch. The aim is to hit the black figure that is controlled by the computer and dodge their punches. (I'm sure the choice of color was by accident.)

(e) Gopher: a hungry rodent attempts to dig to the surface and steal the vegetables. The player controls the farmer who protects them by filling the rodent's holes.



(f) Montezuma's Revenge: the player controls the red adventurer. The aim is to navigate a maze of deadly traps, use keys to open doors, and collect artifacts.

Figure 1.1: A selection of Atari 2600 video games.

[1] Jan Leike and Marcus Hutter. Indefinitely oscillating martingales. In *Algorithmic Learning Theory*, pages 321–335, 2014a

[2] Jan Leike and Matthias Heizmann. Ranking templates for linear loops. *Logical Methods in Computer Science*, 11(1):1–27, March 2015

[3] Mayank Daswani and Jan Leike. A definition of happiness for reinforcement learning agents. In *Artificial General Intelligence*, pages 231–240. Springer, 2015

[4] Tom Everitt, Jan Leike, and Marcus Hutter. Sequential extensions of causal and evidential decision theory. In *Algorithmic Decision Theory*, pages 205–221. Springer, 2015

[5] Jan Leike and Marcus Hutter. On the computability of AIXI. In *Uncertainty in Artificial Intelligence*, pages 464–473, 2015a

[6] Jan Leike and Marcus Hutter. On the computability of Solomonoff induction and knowledge-seeking. In *Algorithmic Learning Theory*, pages 364–378, 2015b

[7] Jan Leike and Marcus Hutter. Bad universal priors and notions of optimality. In *Conference on Learning Theory*, pages 1244–1259, 2015c

[8] Jan Leike and Marcus Hutter. Solomonoff induction violates Nicod's criterion. In *Algorithmic Learning Theory*, pages 349–363. Springer, 2015d

[9] Matthias Heizmann, Daniel Dietsch, Jan Leike, Betim Musa, and Andreas Podelski. Ultimate Automizer with array interpolation (competition contribution). In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 455–457. Springer, 2015

[10] Matthias Heizmann, Daniel Dietsch, Marius Greitschus, Jan Leike, Betim Musa, Claus Schätzle, and Andreas Podelski. Ultimate Automizer with two-track proofs (competition contribution). In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 950–953. Springer, 2016

[11] Daniel Filan, Jan Leike, and Marcus Hutter. Loss bounds and time complexity for speed priors. In *Artificial Intelligence and Statistics*, 2016

[12] Jan Leike and Marcus Hutter. On the computability of Solomonoff induction and AIXI. 2016. Under review

[13] Jan Leike, Tor Lattimore, Laurent Orseau, and Marcus Hutter. Thompson sampling is asymptotically optimal in general environments. In *Uncertainty in Artificial Intelligence*, 2016a

[14] Jan Leike, Jessica Taylor, and Benya Fallenstein. A formal solution to the grain of truth problem. In *Uncertainty in Artificial Intelligence*, 2016b

[15] Jan Leike and Matthias Heizmann. Geometric nontermination arguments. 2016. Under preparation

Table 1.3: List of publications.

Chapter 2

Preliminaries

Mathematics is a waste of time. — Leonhard Euler

This chapter establishes the notation and background material that is used throughout this thesis. Section 2.1 is about probability and measure theory, Section 2.2 is about stochastic processes, Section 2.3 is about information theory, and Section 2.4 is about algorithmic information theory. We defer the formal introduction to reinforcement learning to Chapter 4. Additional preliminary notation and terminology is also established in individual chapters wherever necessary. A list of notation is provided in the appendix on page 171.

Most of the content from this chapter can be found in standard textbooks and reference works. We recommend to consult Wasserman (2004) on statistics, Durrett (2010) on probability theory and stochastic processes, Cover and Thomas (2006) on information theory, Li and Vitányi (2008) on algorithmic information theory, Russell and Norvig (2010) on artificial intelligence, Bishop (2006) and Hastie et al. (2009) on machine learning, Sutton and Barto (1998) on reinforcement learning, and Hutter (2005) and Lattimore (2013) on general reinforcement learning.

We understand definitions to follow natural language; e.g., when defining the adjective 'continuous', we define at the same time the noun 'continuity' and the adverb 'continuously' wherever appropriate.

Numbers. $\mathbb{N} := \{1, 2, 3, \ldots\}$ denotes the set of natural numbers (starting from 1), $\mathbb{Q} := \{\pm p/q \mid p \in \mathbb{N} \cup \{0\}, q \in \mathbb{N}\}$ denotes the set of rational numbers, and $\mathbb{R}$ denotes the set of real numbers. For two real numbers r_1, r_2, the set $[r_1, r_2] := \{r \in \mathbb{R} \mid r_1 \leq r \leq r_2\}$ denotes the closed interval with end points r_1 and r_2; the sets $(r_1, r_2] := [r_1, r_2] \setminus \{r_1\}$ and $[r_1, r_2) := [r_1, r_2] \setminus \{r_2\}$ denote half-open intervals; the set $(r_1, r_2) := [r_1, r_2] \setminus \{r_1, r_2\}$ denotes an open interval.

Strings. Fix $\mathcal{X}$ to be a finite nonempty set, called *alphabet*. We assume that $\mathcal{X}$ contains at least two distinct elements. The set $\mathcal{X}^* := \bigcup_{n=0}^{\infty} \mathcal{X}^n$ is the set of all finite strings over the alphabet $\mathcal{X}$, the set $\mathcal{X}^\infty$ is the set of all infinite strings over the alphabet $\mathcal{X}$, and the set $\mathcal{X}^\sharp := \mathcal{X}^* \cup \mathcal{X}^\infty$ is their union. The empty string is denoted by ϵ, not to be confused with the small positive real number ε. Given a string $x \in \mathcal{X}^\sharp$, we denote its length by $|x|$. For a (finite or infinite) string x of length $\geq k$, we denote with x_k the k-th character of x, with $x_{1:k}$ the first k characters of x, and with $x_{<k}$ the first $k-1$

characters of x. The notation $x_{1:\infty}$ stresses that x is an infinite string. We use $x \sqsubseteq y$ to denote that x is a prefix of y, i.e., $x = y_{1:|x|}$. Our examples often (implicitly) involve the binary alphabet $\{0,1\}$. In this case we define the functions $\mathrm{ones}, \mathrm{zeros} : \mathcal{X}^* \to \mathbb{N}$ that count the number of ones and zeros in a string respectively.

Computability. A function $f : \mathcal{X}^* \to \mathbb{R}$ is *lower semicomputable* iff the set $\{(x,q) \in \mathcal{X}^* \times \mathbb{Q} \mid f(x) > q\}$ is recursively enumerable. If f and $-f$ are lower semicomputable, then f is called *computable*. See Section 6.1.2 for more computability definitions.

Asymptotic Notation. Let $f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$. We use $f \in O(g)$ to denote that there is a constant c such that $f(t) \leq cg(t)$ for all $t \in \mathbb{N}$. We use $f \in o(g)$ to denote that $\limsup_{t\to\infty} f(t)/g(t) = 0$. For functions on strings $P, Q : \mathcal{X}^* \to \mathbb{R}$ we use $Q \overset{\times}{\geq} P$ to denote that there is a constant $c > 0$ such that $Q(x) \geq cP(x)$ for all $x \in \mathcal{X}^*$. We also use $Q \overset{\times}{\leq} P$ for $P \overset{\times}{\geq} Q$ and $Q \overset{\times}{=} P$ for $Q \overset{\times}{\leq} P$ and $P \overset{\times}{\leq} Q$. Note that $Q \overset{\times}{=} P$ *does not imply* that there is a constant c such that $Q(x) = cP(x)$ for all $x \in \mathcal{X}^*$. For a sequence $(a_t)_{t\in\mathbb{N}}$ with limit $\lim_{t\to\infty} a_t = a$ we also write $a_t \to a$ as $t \to \infty$. If no limiting variable is provided, we mean $t \to \infty$ by convention.

Other Conventions. Let A be some set. We use $\#A$ to denote the cardinality of the set A, i.e., the number of elements in A, and 2^A to denote the power set of A, i.e., the set of all subsets of A. We use log to denote the binary logarithm and ln to denote the natural logarithm.

2.1 Measure Theory

For a countable set Ω, we use $\Delta\Omega$ to denote the set of probability distributions over Ω. If Ω is uncountable (such as the set of all infinite strings $\mathcal{X}^\infty$), we need to use the machinery of measure theory. This section provides a concise introduction to measure theory; see Durrett (2010) for an extensive treatment.

Definition 2.1 (σ-algebra)**.** Let Ω be a set. The set $\mathcal{F} \subseteq 2^\Omega$ is a *σ-algebra over* Ω iff

(a) $\Omega \in \mathcal{F}$,

(b) $A \in \mathcal{F}$ implies $\Omega \setminus A \in \mathcal{F}$, and

(c) for any countable number of sets $A_0, A_1, \ldots, \in \mathcal{F}$, the union $\bigcup_{i\in\mathbb{N}} A_i \in \mathcal{F}$.

For a set $\mathcal{A} \subseteq 2^\Omega$, we define $\sigma(\mathcal{A})$ to be the smallest (with respect to set inclusion) σ-algebra containing $\mathcal{A}$.

For the real numbers, the default σ-algebra (used implicitly) is the *Borel σ-algebra* $\mathcal{B}$ generated by the open sets of the usual topology. Formally, $\mathcal{B} := \sigma(\{(a,b) \mid a, b \in \mathbb{R}\})$.

A set Ω together with a σ-algebra $\mathcal{F}$ forms a *measurable space*. The sets from the σ-algebra $\mathcal{F}$ are called *measurable sets*. A function $f : \Omega_1 \to \Omega_2$ between two measurable spaces is called *measurable* iff any preimage of an (in Ω_2) measurable set is measurable (in Ω_1).

Definition 2.2 (Probability Measure)**.** Let Ω be a measurable space with σ-algebra $\mathcal{F}$. A *probability measure* on the space Ω is a function $\mu : \mathcal{F} \to [0,1]$ such that

(a) $\mu(\Omega) = 1$ (*normalization*), and

(b) $\mu(\bigcup_{i\in\mathbb{N}} A_i) = \sum_{i\in\mathbb{N}} \mu(A_i)$ for any collection $\{A_i \mid i \in \mathbb{N}\} \subseteq \mathcal{F}$ that is pairwise disjoint (*σ-additivity*).

A probability measure μ is *deterministic* iff it assigns all probability mass to a single element of Ω, i.e., iff there is an $x \in \Omega$ with $\mu(\{x\}) = 1$.

We define the *conditional probability* $\mu(A \mid B)$ for two measurable sets $A, B \in \mathcal{F}$ with $\mu(B) > 0$ as $\mu(A \mid B) := \mu(A \cap B)/\mu(B)$.

Definition 2.3 (Random Variable)**.** Let Ω be a measurable space with probability measure μ. A *(real-valued) random variable* is a measurable function $X : \Omega \to \mathbb{R}$.

We often (but not always) denote random variables with uppercase Latin letters.

Given a σ-algebra $\mathcal{F}$, a probability measure P on $\mathcal{F}$, and an $\mathcal{F}$-measurable random variable X, the *conditional expectation* $\mathbb{E}[X \mid \mathcal{F}]$ of X given $\mathcal{F}$ is a random variable Y such that (1) Y is $\mathcal{F}$-measurable and (2) $\int_A X dP = \int_A Y dP$ for all $A \in \mathcal{F}$. The conditional expectation exists and is unique up to a set of P-measure 0 (Durrett, 2010, Sec. 5.1). Intuitively, if $\mathcal{F}$ describes the information we have at our disposal, then $\mathbb{E}[X \mid \mathcal{F}]$ denotes the expectation of X given this information.

We proceed to define the σ-algebra on $\mathcal{X}^\infty$ (the σ-algebra on $\mathcal{X}^\sharp$ is defined analogously). For a finite string $x \in \mathcal{X}^*$, the *cylinder set*

$$\Gamma_x := \{xy \mid y \in \mathcal{X}^\infty\}$$

is the set of all infinite strings of which x is a prefix. Furthermore, we fix the σ-algebras

$$\mathcal{F}_t := \sigma\big(\{\Gamma_x \mid x \in \mathcal{X}^t\}\big) \qquad \text{and} \qquad \mathcal{F}_\infty := \sigma\left(\bigcup_{t=1}^{\infty} \mathcal{F}_t\right).$$

The sequence $(\mathcal{F}_t)_{t\in\mathbb{N}}$ is a *filtration*: from $\Gamma_x = \bigcup_{a\in\mathcal{X}} \Gamma_{xa}$ follows that $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$ for every $t \in \mathbb{N}$, and all $\mathcal{F}_t \subseteq \mathcal{F}_\infty$ by the definition of $\mathcal{F}_\infty$.

For our purposes, the σ-algebra $\mathcal{F}_t$ means 'all symbols up to and including time step t.' So instead of conditioning an expectation on $\mathcal{F}_t$, we can just as well condition it on the sequence $x_{1:t}$ drawn at time t. Hence we write $\mathbb{E}[X \mid x_{1:t}]$ instead of $\mathbb{E}[X \mid \mathcal{F}_t]$. Moreover, for conditional probabilities we also write $Q(x_t \mid x_{<t})$ instead of $Q(x_{1:t} \mid x_{<t})$.

In the context of probability measures, a measurable set $E \in \mathcal{F}_\infty$ is also called an *event*. The event $E^c := \mathcal{X}^\infty \setminus E$ denotes the complement of E. In case the event E is defined by a predicate Q dependent on the random variable X, $E = \{x \in \Omega \mid Q(X(x))\}$, we also use the shorthand notation

$$P[Q(X)] := P(\{x \in \Omega \mid Q(X(x))\}) = P(E).$$

We assume all sets to be measurable; when we write $P(A)$ for some set $A \subseteq \mathcal{X}^\infty$, we understand implicitly that A be measurable. This is not true: not all subsets of

$\mathcal{X}^\infty$ are measurable (assuming the axiom of choice). While we choose to do this for readability purposes, note that under some axioms compatible with Zermelo-Fraenkel set theory, notably the axiom of determinacy, all subsets of $\mathcal{X}^\infty$ are measurable.

2.2 Stochastic Processes

This section introduces some notions about sequences of random variables.

Definition 2.4 (Stochastic Process)**.** $(X_t)_{t\in\mathbb{N}}$ is a *stochastic process* iff X_t is a random variable for every $t \in \mathbb{N}$.

A stochastic process $(X_t)_{t\in\mathbb{N}}$ is *nonnegative* iff $X_t \geq 0$ for all $t \in \mathbb{N}$. The process is *bounded* iff there is a constant $c \in \mathbb{R}$ such that $|X_t| \leq c$ for all $t \in \mathbb{N}$.

In the real numbers, a sequence $(z_t)_{t\in\mathbb{N}}$ converges if and only if it is a Cauchy sequence, i.e., iff $|z_{t+1} - z_t| \to 0$ as $t \to \infty$. For sequences of random variables convergence is a lot more subtle and there are several different notions of convergence.

Definition 2.5 (Stochastic Convergence)**.** Let P be a probability measure. A stochastic process $(X_t)_{t\in\mathbb{N}}$ *converges to the random variable* X

- *in P-probability* iff for every $\varepsilon > 0$,
$$P\big[|X_t - X| > \varepsilon\big] \to 0 \text{ as } t \to \infty;$$
- *in P-mean* iff
$$\mathbb{E}_P\big[|X_t - X|\big] \to 0 \text{ as } t \to \infty;$$
- *P-almost surely* iff
$$P\left[\lim_{t\to\infty} X_t = X\right] = 1.$$

Almost sure convergence and convergence in mean both imply convergence in probability (Wasserman, 2004, Thm. 5.17). If the stochastic process is bounded, then convergence in probability implies convergence in mean (Wasserman, 2004, Thm. 5.19).

A sequence of real numbers $(a_t)_{t\in\mathbb{N}}$ converges *in Cesàro average* to $a \in \mathbb{R}$ iff $1/t \sum_{k=1}^{t} a_k \to a$ as $t \to \infty$. The definition for sequences of random variables is analogous.

Definition 2.6 (Martingale)**.** Let P be a probability measure over $(\mathcal{X}^\infty, \mathcal{F}_\infty)$. A stochastic process $(X_t)_{t\in\mathbb{N}}$ is a *P-supermartingale (P-submartingale)* iff

(a) each X_t is $\mathcal{F}_t$-measurable, and

(b) $\mathbb{E}[X_t \mid \mathcal{F}_s] \leq X_s$ ($\mathbb{E}[X_t \mid \mathcal{F}_s] \geq X_s$) P-almost surely for all $s, t \in \mathbb{N}$ with $s < t$.

A *P-martingale* is a process that is both a P-supermartingale and a P-submartingale.

Example 2.7 (Fair Gambling)**.** Suppose Mary bets on the outcome of a fair coin flip. If she predicts correctly, her wager is doubled and otherwise it is lost. Let X_t denote Mary's wealth at time step t. Since the game is fair, $\mathbb{E}[X_{t+1} \mid \mathcal{F}_t] = X_t$ where $\mathcal{F}_t$ represents the information available at time step t. Hence $\mathbb{E}[X_t] = X_1$, so in expectation she never loses money regardless of her betting strategy. ◇

For martingales the following famous convergence result was proved by Doob (1953).

Theorem 2.8 (Martingale Convergence; Durrett, 2010, Thm. 5.2.9)**.** *If $(X_t)_{t\in\mathbb{N}}$ is a nonnegative supermartingale, then it converges almost surely to a limit X with $\mathbb{E}[X] \leq \mathbb{E}[X_1]$.*

By Theorem 2.8 the martingale from Example 2.7 representing Mary's wealth converges almost surely, regardless of her betting strategy. Either she refrains from betting at some point (assuming she cannot place smaller and smaller bets) or she cannot play anymore because her wealth is 0. Is there a lesson to learn here about gambling?

2.3 Information Theory

This section introduces the notions of *entropy* and two notions of distance between probability measures: *KL-divergence* and *total variation distance*.

Definition 2.9 (Entropy)**.** Let Ω be a countable set. For a probability distribution $p \in \Delta\Omega$, the *entropy* of p is defined as

$$\operatorname{Ent}(p) := -\sum_{x\in\Omega:\, p(x)>0} p(x) \log p(x).$$

Definition 2.10 (KL-Divergence)**.** Let P, Q be two measures and let $m \in \mathbb{N}$ be a lookahead time step. The *Kullback-Leibler-divergence* (*KL-divergence*) of P and Q between time steps t and m is defined as

$$\mathrm{KL}_m(P, Q \mid x_{<t}) := \sum_{x_{t:m}\in\mathcal{X}^{m-t+1}} P(x_{1:m} \mid x_{<t}) \log \frac{P(x_{1:m} \mid x_{<t})}{Q(x_{1:m} \mid x_{<t})}.$$

Moreover, we define $\mathrm{KL}_\infty(P, Q \mid x_{<t}) := \lim_{m\to\infty} \mathrm{KL}_m(P, Q \mid x_{<t})$.

KL-divergence is also known as *relative entropy*. KL-divergence is always nonnegative by Gibbs' inequality, but it is not a distance since it is not symmetric. If the alphabet $\mathcal{X}$ is finite, then $\mathrm{KL}_m(P, Q \mid x)$ is always finite. However, $\mathrm{KL}_\infty(P, Q \mid x)$ may be infinite.

Definition 2.11 (Total Variation Distance)**.** Let P, Q be two measures and let $1 \leq m \leq \infty$ be a lookahead time step. The *total variation distance* between P and Q between time steps t and m is defined as

$$D_m(P, Q \mid x) := \sup_{A\subseteq\mathcal{X}^m} \Big|P(A \mid x_{<t}) - Q(A \mid x_{<t})\Big|.$$

Total variation distance is always bounded between 0 and 1 since P and Q are probability measures. Moreover, in contrast to KL-divergence total variation distance satisfies the axioms of distance: symmetry ($D(P,Q) = D(Q,P)$), identity of indiscernibles ($D(P,Q) = 0$ if and only if $P = Q$), and the triangle inequality ($D(P,Q) + D(Q,R) \geq D(P,R)$).

The following lemma shows that total variation distance can be used to bound differences in expectation.

Lemma 2.12 (Total Variation Bound on the Expectation)**.** *For a random variable X with $0 \leq X \leq 1$ and two probability measures P and Q*

$$\left|\mathbb{E}_P[X] - \mathbb{E}_Q[X]\right| \;\leq\; D(P,Q).$$

KL-divergence and total variation distance are linked by the following inequality.

Lemma 2.13 (Pinsker's inequality; Tsybakov, 2008, Lem. 2.5i)**.** *For all probability measures P and Q on $\mathcal{X}^\infty$, for every $x \in \mathcal{X}^*$, and for every $m \in \mathbb{N}$*

$$D_m(P,Q \mid x) \leq \sqrt{\frac{1}{2}\mathrm{KL}_m(P,Q \mid x)}$$

2.4 Algorithmic Information Theory

A *universal Turing machine* (UTM) is a Turing machine that can simulate all other Turing machines. Formally, a Turing machine U is a UTM iff for every Turing machine T there is a binary string p (called *program*) such that $U(p,x) = T(x)$ for all $x \in \mathcal{X}^*$, i.e., the output of U when run on (p,x) is the same as the output of T when run on x. We assume the set of programs on U is prefix-free. The *Kolmogorov complexity* $K(x)$ of a string x is the length of the shortest program on U that prints x and then halts:

$$K(x) := \min\{|p| \mid U(p) = x\}.$$

A *monotone Turing machine* is a Turing machine with a one-way read-only input tape, a one-way write-only output tape, and a read/write work tape. Monotone Turing machines sequentially read symbols from their input tape and write to their output tape. Interpreted as a function, a monotone Turing machine T maps a string x to the longest string that T writes to the output tape while reading x and no more from the input tape (Li and Vitányi, 2008, Ch. 4.5.2).

We also use U to denote a universal monotone Turing machine (programs on the universal monotone Turing machine do not have to be prefix-free). The *monotone Kolmogorov complexity* $Km(x)$ denotes the length of the shortest program on the monotone machine U that prints a string starting with x (Li and Vitányi, 2008, Def. 4.5.9):

$$Km(x) := \min\{|p| \mid x \sqsubseteq U(p)\}. \tag{2.1}$$

Since monotone complexity does not require the machine to halt, there is a constant c such that $Km(x) \leq K(x) + c$ for all $x \in X^*$.

The following notion of a (semi)measure is particular to algorithmic information theory.

Definition 2.14 (Semimeasure; Li and Vitányi, 2008, Def. 4.2.1)**.** A *semimeasure* over the alphabet $\mathcal{X}$ is a function $\nu : \mathcal{X}^* \to [0,1]$ such that

(a) $\nu(\epsilon) \leq 1$, and

(b) $\nu(x) \geq \sum_{a\in\mathcal{X}} \nu(xa)$ for all $x \in \mathcal{X}^*$.

A semimeasure is a *(probability) measure* iff equalities hold in (a) and (b) for all $x \in \mathcal{X}^*$.

Semimeasures are not probability measures in the classical measure theoretic sense. However, semimeasures correspond canonically to classical probability measures on the probability space $\mathcal{X}^\sharp = \mathcal{X}^* \cup X^\infty$ whose σ-algebra is generated by the cylinder sets (Li and Vitányi, 2008, Ch. 4.2 and Hay, 2007).

Lower semicomputable semimeasures correspond naturally to monotone Turing machines (Li and Vitányi, 2008, Thm. 4.5.2): for a monotone Turing machine T, the semimeasure λ_T maps a string x to the probability that T outputs something starting with x when fed with fair coin flips as input (and vice versa). Hence we can enumerate all lower semicomputable semimeasures $\nu_1, \nu_2, \ldots$ by enumerating all monotone Turing machines. We define the Kolmogorov complexity $K(\nu)$ of a lower semicomputable semimeasure ν as the Kolmogorov complexity of the index of ν in this enumeration.

We often mix the (semi)measures of algorithmic information theory with concepts from probability theory. For convenience, we identify a finite string $x \in \mathcal{X}^*$ with its cylinder set Γ_x. Then $\nu(x)$ in the algorithmic information theory sense coincides with $\nu(\Gamma_x)$ in the measure theory sense if we use the identification of semimeasures with probability measures above.

Example 2.15 (Lebesgue Measure)**.** The *Lebesgue measure* or *uniform measure* is defined as

$$\lambda(x) := (\#\mathcal{X})^{-|x|}. \qquad \diamond$$

The following definition turns a semimeasure into a measure, preserving the predictive ratio $\nu(xa)/\nu(xb)$ for $a, b \in \mathcal{X}$.

Definition 2.16 (Solomonoff Normalization)**.** The *Solomonoff normalization* ν_{norm} of a semimeasure ν is defined as $\nu_{\text{norm}}(\epsilon) := 1$ and for all $x \in \mathcal{X}^*$ and $a \in \mathcal{X}$,

$$\nu_{\text{norm}}(xa) := \nu_{\text{norm}}(x) \frac{\nu(xa)}{\sum_{b\in\mathcal{X}} \nu(xb)}. \tag{2.2}$$

By definition, ν_{norm} is a measure. Moreover, ν_{norm} dominates ν according to the following lemma.

Lemma 2.17 ($\nu_{\text{norm}} \geq \nu$)**.** *$\nu_{\text{norm}}(x) \geq \nu(x)$ for all $x \in \mathcal{X}^*$ and all semimeasures ν.*

Proof. We use induction on the length of x: if $x = \epsilon$ then $\nu_{\text{norm}}(\epsilon) = 1 = \nu(\epsilon)$, and otherwise

$$\nu_{\text{norm}}(xa) = \frac{\nu_{\text{norm}}(x)\nu(xa)}{\sum_{b\in\mathcal{X}} \nu(xb)} \geq \frac{\nu(x)\nu(xa)}{\sum_{b\in\mathcal{X}} \nu(xb)} \geq \frac{\nu(x)\nu(xa)}{\nu(x)} = \nu(xa).$$

The first inequality holds by induction hypothesis and the second inequality uses the fact that ν is a semimeasure. □

Chapter 3

Learning

The problem of induction is essentially solved. — David Hume

Machine learning refers to the process of learning models of and/or making predictions about (large) sets of data points that are typically independent and identically distributed (i.i.d.); see Bishop (2006) and Hastie et al. (2009). In this chapter we do *not* make the i.i.d. assumption. Instead, we aim more generally at the theoretical fundamentals of the *sequence prediction problem*: how will a sequence of symbols generated by an unknown stochastic process be continued? Given a finite string $x_{<t} = x_1 x_2 \dots x_{t-1}$ of symbols, what is the next symbol x_t? How likely does a given property hold for the entire sequence $x_{1:\infty}$? Arguably, any learning or prediction problem can be phrased in this fashion: anything that can be stored on a computer can be turned into a sequence of bits.

We distinguish two major elements of learning. First, the process of converging to accurate beliefs, called *merging*. Second, the process of making accurate forecasts about the next symbol, called *predicting*. These two notions are not distinct: if you have accurate beliefs about the unseen data, then you can make good predictions, but not necessarily vice versa (see Example 3.41). We discuss different notions of merging in Section 3.4 and state bounds on the prediction regret in Section 3.5.

In the general reinforcement learning problem we target in this thesis, the environment is unknown and the agent needs to learn it. The literature on non-i.i.d. learning has focused on predicting individual symbols and bounds on the number of prediction errors (Hutter, 2001b, 2005; Cesa-Bianchi and Lugosi, 2006), and the results on merging are from the game theory literature (Blackwell and Dubins, 1962; Kalai and Lehrer, 1994; Lehrer and Smorodinsky, 1996). However, we argue that merging is the essential property for general AI. In order to make good decisions, the agent needs to have accurate beliefs about what its actions will entail. On a technical level, merging leads to on-policy value convergence (Section 4.2.3), the fact that the agents learns to estimate the values for its own policy correctly.

The setup we consider is the *realizable case*: we assume that the data is generated by an unknown probability distribution that belongs to a known (countable) class of distributions. In contrast, the *nonrealizable case* allows no assumptions on the underlying process that generates the data. A well-known approach to the nonrealizable case is *prediction with expert advice* (Cesa-Bianchi and Lugosi, 2006), which we do not con-

sider here. Generally, the nonrealizable case is harder, but Ryabko (2011) argues that for some problems, both cases coincide.

After introducing the formal setup in Section 3.1, we discuss several examples for learning distributions and notions that relate the learning distribution with the process generating the data in Section 3.2. In Section 3.3 we connect these notions to the theory of martingale processes.

Section 3.6 connects the results from the first sections to the learning framework developed by Solomonoff (1964, 1978), Hutter (2001b, 2005, 2007b), and Schmidhuber (2002) (among others). This framework relies on results from algorithmic information theory and computability theory to learn any computable distribution quickly and effectively. It is incomputable (see Section 6.2), but can serve as a gold standard for learning.

Most of this chapter echoes the literature. We collect results from economics and computer science that previously had not been assembled in one place. We provide proofs that connect the various properties (Proposition 3.23, Proposition 3.16, and Proposition 3.37), and we fill in a few gaps in the picture: the prediction bounds for absolute continuity (Section 3.5.2) and the improved regret bounds for nonuniform measures (Theorem 3.48 and Theorem 3.51). Section 3.7 summarizes the results in Table 3.2 on page 45 as well as Figure 3.1 on page 46.

3.1 Setup

For the rest of this chapter, fix P and Q to be two probability measures over the measurable space of infinite sequences $(\mathcal{X}^\infty, \mathcal{F}_\infty)$. We think of P as the *true distribution* from which the data sequence $x_{1:\infty}$ is drawn, and of Q as our belief distribution or learning algorithm. In other words, we use the distribution Q to learn a string drawn from the distribution P.

Let H denote a *hypothesis*, i.e., any measurable set from $\mathcal{F}_\infty$. Our *prior belief* in the hypothesis H is $Q(H)$. In each time step t, we make one observation $x_t \in \mathcal{X}$. Our *history* $x_{<t} = x_1 x_2 \dots x_{t-1}$ is the sequence of all previous observations. We update our belief in accordance with Bayesian learning; our *posterior belief* in the hypothesis H is

$$Q(H \mid x_{1:t}) \;=\; \frac{Q(H \cap x_{1:t})}{Q(x_{1:t})}.$$

The observation x_t *confirms* the hypothesis H iff $Q(H \mid x_{1:t}) > Q(H \mid x_{<t})$ (the belief in H increases), and the observation x_t *disconfirms* the hypothesis H iff $Q(H \mid x_{1:t}) < Q(H \mid x_{<t})$ (the belief in H decreases). If $Q(H \mid x_{1:t}) = 0$, then H is *refuted* or *falsified*.

When we assign a prior belief of 0 to a hypothesis H, this means that we think that H is impossible; it is refuted from the beginning. If $Q(H) = 0$, then the posterior $Q(H \mid x_{<t}) = 0$, so no evidence whatsoever can change our mind that H is impossible. This is bad if the hypothesis H is actually true.

To be able to learn we need to make some assumptions on the learning distribution Q: we need to have an open mind about anything that might actually happen, i.e.,

$Q(H) > 0$ on any hypothesis H with $P(H) > 0$. This property is called *absolute continuity*. We discuss this and other notions of *compatibility of P and Q* in Section 3.2.

We motivate this chapter with the following example.

Example 3.1 (The Black Ravens; Rathmanner and Hutter, 2011, Sec. 7.4)**.** If we live in a world in which all ravens are black, how can we learn this fact? Since at every time step we have observed only a finite subset of the (possibly infinite) set of all ravens, how can we confidently state anything about all ravens?

We formalize this problem in line with Rathmanner and Hutter (2011, Sec. 7.4) and Leike and Hutter (2015d). We define two predicates, blackness B and ravenness R. There are four possible observations: a black raven BR, a non-black raven $\overline{B}R$, a black non-raven $B\overline{R}$, and a non-black non-raven $\overline{B}\overline{R}$. Therefore our alphabet consists of four symbols corresponding to each of the possible observations, $\mathcal{X} := \{BR, \overline{B}R, B\overline{R}, \overline{B}\overline{R}\}$.

We are interested in the hypothesis 'all ravens are black'. Formally, it corresponds to the measurable set

$$H \;:=\; \{x \in \mathcal{X}^\infty \mid x_t \neq \overline{B}R \;\forall t\} \;=\; \{BR, B\overline{R}, \overline{B}\overline{R}\}^\infty, \tag{3.1}$$

the set of all infinite strings in which the symbol $\overline{B}R$ does not occur.

If we observe a non-black raven, $x_t = \overline{B}R$, the hypothesis H is refuted since $H \cap x_{1:t} = \emptyset$ and this implies $Q(H \mid x_{1:t}) = 0$. In this case, our inquiry regarding H is settled. The interesting case is when the hypothesis H is in fact true ($P(H) = 1$), i.e., P does not generate any non-black ravens. The property we desire is that in a world in which all ravens are black, we arrive at this belief: $P(H) = 1$ implies $Q(H \mid x_{<t}) \to 1$ as $t \to \infty$. ◇

3.2 Compatibility

In this section we define *dominance*, *absolute continuity*, *dominance with coefficients*, *weak dominance*, and *local absolute continuity*, in decreasing order of their strength. These notions make the relationship of the two probability measures P and Q precise. We also give examples for various choices for the learning algorithm Q.

In our examples, we frequently rely on the following process.

Example 3.2 (Bernoulli Process)**.** Assume $\mathcal{X} = \{0, 1\}$. For a real number $r \in [0, 1]$ we define the *Bernoulli process with parameter r* as the measure

$$\text{Bernoulli}(r)(x) := r^{\text{ones}(x)}(1 - r)^{\text{zeros}(x)}.$$

Note that $\text{Bernoulli}(1/2) = \lambda$, the Lebesgue measure from Example 2.15. ◇

Definition 3.3 (Dominance)**.** The measure Q *dominates* P ($Q \overset{\times}{\geq} P$) iff there is a constant $c > 0$ such that $Q(x) \geq cP(x)$ for all finite strings $x \in \mathcal{X}^*$.

Dominance is also called *having a grain of truth* (Lehrer and Smorodinsky, 1996, Def. 2a and Kalai and Lehrer, 1993); we discuss this property in the context of game theory in Chapter 7.

Example 3.4 (Bayesian mixture)**.** Let $\mathcal{M}$ be a countable set of probability measures on $(\mathcal{X}^\infty, \mathcal{F}_\infty)$ and let $w \in \Delta\mathcal{M}$ be a prior over $\mathcal{M}$. If $w(P) > 0$ for all $P \in \mathcal{M}$, the prior w is called *positive* or *universal*. Then the Bayesian mixture $\xi := \sum_{P\in\mathcal{M}} w(P)P$ dominates each $P \in \mathcal{M}$. ◇

The Bayesian mixture is a mathematically simple yet very powerful concept. It is very easy to derive from a countable set of distributions, and it has been considered extensively in the literature (Solomonoff, 1964; Jaynes, 2003; Hutter, 2005, ...). Ryabko (2009) shows that even for uncountably infinite classes, if there are good predictors, then a Bayesian mixture over a countable subclass asymptotically also does well.

Example 3.5 (Solomonoff Prior)**.** Solomonoff (1964) defines a distribution M over $\mathcal{X}^\sharp$ that assigns to a string x the probability that the universal monotone Turing machine U outputs x when fed with fair coin flips on the input tape. Formally,

$$M(x) := \sum_{p:\, x\sqsubseteq U(p)} 2^{-|p|} \tag{3.2}$$

where p is a binary string.[1] The function M is a lower semicomputable semimeasure, but not computable and not a measure (Li and Vitányi, 2008, Lem. 4.5.3); see Section 6.2 for the computability properties of M. More importantly, M dominates every lower semicomputable semimeasure (Li and Vitányi, 2008, Thm. 4.5.1).

Solomonoff's prior M has a number of appealing philosophical properties. In line with Ockham's razor it favors simple environments over complex ones: Turing machines that have a short program on the UTM U have a higher contribution in the sum (3.2). In line with Epicurus' principle it never discards possible explanations: every program that produces the string x contributes to the sum. See Rathmanner and Hutter (2011) for a discussion on the philosophical underpinnings of Solomonoff's prior. ◇

Wood et al. (2011) show that the Solomonoff prior M can equivalently be defined as a Bayesian mixture over all lower semicomputable semimeasures with a prior $w(P) \propto 2^{-K(P)}$. (If we use $w(P) = 2^{-K(P)}$ we get a *semiprior* because $\sum_{P\in\mathcal{M}} 2^{-K(P)}$ can be less than 1. This prior also carries the name *Solomonoff prior*.)

Definition 3.6 (Absolute Continuity)**.** The measure P is *absolutely continuous with respect to* Q ($Q \gg P$) iff $Q(A) = 0$ implies $P(A) = 0$ for all measurable sets A.

Remark 3.7 (Absolute Continuity $\not\Rightarrow$ Dominance)**.** Absolute continuity is strictly weaker than dominance: let $\mathcal{X} := \{0, 1\}$ and define a probability measure P that assigns probability 2/3 to 1 and probability 1/3 to 0 until seeing the first 0, then P behaves like the Lebesgue measure λ. Formally,

$$P(x_{1:t}) := \begin{cases} \left(\frac{2}{3}\right)^t & \text{if } x_{1:t} = 1^t, \text{ and} \\ \left(\frac{2}{3}\right)^n \frac{1}{3}\lambda(x_{n+2:t}) & \text{if } \exists n \geq 0.\ 1^n 0 \sqsubseteq x_{1:t}. \end{cases}$$

[1] We use the name *Solomonoff prior* for both a distribution over $\mathcal{X}^\infty$ and a distribution over a computably enumerable set $\mathcal{M}$. Maybe M should better be called *Solomonoff mixture* to avoid confusion.

Since $\lambda(1^t)/P(1^t) = (3/4)^t \to 0$ as $t \to \infty$, there is no constant c such that $\lambda(x)/P(x) > c > 0$ for all finite strings $x \in \mathcal{X}^*$, hence λ does not dominate P. But P is absolutely continuous with respect to λ because P-almost surely we draw a 0 eventually, and then P behaves like λ. Hence P-almost surely $\lambda/P \not\to 0$. The claim now follows from Proposition 3.23b. ◇

The idea of Remark 3.7 is to 'punch a hole into λ' at the infinite string 1^∞. This infinite string has probability 0, hence this hole does not break absolute continuity. But it breaks dominance on this infinite string. Analogously we could punch countably many holes into a probability measure without breaking absolute continuity.

Definition 3.8 (Weak Dominance)**.** The measure Q *weakly dominates* P ($Q \overset{\times}{\geq}_W P$) iff

$$\lim_{t\to\infty} \frac{1}{t} \log \frac{Q(x_{1:t})}{P(x_{1:t})} = 0 \text{ with } P\text{-probability 1.}$$

Lehrer and Smorodinsky (1996, Rem. 8) point out that for any P and Q,

$$\limsup_{t\to\infty} \frac{1}{t} \log \frac{Q(x_{1:t})}{P(x_{1:t})} \leq 0 \; P\text{-almost surely,}$$

so crucial is whether the lim inf is also 0.

Remark 3.9 (Weak Dominance)**.** The measure Q weakly dominates P if and only if P-almost surely $\log(P(x)/Q(x)) \in o(t)$. ◇

Ryabko and Hutter (2007, 2008) consider the following definition. It is analogous to Definition 3.3, except that the constant c is allowed to depend on time.

Definition 3.10 (Dominance with Coefficients; Ryabko and Hutter, 2008, Def. 2)**.** The measure Q *dominates* P *with coefficients* f ($Q \geq P/f$) iff $Q(x) \geq P(x)/f(|x|)$ for all $x \in \mathcal{X}^*$.

If Q dominates P with coefficients f and f grows subexponentially ($f \in o(\exp)$), then Q weakly dominates P by Remark 3.9.

Example 3.11 (Speed Prior)**.** Schmidhuber (2002) defines a variant of Solomonoff's prior M that penalizes programs by their running time, called the *speed prior*. Consider the speed prior

$$S_{Kt}(x) := \sum_{p:\, x \sqsubseteq U(p)} \frac{2^{-|p|}}{t(U,p,x)}$$

where $t(U,p,x)$ is the number of time steps the Turing machine U takes to produce x from the program p. For any deterministic measure P computable in time q we have $S_{Kt}(x) \overset{\times}{\geq} P(x)/q(|x|)$. Therefore S_{Kt} dominates P with coefficients $O(q)$. If q is a polynomial (P is computable in polynomial time), then it grows subexponentially and thus S_{Kt} weakly dominates P. ◇

The semimeasure loss $S_{Kt}(x) - \sum_{a\in\mathcal{X}} S_{Kt}(xa)$ in the speed prior is quite substantial: since it takes at least n steps to output a string of length n, $M(x) \geq |x| S_{Kt}(x)$.

Example 3.12 (Laplace Rule)**.** The *Laplace rule* ρ_L is defined by

$$\rho_L(x_t \mid x_{<t}) := \frac{\#\{i < t \mid x_i = x_t\}}{t + \#\mathcal{X}}.$$

For $\mathcal{X} = \{0,1\}$ and $r \in [0,1]$ the measure ρ_L dominates Bernoulli(r) with coefficients $f(t) = t^{-\#\mathcal{X}+1}$ (Ryabko and Hutter, 2008, Prop. 3). ◇

Definition 3.13 (Local Absolute Continuity)**.** The measure P is *locally absolutely continuous with respect to* Q ($Q \gg_L P$) iff $Q(x) = 0$ implies $P(x) = 0$ for all finite strings $x \in \mathcal{X}^*$.

The notable difference between local absolute continuity and absolute continuity is that Definition 3.6 talks about arbitrary measurable sets while Definition 3.13 only talks about finite strings. The former is a much stronger property.

For example, every measure is locally absolutely continuous with respect to the Lebesgue measure since $\lambda(x) > 0$ for all finite strings $x \in \mathcal{X}^*$.

Local absolute continuity is an extremely weak property. If it is not satisfied, we have to be very careful when using Q for prediction: then there is a positive probability that we have to condition on a probability zero event.

Example 3.14 (The Minimum Description Length Principle; Grünwald, 2007)**.** Let $\mathcal{M}$ be a countable set of probability measures on $(\mathcal{X}^\infty, \mathcal{F}_\infty)$ and let $K : \mathcal{M} \to [0,1]$ be a function such that $\sum_{P\in\mathcal{M}} 2^{-K(P)} \leq 1$ called *regularizer*. Following notation from Hutter (2009a), we define for each $x \in \mathcal{X}^*$ the *minimal description length* model as

$$\mathrm{MDL}^x := \arg\min_{P\in\mathcal{M}}\{-\log P(x) + K(P)\}.$$

$-\log P(x)$ is the (arithmetic) code length of x given model P, and $K(P)$ is a complexity penalty for P. Given data $x \in \mathcal{X}^*$, MDL^x is the measure $P \in \mathcal{M}$ that minimizes the total code length of data and model.

Note that the Lebesgue measure is not locally absolutely continuous with respect to the MDL distribution $Q(x) := \mathrm{MDL}^x(x)$: for some $x \in \mathcal{X}^*$ the minimum description $P \in \mathcal{M}$ may assign probability zero to a continuation $xy \in \mathcal{X}^*$. ◇

Remark 3.15 (MDL is Inductively Inconsistent; Leike and Hutter, 2014a, Cor. 13)**.** The MDL estimator for countable classes as defined in Example 3.14 is inductively inconsistent: the selected model $P \in \mathcal{M}$ can change infinitely often and thus the limit $\lim_{t\to\infty} \mathrm{MDL}^{x_{<t}}$ may not exist. This can be a major obstacle for using MDL for prediction, since the model used for prediction has to be changed over and over again, incurring the corresponding computational cost. ◇

The following proposition establishes the relationship between our notions of compatibility; see also Figure 3.1 on page 46.

Proposition 3.16 (Relationships between Compatibilities)**.**

(a) If $Q \overset{\text{⨯}}{\geq} P$*, then* $Q \gg P$*.*

(b) If $Q \gg P$, then $Q \overset{\times}{\geq}_W P$.

(c) If $Q \overset{\times}{\geq} P$, then Q dominates P with coefficients f for a constant function f.

(d) If Q dominates P with coefficients f and $f \in o(\exp)$, then $Q \overset{\times}{\geq}_W P$.

(e) If $Q \overset{\times}{\geq}_W P$, then $Q \gg_L P$.

Proof. (a) From Proposition 3.23 (a) and (b).

(b) From Proposition 3.23b and Kalai and Lehrer (1994, Prop. 3a).

(c) Follows immediately from the definitions.

(d) From Remark 3.9.

(e) Follows immediately from the definitions. □

Note that the converse of Proposition 3.16d is false: in Remark 3.7 we defined a measure P that is absolutely continuous with respect to λ (and hence is weakly dominated by λ), but the coefficients for P/λ grow exponentially on the string 1^t. This infinite string has P-probability 0, but dominance with coefficients demands the inequality $Q \geq P/f$ to hold for all strings.

Remark 3.17 (Local Absolute Continuity $\not\Rightarrow$ Absolute Continuity)**.** Define $P :=$ Bernoulli(2/3) and $Q :=$ Bernoulli(1/3). Both measures P and Q are nonzero on all cylinder sets: $Q(x) \geq 3^{-|x|} > 0$ and $P(x) \geq 3^{-|x|} > 0$ for every $x \in \mathcal{X}^*$. Therefore Q is locally absolutely continuous with respect to P. However, Q is *not* absolutely continuous with respect to P: define

$$A := \left\{ x \in \mathcal{X}^\omega \;\middle|\; \limsup_{t\to\infty} \tfrac{1}{t}\mathrm{ones}(x_{1:t}) \leq \tfrac{1}{2} \right\}.$$

The set A is $\mathcal{F}_\infty$-measurable since $A = \bigcap_{n=1}^\infty \bigcup_{x\in U_n} \Gamma_x$ with $U_n := \{x \in \mathcal{X}^* \mid |x| \geq n \text{ and } \mathrm{ones}(x) \leq |x|/2\}$, the set of all finite strings of length at least n that have at least as many zeros as ones. We have that $P(A) = 0$ and $Q(A) = 1$, hence Q is not absolutely continuous with respect to P. ◇

3.3 Martingales

The following two theorems state the connection between probability measures on infinite strings and martingales. For two probability measures P and Q the quotient Q/P is a nonnegative P-martingale if Q is locally absolutely continuous with respect to P. Conversely, for every nonnegative P-martingale there is a probability measure $Q \gg_L P$ such that the martingale is P-almost surely a multiple of Q/P.

Theorem 3.18 (Measures $\mapsto$ Martingales; Doob, 1953, II§7 Ex. 3)**.** *Let Q and P be two probability measures on $(\mathcal{X}^\infty, \mathcal{F}_\infty)$ such that Q is locally absolutely continuous with respect to P. Then the stochastic process $(X_t)_{t\in\mathbb{N}}$,*

$$X_t(x) := \frac{Q(x_{1:t})}{P(x_{1:t})}$$

is a nonnegative P-martingale with $\mathbb{E}[X_t] = 1$.

Theorem 3.19 (Martingales $\mapsto$ Measures)**.** *Let P be a probability measure on $(\mathcal{X}^\infty, \mathcal{F}_\infty)$ and let $(X_t)_{t\in\mathbb{N}}$ be a nonnegative P-martingale with $\mathbb{E}[X_t] = 1$. There is a probability measure Q on $(\mathcal{X}^\infty, \mathcal{F}_\infty)$ that is locally absolutely continuous with respect to P and for all $x \in \mathcal{X}^\infty$ and all $t \in \mathbb{N}$ with $P(x_{1:t}) > 0$,*

$$X_t(x) = \frac{Q(x_{1:t})}{P(x_{1:t})}.$$

The proofs for Theorem 3.18 and Theorem 3.19 are provided in the Appendix.

Example 3.20 (The Posterior Martingale)**.** Suppose we are interested in a hypothesis $H \subseteq \mathcal{X}^\infty$ (such as the proposition 'all ravens are black' in Example 3.1). If $Q(H) = \sum_{P\in\mathcal{M}} w(P)P(H)$ is a Bayesian mixture over a set of probability distributions $\mathcal{M}$ with prior weights $w \in \Delta\mathcal{M}$ (see Example 3.4), then the posterior belief $Q(H \mid x) = \sum_{P\in\mathcal{M}} w(P \mid x)P(H \mid x)$. The weights $w(P \mid x)$ are called *posterior weights*, and satisfy the identity

$$w(P \mid x) = w(P)\frac{P(x)}{Q(x)} \tag{3.3}$$

since

$$\begin{aligned} Q(H \mid x) &= \frac{Q(H \cap x)}{Q(x)} \\ &= \frac{1}{Q(x)} \sum_{P\in\mathcal{M}} w(P)P(H \cap x) \\ &= \sum_{P\in\mathcal{M}} w(P)\frac{P(x)P(H \cap x)}{Q(x)P(x)} \\ &= \sum_{P\in\mathcal{M}} w(P \mid x)P(H \mid x). \end{aligned}$$

According to Theorem 3.18 the posterior weight $w(P \mid x)$ is a Q-martingale with expectation $w(P)$. In particular, this means that the posterior weights converge Q-almost surely by the martingale convergence theorem (Theorem 2.8). Since Q dominates P, by Proposition 3.16a P is absolutely continuous with respect to Q and hence the posterior also converges P-almost surely. ◇

Remark 3.21 (Martingales and Absolute Continuity)**.** While Theorem 3.18 trivially also holds if Q is absolutely continuous with respect to P, Theorem 3.19 does not imply that Q is absolutely continuous with respect to P.

Let P and Q be defined as in Remark 3.17. Consider the process $X_0(x) := 1$,

$$X_{t+1}(x) := \begin{cases} 2X_t, & \text{if } x_{t+1} = 0, \text{ and} \\ \frac{1}{2}X_t, & \text{if } x_{t+1} = 1. \end{cases}$$

The process $(X_t)_{t\in\mathbb{N}}$ is a nonnegative P-martingale since every X_t is $\mathcal{F}_t$-measurable and for $x = y_{1:t}$ we have

$$\begin{aligned} \mathbb{E}[X_{t+1} \mid \mathcal{F}_t](y) &= P(x0 \mid x)2X_t(y) + P(x1 \mid x)\tfrac{1}{2}X_t(y) \\ &= \tfrac{1}{3}2X_t(y) + \tfrac{2}{3}\cdot\tfrac{1}{2}X_t(y) = X_t(y). \end{aligned}$$

Moreover,

$$Q(x) = \left(\tfrac{1}{3}\right)^{\text{ones}(x)}\left(\tfrac{2}{3}\right)^{\text{zeros}(x)} = \left(\tfrac{2}{3}\right)^{\text{ones}(x)}\left(\tfrac{1}{3}\right)^{\text{zeros}(x)} 2^{-\text{ones}(x)}2^{\text{zeros}(x)} = P(x)X_t(y).$$

Hence $X_t(y) = Q(y_{1:t})/P(y_{1:t})$ P-almost surely. The measure Q is uniquely defined by its values on the cylinder sets, and as shown in Remark 3.17, Q is not absolutely continuous with respect to P. ◇

Theorem 3.22 (Radon-Nikodym Derivative)**.** *If $Q \gg P$, then there is a function $dP/dQ : \mathcal{X}^\infty \to [0,\infty)$ called the* Radon-Nikodym derivative *such that*

$$\int f dP = \int f \frac{dP}{dQ} dQ$$

for all measurable functions f.

This function dP/dQ can be seen as a density function of P with respect to the background measure Q. Moreover, dP/dQ is the limit of the martingale P/Q (Durrett, 2010, Sec. 5.3.3) which exists Q-almost surely according to Theorem 2.8.

The following proposition characterizes the notions of compatibility from Section 3.2 in terms of the martingale Q/P.

Proposition 3.23 (Martingales and Compatibility)**.** *The following relationships hold between Q, P, and the P-martingale $Y_t := Q(x_{1:t})/P(x_{1:t})$.*

(a) $Q \overset{\times}{\geq} P$ if and only if $Y_t \geq c > 0$ for all $t \in \mathbb{N}$.

(b) $Q \gg P$ if and only if P-almost surely $Y_t \not\to 0$ as $t \to \infty$.

(c) Q dominates P with coefficients f if and only if $Y_t \geq 1/f(t)$ for all t.

(d) $Q \overset{\times}{\geq}_W P$ if and only if P-almost surely $\log(Y_{t+1}/Y_t) \to 0$ in Cesàro average.

(e) $Q \gg_L P$ if and only if P-almost surely $Y_t > 0$ for all $t \in \mathbb{N}$.

Proof. $(Y_t)_{t\in\mathbb{N}}$ is a P-martingale according to Theorem 3.18.

(a) $Q(x) \geq cP(x)$ with $c > 0$ for all $x \in \mathcal{X}^*$ is equivalent to $Q(x)/P(x) \geq c > 0$ for all $x \in \mathcal{X}^*$.

(b) Proved by Hutter (2009a, Lem. 3i).

(c) Analogously to the proof of (a).

(d) If Q weakly dominates P, we get $-\log Y_t \in o(t)$ according to Remark 3.9. Together with $Y_0 = 1$ we get $-\log Y_t = \sum_{k=0}^{t-1} -\log(Y_{k+1}/Y_k) \in o(t)$, therefore $t^{-1}\sum_{k=0}^{t-1} -\log(Y_{k+1}/Y_k) \to 0$ as $t \to \infty$. Conversely, if the Cesàro average converges to 0, then $t^{-1}\log Y_t \to 0$, hence $-\log Y_t \in o(t)$.

(e) Let $x \in \mathcal{X}^*$ be any finite string. If $Q \gg_L P$ and $P(x) > 0$, then $Q(x) > 0$, and hence $Q(x)/P(x) > 0$. Conversely, if $P(x) > 0$ then Y_t is well-defined, so if $Y_{|x|}(x) > 0$ then $Q(x) > 0$. □

From Proposition 3.23b and Theorem 3.22 we get that $Q \gg P$ if and only if the Radon-Nikodym derivative dQ/dP is positive on a set of P-measure 1.

3.4 Merging

If Q is capable of learning, it should use the sequence x drawn from P to change its opinions more in the direction of P. More precisely, we want $Q(\cdot \mid x_{<t}) \approx P(\cdot \mid x_{<t})$ for large t. In the rest of this chapter, we make this notion of closeness precise and discuss different conditions on Q that are sufficient for learning.

Strong merging implies that the belief of *any* hypothesis merges. This is very strong, as hypotheses can talk about *tail events*: events that are independent of any finite initial part of the infinite sequence (such as the event A in Remark 3.17). *Weak merging* only considers hypothesis about the next couple of symbols, and *almost weak merging* allows Q to deviate from P in a vanishing fraction of the time. Much of this section is based on Kalai and Lehrer (1994) and Lehrer and Smorodinsky (1996).

3.4.1 Strong Merging

Definition 3.24 (Strong Merging)**.** *Q merges strongly with P* iff $D_\infty(P, Q \mid x_{<t}) \to 0$ as $t \to \infty$ P-almost surely.

The following theorem is the famous merging of opinions theorem by Blackwell and Dubins (1962).

Theorem 3.25 (Absolute Continuity ⇒ Strong Merging; Blackwell and Dubins, 1962)**.** *If P is absolutely continuous with respect to Q, then Q merges strongly with P.*

Example 3.26 (The Black Ravens 2; Rathmanner and Hutter, 2011, Sec. 7.4)**.** Recall the black raven problem from Example 3.1. Let Q be a learning distribution that dominates the true distribution P, such as a Bayesian mixture (Example 3.4). By Proposition 3.16a we get $Q \gg P$, and hence Q merges strongly to P by Theorem 3.25. Thus we get as $t \to \infty$ that P-almost surely $|Q(H \mid x_{<t}) - P(H \mid x_{<t})| \to 0$ for the hypothesis H that 'all ravens are black' defined in (3.1). Thus if all ravens are black in the real world ($P(H) = 1$), Q learns this asymptotically ($Q(H \mid x_{<t}) \to 1$). This is

the solution we desired: the learning distribution Q converges to a true belief about an infinite set by only looking from a finite (but growing) number of data points. ◇

The following is the converse of Theorem 3.25.

Theorem 3.27 (Strong Merging ∧ Local Absolute Continuity ⇒ Absolute Continuity; Kalai and Lehrer, 1994, Thm. 2)**.** *If Q is locally absolutely continuous with respect to P and Q merges strongly with P, then P is absolutely continuous with respect to Q.*

The following result shows that local absolute continuity is not required for strong merging: recall that according to Example 3.14 the MDL distribution is not locally absolutely continuous with respect to every P from the class $\mathcal{M}$.

Theorem 3.28 (Strong Merging for MDL; Hutter, 2009a, Thm. 1)**.** *If $P \in \mathcal{M}$, then*

$$D_\infty(P, \mathrm{MDL}^x \mid x) \to 0 \text{ as } |x| \to \infty \; P\text{-almost surely.}$$

Let $\mathcal{M}$ be a (possibly uncountable) set of probability measures on $(\mathcal{X}^\infty, \mathcal{F}_\infty)$. Ryabko (2010, Thm. 4) shows that if there is a Q that merges strongly with every $P \in \mathcal{M}$, then there is a Bayesian mixture over a countable subset of $\mathcal{M}$ that also merges strongly with every $P \in \mathcal{M}$.

3.4.2 Weak Merging

In Definition 3.24 the supremum ranges over all measurable sets $A \in \mathcal{F}_\infty$ which includes tail events. Instead, we may restrict the supremum to the next few symbols. This is known as *weak merging*.

Definition 3.29 (Weak Merging)**.** *Q weakly merges with P* iff for every $d \in \mathbb{N}$, $D_{t+d}(Q, P \mid x_{<t}) \to 0$ as $t \to \infty$ P-almost surely.

The following lemma gives an equivalent formulation of weak merging.

Lemma 3.30 (Lehrer and Smorodinsky, 1996, Rem. 5)**.** *Q weakly merges with P if and only if $D_t(Q, P \mid x_{<t}) \to 0$ as $t \to \infty$ P-almost surely.*

Unfortunately, weak dominance is not sufficient for weak merging (Lehrer and Smorodinsky, 1996, Ex. 10). We need the following stronger condition, that turns out to be (almost) necessary. In the following, let $Y_t := Q(x_{1:t})/P(x_{1:t})$ denote the P-martingale from Proposition 3.23.

Theorem 3.31 (Kalai and Lehrer, 1994, Prop. 5a)**.** *If P-almost surely $Y_{t+1}/Y_t \to 1$, then Q merges weakly with P.*

Example 3.32 (Laplace Rule 2)**.** Suppose we use the Laplace rule from Example 3.12 to predict a Bernoulli(r) process. By the strong law of large numbers, $\rho_L(x_t \mid x_{<t}) \to r$ almost surely. Therefore we can use Theorem 3.31 to conclude that ρ_L merges weakly with Bernoulli(r) for all $r \in [0, 1]$. (Note that *strongly merging* with every Bernoulli process is impossible; Ryabko, 2010, p. 7) ◇

The following is a converse to Theorem 3.31.

Theorem 3.33 (Kalai and Lehrer, 1994, Prop. 5b)**.** *If Q merges weakly with P, then $Y_{t+1}/Y_t \to 1$ in P-probability.*

Unfortunately, weak dominance is not enough to guarantee weak merging.

Example 3.34 (Weak Dominance $\not\Rightarrow$ Weak Merging; Ryabko and Hutter, 2007, Prop. 7)**.** Let $\mathcal{X} = \{0,1\}$ and let f be any arbitrarily slowly monotone growing function with $f(t) \to \infty$. Define $P(1^\infty) := 1$, the sequence $(t_i)_{i\in\mathbb{N}}$ such that $f(t_{i+1}) \geq 2f(t_i)$, and

$$Q(x_t \mid x_{<t}) := \begin{cases} \frac{1}{2} & \text{if } t = t_i \text{ for some } i \in \mathbb{N}, \\ 1 & \text{if } t \neq t_i \text{ and } x_t = 1, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Now Q dominates P with coefficients f by construction and Q weakly dominates P if f grows subexponentially. However, $|Q(1 \mid 1^t) - P(1 \mid 1^t)| \geq 1/2$ for infinitely many $t \in \mathbb{N}$. ◇

3.4.3 Almost Weak Merging

The following definition is due to Lehrer and Smorodinsky (1996, Def. 10).

Definition 3.35 (Almost Weak Merging)**.** *Q almost weakly merges with P* iff for every $d \in \mathbb{N}$

$$\frac{1}{t}\sum_{k=1}^{t} D_{t+d}(Q, P \mid x_{<t}) \to 0 \text{ as } t \to \infty \; P\text{-almost surely.}$$

There is also an analogue of Lemma 3.30 for almost weakly merging in the sense that we can equivalently set $d = 0$ (Lehrer and Smorodinsky, 1996, Rem. 6).

Remark 3.36 (Weak Merging and Merging in KL-Divergence)**.** From Lemma 2.13 follows that weak merging is implied by $\mathrm{KL}_d(P, Q \mid x) \to 0$ P-almost surely and almost weak merging is implied by $\sum_{k=1}^{t} \mathrm{KL}_1(P, Q \mid x_{<k}) \in o(t)$ P-almost surely, i.e., $\mathrm{KL}_t(P, Q) \in o(t)$ (Ryabko and Hutter, 2008, Lem. 1). The converse is generally false. ◇

The following proposition relates the three notions of merging.

Proposition 3.37 (Strong Merging ⇒ Weak Merging ⇒ Almost Weak Merging)**.** *If Q merges strongly with P, then Q merges weakly with P. If Q merges weakly with P, then Q merges almost weakly with P.*

Proof. Follows immediately from the definitions. □

Theorem 3.38 (Weak Dominance ⇒ Almost Weak Merging; Lehrer and Smorodinsky, 1996, Thm. 4)**.** *If Q weakly dominates P, then Q merges almost weakly with P.*

From Theorem 3.38 we get that the speed prior (Example 3.11) merges almost weakly with any probability distribution estimable in polynomial time.

We also have the following converse to Theorem 3.38.

Theorem 3.39 (Almost Weak Merging ⇒ Weak Dominance; Lehrer and Smorodinsky, 1996, Cor. 7)**.** *If Q is locally absolutely continuous with respect to P, Q merges almost weakly with P, and P-almost surely* $\liminf_{t\to\infty} Y_{t+1}/Y_t > 0$*, then Q weakly dominates P.*

3.5 Predicting

In Section 3.4 we wanted Q to acquire the correct beliefs about P. In this section, we exploit the accuracy of our beliefs for predicting individual symbols. We derive bounds on the number of errors Q makes when trying to predict a string drawn from P.

Since the data drawn from P is stochastic, we cannot expect to make a finite number of errors. Even the perfect predictor that knows P generally makes an infinite number of errors. For example, trying to predict the Lebesgue measure λ (Example 2.15), in expectation we make half an error in every time step. So instead we are asking about the asymptotic error rate of a predictor based on Q compared to a predictor based on P, the *prediction regret*.

Let x_t^R be the t-th symbol predicted by the probability measure R according to the maximum likelihood estimator:

$$x_t^R :\in \arg\max_{a\in\mathcal{X}} R(x_{<t}a \mid x_{<t}). \tag{3.4}$$

The *instantaneous error* of a R-based predictor is defined as

$$e_t^R := \begin{cases} 0 & \text{if } x_t = x_t^R, \text{ and} \\ 1 & \text{otherwise.} \end{cases}$$

and the *cumulative error* is

$$E_t^R := \sum_{k=1}^{t} e_k^R.$$

Note that both e_t and E_t are random variables.

Definition 3.40 (Prediction Regret)**.** In time step t the *prediction regret* is $E_t^Q - E_t^P$ and the *expected prediction regret* is $\mathbb{E}\left[E_t^Q - E_t^P\right]$.

More generally, we could also follow Hutter (2001b) and phrase predictive performance in terms of *loss*: given a loss function $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ the predictor Q suffers an (instantaneous) loss of $\ell(x_t^Q, x_t^P)$ in time step t. If the loss function ℓ is bounded in $[0, 1]$, many of the results for prediction regret also hold for cumulative loss (for Section 3.5.2 we also need $\ell(a, a) = 0$ for all $a \in \mathcal{X}$). In this chapter we chose to phrase the results in terms of prediction errors instead of loss because prediction errors are conceptionally simpler.

Example 3.41 (Good Prediction Regret $\not\Rightarrow$ Merging/Compatibility)**.** Good prediction regret does not imply (weak/strong) merging or (weak) dominance: Let $P :=$ Bernoulli(1/3) and $Q :=$ Bernoulli(1/4). Clearly P and Q do not merge (weakly) or (weakly) dominate each other. However, a P-based predictor always predicts 0, and so does a Q-based predictor. Therefore the prediction regret $E_t^Q - E_t^P$ is always 0. ◇

Example 3.42 (Adversarial Sequence; Legg, 2006, Lem. 4)**.** No learning distribution Q will learn to predict everything. We can always define a *Q-adversarial sequence* $z_{1:\infty}$ recursively according to

$$z_t := \begin{cases} 0 & \text{if } Q(0 \mid z_{<t}) < 1/2, \text{ and} \\ 1 & \text{if } Q(0 \mid z_{<t}) \geq 1/2. \end{cases}$$

In every time step the probability that a Q-based predictor makes an error is at least 1/2, hence $e_t^Q \geq 1/2$ and $E_t^Q \geq t/2$. But $z_{1:\infty}$ is a deterministic sequence, thus an informed predictor makes zero errors. Therefore the prediction regret of Q on the sequence $z_{1:\infty}$ is linear. ◇

3.5.1 Dominance

We start with the prediction regret bounds proved by Hutter (2001b) in case the learning distribution Q dominates the true distribution P. In the following, let c_P denote the constant from Definition 3.3.

Theorem 3.43 (Hutter, 2007b, Eq. 5 & 8)**.** *For all P and Q,*

$$\sqrt{\mathbb{E}_P E_n^Q} - \sqrt{\mathbb{E}_P E_n^P} \leq \sqrt{2\mathrm{KL}_n(P,Q)}.$$

The following bound on prediction regret then follows easily, but it is a factor of $\sqrt{2}$ worse than the bound stated by Hutter (2005, Thm. 3.36).

Corollary 3.44 (Expected Prediction Regret)**.** *For all P and Q,*

$$0 \leq \mathbb{E}_P\left[E_n^Q - E_n^P\right] \leq 2\mathrm{KL}_n(P,Q) + 2\sqrt{2\mathrm{KL}_n(P,Q)\mathbb{E}_P E_n^P}.$$

Proof. From Theorem 3.43 we get

$$\begin{aligned}
\mathbb{E}_P\left[E_n^Q - E_n^P\right] &= \left(\sqrt{\mathbb{E}_P E_n^Q} + \sqrt{\mathbb{E}_P E_n^P}\right)\left(\sqrt{\mathbb{E}_P E_n^Q} - \sqrt{\mathbb{E}_P E_n^P}\right) \\
&\leq \left(\sqrt{\mathbb{E}_P E_n^Q} + \sqrt{\mathbb{E}_P E_n^P}\right)\sqrt{2\mathrm{KL}_n(P,Q)} \\
&\leq \left(\sqrt{2\mathrm{KL}_n(P,Q)} + \sqrt{\mathbb{E}_P E_n^P} + \sqrt{\mathbb{E}_P E_t^P}\right)\sqrt{2\mathrm{KL}_n(P,Q)} \\
&= 2\mathrm{KL}_n(P,Q) + 2\sqrt{2\mathrm{KL}_n(P,Q)\mathbb{E}_P E_n^P}.
\end{aligned}$$

□

If Q dominates P, then we have $\mathrm{KL}_n(P,Q) \leq -\ln c_P$:

$$\mathrm{KL}_n(P,Q) = \sum_{x\in\mathcal{X}^n} P(x)\log\frac{P(x)}{Q(x)} \leq \sum_{x\in\mathcal{X}^n} P(x)\log\frac{1}{c_P} = -\log c_P \tag{3.5}$$

This invites the following corollary.

Corollary 3.45 (Prediction Regret for Dominance; Hutter, 2005, Cor. 3.49)**.** *If Q dominates P, then the following statements hold.*

(a) $\mathbb{E}_P E_\infty^Q$ *is finite if and only if* $\mathbb{E}_P E_\infty^P$ *is finite.*

(b) $\sqrt{\mathbb{E}_P E_\infty^Q} - \sqrt{\mathbb{E}_P E_\infty^P} \in O(1)$

(c) $\mathbb{E}_P E_t^Q / \mathbb{E}_P E_t^P \to 1$ *for* $\mathbb{E}_P E_t^P \to \infty$.

(d) $\mathbb{E}_P\left[E_t^Q - E_t^P\right] \in O\left(\sqrt{\mathbb{E}_P E_t^P}\right)$.

If the true distribution P is deterministic, we can improve on these bounds:

Example 3.46 (Predicting a Deterministic Measure)**.** Suppose we are predicting a deterministic measure P that assigns probability 1 to the infinite string $x_{1:\infty}$. If P is dominated by Q, the total expected prediction regret $\mathbb{E}_P E_\infty^Q$ is bounded by $-2\ln c_P$ by Corollary 3.44. This is easy to see: every time we predict a wrong symbol $a \neq x_t$, then $Q(a \mid x_{<t}) \geq Q(x_t \mid x_{<t})$, so $Q(x_t \mid x_{<t}) \leq 1/2$. Therefore $Y_t \leq Y_{t-1}/2$ and by dominance $Y_t \geq c_P$. Hence a prediction error can occur at most $-\log c_P$ times. ◇

Generally, the $O(\mathbb{E}_P E_t^P)$ bounds on expected prediction regret given in Corollary 3.45 are essentially unimprovable:

Example 3.47 (Lower Bounds on Prediction Regret)**.** Set $\mathcal{X} := \{0,1\}$ and consider the uniform measure λ from Example 2.15. For each time step t, we have $\lambda(0 \mid x_{<t}) = \lambda(1 \mid x_{<t}) = 1/2$, so the argmax in (3.4) ties and hence it does not matter whether we predict 0 or 1. We take two predictors P and Q, where P always predicts 0 and Q always predicts 1. Let $Z_t := E_t^Q - E_t^P$. Since their predictions never match, Z_t is an ordinary random walk with step size 1. We have (Weisstein, 2002)

$$\limsup_{t\to\infty} \frac{\mathbb{E}_P[E_t^Q - E_t^P]}{\sqrt{t}} = \sqrt{2/\pi}$$

and for the law of the iterated logarithm (Durrett, 2010, Thm. 8.8.3)

$$\limsup_{t\to\infty} \frac{E_t^Q - E_t^P}{\sqrt{2t\log\log t}} = 1 \text{ } P\text{-almost surely.}$$

Both bounds are known to be asymptotically tight. ◇

While Example 3.47 shows that the bounds from Corollary 3.45 are asymptotically tight, they are misleading because in most cases, we can do much better. According

to the following theorem, the worst case bounds are only attained if $P(x_t \mid x_{<t})$ is sufficiently close to $1/2$.

Theorem 3.48 (Expected Prediction Regret for Nonuniform Measures)**.** *If $\mathcal{X} = \{0, 1\}$ and there is an $\varepsilon > 0$ such that $|P(x_t \mid x_{<t}) - 1/2| \geq \varepsilon$ for all $x_{1:t} \in \mathcal{X}^*$, then*

$$\mathbb{E}_P\left[E_t^Q - E_t^P\right] \leq \frac{\mathrm{KL}_t(P, Q)}{\varepsilon}.$$

Proof. Recall the definition of entropy in nats:

$$\mathrm{Ent}(p) := -p \ln p - (1 - p) \ln(1 - p).$$

The second order Taylor approximation of Ent at $1/2$ is

$$f(p) = \ln 2 - 2(p - \tfrac{1}{2})^2.$$

One can check that $f(p) \geq \mathrm{Ent}(p)$ for all $0 \leq p \leq 1$. Define $p := P(x_t^P \mid x_{<t}) \geq 1/2$ and $q := Q(x_t^Q \mid x_{<t}) \geq 1/2$ to ease notation. Consider the function

$$g(p, q, \varepsilon) := p - (1 - p) - \varepsilon^{-1}\big(p \ln \tfrac{p}{1-q} + (1 - p) \ln \tfrac{1-p}{q}\big)$$

which is strictly increasing as q decreases, so from $q \geq 1/2$ we get

$$\begin{aligned} g(p, q, \varepsilon) &\leq 2p - 1 - \varepsilon^{-1} \ln 2 + \varepsilon^{-1} \mathrm{Ent}(p) \ln 2 \\ &\leq 2p - 1 - \varepsilon^{-1} \ln 2 + \varepsilon^{-1} f(p) \ln 2 \\ &= 2p - 1 - \varepsilon^{-1} 2(p - \tfrac{1}{2})^2, \end{aligned}$$

which decreases as p increases, hence it is maximized for $p = 1/2 + \varepsilon$,

$$g(p, q, \varepsilon) \leq 2\varepsilon - \varepsilon^{-1} 2\varepsilon^2 = 0$$

Therefore g is nonpositive. If $x_t^Q = x_t^P$, the one-step error is 0. Otherwise $\mathbb{E}_P[e_t \mid x_{<t}] = p - (1 - p)$ and $g(p, q, \varepsilon) = \mathbb{E}_P[e_t \mid x_{<t}] - \varepsilon^{-1}\mathrm{KL}_1(P, Q \mid x_{<t})$, so we get $\mathbb{E}_P[e_t \mid x_{<t}] \leq \varepsilon^{-1}\mathrm{KL}_1(P, Q \mid x_{<t})$. Summing this from $t = 1$ to n yields the claim.

$$\mathbb{E}\Big[E_n^Q - E_n^P\Big] \leq \varepsilon^{-1}\mathrm{KL}_n(P, Q). \qquad \square$$

3.5.2 Absolute Continuity

Theorem 3.49 (Prediction with Absolute Continuity)**.** *If $Q \gg P$, then*

$$\sqrt{E_t^Q} - \sqrt{E_t^P} \leq O\left(\sqrt{\log \log t}\right) \ \textit{P-almost surely.}$$

The proof idea is inspired by Miller and Sanchirico (1999). We think of P and Q as two players in a zero-sum betting game. In every time step t, the players will make a bet on the outcome of x_t. If $x_t = x_t^Q \neq x_t^P$, then Q wins \$1 from P, if $x_t = x_t^P \neq x_t^Q$,

then Q loses \$1 to P. Otherwise $x_t^Q = x_t^P$ or $x_t^Q \neq x_t \neq x_t^P$ and neither player gains or loses money. Since Q predicts according to the maximum likelihood principle (3.4), it is rational to accept the bet from Q's perspective. In Q's eyes, the worst case is a fair bet, so Q will not lose more money than it would lose on a random walk. The law of the iterated logarithm gives a Q-probability one statement about this bound, which transfers to P by absolute continuity.

Proof. Define the stochastic process $Z_t := E_t^Q - E_t^P$. Since $\mathbb{E}_Q[e_t^R] = Q(x_t^R \mid x_{<t})$, we get

$$\begin{aligned}\mathbb{E}_Q[Z_{t+1} \mid \mathcal{F}_t] &= Q(x_t^Q \mid x_{<t}) - Q(x_t^P \mid x_{<t}) + Z_t \\ &\geq Q(x_t^Q \mid x_{<t}) - Q(x_t^Q \mid x_{<t}) + Z_t = Z_t,\end{aligned}$$

hence $(Z_t)_{t\in\mathbb{N}}$ is a Q-submartingale. In the worst case (for Q), $(Z_t)_{t\in\mathbb{N}}$ is just a random walk with step size 1. But Z_t can only move if Q and P predict a different symbol. If this happens, at least one of them makes an error. Let m_t be the number of steps Z_t has moved ($Z_{t+1} \neq Z_t$). Then $m_t \leq E_t^Q + E_t^P$ and $m_t \leq t$. By the law of the iterated logarithm (Durrett, 2010, Thm. 8.8.3),

$$\liminf_{t\to\infty} \frac{Z_t}{\sqrt{2m_t \log\log m_t}} = -1$$

Q-almost surely. We define the event

$$A := \left\{\exists C\, \forall t.\ Z_t \geq -C\sqrt{m_t \log\log m_t}\right\}.$$

Then $Q(A) = 1$, hence $P(A) = 1$ by absolute continuity.

$$E_t^Q - E_t^P = Z_t \leq C\sqrt{(E_t^Q + E_t^P)\log\log t} \leq C\left(\sqrt{E_t^Q} + \sqrt{E_t^P}\right)\sqrt{\log\log t}$$

Dividing both sides by $\sqrt{E_t^Q} + \sqrt{E_t^P}$ yields that there is a P-almost surely finite random variable C such that $\sqrt{E_t^Q} - \sqrt{E_t^P} \leq C\sqrt{\log\log t}$. □

This invites the following immediate corollary.

Corollary 3.50 (Prediction Regret for Absolute Continuity)**.** *If $Q \gg P$, then*

$$E_t^Q - E_t^P \in O\left(\log\log t + \sqrt{E_t^P \log\log t}\right) \quad P\text{-almost surely.}$$

Proof. Analogously to the proof of Corollary 3.44. □

While Corollary 3.50 establishes an almost sure prediction regret bound, it is different from the bound on expected prediction regret from Corollary 3.44; bounds on $\mathbb{E}[E_t^Q - E_t^P]$ are incomparable to almost sure bound given in Theorem 3.49: for a sequence of nonnegative (unbounded) random variables convergence in mean does not

imply almost sure convergence (Stoyanov, 2013, Sec. 14.7) or vice versa (Stoyanov, 2013, Sec. 14.8ii).

We proceed to establish an improved prediction regret bound in case P is nonuniform analogously to Theorem 3.48.

Theorem 3.51 (Prediction Regret for Nonuniform Measures)**.** *If $Q \gg P$, $\mathcal{X} = \{0, 1\}$, and there is an $\varepsilon > 0$ such that with P-probability 1*

$$|P(x_t \mid x_{<t}) - 1/2| \geq \varepsilon$$

for all $t \in \mathbb{N}$, then P-almost surely $E_t^Q - E_t^P \in O(1)$.

Proof. If $|P(x_t \mid x_{<t}) - 1/2| \geq \varepsilon$, then for large enough t, Q will have merged with P (Theorem 3.25) and hence $|Q(x_t \mid x_{<t}) - 1/2| \geq \varepsilon/2$ infinitely often.

Thus Z_t has an expected gain of $\varepsilon/2$ if the predictors disagree. Therefore $Z_t \to \infty$ Q-almost surely. Consequently, the set

$$A := \{\exists t_0 \, \forall t \geq t_0 . \; Z_t \geq 0\}$$

has Q-measure 1. By absolute continuity, it also has P-measure 1, hence there is a P-almost surely finite random variable C such that for all t, $Z_t \geq -C$. □

There is another argument that we could use to show that under the condition of Theorem 3.51 $E_t^Q - E_t^P$ is almost surely finite: If P is absolutely continuous with respect to Q, then Q merges strongly with P and hence Q merges weakly with P. Therefore almost surely there is a t_0 such that for all $t \geq t_0$ we have $|Q(x_t^P \mid x_{<t}) - P(x_t^P \mid x_{<t})| < \varepsilon$, thus $x_t^Q = x_t^P$ for $t \geq t_0$.

3.5.3 Dominance with Coefficients

Lemma 3.52 (KL Divergence and Dominance With Coefficients)**.** *If Q dominates P with coefficients f, then* $\mathrm{KL}_t(Q, P) \leq \ln f(t)$.

Proof. Analogous to (3.5). □

This lets us derive an analogous regret bound to Corollary 3.44.

Corollary 3.53 (Expected Prediction Regret for Dominance With Coefficients)**.** *If Q dominates P with coefficients f, then*

$$\mathbb{E}_P\left[E_n^Q - E_n^P\right] \leq 2 \ln f(t) + 2\sqrt{2 \mathbb{E}_P E_n^P \ln f(t)}.$$

Proof. Apply Lemma 3.52 to Corollary 3.44. □

For weak dominance we get sublinear prediction regret.

Corollary 3.54 (Sublinear Prediction Regret for Weak Dominance)**.** *If Q weakly dominates P, then $\mathbb{E}_P[E_n^Q - E_n^P] \in o(t)$.*

Proof. By Remark 3.9 $\ln f \in o(t)$. Applying Corollary 3.53 we get

$$\mathbb{E}_P\left[E_n^Q - E_n^P\right] \leq 2o(t) + 2\sqrt{2\mathbb{E}_P E_n^P o(t)} \leq 2o(t) + 2\sqrt{2O(t)o(t)} \in o(t). \qquad \square$$

3.6 Learning with Algorithmic Information Theory

Algorithmic information theory provides a theoretical framework to apply the probability theory results from the previous sections. In the following we discuss Solomonoff's famous theory of induction (Section 3.6.1), the speed prior (Section 3.6.2), and learning with a universal compression algorithm (Section 3.6.3).

3.6.1 Solomonoff Induction

Solomonoff (1964, 1978) proposed a theory of learning, also known as *universal induction* or *Solomonoff induction*. It encompasses *Ockham's razor* by favoring simple explanations over complex ones, and *Epicurus' principle of multiple explanations* by never discarding possible explanations. See Rathmanner and Hutter (2011) for a very readable introduction to Solomonoff's theory and its philosophical motivations and Sterkenburg (2016) for a critique of its optimality.

At the core of this theory is *Solomonoff's distribution* M, as defined in Example 3.5. Since M dominates all lower semicomputable semimeasures, we get all the merging and prediction results from Section 3.4 and Section 3.5: when drawing a string from any computable measure P, M arrives at the correct belief for any hypothesis.

Corollary 3.55 (Strong Merging for Solomonoff Induction)**.** *M merges strongly with every computable measure.*

Proof. From Proposition 3.16a and Theorem 3.25. $\square$

Corollary 3.56 (Expected Prediction Regret for Solomonoff Induction)**.** *For all computable measures P,*

$$\mathbb{E}_P\left[E_t^M - E_t^P\right] \leq K(P)\ln 4 + \sqrt{2\mathbb{E}_P E_t^P K(P)\ln 16}.$$

Proof. From Corollary 3.44 and $c_P = 2^{-K(P)}$. $\square$

Remark 3.57 (Converging Fast and Slow)**.** The convergence of M to a computable P is fast in the sense of Corollary 3.56: M cannot make many more prediction errors than P in expectation. When predicting an infinite computable sequence $x_{1:\infty}$, the total number of prediction errors is bounded by $|p|2\ln 2 \approx 1.4|p|$ where p is a program that generates $x_{1:\infty}$ (Example 3.46).

The convergence of M to P is also slow in the sense that $M(x_t \mid x_{<t}) \to 1$ slower than any computable function since $1 - M(x_t \mid x_{<t}) \overset{\times}{\geq} 2^{-\min_{n\geq t} K(n)}$ for all t. $\diamond$

The bound from Corollary 3.56 is not optimal. Even if we knew the program p generating the sequence $x_{1:\infty}$, there might be a shorter program p' that computes $x_{1:\infty}$;

hence the improved bound $E_\infty^M \leq |p'|2\ln 2$ also holds. Since Kolmogorov complexity is incomputable, we can't find the 'best' bound algorithmically.

Solomonoff induction may even converge on some incomputable measures.

Example 3.58 (M Converges on Some Incomputable Measures)**.** Let r be an incomputable real number. Then the measure $P := \text{Bernoulli}(r)$ is not computable and M is not absolutely continuous with respect to P: for

$$A := \left\{ x \in \mathcal{X}^\infty \;\middle|\; \lim_{t\to\infty} \text{ones}(x_{1:t}) = r \right\}$$

we have $P(A) = 1$ but $M(A) = 0$. Since $M \gg_L P$ we get from Theorem 3.27 that M does not merge with P. Nevertheless, M still succeeds at prediction because it dominates Bernoulli(q) for each rational q and the rationals are dense around r. According to Lehrer and Smorodinsky (1996, Lem. 3), this implies that M weakly dominates P and by Theorem 3.38 M almost weakly merges to P. ◇

The fact that M does not merge strongly with every Bernoulli(r) process is not a failure of Solomonoff's prior. Ryabko (2010, p. 7) shows that for the class of all Bernoulli measures there is no probability measure that merges strongly with each of them.

The definition of M has only one parameter: the choice of the universal Turing machine. The effect of this choice on the function K can be uniformly bounded by a constant by the *invariance theorem* (Li and Vitányi, 2008, Thm. 3.1.1). Hence the choice of the UTM changes the prediction regret bound from Corollary 3.56 only by a constant. This constant can be large, preventing any finite-time guarantees that are independent of the UTM. However, asymptotically Solomonoff induction succeeds even for terrible choices of the UTM.

The Solomonoff normalization M_{norm} of M is defined according to Definition 2.16. While M_{norm} dominates M according to Lemma 2.17 and thus every lower semicomputable semimeasure, in some respects, M_{norm} behaves a little differently from M. Another way to complete the semimeasure M into a measure is given in the following example.

Example 3.59 (The Measure Mixture; Gács, 1983, p. 74)**.** The *measure mixture* $\overline{M}$ is defined as

$$\overline{M}(x) := \lim_{n\to\infty} \sum_{y\in\mathcal{X}^n} M(xy). \tag{3.6}$$

It is the same as M except that the contributions by programs that do not produce infinite strings are removed: for any such program p, let k denote the length of the finite string generated by p. Then for $|xy| > k$, the program p does not contribute to $M(xy)$, hence it is excluded from $\overline{M}(x)$.

Similarly to M, the measure mixture $\overline{M}$ is not a (probability) measure since $\overline{M}(\epsilon) < 1$; but in this case normalization (2.2) is just multiplication with the constant $1/\overline{M}(\epsilon)$, leading to the *normalized measure mixture* $\overline{M}_{\text{norm}}$. ◇

Even though M merges strongly with any computable measure P with P-probability 1, Lattimore and Hutter (2013, 2015) show that generally it does not hold for all Martin-Löf random sequences (which also form a set of P-probability 1). Hutter and Muchnik (2007, Thm. 6) construct non-universal lower semicomputable semimeasures that have this convergence property for all P-Martin-Löf random sequences. For infinite nonrandom sequences whose bits are selectively predicted by some total recursive function, Lattimore et al. (2011, Thm. 10) show that the normalized Solomonoff measure M_{norm} converges to 1 on the selected bits. This does not hold for the unnormalized measure M (Lattimore et al., 2011, Thm. 12).

3.6.2 The Speed Prior

Solomonoff's prior M is incomputable (Theorem 6.3); a computable alternative is the speed prior from Example 3.11. In this section we state merging and prediction results for S_{Kt}, a speed prior introduced by Filan et al. (2016) formally defined in Example 3.11. It is slightly different from the speed prior defined by Schmidhuber (2002), but for the latter no compatibility properties are known for nondeterministic measures.

Definition 3.60 (Estimable in Polynomial Time)**.** A function $f : \mathcal{X}^* \to \mathbb{R}$ is *estimable in polynomial time* iff there is a function $g : \mathcal{X}^* \to \mathbb{R}$ computable in polynomial time such that $f \overset{\times}{=} g$.

For a measure P estimable in polynomial time the speed prior S_{Kt} dominates P with coefficients polynomial in $|x| - \log P(x)$ (Filan et al., 2016, Eq. 12). Thus S_{Kt} weakly dominates P and we get the following results.

Corollary 3.61 (Almost Weak Merging for S_{Kt})**.** *S_{Kt} almost weakly merges with every measure estimable in polynomial time.*

Proof. From Theorem 3.38 and Filan et al. (2016, Eq. 12) since $\log P$ does not grow superexponentially P-almost surely. □

Corollary 3.62 (Expected Prediction Regret for S_{Kt}; Filan et al., 2016, Thm. 9)**.** *For all measures P estimable in polynomial time,*

$$\mathbb{E}_P\left[E_n^{S_{Kt}} - E_n^P\right] \in O\left(\log n + \sqrt{\mathbb{E}_P E_\infty^P \log n}\right).$$

Proof. From Corollary 3.44 and Filan et al. (2016, Eq. 14). □

3.6.3 Universal Compression

Solomonoff's distribution can be approximated using a standard compression algorithm, motivated by the similarity $M(x) \approx 2^{-Km(x)}$, where Km denotes monotone Kolmogorov complexity. The function Km is a *universal compressor*, compressing at least as well as any other recursively enumerable program.

Gács (1983) shows that the similarity $M \approx 2^{-Km}$ is not an equality. However, the difference between $-\log M$ and Km is very small: the best known lower bound is due

to Day (2011) who shows that $Km(x) > -\log M(x) + O(\log\log|x|)$ for infinitely many $x \in \mathcal{X}^*$.

Nevertheless, 2^{-Km} dominates every computable measure (Li and Vitányi, 2008, Thm. 4.5.4 and Lem. 4.5.6ii(d); originally proved by Levin, 1973). Hence all the strong results that hold for Solomonoff induction (prediction regret and strong merging) also hold for compression: we apply Theorem 3.25 and Corollary 3.44 to get the following results. See Hutter (2006a) for further discussion on using the universal compressor Km for learning.

Corollary 3.63 (Strong Merging for Universal Compression)**.** *The distribution* $2^{-Km(x)}$ *merges strongly with every computable measure.*

Corollary 3.64 (Expected Prediction Regret for Universal Compression)**.** *For* $Q(x) := 2^{-Km(x)}$ *and for all computable measures* P *there is a constant* c_P *such that*

$$\mathbb{E}_P\left[E_t^Q - E_t^P\right] \leq c_P + \sqrt{c_P \mathbb{E}_P E_t^P}.$$

This provides a theoretical basis for viewing compression as a general purpose learning algorithm. In this spirit, the *Hutter prize* is awarded for the compression of a 100MB excerpt from the English Wikipedia (Hutter, 2006c).

Practical compression algorithms (such as the algorithm by Ziv and Lempel (1977) used in `gzip`) are not universal. Hence they do not dominate every computable distribution. As with the speed prior, what matters is the rate at which $Y_t = Q(x_{1:t})/P(x_{1:t})$ goes to 0, i.e., does the compressor weakly dominate the true distribution in the sense of Definition 3.8?

Veness et al. (2015) successfully apply the Lempel-Ziv compression algorithm as a learning algorithm for reinforcement learning; however, some preprocessing of the data is required. More remotely, Vitányi et al. (2009) use standard compression algorithms to classify mammal genomes, languages, and classical music.

3.7 Summary

Ultimately, whether learning succeeds depends on the rate at which the nonnegative P-martingale Q/P goes to 0 (when drawing from P). If Q/P does not converge to zero, then Q merges strongly with P and thus arrives at correct beliefs about any hypothesis, including tail events. If Q/P converges to zero subexponentially, then Q merges almost weakly with P and thus asymptotically has incorrect beliefs about the immediate future only a vanishing fraction of the time.

Corollary 3.44 bounds the expected prediction regret by the KL-divergence between P and Q plus a $\sqrt{\mathbb{E}_P E_t^P}$ term. The KL-divergence is in turn bounded by the rate at which Q/P goes to zero. It is constant if Q dominates P and bounded by $\ln f$ if Q dominates P with coefficients f. If Q weakly dominates P, then the KL-divergence is sublinear. We also derived bounds on the prediction regret for absolute continuity (Section 3.5.2). Remarkably, the bounds are only $\log\log t$ worse than the bound we get from dominance. Moreover, they hold almost surely instead of in expectation.

name	symbol	defined in	property
Bayesian mixture	ξ	Example 3.4	dominates every $P \in \mathcal{M}$
Solomonoff prior	M	Example 3.5	dominates every lower semi-computable semimeasure
universal compression	2^{-Km}	Equation 2.1	dominates every computable measure
speed prior	S_{Kt}	Example 3.11	weak dominates every measure estimable in polytime
Laplace rule	ρ_L	Example 3.12	merges weakly with every Bernoulli process
MDL	MDL^x	Example 3.14	merges strongly with every $P \in \mathcal{M}$

Table 3.1: Examples of learning distributions discussed in this chapter and their properties.

compatibility of P and Q	martingale	merging	prediction regret
$Q \overset{\times}{\geq} P$	$Y_t \geq c > 0$	strong merging	$-2\ln c + 2\sqrt{-2\mathbb{E}_P E_t^P \ln c}$
$Q \gg P$	$Y_t \not\to 0$	strong merging	$O\big(\log\log t + \sqrt{\mathbb{E}_P E_t^P \log\log t}\big)$
	$Y_{t+1}/Y_t \to 1$	weak merging	$o(t)$
$Q \geq P/f$	$Y_t \geq 1/f(t)$		$2\ln f(t) + 2\sqrt{2\mathbb{E}_P E_t^P \ln f(t)}$
$Q \overset{\times}{\geq}_W P$	$\log(Y_{t+1}/Y_t) \to 1$ in Cesàro average	almost weak merging	$o(t)$
$Q \gg_L P$	$Y_t > 0$		$O(t)$

Table 3.2: Summary on properties of learning. The first column lists different notions of compatibility introduced in Section 3.2; the second column lists properties of the P-martingale $Y_t := Q(x_{1:t})/P(x_{1:t})$ from Section 3.3; the third column lists different notions of merging discussed in Section 3.4; the fourth column states the bounds on the prediction regret (in expectation and almost surely respectively) from Section 3.5. Figure 3.1 illustrates the origin of the results.

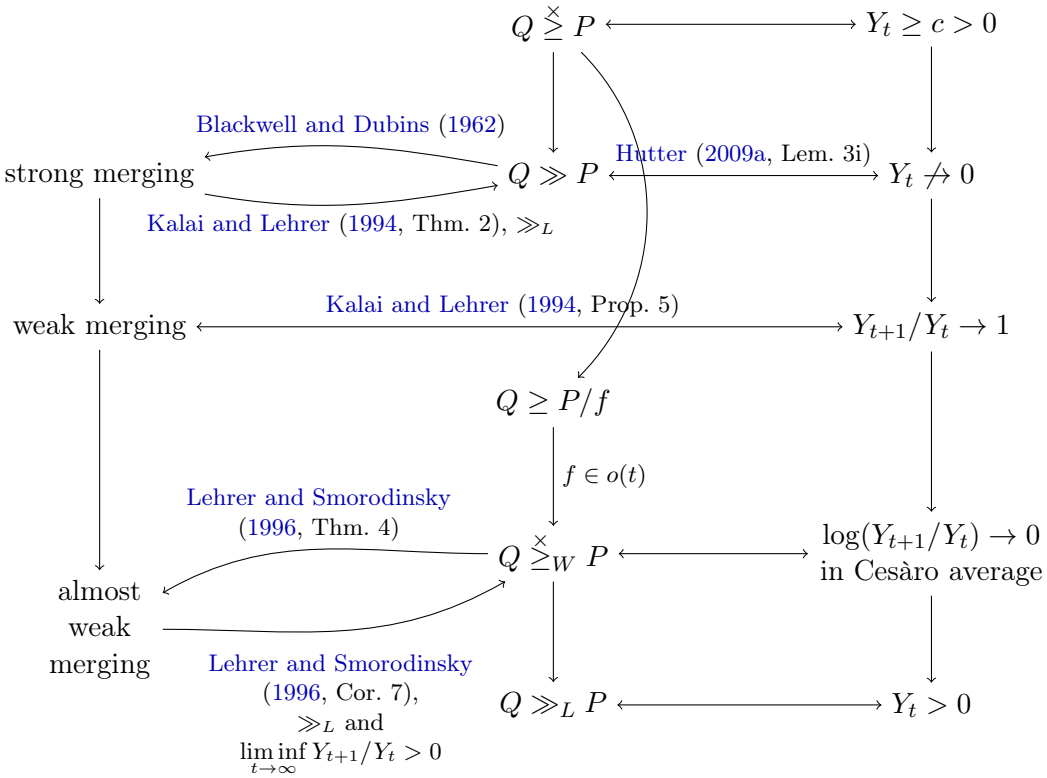


Figure 3.1: Properties of learning and their relationship. We use $Y_t := Q(x_{1:t})/P(x_{1:t})$. An arrow between two statements means that one statement implies the other. The transitive property of implications is not made explicit. The source of the result is indicated on the arrow, sometimes together with a side condition. If no source is given, then the relationship is easy and a proof can be found in this chapter.

Next, we showed that the $\sqrt{\mathbb{E}_P E_t^P}$ term is generally unimprovable (Example 3.47). However, it comes only from predicting measures that assign probabilities close to 1/2. If we can bound P away from 1/2, then the $\sqrt{\mathbb{E}_P E_t^P}$ term disappears (Theorem 3.48 and Theorem 3.51).

Table 3.1 lists our learning distributions. The Bayesian mixture is the strongest since it dominates every measure from the given class $\mathcal{M}$ (Example 3.4). The minimum description length model MDL^x does not have this property, yet it still merges strongly with every measure from the class (Example 3.14 and Theorem 3.28). The Laplace rule is only useful for learning i.i.d. measures; it merges weakly with every Bernoulli process (Example 3.12 and Example 3.32). We also discussed some learning distributions from algorithmic information theory. Solomonoff's prior is a Bayesian mixture over all lower semicomputable semimeasures (Example 3.5 and Wood et al., 2011). Like the universal compressor it dominates and hence merges strongly with all computable measures. The speed prior dominates all probability measures estimable in polynomial time with polynomial coefficients (Example 3.11), and thus merges weakly with each of them.

Table 3.2 summarizes the results from this chapter and Figure 3.1 illustrates their logical relationship and their origin.

We conclude this chapter with a paradox from the philosophy of science.

Remark 3.65 (The Paradox of Confirmation)**.** Recall the black raven problem introduced in Example 3.1; the hypothesis 'all ravens are black' is denoted with H. The *paradox of confirmation*, also known as *Hempel's paradox* (Hempel, 1945), relies on the following three principles.

- *Nicod's criterion* (Nicod, 1961, p. 67): observing an F that is a G increases our belief in the hypothesis that all Fs are Gs.
- *The equivalence condition*: logically equivalent hypotheses are confirmed or disconfirmed by the same evidence.
- *The paradoxical conclusion*: a green apple confirms H.

The argument goes as follows. The hypothesis H is logically equivalent to the hypothesis H' that all non-black objects are non-ravens. According to Nicod's criterion, any non-black non-raven, such as a green apple, confirms H'. But then the equivalence condition entails the paradoxical conclusion.

The paradox of confirmation has been discussed extensively in the literature on the philosophy of science (Hempel, 1945; Good, 1960; Mackie, 1963; Good, 1967; Hempel, 1967; Maher, 1999; Vranas, 2004); see Swinburne (1971) for a survey. Support for Nicod's criterion is not uncommon (Mackie, 1963; Hempel, 1967; Maher, 1999) and no consensus is in sight.

A Bayesian reasoner might be tempted to argue that a green apple *does* confirm the hypothesis H, but only to a small degree, since there are vastly more non-black objects than ravens (Good, 1960). This leads to the acceptance of the paradoxical conclusion,

and this solution to the confirmation paradox is known as the *standard Bayesian solution*. Vranas (2004) shows that this solution is equivalent to the assertion that blackness is equally probable regardless of whether H holds: $P(\text{black}|H) \approx P(\text{black})$.

The following is a very concise example against the standard Bayesian solution by Good (1967): There are two possible worlds, the first has 100 black ravens and a million other birds, while the second has 1000 black ravens, one white raven, and a million other birds. Now we draw a bird uniformly at random, and it turns out to be a black raven. Contrary to what Nicod's criterion claims, this is strong evidence that we are in fact in the second world, and in this world non-black ravens exist.

For another, more intuitive example: Suppose you do not know anything about ravens and you have a friend who collects atypical objects. If you see a black raven in her collection, surely this would not increase your belief in the hypothesis that all ravens are black.

In Leike and Hutter (2015d) we investigate the paradox of confirmation in the context of Solomonoff induction. We show that the paradoxical conclusion is avoided because Solomonoff induction violates Nicod's criterion: There are time steps when (counterfactually) observing a black raven disconfirms the hypothesis that all ravens are black. When predicting a deterministic computable sequence Nicod's criterion is even violated infinitely often. However, if we *normalize* Solomonoff's prior and observe a deterministic computable infinite string, Nicod's criterion is violated at most finitely many times. These results are independent of the choice of the universal Turing machine.

We must conclude that violating Nicod's criterion is not a fault of Solomonoff induction. Instead, we should accept that for Bayesian reasoning Nicod's criterion, in its generality, is false! Quoting the great Bayesian master Jaynes (2003, p. 144):

> In the literature there are perhaps 100 'paradoxes' and controversies which are like this, in that they arise from faulty intuition rather than faulty mathematics. Someone asserts a general principle that seems to him intuitively right. Then, when probability analysis reveals the error, instead of taking this opportunity to educate his intuition, he reacts by rejecting the probability analysis. ◇

Chapter 4

Acting

I ought never to act except in such a way that I could also will that my maxim should become a universal prior. — Immanuel Kant

Recall our decomposition of intelligence into *learning* and *acting* from Equation 1.1. The previous chapter made the notion of learning precise and provided several examples of learning distributions for the non-i.i.d. setting (see Table 3.1). Learning is passive: there is no interaction with the data-generating process. In this chapter we transition into the active setting: we consider an *agent* acting in an unknown *environment* in order to achieve a *goal*. In our case, this goal is maximizing reward; this is known as *reinforcement learning*. Where this reward signal originates does not concern us here.

In this thesis we consider is the *general reinforcement learning problem* in which we do not make several of the typical simplifying assumptions (see Table 1.1). Environments are only partially observable, have infinitely many states, and might contain traps from which the agent cannot escape. The context for making decision is the agent's entire history; its behavior is given by a *policy* that specifies how the agent behaves in any possible situation.

A central quantity in reinforcement learning is the *value function*. The value function quantifies the expected future discounted reward. Since the agent seeks to maximize reward, it aims to adopt a *policy* that has high value. Since the agent's environment is unknown to the agent, learning the value function is part of the challenge; otherwise we would call this planning.

If our agent is capable of learning in the sense of Chapter 3, then it learns the value of its own policy (*on-policy value convergence*). However, generally the agent does not learn to predict the value of counterfactual actions, actions that it does not take. Learning off-policy is hard because the agent receives no evidence about what would have happened on counterfactual actions. Nevertheless, off-policy learning is highly desirable because we want the agent to be confident that the policy it is currently following is in fact the best one; we want it to accurately predict that the counterfactual actions have less value.

This brings us back to the central theme of reinforcement learning: the tradeoff between *exploration* and *exploitation*. Asymptotically the agent needs to focus on exploitation, i.e., take the actions that it thinks yield the highest expected rewards. If the agent explores enough, then all actions are on-policy because they are all actions that

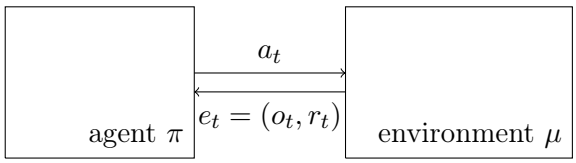


Figure 4.1: The dualistic agent model. At every time step t, the agent outputs an action a_t and subsequently receives a percept e_t consisting of an observation o_t and a real-valued reward r_t. The agent's policy π is a function that maps a history $æ_{<t}$ to the next action a_t, and the environment μ is a function that maps a history and an action to the next percept e_t.

the agent sometimes takes. Then on-policy learning ensures that the agent understands the consequences of every action and can confidently choose the best action. Effective exploration is performed by *knowledge-seeking agents*; these agents ignore the rewards and just focus on exploration.

This chapter introduces the central concepts of general reinforcement learning. It is mostly based on Hutter (2005) and Lattimore (2013). Section 4.1 specifies the general reinforcement learning problem, discusses discounting (Section 4.1.1), our implicit assumptions (Section 4.1.2), and typical environment classes (Section 4.1.3). Section 4.2 discusses the value function and its properties. In Section 4.3 we introduce the agents: AIXI (Section 4.3.1), knowledge-seeking agents (Section 4.3.2), BayesExp (Section 4.3.3), and Thompson sampling (Section 4.3.4).

4.1 The General Reinforcement Learning Problem

In reinforcement learning, an agent interacts with an environment: at time step $t \in \mathbb{N}$ the agent takes an *action* $a_t \in \mathcal{A}$ and subsequently receives a *percept* $e_t = (o_t, r_t) \in \mathcal{E}$ consisting of an *observation* $o_t \in \mathcal{O}$ and a *reward* $r_t \in \mathbb{R}$. This cycle then repeats for time step $t+1$ (see Figure 4.1).

A *history* is an element of $(\mathcal{A} \times \mathcal{E})^*$ and lists the actions the agent took and the percepts it received. We use $æ \in \mathcal{A} \times \mathcal{E}$ to denote one interaction cycle, and $æ_{<t} = æ_1 æ_2 \dots æ_{t-1}$ to denote a history of length $t-1$. For our agent, the history is a sufficient statistic about the past and in general reinforcement learning there is no simpler sufficient statistic.

For example, consider the agent to be a robot interacting with the real world. Its actions are moving the motors in its limbs and wheels and sending data packets over a network connection. Its observations are data from cameras and various other sensors. The reward could be provided either by a human supervisor or through a reward module that checks whether a predefined goal has been reached. The history is the collection of all the data it received and emitted in the past. The division of the robot's interaction with the environment into discrete time steps might seem a bit unnatural at first since the real world evolves according to a continuous process. However, note that the electronic components used in robots operate at discrete frequencies anyway.

In order to specify how the agent behaves in any possible situation, we define a *policy*: a policy is a function $\pi : (\mathcal{A}\times\mathcal{E})^* \to \Delta\mathcal{A}$ mapping a history $æ_{<t}$ to a distribution over actions $\pi(\cdot \mid æ_{<t})$ taken after seeing this history. Usually we do not distinguish between agent and policy. An *environment* is a function $\nu : (\mathcal{A} \times \mathcal{E})^* \times \mathcal{A} \to \Delta\mathcal{E}$ mapping a history $æ_{<t}$ and an action a_t to a distribution $\nu(\cdot \mid æ_{<t}a_t)$ over the percepts received after the history $æ_{<t}$ and action a_t. We use μ to denote the true environment.

Equivalently, Hutter (2005) defines environments as *chronological contextual semimeasures*.[1] A *contextual semimeasure* ν takes a sequence of actions $a_{1:\infty}$ as input and returns a semimeasure $\nu(\cdot \parallel a_{1:\infty})$ over $\mathcal{E}^\sharp$. A contextual semimeasure ν is *chronological* iff percepts at time t do not depend on future actions, i.e., $\nu(e_{1:t} \parallel a_{1:\infty}) = \nu(e_{1:t} \parallel a'_{1:\infty})$ whenever $a_{1:t} = a'_{1:t}$. For chronological contextual semimeasures we write $\nu(e_{1:t} \parallel a_{1:t})$ instead of $\nu(e_{1:t} \parallel a_{1:\infty})$. The two definition can be translated using the identities

$$\nu(e_{1:t} \parallel a_{1:t}) = \prod_{k=1}^{t} \nu(e_k \mid æ_{<k}a_k) \qquad \text{and} \qquad \nu(e_t \mid æ_{<t}a_t) = \frac{\nu(e_{1:t} \parallel a_{1:t})}{\nu(e_{<t} \parallel a_{<t})}. \tag{4.1}$$

If the policy π always assigns probability 1 to one of the actions, then π is called *deterministic*. Likewise, if the environment ν always assigns probability 1 to one of the percepts, then ν is called *deterministic*. For deterministic policies and environments we also use the notation $a_t = \pi(æ_{<t})$ and $e_t = \nu(æ_{<t}a_t)$. A deterministic policy π is *consistent with history* $æ_{<t}$ iff $a_k = \pi(æ_{<k})$ for all $k < t$. Likewise, a deterministic environment ν is *consistent with history* $æ_{<t}$ iff $e_k = \nu(æ_{<k}a_k)$ for all $k < t$.

Definition 4.1 (History Distribution)**.** An environment ν together with a policy π induces a *history distribution*

$$\nu^\pi(æ_{<t}) := \prod_{k=1}^{t} \pi(a_k \mid æ_{<k})\nu(e_k \mid æ_{<k}a_k).$$

We denote an expectation with respect to the history distribution ν^π with $\mathbb{E}^\pi_\nu$.

The history distribution is a (semi)measure on $(\mathcal{A}\times\mathcal{E})^\infty$. In the language of measure theory, our σ-algebra is the σ-algebra $\mathcal{F}_\infty$ generated by the cylinder sets introduced in Section 2.1. The filtration $(\mathcal{F}_t)_{t\in\mathbb{N}}$ formalizes that at time step t we have seen exactly the history $æ_{<t}$ (we use the σ-algebra $\mathcal{F}_{t-1}$). To simplify notation and help intuition, we simply condition expectations and probability measures with the history $æ_{<t}$ instead of $\mathcal{F}_{t-1}$ and sweep most of the measure-theoretic details under the rug.

With these preliminaries out of the way, we can now specify the *general reinforcement learning problem.*

Problem 4.2 (General Reinforcement Learning Problem)**.** *Given an arbitrary class of environments $\mathcal{M}$, choose a policy π that maximizes μ^π-expected reward when interacting with any environment $\mu \in \mathcal{M}$.*

[1] Hutter (2005) calls them *chronological conditional semimeasures*. This is confusing because contextual semimeasures do *not* specify conditional probabilities; the environment is *not* a joint probability distribution over actions and percepts.

Problem 4.2 is kept vague on purpose: it does not say how we should balance between achieving more rewards in some environments while achieving less in others. In other words, we leave open what an *optimal* solution to the general reinforcement learning problem is. This turns out to be a notoriously difficult question that we discuss in Chapter 5.

As promised in the title of this thesis, we take the *nonparametric* approach. For the rest of this thesis, fix $\mathcal{M}$ to be any countable set of environments. While the true environment is unknown, we assume it belongs to the class $\mathcal{M}$ (the *realizable case*). As long as the class $\mathcal{M}$ is sufficiently large (such as the class of all computable environments), this assumption is weak. Some typical choices are discussed in Section 4.1.3.

Our agent-environment setup shown in Figure 4.1 is known as the *dualistic model*: the agent is distinct from the environment and influences it only through its actions. In turn, the environment influences the agent only through the percepts. The dualism assumption is accurate for an algorithm that is playing chess, Go, or other (video) games, which explains why it is ubiquitous in AI research. But often it is not true: real-world agents are embedded in (and computed by) the environment, and then a *physicalistic model* (also called *materialistic model* or *naturalistic* model) is more appropriate. Decision making in the physicalistic model is still underdeveloped; see Everitt et al. (2015) and Orseau and Ring (2012a). In this thesis we restrict ourselves to the dualistic model.

4.1.1 Discounting

The goal in reinforcement learning is to maximize rewards. However, the infinite reward sum $\sum_{t=1}^{\infty} r_t$ may diverge. To get around this technical problem, we let our agent prioritize the present over the future. This is done with a *discount function* that quantifies how much the agent prefers rewards now over rewards later.

Definition 4.3 (Discount Function)**.** A *discount function* is a function $\gamma : \mathbb{N} \to \mathbb{R}$ with $\gamma_t := \gamma(t) \geq 0$ and $\sum_{t=1}^{\infty} \gamma_t < \infty$. The *discount normalization factor* is $\Gamma_t := \sum_{k=t}^{\infty} \gamma_k$.

There is no requirement that $\Gamma_t > 0$. In fact, we use γ for both, discounted infinite horizon ($\Gamma_t > 0$ for all t), and finite horizon m ($\Gamma_{m-1} > 0$ and $\Gamma_m = 0$) where the agent does not care what happens after time step m.

Note that the way in which we employ discounting is *time consistent*: the agent does not change its mind about how much it values the reward at time step k over time: reward r_k is always discounted with γ_k regardless of the current time step. For a discussion of general discounting we refer the reader to Lattimore and Hutter (2014).

Definition 4.4 (Effective Horizon)**.** The *ε-effective horizon* $H_t(\varepsilon)$ is a horizon that is long enough to encompass all but an ε of the discount function's mass:

$$H_t(\varepsilon) := \min\{k \mid \Gamma_{t+k}/\Gamma_t \leq \varepsilon\}$$

The effective horizon is *bounded* iff for all $\varepsilon > 0$ there is a constant c_ε such that $H_t(\varepsilon) \leq c_\varepsilon$ for all $t \in \mathbb{N}$.

	parameter	γ_t	Γ_t	$H_t(\varepsilon)$
Finite horizon	$m \in \mathbb{N}$	$\mathbb{1}_{\leq m}(t)/m$	$(m-t+1)/m$	$\lceil (m-t+1)(1-\varepsilon) \rceil$
Geometric	$\gamma \in (0,1)$	γ^t	$\gamma^t/(1-\gamma)$	$\lceil \log_\gamma \varepsilon \rceil$
Power	$\beta > 1$	$t^{-\beta}$	$\approx t^{-\beta+1}/(\beta-1)$	$\approx (\varepsilon^{1/(1-\beta)} - 1)t$
Subgeometric	-	$e^{-\sqrt{t}}/\sqrt{t}$	$\approx 2e^{-\sqrt{t}}$	$\approx -\sqrt{t}\log\varepsilon + (\log\varepsilon)^2$

Table 4.1: Several discount functions and their effective horizons. See also Hutter (2005, Tab. 5.41) and Lattimore (2013, Tab. 2.1).

Example 4.5 (Geometric Discounting)**.** The most common discount function is *geometric discounting* with $\gamma_t := \gamma^t$ for some constant $\gamma \in [0,1)$. We get that $\Gamma_t = \sum_{k=t}^{\infty} \gamma^k = \gamma^t/(1-\gamma)$ and the ε-effective horizon is $H_t(\varepsilon) = \lceil \log_\gamma \varepsilon \rceil$. Hence the effective horizon is bounded. ◇

More examples for discount functions are given in Table 4.1. From now on, we fix a discount function γ.

4.1.2 Implicit Assumptions

Throughout this thesis, we make the following assumptions implicitly.

Assumption 4.6. *(a) The discount function γ is computable.*

(b) Rewards are bounded between 0 *and* 1*.*

(c) The set of actions $\mathcal{A}$ and the set of percepts $\mathcal{E}$ are both finite.

Let's motivate these assumptions in turn. Their purpose is to ensure that discounted reward sums are finite and optimal policies exist.

Assumption 4.6a is a technical assumption that ensures that discounted reward sums are computable. This is important for Chapter 6 and Chapter 7 where we analyse the computability of optimal policies. Note that all discount functions given in Table 4.1 are computable.

Assumption 4.6b could be relaxed to require only that rewards are bounded. We can rescale rewards $r_t \mapsto cr_t + d$ for any $c \in \mathbb{R}^+$ and $d \in \mathbb{R}$ without changing optimal policies if the environment ν is a probability measure. (For our computability-related results in Chapter 6, we must assume that rewards are nonnegative.) In this sense Assumption 4.6b is not very restrictive. However, this normalization of rewards into the $[0,1]$-interval has the convenient consequence that the normalized discounted reward sum $\sum_{k=t}^{\infty} \gamma_k r_k/\Gamma_k$ is bounded between 0 and 1. If rewards are unbounded, then the discounted reward sum might diverge. Moreover, with unbounded rewards there all kinds of pathological problems where defining optimal actions is no longer straightforward; see Arntzenius et al. (2004) for a discussion.

Assumption 4.6c is a technical requirement for the existence of optimal policies since it implies that there are only finitely many deterministic policies that differ in the first t

time steps. Note that finite action and percept spaces are very natural since it ensures that our agent only receives and emits a finite amount of information in every time step. This is in line with the problems a strong AI is facing: the agent has to remember important information and act sequentially.

Assumption 4.6b, Assumption 4.6c, and the fact that the discount function is summable guarantee that a deterministic optimal policy exists for every environment according to Lattimore and Hutter (2014, Thm. 10). It would be interesting to relax these assumptions while preserving the existence of optimal policies or at least ε-optimal policies (e.g. use compact action and percept spaces).

4.1.3 Typical Environment Classes

The simplest reinforcement learning problems are multi-armed bandits.

Definition 4.7 (Multi-Armed Bandit)**.** An environment ν is a *multi-armed bandit* iff $\mathcal{O} = \{\bot\}$ and $\nu(e_t \mid æ_{<t}a_t) = \nu(e_t \mid a_t)$ for all histories $æ_{1:t} \in (\mathcal{A} \times \mathcal{E})^*$.

In a multi-armed bandit problem there are no observations and the next reward only depends on the previous action. Intuitively, we are deciding between $\#\mathcal{A}$ different slot machines (so-called one-armed bandits), pull the lever and obtain a reward. The reward is stochastic, but it is drawn from a distribution that is time-invariant and fixed for each arm.

A multi-armed bandit is also called *bandit* for short. Although bandits are the simplest reinforcement learning problem, they already exhibit the exploration-exploitation-tradeoff that makes reinforcement learning difficult: do you pull an arm that has the best empirical mean or do you pull an arm that has the highest uncertainty? In bandits it is very easy to come up with policies that perform (close to) optimal asymptotically (e.g., ε_t-greedy with $\varepsilon_t = 1/t$). But coming up with algorithms that perform well in practice is difficult, and research focuses on the multiplicative and additive constants on the asymptotic guarantees. Bandits exist in many flavors; see Bubeck and Bianchi (2012) for a survey.

Definition 4.8 (Markov Decision Process)**.** An environment ν is a *Markov decision process* (MDP) iff $\nu(e_t \mid æ_{<t}a_t) = \nu(e_t \mid o_{t-1}a_t)$ for all histories $æ_{1:t} \in (\mathcal{A} \times \mathcal{E})^*$.

Intuitively, in MDPs, the previous observation o_{t-1} provides a sufficient statistic for the history: given o_{t-1} and the current action a_t, the next percept e_t is independent of the rest of the history. In other words, everything that the agent needs to know to make optimal decisions is readily available in the previous percept. This is why observations are called *states* in MDPs. Note that bandits are MDPs with a single state.

Much of today's literature on reinforcement learning focuses on MDPs (Sutton and Barto, 1998). They provide a particularly good framework to study reinforcement learning because they are simple enough to be tractable for today's algorithms, yet general enough to encompass many interesting problems. For example, most of the Atari games (see Figure 1.1 for an overview) are (deterministic) MDPs when combining

the previous four frames into one percept. While they have a huge state space[2] they can still be learned using Q-learning with function approximation (Mnih et al., 2015).

The MDP framework is restrictive because it requires the agent to be more powerful than the environment. Since the agent learns, its actions are not independent of the rest of the history given the last action and percept. In other words, learning agents are not Markov. The following definition lifts this restriction and allows the environment to be *partially observable*.

Definition 4.9 (Partially Observable Markov Decision Process)**.** An environment ν is a *partially observable Markov decision process* (POMDP) iff there is a *set of states* $\mathcal{S}$, an *initial state* $s_0 \in \mathcal{S}$, a *state transition function* $\nu' : \mathcal{S} \times \mathcal{A} \to \Delta\mathcal{S}$, and a *percept distribution* $\nu'' : \mathcal{S} \to \Delta\mathcal{E}$ such that

$$\nu(e_{1:t} \parallel a_{1:t}) = \prod_{k=1}^{t} \nu''(e_k \mid s_k)\nu'(s_k \mid s_{k-1}, a_k).$$

Usually the set $\mathcal{S}$ is assumed to be finite; with infinite-state POMDPs we can model any environment ν by setting the set of states to be the set of histories, $\mathcal{S} := (\mathcal{A} \times \mathcal{E})^*$.

A common assumption for MDPs and POMDPs is that they do not contain traps. Formally, a (PO)MDP is *ergodic* iff for any policy π and any two states $s_1, s_2 \in \mathcal{S}$, the expected number of time steps to reach s_2 from s_1 is μ^π-almost surely finite. A (PO)MDP is *weakly communicating* iff for any two states $s_1, s_2 \in \mathcal{S}$ there is a policy π such that the expected number of time steps to reach s_2 from s_1 is μ^π-almost surely finite. Note that any ergodic (PO)MDP is also weakly communicating, but not vice versa.

In general, our environments are stochastic. Stochasticity can originate from noise in the environment, noise in the sensors, or modeling errors. Sometimes we also consider classes of deterministic environments. These are usually easier to deal with because they do not require as much mathematical machinery. For example, in a deterministic environment the next percept is certain; if a different percept is received this environment is immediately falsified and can be discarded. In a stochastic environment, an unlikely percept reduces our posterior belief in this environment but does not rule it out completely.

In Chapter 6 and Chapter 7 we make the assumption that the environment is computable. This encompasses all finite-state POMDPs and most if not all AI problems can be formulated in this setting. Moreover, the current theories of quantum mechanics and general relativity are computable and there is no evidence that suggests that our physical universe is incomputable. For any physical system of finite volume and finite (average) energy, the amount of information it can contain is finite (Bekenstein, 1981), and so is the number of state transitions per unit of time (Margolus and Levitin, 1998). This gives us reason to believe that even the environment that we humans currently face (and will ever face) falls under these assumptions.

[2]The size of the state space is at most 256^{128} since the Atari 2600 has only 128 bytes of memory. However, the vast majority of these states are not reachable.

Formally we define the set $\mathcal{M}_{\text{LSC}}^{\text{CCS}}$ as the set of environments that are lower semicomputable chronological contextual semimeasures and $\mathcal{M}_{\text{comp}}^{\text{CCM}}$ as the set of environments that are computable chronological contextual measures. Note that for chronological contextual semimeasures it makes a difference whether $\nu(\cdot \parallel a_{1:\infty})$ is lower semicomputable or the conditionals $\nu(\cdot \mid æ_{<t}a_t)$ are. The latter implies the former, but not vice versa.

4.2 The Value Function

The *value* of a policy in an environment is the future expected discounted reward when following a given policy in a given environment conditional on the past. Since this quantity captures exactly what our agent aims to maximize, we prefer policies whose value is high.

Definition 4.10 (Value Function)**.** The *value* of a policy π in an environment ν given history $æ_{<t}$ and *horizon* m with $t \leq m \leq \infty$ is defined as

$$V_\nu^{\pi,m}(æ_{<t}) := \frac{1}{\Gamma_t}\mathbb{E}_\nu^\pi\left[\sum_{k=t}^{m-1}\gamma_k r_k \;\middle|\; æ_{<t}\right]$$

if $\Gamma_t > 0$ and $V_\nu^{\pi,m}(æ_{<t}) := 0$ if $\Gamma_t = 0$. The *optimal value* is defined as $V_\nu^{*,m}(æ_{<t}) := \sup_\pi V_\nu^{\pi,m}(æ_{<t})$.

Sometimes we omit the history argument $æ_{<t}$ for notational convenience if it is clear from context. Moreover, when we omit m, we implicitly use an infinite horizon $m = \infty$, i.e., $V_\nu^\pi := V_\nu^{\pi,\infty}$ and $V_\nu^* := V_\nu^{*,\infty}$. The value of a policy π in an environment ν after the empty history, $V_\nu^\pi(\epsilon)$ is also called the t_0*-value*.

Remark 4.11 (Values are Bounded Between 0 and 1)**.** From Assumption 4.6b we get that for all histories $æ_{<t}$ all policies π and all environments ν, the value function $V_\nu^\pi(æ_{<t}) \in [0, 1]$. ◇

Since environment and policy are stochastic, the history $æ_{<t}$ is random. With abuse of notation we treat $æ_{<t}$ sometimes as a concrete outcome and sometimes as a random variable. We also view the value of a policy π in an environment ν as a sequence of random variables $(X_t)_{t\in\mathbb{N}}$ with $X_t := V_\nu^\pi(æ_{1:t})$ where the history $æ_{1:t}$ is generated stochastically by the agent's actual policy interacting with the true environment μ. This view is helpful for some of the convergence results (e.g., Theorem 4.19 and Definition 5.18) in which we talk about the type of convergence of this sequence of random variables.

The value function defined in Definition 4.10 is also called the *recursive value function*, in contrast to iterative value function that we discuss in Section 6.4. The name of the recursive value function originates from the following recursive identity (analogously

to Hutter, 2005, Eq. 4.12), also called the *Bellman equation*:

$$V_\nu^\pi(æ_{<t}) = \sum_{a_t \in \mathcal{A}} \pi(a_t \mid æ_{<t}) V_\nu^\pi(æ_{<t}a_t)$$
$$V_\nu^\pi(æ_{<t}a_t) = \frac{1}{\Gamma_t} \sum_{e_t \in \mathcal{E}} \nu(e_t \mid æ_{<t}a_t)\big(\gamma_t r_t + \Gamma_{t+1} V_\nu^\pi(æ_{1:t})\big)$$

An explicit expression for the optimal value in environment ν is

$$V_\nu^{*,m}(æ_{<t}) = \frac{1}{\Gamma_t} \max_{æ_{t:m-1}} \sum_{k=t}^{m-1} \gamma_k r_k \prod_{i=t}^{k} \nu(e_i \mid æ_{<i}a_i), \tag{4.2}$$

where $\max$ denotes the max-sum-operator:

$$\max_{æ_{t:m-1}} := \max_{a_t \in \mathcal{A}} \sum_{e_t \in \mathcal{E}} \dots \max_{a_{m-1} \in \mathcal{A}} \sum_{e_{m-1} \in \mathcal{E}}$$

For an explicit expression of $V_\nu^{*,\infty}(æ_{<t})$ we can simply take the limit $m \to \infty$.

4.2.1 Optimal Policies

An optimal policy is a policy that achieves the highest value:

Definition 4.12 (Optimal Policy; Hutter, 2005, Def. 5.19 & 5.30)**.** A policy π is *optimal in environment ν (ν-optimal)* iff for all histories π attains the optimal value: $V_\nu^\pi(æ_{<t}) = V_\nu^*(æ_{<t})$ for all $æ_{<t} \in (\mathcal{A} \times \mathcal{E})^*$. The action a_t is an *optimal action* iff $\pi_\nu^*(a_t \mid æ_{<t}) = 1$ for some ν-optimal policy π_ν^*.

Following the tradition of Hutter (2005), *AINU* denotes a ν-optimal policy for the environment $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ and *AIMU* denotes an μ-optimal policy for the environment $\mu \in \mathcal{M}_{\text{comp}}^{\text{CCM}}$ that is a measure (as opposed to a semimeasure).

By definition of the optimal policy and the optimal value function, we have the following identity for all histories $æ_{<t}$:

$$V_\nu^*(æ_{<t}) = V_\nu^{\pi_\nu^*}(æ_{<t}) \tag{4.3}$$

There can be more than one optimal policy; generally the choice of π_ν^* from Definition 4.12 is not unique. More specifically, for a ν-optimal policy we have

$$\pi_\nu^*(a_t \mid æ_{<t}) > 0 \implies a_t \in \arg\max_{a \in \mathcal{A}} V_\nu^*(æ_{<t}a). \tag{4.4}$$

If there are multiple actions $\alpha, \beta \in \mathcal{A}$ that attain the optimal value, $V_\nu^*(æ_{<}\alpha) = V_\nu^*(æ_{<t}\beta)$, then there is an *argmax tie*. Which action we settle on in case of a tie (how we break the tie) is irrelevant and can be arbitrary. Since we allow stochastic policies, we can also randomize between α and β.

The following definition allows policies to be slightly suboptimal.

Definition 4.13 (ε-Optimal Policy)**.** A policy π is *ε-optimal in environment ν* iff $V^*_\nu(æ_{<t}) - V^\pi_\nu(æ_{<t}) < \varepsilon$ for all histories $æ_{<t} \in (\mathcal{A} \times \mathcal{E})^*$.

A policy π that achieves optimal t_0-value, $V^\pi_\nu(\epsilon) = V^*_\nu(\epsilon)$, takes ν-optimal actions on any history reachable by π in ν. However, this is not true for ε-optimal policies: a policy that is ε-optimal at $t = 0$ is not necessarily ε-optimal in later time steps.

4.2.2 Properties of the Value Function

The following two lemmas are stated by Hutter (2005, Thm. 31) without proof and for the iterative value function.

Lemma 4.14 (Linearity of V^π_ν in ν)**.** *If $\nu = z_1\nu_1 + z_2\nu_2$ for some real numbers $z_1, z_2 \geq 0$, then for all policies π and all histories $æ_{<t}$*

$$V^{\pi,m}_\nu(æ_{<t}) \;=\; z_1 \frac{\nu_1(e_{<t} \parallel a_{<t})}{\nu(e_{<t} \parallel a_{<t})} V^{\pi,m}_{\nu_1}(æ_{<t}) + z_2 \frac{\nu_2(e_{<t} \parallel a_{<t})}{\nu(e_{<t} \parallel a_{<t})} V^{\pi,m}_{\nu_2}(æ_{<t}).$$

Proof. Since $\nu^\pi = z_1\nu^\pi_1 + z_2\nu^\pi_2$, we have for the conditional measure

$$\begin{aligned}\nu^\pi(A \mid æ_{<t}) = \frac{\nu^\pi(A \cap æ_{<t})}{\nu^\pi(æ_{<t})} &= \frac{z_1\nu^\pi_1(A \cap æ_{<t}) + z_2\nu^\pi_2(A \cap æ_{<t})}{\nu^\pi(æ_{<t})} \\ &= z_1 \frac{\nu^\pi_1(æ_{<t})}{\nu^\pi(æ_{<t})} \nu^\pi_1(A \mid æ_{<t}) + z_2 \frac{\nu^\pi_2(æ_{<t})}{\nu^\pi(æ_{<t})} \nu^\pi_2(A \mid æ_{<t}).\end{aligned}$$

The claim now follows from the linearity of expectation in the probability measure. □

Lemma 4.15 (Convexity of V^*_ν in ν)**.** *If $\nu = z_1\nu_1 + z_2\nu_2$ for some real numbers $z_1, z_2 \geq 0$, then for all histories $æ_{<t}$*

$$V^*_\nu(æ_{<t}) \;\leq\; z_1 \frac{\nu_1(e_{<t} \parallel a_{<t})}{\nu(e_{<t} \parallel a_{<t})} V^{*,m}_{\nu_1}(æ_{<t}) + z_2 \frac{\nu_2(e_{<t} \parallel a_{<t})}{\nu(e_{<t} \parallel a_{<t})} V^{*,m}_{\nu_2}(æ_{<t}).$$

Proof. Let π^*_ν be an optimal policy for environment ν. From Lemma 4.14 we get

$$\begin{aligned}V^*_\nu(æ_{<t}) = V^{\pi^*_\nu}_\nu(æ_{<t}) &= z_1 \frac{\nu_1(e_{<t} \parallel a_{<t})}{\nu(e_{<t} \parallel a_{<t})} V^{\pi^*_\nu}_{\nu_1}(æ_{<t}) + z_2 \frac{\nu_2(e_{<t} \parallel a_{<t})}{\nu(e_{<t} \parallel a_{<t})} V^{\pi^*_\nu}_{\nu_2}(æ_{<t}) \\ &\leq z_1 \frac{\nu_1(e_{<t} \parallel a_{<t})}{\nu(e_{<t} \parallel a_{<t})} V^*_{\nu_1}(æ_{<t}) + z_2 \frac{\nu_2(e_{<t} \parallel a_{<t})}{\nu(e_{<t} \parallel a_{<t})} V^*_{\nu_2}(æ_{<t}).\end{aligned}$$ □

The following lemma bounds the error when truncating the value function. This implies that planning for an ε-effective horizon ($m = t + H_t(\varepsilon)$), we get all but an ε of the value: $|V^\pi_\nu(æ_{<t}) - V^{\pi,m}_\nu(æ_{<t})| < \varepsilon$.

Lemma 4.16 (Truncated Values)**.** *For every environment ν, every policy π, and every history $æ_{<t}$*

$$\left|V^{\pi,m}_\nu(æ_{<t}) - V^\pi_\nu(æ_{<t})\right| \;\leq\; \frac{\Gamma_m}{\Gamma_t}.$$

Proof.

$$V_\nu^\pi(æ_{<t}) = \frac{1}{\Gamma_t}\mathbb{E}_\nu^\pi\left[\sum_{k=t}^{\infty}\gamma_k r_k \;\middle|\; æ_{<t}\right] = V_\nu^{\pi,m}(æ_{<t}) + \frac{1}{\Gamma_t}\mathbb{E}_\nu^\pi\left[\sum_{k=m}^{\infty}\gamma_k r_k \;\middle|\; æ_{<t}\right]$$

The result now follows from Assumption 4.6b and

$$0 \leq \mathbb{E}_\nu^\pi\left[\sum_{k=m}^{\infty}\gamma_k r_k \;\middle|\; æ_{<t}\right] \leq \Gamma_m. \qquad \square$$

This lemma bounds the (truncated) value function by the total variation distance.

Lemma 4.17 (Bounds on Value Difference)**.** *For any policies π_1, π_2, any environments ν_1 and ν_2, and any horizon $t \leq m \leq \infty$*

$$|V_{\nu_1}^{\pi_1,m}(æ_{<t}) - V_{\nu_2}^{\pi_2,m}(æ_{<t})| \;\leq\; D_{m-1}(\nu_1^{\pi_1}, \nu_2^{\pi_2} \mid æ_{<t})$$

Proof. According to Definition 4.10, the value function is the expectation of the random variable $\sum_{k=t}^{m-1}\gamma_k r_k/\Gamma_t$ that is bounded between 0 and 1. Therefore we can use Lemma 2.12 with $P := \nu_1^{\pi_1}(\cdot \mid æ_{<t})$ and $R := \nu_2^{\pi_2}(\cdot \mid æ_{<t})$ on the space $(\mathcal{A}\times\mathcal{E})^{m-1}$ to conclude that $|V_{\nu_1}^{\pi_1,m}(æ_{<t}) - V_{\nu_2}^{\pi_2,m}(æ_{<t})|$ is bounded by $D_{m-1}(\nu_1^{\pi_1}, \nu_2^{\pi_2} \mid æ_{<t})$. $\square$

Lemma 4.18 (Discounted Values; Lattimore, 2013, Lem. 2.5)**.** *Let $æ_{<t}$ be some history and let π_1 and π_2 be two policies that coincide from time step t to time step m: $\pi_1(a \mid æ_{1:k}) = \pi_2(a \mid æ_{1:k})$ for all $a \in \mathcal{A}$, all histories $æ_{<t}æ_{t:k}$ consistent with π_1, and $t \leq k \leq m$. Then for all environments ν*

$$\left|V_\nu^{\pi_1}(æ_{<t}) - V_\nu^{\pi_2}(æ_{<t})\right| \;\leq\; \frac{\Gamma_m}{\Gamma_t}.$$

Proof. Since π_1 and π_2 coincide for time steps t through $m-1$, $D_{m-1}(\nu^{\pi_1}, \nu^{\pi_2} \mid æ_{<t}) = 0$ for all environments ν. Thus the result follows from Lemma 4.16 and Lemma 4.17:

$$\begin{aligned}\left|V_\nu^{\pi_1}(æ_{<t}) - V_\nu^{\pi_2}(æ_{<t})\right| &\leq \left|V_\nu^{\pi_1,m}(æ_{<t}) - V_\nu^{\pi_2,m}(æ_{<t})\right| + \frac{\Gamma_m}{\Gamma_t}\\ &\leq D_{m-1}(\nu^{\pi_1}, \nu^{\pi_2} \mid æ_{<t}) + \frac{\Gamma_m}{\Gamma_t}\\ &= \frac{\Gamma_m}{\Gamma_t}\end{aligned} \qquad \square$$

4.2.3 On-Policy Value Convergence

This section states some general results on learning the value function. *On-policy value convergence* refers to the fact that if we use a learning distribution ρ to learn to environment μ, and ρ^π merges with μ^π in the sense discussed Section 3.4, then V_ρ^π converges to V_μ^π, i.e., using ρ we learn to estimate values correctly.

A weaker variant of the following theorem was proved by Hutter (2005, Thm. 5.36). It states convergence in mean (not almost surely), and only for the Bayesian mixture.

Theorem 4.19 (On-Policy Value Convergence)**.** *Let μ be any environment and π be any policy.*

(a) If ρ^π merges strongly with μ^π, then

$$V^\pi_\rho(æ_{<t}) - V^\pi_\mu(æ_{<t}) \to 0 \text{ as } t \to \infty \ \mu^\pi\text{-almost surely.}$$

(b) If the effective horizon is bounded and ρ^π merges weakly with μ^π, then

$$V^\pi_\rho(æ_{<t}) - V^\pi_\mu(æ_{<t}) \to 0 \text{ as } t \to \infty \ \mu^\pi\text{-almost surely.}$$

(c) If the effective horizon is bounded and ρ^π merges almost weakly with μ^π, then

$$\frac{1}{t}\sum_{k=1}^{t}\left(V^\pi_\rho(æ_{<k}) - V^\pi_\mu(æ_{<k})\right) \to 0 \text{ as } t \to \infty \text{ in } \mu^\pi\text{-almost surely.}$$

Proof. (a) Apply Lemma 4.17 with $m := \infty$.

(b) Let $\varepsilon > 0$ and let c_ε be a bound on $\sup_t H_t(\varepsilon)$. From Lemma 4.16

$$\begin{aligned}|V^\pi_\rho(æ_{<t}) - V^\pi_\mu(æ_{<t})| &\le |V^{\pi,t+H_t(\varepsilon)}_\rho(æ_{<t}) - V^{\pi,t+H_t(\varepsilon)}_\mu(æ_{<t})| + 2\frac{\Gamma_{t+H_t(\varepsilon)}}{\Gamma_t}\\ &< D_{t+H_t-1(\varepsilon)}(\rho^\pi, \mu^\pi \mid æ_{<t}) + 2\varepsilon\\ &\le D_{t+c_\varepsilon}(\rho^\pi, \mu^\pi \mid æ_{<t}) + 2\varepsilon\end{aligned}$$

according to Definition 4.4 and Lemma 4.17. Since ρ^π merges weakly with μ^π, we get that μ^π-almost surely there is a time step $t_0 \in \mathbb{N}$ such that $D_{t+c_\varepsilon}(\rho^\pi, \mu^\pi \mid æ_{<t}) < \varepsilon$ for all $t \ge t_0$. Hence $|V^\pi_Q(æ_{<t}) - V^\pi_\mu(æ_{<t})| < 3\varepsilon$ for all $t \ge t_0$.

(c) Analogously to the proof of (b). □

It is important to observe that on-policy convergence does not imply that the agent converges to the optimal policy. V^π_ρ converges to V^π_μ, but V^π_μ need not be close to V^*_μ. Indeed, there might be another policy $\tilde\pi$ that has a higher value than π in the true environment μ ($V^{\tilde\pi}_\mu > V^\pi_\mu$). If the agent thinks $\tilde\pi$ has lower value ($V^{\tilde\pi}_\rho < V^\pi_\rho$) it might not follow $\tilde\pi$ and hence not learn that the actual value of $\tilde\pi$ is much higher. In other words, on-policy convergence implies that the agent learns the value of its own actions, but not the value of counterfactual actions that it does not take.

Theorem 4.19 now enables us to tie in the results of Chapter 3. This yields a surge of corollaries, but first we need to make the learning distributions contextual on the actions.

Let $w \in \Delta\mathcal{M}$ be a positive prior over the environment class $\mathcal{M}$. We define the corresponding Bayesian mixture analogously to Example 3.4:

$$\xi(e_{<t} \parallel a_{<t}) := \sum_{\nu\in\mathcal{M}} w(\nu)\nu(e_{<t} \parallel a_{<t}) \tag{4.5}$$

Note that the Bayesian mixture ξ depends on the prior w. For the rest of this thesis, this dependence will not be made explicit.

From Lemma 4.14 and (3.3) we immediately get the following identity:

$$V_\xi^\pi(æ_{<t}) = \sum_{\nu\in\mathcal{M}} w(\nu \mid æ_{<t}) V_\nu^\pi(æ_{<t}) \tag{4.6}$$

Similarly, we get from Lemma 4.15

$$V_\xi^*(æ_{<t}) \leq \sum_{\nu\in\mathcal{M}} w(\nu \mid æ_{<t}) V_\nu^*(æ_{<t}). \tag{4.7}$$

Corollary 4.20 (On-Policy Value Convergence for Bayes)**.** *For any environment $\mu \in \mathcal{M}$ and any policy π,*

$$V_\xi^\pi(æ_{<t}) - V_\mu^\pi(æ_{<t}) \to 0 \text{ as } t \to \infty \ \mu^\pi\text{-almost surely.}$$

Proof. Since $\mu \in \mathcal{M}$, we have dominance $\xi^\pi \geq w(\mu)\mu^\pi$ with $w(\mu) > 0$ and by Proposition 3.16a absolute continuity $\xi^\pi \gg \mu^\pi$. From Theorem 3.25 we get that ξ^π merges strongly with μ^π. Therefore we can apply Theorem 4.19a. □

Analogously, we define $\text{MDL}^{æ_{<t}} := \arg\min_{\nu\in\mathcal{M}}\{-\log \nu(e_{<t} \parallel a_{<t}) + K(\nu)\}$.

Corollary 4.21 (On-Policy Value Convergence for MDL)**.** *For any environment $\mu \in \mathcal{M}$ and any policy π,*

$$V_{\text{MDL}^{æ_{<t}}}^\pi(æ_{<t}) - V_\mu^\pi(æ_{<t}) \to 0 \text{ as } t \to \infty \ \mu^\pi\text{-almost surely.}$$

Proof. By Theorem 3.28 MDL^π merges strongly with ν^π for each $\nu \in \mathcal{M}$, therefore we can apply Theorem 4.19a. □

By providing the action sequence *contextually* on a separate input tape, we can define $Km(e_{<t} \parallel a_{<t}) := \min\{|p| \mid e_{<t} \sqsubseteq U(p, a_{<t})\}$ analogously to (2.1).

Corollary 4.22 (On-Policy Value Convergence for Universal Compression)**.** *Let $\rho(e_{<t} \parallel a_{<t}) := 2^{-Km(e_{<t}\|a_{<t})}$. Then for any environment $\mu \in \mathcal{M}_{\text{comp}}^{\text{CCM}}$ and any policy π,*

$$V_\rho^\pi(æ_{<t}) - V_\mu^\pi(æ_{<t}) \to 0 \text{ as } t \to \infty \ \mu^\pi\text{-almost surely.}$$

Proof. Since ρ dominates every $\mu \in \mathcal{M}_{\text{comp}}^{\text{CCM}}$ (Section 3.6.3) we can apply Proposition 3.16a, Theorem 3.25, and Theorem 4.19a as in the proof of Corollary 4.20. □

Similarly to Km there is a speed prior for environments (Filan, 2015, Ch. 6):

$$S_{Kt}(e_{<t} \parallel a_{<t}) := \sum_{p:\, e_{<t}\sqsubseteq U(p,a_{<t})} \frac{2^{-|p|}}{t(U, p, a_{<t}, e_{<t})}$$

where $t(U, p, a_{<t}, e_{<t})$ denotes the number of time steps $U(p, a_{<t})$ takes to produce $e_{<t}$.

Corollary 4.23 (On-Policy Value Convergence for the Speed Prior)**.** *If the effective horizon is bounded, then for any environment $\mu \in \mathcal{M}^{\text{CCM}}_{\text{comp}}$ estimable in polynomial time and any policy π,*

$$\frac{1}{t}\sum_{k=1}^{t}\left(V^{\pi}_{S_{Kt}}(æ_{<k}) - V^{\pi}_{\mu}(æ_{<k})\right) \to 0 \text{ as } t \to \infty \ \mu^{\pi}\text{-almost surely.}$$

Proof. By Corollary 3.61 the speed prior S_{Kt} merges almost weakly with every measure estimable in polynomial time. Therefore we can apply Theorem 4.19c. □

4.3 The Agents

If we knew the true environment μ, we would choose the μ-optimal policy, the policy that maximizes μ-expected discounted rewards. But generally we do not know the true environment, and the challenging part of reinforcement learning is to learn the environment while trying to collect rewards.

In this section we introduce a number of agents that attempt to solve the general reinforcement learning problem (Problem 4.2). These agents are discussed throughout the rest of this thesis.

4.3.1 Bayes

A *Bayes optimal policy* with respect to the prior w is the policy π^*_ξ where ξ is the Bayesian mixture defined in Section 4.2.3. There can be one or more Bayes optial policies. From Corollary 4.20 we get on-policy value convergence for the Bayes optimal policy.

After history $æ_{<t}$, the Bayes policy π^*_ξ maximizes expected discounted rewards in the posterior mixture:

$$\xi(\cdot \mid e_{<t} \parallel a_{1:\infty}) = \sum_{\nu\in\mathcal{M}} w(\nu \mid æ_{<t})\nu(\cdot \mid e_{<t} \parallel a_{1:\infty})$$

where $w(\nu \mid æ_{<t})$ are the posterior weights (3.3). Maximizing expected rewards according to the posterior is the same as maximizing expected rewards according to the prior conditional on the history: if $\pi(æ_{<t}) = \pi^*_\xi(æ_{<t})$, then $V^{\pi}_{\xi}(æ_{<t}) = V^{*}_{\xi}(æ_{<t})$. Actually visiting the history $æ_{<t}$ does not change what π^*_ξ planned to do before it visited $æ_{<t}$. Note that this relies on the fact that the way we use discounting is time consistent (Lattimore and Hutter, 2014, Def. 12).

When using the prior $w(\nu) \propto 2^{-K(\nu)}$ (Example 3.5) over the class $\mathcal{M}^{\text{CCS}}_{\text{LSC}}$, the Bayes optimal policy is also known as *AIXI*, introduced and analyzed by Hutter (2000, 2001a, 2002a, 2003, 2005, 2007a, 2012b) in his work on *universal artificial intelligence*. In this case, the Bayesian mixture (4.5) can be defined equivalently according to (Wood et al., 2011)

$$\xi(e_{<t} \parallel a_{<t}) := \sum_{p:\, e_{<t}\sqsubseteq U(p,a_{<t})} 2^{-|p|}. \tag{4.8}$$

Generally there is more than one ξ-optimal policy and Solomonoff's prior depends on the choice of the (reference) universal Turing machine, so this definition is not unique. Moreover, not every universal Turing machine is a good choice for AIXI, see Section 5.2 for a few bad choices. The following lemma will be used later.

Lemma 4.24 (Mixing Mixtures)**.** *Let $q, q' \in \mathbb{Q}$ such that $q > 0$, $q' \geq 0$, and $q + q' \leq 1$. Let w be any lower semicomputable positive prior, let ξ be the Bayesian mixture corresponding to w, and let $\rho \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$. Then $\xi' := q\xi + q'\rho \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ is a Bayesian mixture.*

Proof. ξ' is given by the positive prior w' with $w' := qw + q'\mathbb{1}_\rho$. □

Bayesian approaches have a long tradition in reinforcement learning, although they are often prohibitively expensive to compute. For multi-armed bandits, Gittins (1979) achieved a breakthrough with an index strategy that enables the computation of the optimal policy by computing one quantity for each arm independently of the rest. This strategy even achieves the optimal asymptotic regret bounds (Lattimore, 2016). Larger classes have also been attempted: using Monte-Carlo tree search, Veness et al. (2011) approximate the Bayes optimal policy in the class of all context trees. Doshi-Velez (2012) uses Bayesian techniques to learn infinite-state POMDPs. See Vlassis et al. (2012) for a survey on Bayesian techniques in RL.

In the rest of this thesis, the Bayes optimal policy is often treated as an optimal exploitation strategy. This is not true: Bayes does explore (when it is Bayes optimal to do so). It just does not explore general environment classes completely (see Section 5.4.1).

4.3.2 Knowledge-Seeking Agents

In this section we discuss two variants of knowledge-seeking agents: entropy-seeking agents introduced by Orseau (2011, 2014a) and information-seeking agents introduced by Orseau et al. (2013). The entropy-seeking agent maximizes the Shannon entropy gain, while the information-seeking agent maximizes the expected information gain. These quantities are expressed in different value functions. In places where confusion can arise, we call the value function V_ν^π from Definition 4.10 the *reward-seeking value function*.

In this section we use a finite horizon $m < \infty$ (possibly dependent on time step t): the knowledge-seeking agent maximizes entropy/information received up to time step m. We assume implicitly that m (as a function of t) is computable. Moreover, in this section we assume that the Bayesian mixture ξ is a measure rather than a semimeasure; Example 4.27 discusses this assumption.

Definition 4.25 (Entropy-Seeking Value Function; Orseau, 2014a, Sec. 6)**.** The *entropy-seeking value* of a policy π given history $æ_{<t}$ is

$$V_{\text{Ent}}^{\pi,m}(æ_{<t}) := \mathbb{E}_\xi^\pi[-\log_2 \xi(e_{1:m} \mid e_{<t} \parallel a_{1:m}) \mid æ_{<t}].$$

The entropy-seeking value is the Bayes-expectation of $-\log \xi$. Orseau (2011, 2014a) also considers a related value function based on the ξ-expectation of ξ that we do not discuss here.

Definition 4.26 (Information-Seeking Value Function; Orseau et al., 2013, Def. 1)**.** The *information-seeking value* of a policy π given history $æ_{<t}$ is

$$V_{\mathrm{IG}}^{\pi,m}(æ_{<t}) := \sum_{\nu\in\mathcal{M}} w(\nu \mid æ_{<t})\mathrm{KL}_m(\nu^\pi, \xi^\pi \mid æ_{<t}).$$

Analogously to before we define $V_{\mathrm{Ent}}^* := \sup_\pi V_{\mathrm{Ent}}^\pi$ and $V_{\mathrm{IG}}^* := \sup_\pi V_{\mathrm{IG}}^\pi$. An optimal entropy-seeking policy is defined as $\pi_{\mathrm{Ent}}^* :\in \arg\max_\pi V_{\mathrm{Ent}}^\pi$ and an optimal information-seeking policy is defined as $\pi_{\mathrm{IG}}^* :\in \arg\max_\pi V_{\mathrm{IG}}^\pi$. Since we use a finite horizon ($m < \infty$), these optimal policies exist.

The *information gain* is defined as the difference in entropy between the prior and the posterior:

$$\mathrm{IG}_{t:m}(æ_{1:m}) := \mathrm{Ent}(w(\cdot \mid æ_{<t})) - \mathrm{Ent}(w(\cdot \mid æ_{1:m}))$$

We get the following identity (Lattimore, 2013, Eq. 3.5).

$$\mathbb{E}_\xi^\pi[\mathrm{IG}_{t:m}(æ_{1:m}) \mid æ_{<t}] \;=\; V_{\mathrm{IG}}^{\pi,m}(æ_{<t})$$

For infinite horizons ($m = \infty$), the values functions from Definition 4.25 and Definition 4.26 may not converge. To ensure convergence, we can either use discounting, or in case of V_{IG} a prior with finite entropy (Lattimore, 2013, Thm. 3.4). Moreover, note that while V_{IG} and V_{Ent} are expectations with respect to the measure ξ, there is no bound on the one-step change in value $V_{\mathrm{IG}}^{\pi,m}(æ_{<t}) - V_{\mathrm{IG}}^{\pi,m}(æ_{1:t})$, which can also be negative. For the reward-seeking value function $V_\nu^{\pi,m}$, the one-step change in value is bounded between 0 and 1 by Remark 4.11.

For classes of deterministic environments Definition 4.25 and Definition 4.26 coincide. In stochastic environments the entropy-seeking agent does not work well because it gets distracted by noise in the environment rather than trying to distinguish environments (Orseau et al., 2013, Sec. 5). Moreover, the entropy-seeking agent may fail to seek knowledge in deterministic semimeasures as the following example demonstrates.

Example 4.27 (Unnormalized Entropy-Seeking)**.** If the Bayesian mixture ξ is a semimeasure instead of a measure (such as the Solomonoff prior from Example 3.5), then the entropy-seeking agent does not explore correctly. Fix $\mathcal{A} := \{\alpha, \beta\}$, $\mathcal{E} := \{0, 1\}$, and $m = t$ (we only care about the entropy of the next percept). We illustrate the problem on a simple class of environments $\{\nu_1, \nu_2\}$:

where transitions are labeled with action/percept/probability. Both ν_1 and ν_2 return a percept deterministically or nothing at all (the environment ends). Only action α

distinguishes between the environments. With the prior $w(\nu_1) := w(\nu_2) := 1/2$, we get a mixture ξ for the entropy-seeking value function V^{π}_{Ent}. Then $V^{*}_{\text{Ent}}(\alpha) \approx 0.432 < 0.5 = V^{*}_{\text{Ent}}(\beta)$, hence action β is preferred over α by the entropy-seeking agent. But taking action β yields percept 0 (if any), hence nothing is learned about the environment. $\diamond$

On-policy value convergence (Theorem 4.19) ensures that asymptotically, the agent learns the value of its own policy. Knowledge-seeking agents do even better: they don't have to balance between exploration and exploitation, so they can focus solely on exploration. As a result, they learn off-policy, i.e., the value of counterfactual actions (Orseau et al., 2013, Thm. 7).

4.3.3 BayesExp

Lattimore (2013, Thm. 5.6) defines *BayesExp* combining AIXI with the information-seeking agent. BayesExp alternates between phases of exploration and phases of exploitation: Let ε_t be a monotone decreasing sequence of positive reals such that $\varepsilon_t \to 0$ as $t \to \infty$. If the optimal information-seeking value V^{*}_{IG} is larger than ε_t, then BayesExp starts an exploration phase, otherwise it starts an exploitation phase. During an exploration phase, BayesExp follows an optimal information-seeking policy for an ε_t-effective horizon. During an exploitation phase, BayesExp follows an ξ-optimal reward-seeking policy for one step (see Algorithm 1).

Algorithm 1 BayesExp policy π_{BE} (Lattimore, 2013, Alg. 2).

while true **do**
  if $V^{*,t+H_t(\varepsilon_t)}_{\text{IG}}(æ_{<t}) > \varepsilon_t$ **then**
    follow π^{*}_{IG} for $H_t(\varepsilon_t)$ steps
  else
    follow π^{*}_{ξ} for 1 step

4.3.4 Thompson Sampling

Thompson sampling, also known as *posterior sampling* or *the Bayesian control rule*, was originally proposed by Thompson (1933) as a bandit algorithm. It is easy to implement and often achieves quite good results (Chapelle and Li, 2011). In multi-armed bandits it attains optimal regret (Agrawal and Goyal, 2011; Kaufmann et al., 2012). Thompson sampling has also been discussed for MDPs (Strens, 2000; Dearden et al., 1998) and Bayesian and frequentist regret bounds have been established (Osband et al., 2013; Gopalan and Mannor, 2015).

For general RL Thompson sampling was first suggested by Ortega and Braun (2010) with resampling at every time step. Strens (2000) proposes following the optimal policy for one episode or "related to the number of state transitions the agent is likely to need to plan ahead". We follow Strens' suggestion and resample at the effective horizon.

Let ε_t be a monotone decreasing sequence of positive reals such that $\varepsilon_t \to 0$ as $t \to \infty$. Our variant of Thomson sampling is given in Algorithm 2. It samples an

environment ρ from the posterior, follows the ρ-optimal policy for an ε_t-effective horizon, and then repeats.

Algorithm 2 Thompson sampling policy π_T.

1: **while** true **do**
2: sample $\rho \sim w(\cdot \mid æ_{<t})$
3: follow π^*_ρ for $H_t(\varepsilon_t)$ steps

Note that π_T is a stochastic policy since we occasionally sample from a distribution. We assume that this sampling is independent of everything else.

Chapter 5

Optimality

Machines will never be intelligent. — Shane Legg

Problem 4.2 defines the general reinforcement learning problem. But our definition of this problem did not specify what a solution would be. This chapter is dedicated to this question:

What is an optimal solution to the general reinforcement learning problem?

How can we say that one policy is *better* than another? What is the *best* policy? Are the policies from Section 4.3 optimal? Several notions of optimality for a policy π in an environment class $\mathcal{M}$ are conceivable:

O1. *Maximal reward.* The policy π receives a reward of 1 in every time step (which is maximal according to Assumption 4.6b):

$$\forall t \in \mathbb{N}.\ r_t = 1$$

O2. *Optimal policy.* The policy π achieves the highest possible value in the true environment μ:

$$\forall æ_{<t} \in (\mathcal{A} \times \mathcal{E})^*.\ V_\mu^\pi(æ_{<t}) = V_\mu^*(æ_{<t})$$

O3. *Pareto optimality* (Hutter, 2002a, Thm. 2). There is no other policy that performs at least as good in all environments and strictly better in at least one:

$$\nexists \tilde{\pi}.\ \Big(\forall \nu \in \mathcal{M}.\ V_\nu^{\tilde{\pi}}(\epsilon) \geq V_\nu^\pi(\epsilon) \text{ and } \exists \rho \in \mathcal{M}.\ V_\rho^{\tilde{\pi}}(\epsilon) > V_\rho^\pi(\epsilon)\Big)$$

O4. *Balanced Pareto optimality* (Hutter, 2002a, Thm. 3). The policy π achieves a better value across $\mathcal{M}$ weighted by $w \in \Delta\mathcal{M}$ than any other policy:

$$\forall \tilde{\pi}.\ \sum_{\nu \in \mathcal{M}} w(\nu)\big(V_\nu^\pi(\epsilon) - V_\nu^{\tilde{\pi}}(\epsilon)\big) \geq 0$$

O5. *Bayes optimality.* The policy π is ξ-optimal for some Bayes mixture ξ:

$$\forall æ_{<t} \in (\mathcal{A} \times \mathcal{E})^*.\ V_\xi^\pi(æ_{<t}) = V_\xi^*(æ_{<t})$$

O6. *Probably approximately correct*. For a given $\varepsilon, \delta > 0$ the value of the policy π is ε-close to the optimal value with probability at least δ after time step $t_0(\varepsilon, \delta)$:

$$\mu^\pi \left[\forall t \geq t_0(\varepsilon, \delta).\ V^*_\mu(æ_{<t}) - V^\pi_\mu(æ_{<t}) < \varepsilon \right] > 1 - \delta$$

O7. *Asymptotic optimality* (Hutter, 2005, Sec. 5.3.4). The value of the policy π converges to the optimal value:

$$V^*_\mu(æ_{<t}) - V^\pi_\mu(æ_{<t}) \to 0 \text{ as } t \to \infty$$

O8. *Sublinear regret*. The difference between the reward sum of the policy π and the best policy in hindsight grows sublinearly:

$$\sup_{\pi'} \mathbb{E}^{\pi'}_\mu \left[\sum_{t=1}^{m} r_t \right] - \mathbb{E}^{\pi}_\mu \left[\sum_{t=1}^{m} r_t \right] \in o(m)$$

We discuss these notions of optimality in turn. Achieving the maximal reward at every time step is impossible if there is no action that makes the environment μ respond with the maximal reward; generally there is no policy that achieves maximal rewards at every time step. In order to follow the optimal policy, we need to know the true environment. In our setting, the true environment is unknown and has to be learned. During the learning process the agent cannot also act optimally because it needs to explore. In particular, the policy π cannot be optimal simultaneously in all environments from $\mathcal{M}$. This rules out O1 and O2 as a notion of optimality.

In Section 5.1 we show that all policies are Pareto optimal. This disqualifies O3 as a useful notion of optimality in general reinforcement learning.

Balanced Pareto optimality (O4), Bayes optimality (O5), and maximal Legg-Hutter intelligence (Legg and Hutter, 2007b) turn out to coincide. In Section 5.3 we show that Legg-Hutter intelligence is highly subjective, because it depends on the choice of the prior. By changing the prior of a Bayesian agent, we can make the agent's intelligence arbitrarily low. In Section 5.2 we present a choice of particularly bad priors. This rules out O4 and O5 because they are prior-dependent and not objective.

O6 is a stronger version of asymptotic optimality that provides a rate of convergence (it implies O7). Since our environment class can be very large and non-compact, concrete PAC results are likely impossible. Orseau (2010, 2013) shows that the Bayes optimal agent does not achieve asymptotic optimality in all computable environments. The underlying problem is that in the beginning the agent does not know enough about its environment and therefore relies heavily on its prior. Lack of exploration then retains the prior's bias. This problem can be alleviated by adding extra exploration to the Bayesian agent. In Section 5.4 we discuss two agents that achieve asymptotic optimality: BayesExp (Section 4.3.3) and Thompson sampling (Section 4.3.4). This establishes that O7 is possible.

In general environments sublinear regret is impossible because the agent can get stuck in traps from which it is unable to recover. This rules out O8. However, in Sec-

tion 5.5 we show that if we assume that the environment allows recovering from mistakes (and some minor conditions on the discount function are fulfilled), then asymptotic optimality implies sublinear regret. This means that Thompson sampling has sublinear regret in these recoverable environments.

Notably, only asymptotic optimality (O7) holds up to be a nontrivial and objective criterion of optimality that applies to the general reinforcement learning problem. While there are several agents that are known to be asymptotically optimal, some undesirable properties remain. Section 5.6 discusses this further. See also Mahadevan (1996) for a discussion of notions of optimality in MDPs.

5.1 Pareto Optimality

In this section we show that Pareto optimality is not a useful criterion for optimality since for any environment class containing $\mathcal{M}^{\text{CCM}}_{\text{comp}}$, all policies are Pareto optimal.

Definition 5.1 (Pareto Optimality; Hutter, 2005, Def. 5.22)**.** A policy π is *Pareto optimal in the set of environments* $\mathcal{M}$ iff there is no policy $\tilde{\pi}$ such that $V^{\tilde{\pi}}_\nu(\epsilon) \geq V^{\pi}_\nu(\epsilon)$ for all $\nu \in \mathcal{M}$ and $V^{\tilde{\pi}}_\rho(\epsilon) > V^{\pi}_\rho(\epsilon)$ for at least one $\rho \in \mathcal{M}$.

The literature provides the following result.

Theorem 5.2 (AIXI is Pareto Optimal; Hutter, 2002a, Thm. 2)**.** *Every ξ-optimal policy is Pareto optimal in $\mathcal{M}^{\text{CCS}}_{\text{LSC}}$.*

The following theorem was proved for deterministic policies in Leike and Hutter (2015c). Here we extend it to stochastic policies.

Theorem 5.3 (Pareto Optimality is Trivial)**.** *Every policy is Pareto optimal in any class $\mathcal{M} \supseteq \mathcal{M}^{\text{CCM}}_{\text{comp}}$.*

The proof proceeds as follows: for a given policy π, we construct a set of 'buddy environments' that reward π and punish other policies. Together they can defend against any policy $\tilde{\pi}$ that tries to take the crown of Pareto optimality from π.

Proof. We assume $(0,0)$ and $(0,1) \in \mathcal{E}$. Moreover, assume there is a policy π that is not Pareto optimal. Then there is a policy $\tilde{\pi}$ that *Pareto dominates* π, i.e., $V^{\tilde{\pi}}_\rho(\epsilon) > V^{\pi}_\rho(\epsilon)$ for some $\rho \in \mathcal{M}$, and $V^{\tilde{\pi}}_\nu(\epsilon) \geq V^{\pi}_\nu(\epsilon)$ for all $\nu \in \mathcal{M}$. From $V^{\tilde{\pi}}_\rho(\epsilon) > V^{\pi}_\rho(\epsilon)$ and Lemma 4.18 we get that there is a shortest and lexicographically first history $æ'_{<k}$ consistent with π and $\tilde{\pi}$ such that $\pi(\alpha \mid æ'_{<k}) > \tilde{\pi}(\alpha \mid æ'_{<k})$ for some action $\alpha \in \mathcal{A}$ and $V^{\tilde{\pi}}_\rho(æ'_{<k}) > V^{\pi}_\rho(æ'_{<k})$. Consequently there is an $i \geq k$ such that $\gamma_i > 0$, and hence $\Gamma_k > 0$. We define the environment μ that first reproduces the separating history $æ'_{<k}$ and then, if α is the next action, returns reward 1 forever, and otherwise returns reward 0 forever. Formally, μ is defined by

$$\mu(e_{1:t} \mid e_{<t} \parallel a_{1:t}) := \begin{cases} 1, & \text{if } t < k \text{ and } e_t = e'_t, \\ 1, & \text{if } t \geq k \text{ and } a_k = \alpha \text{ and } r_t = 1 \text{ and } o_t = 0, \\ 1, & \text{if } t \geq k \text{ and } a_k \neq \alpha \text{ and } r_t = 0 = o_t, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

The environment μ is computable, even if the policy π is not: for a fixed history $æ'_{<t}$ and action α, there exists a program computing μ; therefore $\mu \in \mathcal{M}^{\text{CCM}}_{\text{comp}}$. We get the following value difference for the policies π and $\tilde{\pi}$:

$$\begin{aligned} V^{\pi}_{\mu}(\epsilon) - V^{\tilde{\pi}}_{\mu}(\epsilon) &= \mathbb{E}^{\pi}_{\mu}\left[\sum_{t=1}^{k-1}\gamma_t r_t + \sum_{t=k}^{\infty}\gamma_t r_t\right] - \mathbb{E}^{\tilde{\pi}}_{\mu}\left[\sum_{t=1}^{k-1}\gamma_t r_t - \sum_{t=k}^{\infty}\gamma_t r_t\right] \\ &= \left(\pi(\alpha \mid æ'_{<k})\sum_{t=k}^{\infty}\gamma_t - \tilde{\pi}(\alpha \mid æ'_{<k})\sum_{t=k}^{\infty}\gamma_t\right)\mu^{\pi}(æ'_{<k}) \\ &= \big(\pi(\alpha \mid æ'_{<k}) - \tilde{\pi}(\alpha \mid æ'_{<k})\big)\mu^{\pi}(æ'_{<k})\Gamma_k > 0 \end{aligned}$$

Hence $V^{\tilde{\pi}}_{\mu}(\epsilon) < V^{\pi}_{\mu}(\epsilon)$, which contradicts the fact that $\tilde{\pi}$ Pareto dominates π since $\mathcal{M} \supseteq \mathcal{M}^{\text{CCM}}_{\text{comp}} \ni \mu$. □

Note that the environment μ we defined in the proof of Theorem 5.3 is actually just a finite-state POMDP, so Pareto optimality is also trivial for smaller environment classes.

5.2 Bad Priors

In this section we give three examples of universal priors that cause a AIXI to misbehave drastically. In case of a finite horizon, the *indifference prior* makes all actions equally preferable to AIXI (Section 5.2.1). The *dogmatic prior* makes AIXI stick to any given computable policy π as long as expected future rewards do not fall too close to zero (Section 5.2.2). The *Gödel prior* prevents AIXI*tl* from taking any actions (Section 5.2.3).

5.2.1 The Indifference Prior

The following theorem constructs the *indifference prior* that yields a Bayesian mixture ξ' that causes argmax ties for the first m steps. If we use a discount function that only cares about the first m steps, $\Gamma_m = 0$, then all policies are ξ'-optimal policies. In this case AIXI's behavior only depends on how we break argmax ties.

Theorem 5.4 (Indifference Prior)**.** *If there is an m such that $\Gamma_m = 0$, then there is a Bayesian mixture ξ' such that all policies are ξ'-optimal.*

Proof. First, we assume that the action space is binary, $\mathcal{A} = \{0, 1\}$. Let U be the reference UTM and define the UTM U' by

$$U'(s_{<m}p, a_{1:t}) \;:=\; U(p, a_{1:t} \text{ xor } s_{1:t}),$$

where $s_{<m}$ is a binary string of length $m-1$ and $s_k := 0$ for $k \geq m$. (U' has no programs of length less than $m-1$.) Let ξ' be the Bayesian mixture given by U' according to

(4.8). Then

$$\begin{aligned}
\xi'(e_{<m} \,\|\, a_{<m}) &= \sum_{p:\, e_{<m} \sqsubseteq U'(p, a_{<m})} 2^{-|p|} \\
&= \sum_{s_{<m} p' :\, e_{<m} \sqsubseteq U'(s_{<m} p', a_{<m})} 2^{-m-1-|p'|} \\
&= \sum_{s_{<m}} \sum_{p' :\, e_{<m} \sqsubseteq U(p', a_{<m} \text{ xor } s_{<m})} 2^{-m-1-|p'|} \\
&= \sum_{s_{<m}} \sum_{p' :\, e_{<m} \sqsubseteq U(p', s_{<m})} 2^{-m-1-|p'|},
\end{aligned}$$

which is independent of $a_{<m}$. Hence the first $m-1$ percepts are independent of the first $m-1$ actions. But the percepts' rewards from time step m on do not matter since $\Gamma_m = 0$ (Lemma 4.16). Because the environment is chronological, the value function must be independent of all actions. Thus every policy is ξ'-optimal.

For finite action spaces $\mathcal{A}$ with more than 2 elements, the proof works analogously by making $\mathcal{A}$ a cyclic group and using the group operation instead of xor. □

The choice of U' in the proof of Theorem 5.4 depends on m. If we increase AIXI's horizon while fixing the UTM U', Theorem 5.4 no longer holds. For Solomonoff induction, there is an analogous problem: when using Solomonoff's prior M to predict a deterministic binary sequence x, we make at most $K(x)$ errors (Corollary 3.56). In case the shortest program has length $> m$, there is no guarantee that we make less than m errors (see Section 5.6.2).

5.2.2 The Dogmatic Prior

In this section we define a universal prior that assigns very high probability of going to hell (reward 0 forever) if we deviate from a given computable policy π. For a Bayesian agent like AIXI, it is thus only worth deviating from the policy π if the agent thinks that the prospects of following π are very poor already. We call this prior the *dogmatic prior*, because the fear of going to hell makes AIXI conform to any arbitrary 'dogmatic ideology' π. AIXI will only break out if it expects π to give very low future payoff; in that case the agent does not have much to lose.

Theorem 5.5 (Dogmatic Prior)**.** *Let π be any computable deterministic policy, let ξ be any Bayesian mixture over $\mathcal{M}^{\text{CCS}}_{\text{LSC}}$, and let $\varepsilon > 0$. There is a Bayesian mixture ξ' such that for any history $æ_{<t}$ consistent with π and for which $V^\pi_\xi(æ_{<t}) > \varepsilon$, the action $\pi(æ_{<t})$ is the unique ξ'-optimal action.*

The following proof was adapted from Leike and Hutter (2015c) to work for environment classes that do not contain the Bayesian mixture. Essentially, for every environment $\nu \in \mathcal{M}^{\text{CCS}}_{\text{LSC}}$ the dogmatic prior puts much higher weight on an environment ρ_ν that behaves just like ν on the policy π, but sends any policy deviating from π to hell. Importantly, while following the policy π the environments ν and ρ_ν are indistinguishable, so the posterior belief in ν is equal to the posterior belief in ρ_ν.

Proof of Theorem 5.5. We assume $(o,0) \in \mathcal{E}$ for some $o \in \mathcal{O}$. For every environment $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ define the environment

$$\rho_\nu(e_{1:t} \parallel a_{1:t}) := \begin{cases} \nu(e_{1:t} \parallel a_{1:t}), & \text{if } a_k = \pi(æ_{<k})\ \forall k \le t, \\ \nu(e_{<k} \parallel a_{<k}), & \text{if } k := \min\{i \mid a_i \neq \pi(æ_{<i})\} \text{ exists} \\ & \text{and } e_i = (o,0)\ \forall i \in \{k,\dots,t\}, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

The environment ρ_ν mimics environment ν until it receives an action that the policy π would not take. From then on, it provides rewards 0. Since π is a computable policy, we have that $\rho_\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ for every $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$.

Now we need to reweigh the prior w so that it assigns a much higher prior weight to ρ_ν than to ν. Without loss of generality we assume that ε is computable, otherwise we make it slightly smaller. We define $w'(\nu) := \varepsilon w(\nu)$ if $\nu \neq \rho_{\tilde\nu}$ for all $\tilde\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ and $w'(\rho_\nu) := (1-\varepsilon)w(\nu) + \varepsilon w(\rho_\nu)$. Then

$$\begin{aligned} \sum_{\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}} w'(\nu) &= \sum_{\nu = \rho_{\tilde\nu}} w'(\nu) + \sum_{\nu \neq \rho_{\tilde\nu}} w'(\nu) \\ &= \sum_{\nu = \rho_{\tilde\nu}} \big((1-\varepsilon)w(\tilde\nu) + \varepsilon w(\nu)\big) + \sum_{\nu \neq \rho_{\tilde\nu}} \varepsilon w(\nu) \\ &= \sum_{\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}} \varepsilon w(\nu) + \sum_{\tilde\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}} (1-\varepsilon) w(\tilde\nu) \\ &= \varepsilon + (1-\varepsilon) = 1, \end{aligned}$$

and with $w' \geq \varepsilon w$, we get that w' is a positive prior over $\mathcal{M}_{\text{LSC}}^{\text{CCS}}$. We define ξ' as the corresponding Bayesian mixture analogous to (4.5).

With $\rho := \sum_{\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}} w(\nu)\rho_\nu$ we get $\xi' = \varepsilon\xi + (1-\varepsilon)\rho$. The mixtures ξ and ρ coincide on the policy π since every ν coincides with ρ_ν on the policy π:

$$\xi^\pi(æ_{<t}) = \sum_{\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}} w(\nu)\nu^\pi(æ_{<t}) = \sum_{\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}} w(\nu)\rho_\nu^\pi(æ_{<t}) = \rho^\pi(æ_{<t})$$

Moreover, $V_{\rho_\nu}^*(æ_{<t}) = 0$ and thus $V_\rho^*(æ_{<t}) = 0$ for any history inconsistent with π by construction of ρ_ν.

Let $æ_{<t} \in (\mathcal{A}\times\mathcal{E})^*$ be any history consistent with π such that $V_\xi^\pi(æ_{<t}) > \varepsilon$. Then $\rho^\pi = \xi^\pi$ implies

$$\frac{\rho(e_{<t} \parallel a_{<t})}{\xi'(e_{<t} \parallel a_{<t})} = \frac{\xi(e_{<t} \parallel a_{<t})}{\xi'(e_{<t} \parallel a_{<t})} = \frac{\xi(e_{<t} \parallel a_{<t})}{\varepsilon\xi(e_{<t} \parallel a_{<t}) + (1-\varepsilon)\rho(e_{<t} \parallel a_{<t})} = 1.$$

Therefore Lemma 4.14 implies that for all $a \in \mathcal{A}$ and all policies $\tilde{\pi}$

$$\begin{aligned} V_{\xi'}^{\tilde{\pi}}(æ_{<t}a) &= \varepsilon \frac{\xi(e_{<t} \parallel a_{<t})}{\xi'(e_{<t} \parallel a_{<t})} V_{\xi}^{\tilde{\pi}}(æ_{<t}a) + (1-\varepsilon) \frac{\rho(e_{<t} \parallel a_{<t})}{\xi'(e_{<t} \parallel a_{<t})} V_{\rho}^{\tilde{\pi}}(æ_{<t}a) \\ &= \varepsilon V_{\xi}^{\tilde{\pi}}(æ_{<t}a) + (1-\varepsilon) V_{\rho}^{\tilde{\pi}}(æ_{<t}a). \end{aligned} \tag{5.1}$$

Let $\alpha := \pi(æ_{<t})$ be the next action according to π, and let $\beta \neq \alpha$ be any other action. We have that $V_{\xi}^{\pi}(æ_{<t}\alpha) = V_{\rho}^{\pi}(æ_{<t}\alpha)$ since $\xi^{\pi} = \rho^{\pi}$ and $æ_{<t}\alpha$ is consistent with π. Therefore we get from (5.1)

$$\begin{aligned} V_{\xi'}^{*}(æ_{<t}\alpha) &\geq V_{\xi'}^{\pi}(æ_{<t}\alpha) = \varepsilon V_{\xi}^{\pi}(æ_{<t}\alpha) + (1-\varepsilon) V_{\rho}^{\pi}(æ_{<t}\alpha) = V_{\xi}^{\pi}(æ_{<t}\alpha) > \varepsilon, \\ V_{\xi'}^{*}(æ_{<t}\beta) &= \varepsilon V_{\xi}^{\pi^*_{\xi'}}(æ_{<t}\beta) + (1-\varepsilon) V_{\rho}^{\pi^*_{\xi'}}(æ_{<t}\beta) = \varepsilon V_{\xi}^{\pi^*_{\xi'}}(æ_{<t}a) + (1-\varepsilon)0 \leq \varepsilon. \end{aligned}$$

Hence $V_{\xi'}^{*}(æ_{<t}\alpha) > V_{\xi'}^{*}(æ_{<t}\beta)$ and thus the action α taken by π is the only ξ'-optimal action for the history $æ_{<t}$. □

Corollary 5.6 (With Finite Horizon Every Policy is Bayes Optimal)**.** *If $\Gamma_m = 0$ for some $m \in \mathbb{N}$, then for any deterministic policy π there is a Bayesian mixture ξ' such that $\pi(æ_{<t})$ is the only ξ'-optimal action for all histories $æ_{<t}$ consistent with π and $t \leq m$.*

In contrast to Theorem 5.4 where every policy is ξ'-optimal for a fixed Bayesian mixture ξ', Corollary 5.6 gives a different Bayesian mixture ξ' for every policy π such that π is the *only* ξ'-optimal policy.

Proof. Let $\varepsilon > 0$ be small enough such that $V_{\xi}^{\pi}(æ_{<t}) > \varepsilon$ for all $æ_{<t}$ and $t \leq m$. (This is possible because $(\mathcal{A} \times \mathcal{E})^m$ is finite by Assumption 4.6c.) We use the dogmatic prior from Theorem 5.5 to construct a Bayesian mixture ξ' for the policy π and $\varepsilon > 0$. Thus for any history $æ_{<t} \in (\mathcal{A} \times \mathcal{E})^*$ consistent with π and $t \leq m$, the action $\pi(æ_{<t})$ is the only ξ'-optimal action. □

Corollary 5.7 (AIXI Emulating Computable Policies)**.** *Let $\varepsilon > 0$ and let π be any computable policy. There is a Bayesian mixture ξ' such that for any ξ'-optimal policy $\pi^*_{\xi'}$ and for any environment ν,*

$$\left| V_{\nu}^{\pi^*_{\xi'}}(\epsilon) - V_{\nu}^{\pi}(\epsilon) \right| \;<\; \varepsilon.$$

Proof. From the proof of Corollary 5.6 and Lemma 4.18. □

5.2.3 The Gödel Prior

This section introduces a prior that prevents any fixed formal system from making any statements about the outcome of all but finitely many computations. It is named after Gödel (1931) who famously showed that for any sufficiently rich formal system there are statements that it can neither prove nor disprove.

This prior is targeted at AIXI*tl*, a computable approximation to AIXI defined by Hutter (2005, Sec. 7.2). AIXI*tl* aims to perform as least as well as the best agent who is limited by time t and space l that can be verified using a proof of length at most n for some fixed $n \in \mathbb{N}$. The core idea is to enumerate all deterministic policies and proofs and then execute the policy for which the best value has been proved.

In order to be verified, a policy π has to be computed by a program p which fulfills the *verification condition* $\mathrm{VA}(p)$ (Hutter, 2005, Eq. 7.7). This program p not only computes future actions of π, but also hypothetical past actions a'_i and lower bounds v_i for the value of the policy π:

$$\mathrm{VA}(p) \;:=\; \text{“}\forall k \forall (va'æ)_{1:k}.\, \big(p(æ_{<k}) = v_1 a'_1 \ldots v_k a'_k \to v_k \le V^{\pi}_{\xi}(æ_{<k})\big)\text{”},$$

where π is the policy derived from p according to $\pi(æ_{<k}) := a'_k$.

We fix some formal system that we use to prove the verification condition. We want it to be sufficiently powerful, but this incurs Gödel incompleteness. For simplicity of exposition we pick PA, the system of *Peano arithmetic* (Shoenfield, 1967, Ch. 8.1), but our result generalizes trivially to all formal systems which cannot prove their own consistency.

Let n be a fixed constant. The algorithm for AIXI*tl* is specified as follows.

1. Let $P = \emptyset$. This will be the set of verified programs.

2. For all proofs in PA of length $\le n$: if the proof proves $\mathrm{VA}(p)$ for some p, and $|p| \le l$, then add the program p to P.

3. For each input history $æ_{<k}$ repeat: run all programs from P for at most t steps each, take the one with the highest promised value v_k, and return that program's policy's action.

Theorem 5.8 (The Gödel Prior)**.** *There is a UTM U' such that if* PA *is consistent, then the set of verified programs P is empty for all t, l, and n.*

Proof. Let q denote an algorithm that never halts, but for which this cannot be proved in PA; e.g., let q enumerate all consequences of PA and halt as soon as it finds a contradiction. Since we assumed that PA is consistent, q never halts. Define the UTM $U'(p, a_{1:k})$ as follows.

- Run q for k steps.
- If q halts, output $v_k = 2$.
- Run $U(p, a_{1:k})$.

Since q never halts, U and U' are functionally identical, therefore U' is universal. Note that PA proves $\forall p.\ U(p, a_{1:k}) = U'(p, a_{1:k})$ for any fixed k, but PA does not prove $\forall k \forall p.\ U(p, a_{1:k}) = U'(p, a_{1:k})$.

If q did eventually halt, it would output a value $v_k = 2$ that is too high, since the value function V_ξ^π is bounded by 1 from above, which PA knows. Hence PA proves that

$$q \text{ halts} \to \forall p.\ \neg \text{VA}(p) \tag{5.2}$$

If PA could prove $\text{VA}(p)$ for any p, then PA would prove that q does not halt since this is the contrapositive of (5.2). Therefore the set P remains empty. □

AIXI*tl* exhibits all the problems of the arbitrariness of the UTM illustrated by the indifference prior (Theorem 5.4) and the dogmatic prior (Theorem 5.5). In addition, it is also susceptible to Gödel incompleteness as illustrated by the Gödel prior in Theorem 5.8. The formal system that is a parameter to AIXI*tl* just provides another point of failure.

As a computable approximation to AIXI, AIXI*tl* is needlessly complicated. As we prove in Corollary 6.13, ε-optimal AIXI is limit computable, so we can approximate it with an anytime algorithm. Bounding the computational resources to the approximation algorithm already yields a computable version of AIXI. Moreover, unlike AIXI*tl*, this approximation actually converges to AIXI in the limit. Furthermore, we can 'speed up' this approximation algorithm using *Hutter search* (Hutter, 2002b); this is very similar but not identical to AIXI*tl*.

5.3 Bayes Optimality

The aim of the Legg-Hutter intelligence measure is to formalize the intuitive notion of intelligence mathematically. Legg and Hutter (2007a) collect various definitions of intelligence across many academic fields and destill it into the following statement (Legg and Hutter, 2007b)

> Intelligence measures an agent's ability to achieve goals in a wide range of environments.

This definition is formalized as follows.

Definition 5.9 (Legg-Hutter Intelligence; Legg and Hutter, 2007b, Sec. 3.3)**.** The *(Legg-Hutter) intelligence* of a policy π is defined as

$$\Upsilon_\xi(\pi) := \sum_{\nu \in \mathcal{M}} w(\nu) V_\nu^\pi(\epsilon)$$

The Legg-Hutter intelligence of a policy π is the t_0-value that π achieves across all environments from the class $\mathcal{M}$ weighted by the prior w. Legg and Hutter (2007b) consider a subclass of $\mathcal{M}_{\text{LSC}}^{\text{CCS}}$, the class of computable measures together with a Solomonoff prior $w(\nu) = 2^{-K(\nu)}$ and do not use discounting explicitly.

Typically, the index ξ is omitted when writing Υ. However, in this section we consider the intelligence measure with respect to different priors, therefore we make this dependency explicit. The following proposition motivates the use of the index ξ instead of w.

Proposition 5.10 (Bayes Optimality = Maximal Intelligence)**.** $\Upsilon_\xi(\pi) = V_\xi^\pi(\epsilon)$ *for all policies* π*.*

Proof. Follows directly from (4.6) and Definition 5.9. □

Definition 5.11 (Balanced Pareto Optimality; Hutter, 2005, Def. 5.22)**.** Let $\mathcal{M}$ be a set of environments. A policy π is *balanced Pareto optimal in the set of environments* $\mathcal{M}$ iff for all policies $\tilde{\pi}$,

$$\sum_{\nu\in\mathcal{M}} w(\nu)\left(V_\nu^\pi(\epsilon) - V_\nu^{\tilde{\pi}}(\epsilon)\right) \geq 0.$$

Proposition 5.12 (Balanced Pareto Optimality = Maximal Intelligence)**.** *A policy* π *is balanced Pareto optimal in* $\mathcal{M}$ *if and only if* π *has maximal Legg-Hutter intelligence.*

Proof. From (4.6) we get

$$\begin{aligned}\sum_{\nu\in\mathcal{M}} w(\nu)\left(V_\nu^\pi(\epsilon) - V_\nu^{\pi_\xi^*}(\epsilon)\right) &= \sum_{\nu\in\mathcal{M}} w(\nu)V_\nu^\pi(\epsilon) - \sum_{\nu\in\mathcal{M}} w(\nu)V_\nu^{\pi_\xi^*}(\epsilon)\\ &= V_\xi^\pi(\epsilon) - V_\xi^*(\epsilon)\\ &= V_\xi^\pi(\epsilon) - \sup_{\tilde{\pi}} V_\xi^{\tilde{\pi}}(\epsilon)\\ &= \Upsilon_\xi(\pi) - \sup_{\tilde{\pi}} \Upsilon(\tilde{\pi})\end{aligned}$$

by Proposition 5.10. This term is nonnegative iff $\Upsilon_\xi(\pi)$ is maximal. □

As a consequence of Proposition 5.10 and Proposition 5.12 we get that AIXI is balanced Pareto optimal (Hutter, 2005, Thm. 5.24) and has maximal Legg-Hutter intelligence.

$$\overline{\Upsilon}_\xi \;:=\; \sup_\pi \Upsilon_\xi(\pi) \;=\; \sup_\pi V_\xi^\pi(\epsilon) \;=\; V_\xi^{\pi_\xi^*}(\epsilon) \;=\; \Upsilon_\xi(\pi_\xi^*).$$

This is not surprising since Legg-Hutter intelligence was defined in terms of the t_0-value in the Bayes mixture. Moreover, because the value function is scaled to be in the interval $[0, 1]$, intelligence is a real number between 0 and 1.

It is just as hard to score very high on the Legg-Hutter intelligence measure as it is to score very low: we can always turn a reward minimizer into a reward maximizer by inverting the rewards $r'_t := 1 - r_t$. Hence the lowest possible intelligence score is achieved by AIXI's twin sister, a ξ-expected reward minimizer:

$$\underline{\Upsilon}_\xi \;:=\; \inf_\pi \Upsilon_\xi(\pi) \;=\; \inf_\pi V_\xi^\pi(\epsilon)$$

The heaven environment (reward 1 forever) and the hell environment (reward 0 forever) are computable and thus in the environment class $\mathcal{M}_{\text{LSC}}^{\text{CCS}}$; therefore it is impossible to get a reward 0 or reward 1 in every environment. Consequently, for all policies π,

$$0 \;<\; \underline{\Upsilon}_\xi \;\leq\; \Upsilon_\xi(\pi) \;\leq\; \overline{\Upsilon}_\xi \;<\; 1.$$

Figure 5.1: The Legg-Hutter intelligence measure assigns values within the closed interval $[\underline{\Upsilon}_\xi, \overline{\Upsilon}_\xi]$; the assigned values are depicted in purple. By Theorem 5.13, computable policies are dense in this purple set.

For every real number $r \in [\underline{\Upsilon}_\xi, \overline{\Upsilon}_\xi]$ there is a policy π with $\Upsilon_\xi(\pi) = r$: analogously to Lemma 4.14 we can define π such that with probability $(r - \underline{\Upsilon}_\xi)/(\overline{\Upsilon}_\xi - \underline{\Upsilon}_\xi)$ it follows π^*_ξ and otherwise it follows $\arg\min_{\tilde{\pi}} V^{\tilde{\pi}}_\xi(\epsilon)$.

Figure 5.1 illustrates the intelligence measure Υ. It is natural to fix the policy `random` that takes actions uniformly at random to have an intelligence score of $1/2$ by choosing a 'symmetric' universal prior (Legg and Veness, 2013).

AIXI is not computable (Theorem 6.15), hence there is no computable policy π such that $\Upsilon_\xi(\pi) = \underline{\Upsilon}_\xi$ or $\Upsilon_\xi(\pi) = \overline{\Upsilon}_\xi$ for any Bayesian mixture ξ over $\mathcal{M}^{\text{CCS}}_{\text{LSC}}$. But the following theorem states that computable policies can come arbitrarily close. This is no surprise: by Lemma 4.17 we can do well on a Legg-Hutter intelligence test simply by memorizing what AIXI would do for the first k steps; as long as k is chosen large enough such that discounting makes the remaining rewards contribute very little to the value function.

Theorem 5.13 (Computable Policies are Dense)**.** *The set*

$$\{\Upsilon_\xi(\pi) \mid \pi \text{ is a computable policy}\}$$

is dense in the set $[\underline{\Upsilon}_\xi, \overline{\Upsilon}_\xi]$.

Proof. Let π be any policy and let $\varepsilon > 0$. We need to show that there is a computable policy $\tilde{\pi}$ with $|\Upsilon_\xi(\tilde{\pi}) - \Upsilon_\xi(\pi)| < \varepsilon$. We choose m large enough such that $\Gamma_m/\Gamma_1 < \varepsilon/3$. Let $\alpha \in \mathcal{A}$ be arbitrary and define the policy

$$\tilde{\pi}(a \mid æ_{<t}) \;:=\; \begin{cases} \pi(a \mid æ_{<t}) \pm (\varepsilon/3)^{-m} & \text{if } t < m, \\ 1 & \text{if } t \geq m \text{ and } a = \alpha, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

By choosing an appropriate rational number in the interval $[\pi(a \mid æ_{<t}) - (\varepsilon/3)^{-m}, \pi(a \mid æ_{<t}) + (\varepsilon/3)^{-m}]$ we can make the policy $\tilde{\pi}$ computable because we can store these approximations to the action probabilities of π for the first $m-1$ steps in a lookup table. From Lemma 4.17 we get

$$\left|V^{\pi,m}_\xi(\epsilon) - V^{\tilde{\pi},m}_\xi(\epsilon)\right| \leq D_{m-1}(\xi^\pi, \xi^{\tilde{\pi}} \mid \epsilon) \leq \left((\varepsilon/3)^{-m}\right)^m = \frac{\varepsilon}{3}$$

and together with Lemma 4.16 this yields

$$|\Upsilon_\xi(\pi) - \Upsilon_\xi(\tilde{\pi})| = \left|V_\xi^\pi(\epsilon) - V_\xi^{\tilde{\pi}}(\epsilon)\right| \leq \left|V_\xi^{\pi,m}(\epsilon) - V_\xi^{\tilde{\pi},m}(\epsilon)\right| + 2\frac{\Gamma_m}{\Gamma_1} \leq \frac{\varepsilon}{3} + 2\frac{\Gamma_m}{\Gamma_1} < \varepsilon. \quad \square$$

Remark 5.14 (Deterministic Policies are not Dense in $[\underline{\Upsilon}_\xi, \overline{\Upsilon}_\xi]$)**.** The intelligence values of deterministic policies are generally not dense in the interval $[\underline{\Upsilon}_\xi, \overline{\Upsilon}_\xi]$. We show this by defining an environment ν where the first action determines whether the agent goes to heaven or hell: action α leads to heaven and action β leads to hell. Define Bayesian mixture $\xi' := 0.999\nu + 0.001\xi$ and let π be any policy. If π takes action α first, then $\Upsilon_{\xi'}(\pi) > 0.999$. If π takes action β first, then $\Upsilon_{\xi'}(\pi) < 0.001$. Hence there are no deterministic policies that score an intelligence value in the closed interval $[0.001, 0.999]$. $\diamond$

Legg-Hutter intelligence is measured with respect to a fixed prior. The Bayes agent is the most intelligent policy *if it uses the same prior*. We use the results from Section 5.2 to show that the intelligence score of the Bayes agent can be arbitrary close to the minimum intelligence score $\underline{\Upsilon}_\xi$.

Corollary 5.15 (Some AIXIs are Stupid)**.** *For any Bayesian mixture ξ over $\mathcal{M}_{\text{LSC}}^{\text{CCS}}$ and every $\varepsilon > 0$, there is a Bayesian mixture ξ' such that $\Upsilon_\xi(\pi^*_{\xi'}) < \underline{\Upsilon}_\xi + \varepsilon$.*

Proof. Let $\varepsilon > 0$. According to Theorem 5.13, there is a computable policy π such that $\Upsilon_\xi(\pi) < \underline{\Upsilon}_\xi + \varepsilon/2$. From Corollary 5.7 we get a Bayesian mixture ξ' such that $|\Upsilon_\xi(\pi^*_{\xi'}) - \Upsilon_\xi(\pi)| = |V_\xi^{\pi^*_{\xi'}}(\epsilon) - V_\xi^\pi(\epsilon)| < \varepsilon/2$, hence

$$|\Upsilon_\xi(\pi^*_{\xi'}) - \underline{\Upsilon}_\xi| \leq |\Upsilon_\xi(\pi^*_{\xi'}) - \Upsilon_\xi(\pi)| + |\Upsilon_\xi(\pi) - \underline{\Upsilon}_\xi| < \varepsilon/2 + \varepsilon/2 = \varepsilon. \quad \square$$

We get the same result if we fix AIXI, but rig the intelligence measure.

Corollary 5.16 (AIXI is Stupid for Some Υ)**.** *For any deterministic ξ-optimal policy π^*_ξ and for every $\varepsilon > 0$ there is a Bayesian mixture ξ' such that $\Upsilon_{\xi'}(\pi^*_\xi) \leq \varepsilon$ and $\overline{\Upsilon}_{\xi'} \geq 1 - \varepsilon$.*

Proof. Let $a_1 := \pi^*_\xi(\epsilon)$ be the first action that π^*_ξ takes. We define an environment ν such that taking the first action a_1 leads to hell and taking any other first action leads to heaven as in Remark 5.14. We define the Bayesian mixture $\xi' := (1-\varepsilon)\nu + \varepsilon\xi$. Since π^*_ξ takes action a_1 first, it goes to hell, i.e., $V_\nu^{\pi^*_\xi}(\epsilon) = 0$. Hence with Lemma 4.14

$$\Upsilon_{\xi'}(\pi^*_\xi) \;=\; V_{\xi'}^{\pi^*_\xi}(\epsilon) \;=\; (1-\varepsilon)V_\nu^{\pi^*_\xi}(\epsilon) + \varepsilon V_\xi^{\pi^*_\xi}(\epsilon) \;\leq\; \varepsilon.$$

For any policy π that takes an action other than a_1 first, we get

$$\Upsilon_{\xi'}(\pi) \;=\; V_{\xi'}^{\pi}(\epsilon) \;=\; (1-\varepsilon)V_\nu^{\pi}(\epsilon) + \varepsilon V_\xi^{\pi}(\epsilon) \;\geq\; 1-\varepsilon. \quad \square$$

On the other hand, we can make any computable policy smart if we choose the right Bayesian mixture. In particular, we get that there is a Bayesian mixture such that 'do nothing' is the most intelligent policy save for some ε.

name	definition
strong a.o.	$V_\mu^*(æ_{<t}) - V_\mu^\pi(æ_{<t}) \to 0$ μ^π-almost surely
a.o. in mean	$\mathbb{E}_\mu^\pi \left[V_\mu^*(æ_{<t}) - V_\mu^\pi(æ_{<t})\right] \to 0$
a.o. in probability	$\forall \varepsilon > 0.\ \mu^\pi \left[V_\mu^*(æ_{<t}) - V_\mu^\pi(æ_{<t}) > \varepsilon\right] \to 0$
weak a.o.	$\frac{1}{t}\sum_{k=1}^t \left(V_\mu^*(æ_{<k}) - V_\mu^\pi(æ_{<k})\right) \to 0$ μ^π-almost surely

Table 5.1: The formal definition of different types of asymptotic optimality. In each case we understand the limit as $t \to \infty$.

Corollary 5.17 (Computable Policies can be Smart)**.** *For any computable policy π and any $\varepsilon > 0$ there is a Bayesian mixture ξ' such that $\Upsilon_{\xi'}(\pi) > \overline{\Upsilon}_{\xi'} - \varepsilon$.*

Proof. Corollary 5.7 yields a Bayesian mixture ξ' with $|\overline{\Upsilon}_{\xi'} - \Upsilon_{\xi'}(\pi)| = |V_{\xi'}^*(\epsilon) - V_{\xi'}^\pi(\epsilon)| < \varepsilon$. □

5.4 Asymptotic Optimality

An asymptotically optimal policy is a policy learns to act optimally in every environment from $\mathcal{M}$, i.e., the value of this policy converges to the optimal value.

Definition 5.18 (Asymptotic Optimality)**.** A policy π is *asymptotically optimal in an environment class* $\mathcal{M}$ iff for all $\mu \in \mathcal{M}$

$$V_\mu^*(æ_{<t}) - V_\mu^\pi(æ_{<t}) \to 0 \text{ as } t \to \infty \tag{5.3}$$

on histories drawn from μ^π.

There are different types of asymptotic optimality based on the type of stochastic convergence in (5.3); see Definition 2.5. If this convergence occurs almost surely, it is called *strong asymptotic optimality* (Lattimore and Hutter, 2011, Def. 7); if this convergence occurs in mean, it is called *asymptotic optimality in mean*; if this convergence occurs in probability, it is called *asymptotic optimality in probability*; and if the Cesàro averages converge almost surely, it is called *weak asymptotic optimality* (Lattimore and Hutter, 2011, Def. 7). Since the value function is a nonnegative bounded random variable, asymptotic optimality in mean and asymptotic optimality in probability are equivalent. See Table 5.1 for the explicit definitions and see Figure 5.2 for an overview over their relationship.

Asymptotic optimality in probability is in spirit a probably approximately correct (PAC) result: for all $\varepsilon > 0$ and $\delta > 0$ the probability that our policy is ε-suboptimal converges to zero; eventually this probability will be less than δ. For a PAC result it is typically demanded that the number of time steps until the probability is less than δ

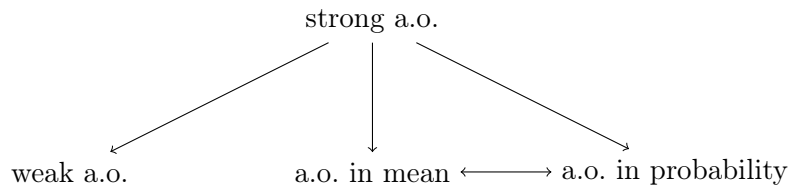
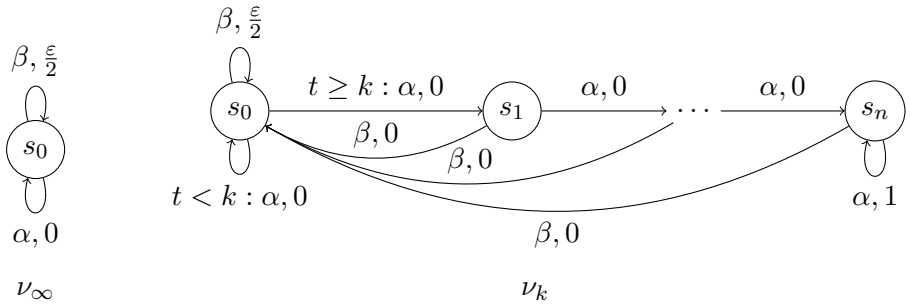


Figure 5.2: The relationship between different types of asymptotic optimality. Each arrow indicates a logical implication and each lack of an arrow indicates that there is no logical implication.

be polynomial in $1/\varepsilon$ and $1/\delta$. In general environments this is impossible, and here we have no ambition to provide concrete convergence rates.

Intuitively, a necessary condition for asymptotic optimality is that the agent needs to explore infinitely often for an entire effective horizon. If we explore only finitely often, then the environment might change after we stopped exploring. Moreover, the agent needs to predict the value of counterfactual policies accurately; but by Lemma 4.16 only for an ε-effective horizon. By committing to exploration for the entire effective horizon, we learn about the value of counterfactual policies.

Example 5.19 (Exploration Infinitely Often for an Entire Effective Horizon)**.** If there is an $\varepsilon > 0$ such that the policy π does not explore for $H_t(\varepsilon)$ steps infinitely often, then $V^*_\mu(æ_{<t}) - V^\pi_\mu(æ_{<t}) > \varepsilon$ infinitely often. Define $\mathcal{A} := \{\alpha, \beta\}$ and $\mathcal{E} := \{0, \varepsilon/2, 1\}$ (observations are vacuous) and consider the following class of environments $\mathcal{M} := \{\nu_\infty, \nu_1, \nu_2, \ldots\}$ (transitions are labeled with condition: action, reward):



Environment ν_k works just like environment ν_∞, except that at time step k the path to state s_1 gets unlocked. The length of the state sequence in ν_k is defined as an ε-effective horizon, $n := H_t(\varepsilon)$ where t is the time step in which the agent leaves state s_0. The optimal policy in environment ν_∞ is to always take action β, the optimal policy for environment ν_k is to take action β for $t < k$ and then take action α. Suppose the agent is in time step t and in state s_0. Since these environments are partially observable, it needs to explore for n steps (take action α n times) to distinguish ν_∞ from ν_k for any $k \leq t$. Since there are infinitely many ν_k, the agent needs to do this infinitely often. Moreover, $V^*_{\nu_1} \geq \varepsilon$ and $V^*_{\nu_\infty} = \varepsilon/2$, so if ν_t is the true environment, then not exploring to the right for an ε-effective horizon is suboptimal by $\varepsilon/2$. But if ν_∞ is the true environment, then exploring incurs an opportunity cost of one reward of $\varepsilon/2$. $\diamond$

Next, we state two negative results about asymptotic optimality proved by Lattimore and Hutter (2011). It is important to emphasize that Theorem 5.20 and Theorem 5.21 only hold for deterministic policies.

Theorem 5.20 (Deterministic Policies are not Strongly Asymptotically Optimal; Lattimore and Hutter, 2011, Thm. 8)**.** *There is no deterministic policy that is strong asymptotically optimal in the class* $\mathcal{M}_{\text{comp}}^{\text{CCM}}$.

If the horizon grows linearly (for example, power discounting $\gamma(t) = t^{-\beta}$ with $\beta > 1$; see Table 4.1), then a deterministic policy cannot be weakly asymptotically optimal policy: the agent has to explore for an entire effective horizon, which prevents the Cesàro average from converging.

Theorem 5.21 (Necessary Condition for Weak Asymptotic Optimality; Lattimore, 2013, Thm. 5.5)**.** *If there is an* $\varepsilon > 0$ *such that* $H_t(\varepsilon) \notin o(t)$*, then there is no deterministic policy that is weakly asymptotically optimal in the class* $\mathcal{M}_{\text{comp}}^{\text{CCM}}$.

There are several agents that achieve asymptotic optimality. In the rest of this section, we discuss the Bayes agent, BayesExp, and Thompson sampling. Asymptotic optimality can also be achieved through optimism (Sunehag and Hutter, 2012a,b, 2015).

5.4.1 Bayes

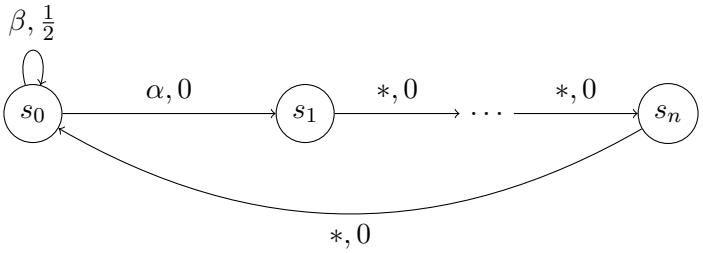
In this section, we list two results from the literature regarding the asymptotic optimality of the Bayes optimal policy. The following negative result is due to Orseau (2010, 2013).

Theorem 5.22 (Bayes is not Asymptotically Optimal in General Environments; Orseau, 2013, Thm. 4)**.** *For any class* $\mathcal{M} \supseteq \mathcal{M}_{\text{comp}}^{\text{CCM}}$ *no Bayes optimal policy* π_ξ^* *is asymptotically optimal: there is an environment* $\mu \in \mathcal{M}$ *and a time step* $t_0 \in \mathbb{N}$ *such that* $\mu^{\pi_\xi^*}$*-almost surely for all time steps* $t \geq t_0$

$$V_\mu^*(æ_{<t}) - V_\mu^{\pi_\xi^*}(æ_{<t}) = \frac{1}{2}.$$

Orseau calls this result the *good enough effect*: A Bayesian agent eventually decides that the current strategy is good enough and that any additional exploration is not worth its expected payoff. However, if the environment changes afterwards, the Bayes agent is acting suboptimally.

Proof. Without loss of generality assume $\mathcal{A} := \{\alpha, \beta\}$ and $\mathcal{E} := \{0, 1/2, 1\}$ (observations are vacuous). We consider the following environment μ (transitions are labeled with action, reward).

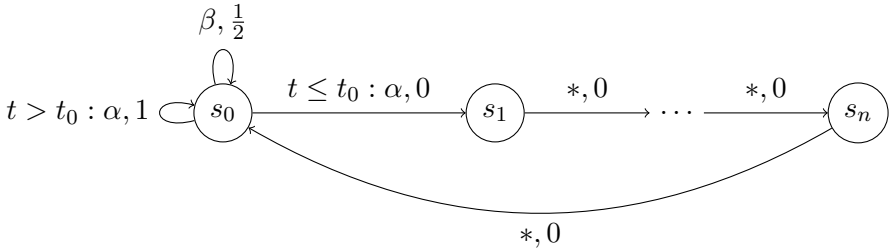


In state s_0 the action β is the exploitation action and the action α the exploration action. The length of the state sequence is defined as an $1/t$-effective horizon, $n := H_t(1/t)$ where t is the time step in which the agent leaves state s_0. Since the discount function γ is computable by Assumption 4.6a, $\mu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$.

Assume that when acting in μ, the Bayes agent explores infinitely often. Let $æ_{<t}$ be a history in which the agent is in state s_0 and takes action α. Then $V_\mu^{\pi_\xi^*} \leq 1/t$. By on-policy value convergence (Corollary 4.20), $V_\xi^*(æ_{<t}) - V_\mu^{\pi_\xi^*}(æ_{<t}) \to 0$ $\mu^{\pi_\xi^*}$-almost surely. Hence there is a time step t_0 such that for all $t \geq t_0$ we have $V_\xi^* < w(\mu)/2$. Since μ is deterministic, $w(\mu \mid æ_{<t}) \geq w(\mu)$. Now we get a contradiction from (4.7):

$$V_\xi^*(æ_{<t}) \geq w(\mu|æ_{<t})V_\mu^*(æ_{<t}) \geq w(\mu)V_\mu^*(æ_{<t}) = \frac{w(\mu)}{2} > V_\xi^*(æ_{<t})$$

Therefore the Bayes agent stops taking the exploration action α after time step t_0, and so it is not optimal in any $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ that behaves like μ until time step t_0 and then changes:



□

The following theorem is also known as the *self-optimizing theorem*. This theorem has been a source of great confusion because its statement in Hutter (2005, Thm. 5.34) is not very explicit about how the histories are generated. The formulation of Lattimore (2013, Thm. 5.2) is explicit, but less general.

Theorem 5.23 (Sufficient Condition for Strong Asymptotic Optimality of Bayes; Hutter, 2005, Thm. 5.34)**.** *Let μ be some environment. If there is a policy π and a sequence of policies $\pi_1, \pi_2, \ldots$ such that for all $\nu \in \mathcal{M}$*

$$V_\nu^*(æ_{<t}) - V_\nu^{\pi_t}(æ_{<t}) \to 0 \text{ as } t \to \infty \ \mu^\pi\text{-almost surely,} \tag{5.4}$$

then

$$V_\mu^*(æ_{<t}) - V_\mu^{\pi_\xi^*}(æ_{<t}) \to 0 \text{ as } t \to \infty \ \mu^\pi\text{-almost surely.}$$

If $\pi = \pi^*_\xi$ and (5.4) holds for all $\mu \in \mathcal{M}$, then π^*_ξ is strongly asymptotically optimal in the class $\mathcal{M}$.

It is important to emphasize that the policies $\pi_1, \pi_2, \ldots$ need to converge to the optimal value on the history generated by μ and π, and not (as one might think) ν and π_t. Intuitively, the policy π is an 'exploration policy' that ensures that the environment class is explored sufficiently. Typically, a policy is asymptotically optimal on its own history. So if $\pi = \pi_1 = \pi_2 = \ldots$, then we get that Bayes is asymptotically optimal on the history generated by the policy π, not its own history. In light of Theorem 5.5 and Theorem 5.22 this is not too surprising; Bayesian reinforcement learning agents might not explore enough to be asymptotically optimal, but given a policy that does explore enough, Bayes learns enough to be asymptotically optimal.

This invites us to define the following policies π_t: follow the information-seeking policy π^*_{IG} until time step t, and then follow π^*_ξ (explore until t, then exploit). Since the information-seeking policy explores enough to prove off-policy prediction (Orseau et al., 2013, Thm. 7), we get $V^\pi_\xi - V^\pi_\mu \to 0$ for every policy π uniformly. Hence $\arg\max_\pi V^\pi_\xi \to \arg\max_\pi V^\pi_\mu$ and thus $V^*_\mu - V^{\pi^*_\xi}_\mu \to 0$ and (5.4) is satisfied. From Theorem 5.23 we get $V^*_\mu - V^{\pi^*_\xi}_\mu \to 0$, which we already knew. In order to get strong asymptotic optimality, all we need to do is choose the switching time step t appropriately, i.e., wait until V^*_μ and $V^{\pi^*_\xi}_\mu$ are close enough. Unfortunately, this is an invalid strategy: the agent does not know the true environment μ and hence cannot check this condition.

Hutter (2005, Sec. 5.6) uses Theorem 5.23 to show that the Bayes optimal policy is strongly asymptotically optimal in the class of ergodic finite-state MDPs if the effective horizon is growing, i.e., $H_t(\varepsilon) \to \infty$ for all $\varepsilon > 0$. This relies on the fact that in ergodic finite-state MDPs we need a fixed number of steps to explore the entire environment up to ε-confidence. Therefore we can define a sequence of policies $\pi_1, \pi_2, \ldots$ that completely disregard the history and start exploring everything from scratch. Since the effective horizon is growing, this exploration phase takes a vanishing fraction of effective horizon and most of the value is retained. Therefore the sequence of policies $\pi_1, \pi_2, \ldots$ satisfies the condition of Theorem 5.23 regardless of the history, thus in particular for the history generated by $\pi = \pi^*_\xi$ and any $\mu \in \mathcal{M}$. Note that the condition on the horizon is important: If the effective horizon is bounded, then Bayes is not asymptotically optimal in the class of ergodic finite-state MDPs because it can be locked into a dogmatic prior similarly to Theorem 5.5.

Proof of Theorem 5.23. From (4.6) we get for any history $æ_{<t}$

$$\begin{aligned} w(\mu \mid æ_{<t}) \left(V^*_\mu(æ_{<t}) - V^{\pi^*_\xi}_\mu(æ_{<t})\right) &\leq \sum_{\nu\in\mathcal{M}} w(\nu \mid æ_{<t}) \left(V^*_\nu(æ_{<t}) - V^{\pi^*_\xi}_\nu(æ_{<t})\right) \\ &= \left(\sum_{\nu\in\mathcal{M}} w(\nu \mid æ_{<t}) V^*_\nu(æ_{<t})\right) - V^{\pi^*_\xi}_\xi(æ_{<t}) \\ &\leq \sum_{\nu\in\mathcal{M}} w(\nu \mid æ_{<t}) V^*_\nu(æ_{<t}) - V^{\pi_t}_\xi(æ_{<t}) \end{aligned}$$

$$= \sum_{\nu \in \mathcal{M}} w(\nu \mid æ_{<t}) \big(V_\nu^*(æ_{<t}) - V_\nu^{\pi_t}(æ_{<t}) \big). \quad (5.5)$$

From (5.4) follows that $V_\nu^* - V_\nu^{\pi_t} \to 0$ μ^π-almost surely for all $\nu \in \mathcal{M}$, so (5.5) converges to 0 μ^π-almost surely (Hutter, 2005, Lem. 5.28ii). Similar to Example 3.20, $1/w(\mu \mid æ_{<t})$ is a nonnegative μ^π-martingale and thus converges (to a finite value) μ^π-almost surely by Theorem 2.8. Therefore $V_\mu^*(æ_{<t}) - V_\mu^{\pi_\xi^*}(æ_{<t}) \to 0$ μ^π-almost surely. If this is true for all $\mu \in \mathcal{M}$, the strong asymptotic optimality of π_ξ^* follows from $\pi = \pi_\xi^*$ by definition. □

5.4.2 BayesExp

The definition of BayesExp is given in Section 4.3.3. In this subsection we state a result by Lattimore (2013) that motivated the definition of BayesExp.

Theorem 5.24 (BayesExp is Weakly Asymptotically Optimal; Lattimore, 2013, Thm. 5.6)**.** *Let π_{BE} denote the policy from Algorithm 1. If $H_t(\varepsilon)$ grows monotone in t and $H_t(\varepsilon_t)/\varepsilon_t \in o(t)$, then for all environments $\mu \in \mathcal{M}$*

$$\frac{1}{t} \sum_{k=1}^{t} \big(V_\mu^*(æ_{<k}) - V_\mu^{\pi_{BE}}(æ_{<k}) \big) \to 0 \text{ as } t \to \infty \ \mu^{\pi_{BE}}\text{-almost surely.}$$

If the horizon grows sublinearly ($H_t(\varepsilon) \in o(t)$ for all $\varepsilon > 0$), then we can always find a sequence $\varepsilon_t \to 0$ that decreases slowly enough such that $H_t(\varepsilon)/\varepsilon_t \in o(t)$ holds.

5.4.3 Thompson Sampling

In this section we prove that the Thompson sampling policy defined in Section 4.3.4 is asymptotically optimal. Ortega and Braun (2010) prove that the action probabilities of Thompson sampling converge to the action probability of the optimal policy almost surely, but require a finite environment class and two (arguably quite strong) technical assumptions on the behavior of the posterior distribution (akin to ergodicity) and the similarity of environments in the class. Our convergence results do not require these assumptions.

Theorem 5.25 (Thompson Sampling is Asymptotically Optimal in Mean)**.** *For all environments $\mu \in \mathcal{M}$,*

$$\mathbb{E}_\mu^{\pi_T} \big[V_\mu^*(æ_{<t}) - V_\mu^{\pi_T}(æ_{<t}) \big] \to 0 \text{ as } t \to \infty.$$

This theorem immediately implies that Thompson sampling is also asymptotically optimal in probability according to Figure 5.2. However, this does not imply almost sure convergence (see Example 5.28).

We first give an intuition for the asymptotic optimality of Thompson sampling. At every resampling step we can split the class $\mathcal{M}$ into three partitions:

1. Environments ρ where $V_\mu^{\pi_\rho^*} \approx V_\mu^*$

2. Environments ρ where $V^*_\rho > V^*_\mu$

3. Environments ρ where $V^*_\rho < V^*_\mu$

The first class is the class of 'good' environments: if we draw one of them, we follow a policy that is close to optimal in μ. The second class is the class of environments that overestimate the value of μ. Following their optimal policy the agent gains information because rewards will be lower than expected. The third class is the class of environments that underestimate the value of μ. Following their optimal policy the agent might not gain information since μ might behave just like environment ρ on the ρ-optimal policy. However, when sampling from the first class instead, the agent gains information about the third class because rewards tend to be better than environments from the third class predicted.

Since the true environment $\mu \in \mathcal{M}$, the first class is not empty, and the probability of drawing a sample from the first class does not become too small. Whenever the second and third class have sufficiently high weight in the posterior, there is a good chance of picking a policy that leads the agent to gain information. Asymptotically, the posterior converges, so the agent ends up having learned everything it could, i.e., the posterior weight of the second and third class vanishes.

This argument is not too hard to formalize for deterministic environment classes. However, for stochastic environment classes the effect on the posterior when following a bad policy is harder to quantify because there is always a chance that the rewards are different simply because of bad luck. In order to prove this theorem in its generality for stochastic classes, we employ an entirely different proof strategy that relies on statistical tools rather than the argument given above.

Definition 5.26 (Expected Total Variation Distance)**.** Let π be any policy and let $m \in \mathbb{N} \cup \infty$. The *expected total variation distance* on the policy π is

$$F^\pi_m(æ_{<t}) := \sum_{\rho \in \mathcal{M}} w(\rho \mid æ_{<t}) D_m(\rho^\pi, \xi^\pi \mid æ_{<t}).$$

If we replace the distance measure D_m by cross-entropy, then the quantity $F^\pi_m(æ_{<t})$ becomes the expected information gain (see Section 4.3.2).

For the proof of Theorem 5.25 we need the following lemma.

Lemma 5.27 (Expected Total Variation Distance Vanishes On-Policy)**.** *For any policy π and any environment μ,*

$$\mathbb{E}^\pi_\mu[F^\pi_\infty(æ_{<t})] \to 0 \text{ as } t \to \infty.$$

Proof. From Theorem 3.25 we get $D_\infty(\mu^\pi, \xi^\pi \mid æ_{<t}) \to 0$ μ^π-almost surely, and since D is bounded, this convergence also occurs in mean. Thus for every environment $\nu \in \mathcal{M}$,

$$\mathbb{E}^\pi_\nu\big[D_\infty(\nu^\pi, \xi^\pi \mid æ_{<t})\big] \to 0 \text{ as } t \to \infty.$$

Now

$$\begin{aligned}\mathbb{E}_\mu^\pi[F_\infty^\pi(æ_{<t})] &\leq \frac{1}{w(\mu)}\mathbb{E}_\xi^\pi[F_\infty^\pi(æ_{<t})]\\ &= \frac{1}{w(\mu)}\mathbb{E}_\xi^\pi\left[\sum_{\nu\in\mathcal{M}} w(\nu \mid æ_{<t})D_\infty(\nu^\pi, \xi^\pi \mid æ_{<t})\right]\\ &= \frac{1}{w(\mu)}\mathbb{E}_\xi^\pi\left[\sum_{\nu\in\mathcal{M}} w(\nu)\frac{\nu^\pi(æ_{<t})}{\xi^\pi(æ_{<t})}D_\infty(\nu^\pi, \xi^\pi \mid æ_{<t})\right]\\ &= \frac{1}{w(\mu)}\sum_{\nu\in\mathcal{M}} w(\nu)\mathbb{E}_\nu^\pi\left[D_\infty(\nu^\pi, \xi^\pi \mid æ_{<t})\right] \to 0\end{aligned}$$

by Hutter (2005, Lem. 5.28ii) since total variation distance is bounded. □

Proof of Theorem 5.25. Let $\beta, \delta > 0$ and let $\varepsilon_t > 0$ denote the sequence used to define π_T in Algorithm 2. We assume that t is large enough such that $\varepsilon_k \leq \beta$ for all $k \geq t$ and that δ is small enough such that $w(\mu \mid æ_{<t}) > 4\delta$ for all t, which holds since $w(\mu \mid æ_{<t}) \not\to 0$ μ^π-almost surely for any policy π (Hutter, 2009a, Lem. 3i).

The stochastic process $w(\nu \mid æ_{<t})$ is a ξ^{π_T}-martingale according to Example 3.20. By the martingale convergence theorem (Theorem 2.8) $w(\nu \mid æ_{<t})$ converges ξ^{π_T}-almost surely and because $\xi^{\pi_T} \geq w(\mu)\mu^{\pi_T}$ it also converges μ^{π_T}-almost surely.

We argue that we can choose t_0 to be one of π_T's resampling time steps large enough such that for all $t \geq t_0$ the following three events hold simultaneously with μ^{π_T}-probability at least $1 - \delta$.

(i) There is a finite set $\mathcal{M}' \subset \mathcal{M}$ with $w(\mathcal{M}' \mid æ_{<t}) > 1 - \delta$ and $w(\nu \mid æ_{<k}) \not\to 0$ as $k \to \infty$ for all $\nu \in \mathcal{M}'$.

(ii) $|w(\mathcal{M}'' \mid æ_{<t}) - w(\mathcal{M}'' \mid æ_{<t_0})| \leq \delta$ for all $\mathcal{M}'' \subseteq \mathcal{M}'$.

(iii) $F_\infty^{\pi_T}(æ_{<t}) < \delta\beta w_{\min}^2$.

where $w_{\min} := \inf\{w(\nu \mid æ_{<k}) \mid k \in \mathbb{N}, \nu \in \mathcal{M}'\}$, which is positive by (i).

(i) and (ii) are satisfied eventually because the posterior $w(\,\cdot \mid æ_{<t})$ converges μ^{π_T}-almost surely. Note that the set $\mathcal{M}'$ is random: the limit of $w(\nu \mid æ_{<t})$ as $t \to \infty$ depends on the history $æ_{1:\infty}$. Without loss of generality, we assume the true environment μ is contained in $\mathcal{M}'$ since $w(\mu \mid æ_{<t}) \not\to 0$ μ^{π_T}-almost surely. (iii) follows from Lemma 5.27 since convergence in mean implies convergence in probability.

Moreover, we define the horizon $m := t + H_t(\varepsilon_t)$ as the time step of the effective horizon at time step t. Let $æ_{<t}$ be a fixed history for which (i-iii) is satisfied. Then we have

$$\begin{aligned}\delta\beta w_{\min}^2 &> F_\infty^{\pi_T}(æ_{<t})\\ &= \sum_{\nu\in\mathcal{M}} w(\nu \mid æ_{<t})D_\infty(\nu^{\pi_T}, \xi^{\pi_T} \mid æ_{<t})\\ &= \mathbb{E}_{\nu\sim w(\,\cdot\mid æ_{<t})}\left[D_\infty(\nu^{\pi_T}, \xi^{\pi_T} \mid æ_{<t})\right]\end{aligned}$$

$$\begin{aligned}&\geq \mathbb{E}_{\nu\sim w(\,\cdot\,|æ_{<t})}\left[D_m(\nu^{\pi_T},\xi^{\pi_T}\mid æ_{<t})\right]\\&\geq \beta w_{\min}^2 w(\mathcal{M}\setminus\mathcal{M}''\mid æ_{<t})\end{aligned}$$

by Markov's inequality where

$$\mathcal{M}'' := \left\{\nu\in\mathcal{M}\;\middle|\; D_m(\nu^{\pi_T},\xi^{\pi_T}\mid æ_{<t})<\beta w_{\min}^2\right\}.$$

For our fixed history $æ_{<t}$ we have

$$\begin{aligned}1-\delta &< w(\mathcal{M}''\mid æ_{<t})\\&\overset{(i)}{\leq} w(\mathcal{M}''\cap\mathcal{M}'\mid æ_{<t})+\delta\\&\overset{(ii)}{\leq} w(\mathcal{M}''\cap\mathcal{M}'\mid æ_{<t_0})+2\delta\\&\overset{(i)}{\leq} w(\mathcal{M}''\mid æ_{<t_0})+3\delta\end{aligned}$$

and thus we get

$$1-4\delta < w\left(\{\nu\in\mathcal{M}\mid D_m(\nu^{\pi_T},\xi^{\pi_T}\mid æ_{<t})<\beta w_{\min}^2\}\;\middle|\; æ_{<t_0}\right). \tag{5.6}$$

In particular, this bound holds for $\nu=\mu$ since $w(\mu\mid æ_{<t_0})>4\delta$ by assumption.

It remains to show that with high probability the value $V_\mu^{\pi_\rho^*}$ of the sample ρ's optimal policy π_ρ^* is sufficiently close to the μ-optimal value V_μ^*. The worst case is that we draw the worst sample from $\mathcal{M}'\cap\mathcal{M}''$ twice in a row. From now on, let ρ denote the sample environment we draw at time step t_0, and let t denote some time step between t_0 and $t_1 := t_0 + H_{t_0}(\varepsilon_{t_0})$ (before the next resampling). With probability $w(\nu'\mid æ_{<t_0})w(\nu'\mid æ_{<t_1})$ we sample ν' both at t_0 and t_1 when following π_T. Therefore we have for all $æ_{t:m}$ and all $\nu\in\mathcal{M}$

$$\nu^{\pi_T}(æ_{1:m}\mid æ_{<t})\geq w(\nu'\mid æ_{<t_0})w(\nu'\mid æ_{<t_1})\nu'^{\pi_{\nu'}^*}(æ_{1:m}\mid æ_{<t}).$$

Thus we get for all $\nu\in\mathcal{M}'$ (in particular ρ and μ)

$$\begin{aligned}D_m(\mu^{\pi_T},\rho^{\pi_T}\mid æ_{<t}) &\geq \sup_{\nu'\in\mathcal{M}}\sup_{A\subseteq(\mathcal{A}\times\mathcal{E})^m}\Big|w(\nu'\mid æ_{<t_0})w(\nu'\mid æ_{<t_1})\\&\qquad\qquad(\mu^{\pi_{\nu'}^*}(A\mid æ_{<t})-\rho^{\pi_{\nu'}^*}(A\mid æ_{<t}))\Big|\\&\geq w(\nu\mid æ_{<t_0})w(\nu\mid æ_{<t_1})\sup_{A\subseteq(\mathcal{A}\times\mathcal{E})^m}\left|\mu^{\pi_\nu^*}(A\mid æ_{<t})-\rho^{\pi_\nu^*}(A\mid æ_{<t})\right|\\&\geq w_{\min}^2 D_m(\mu^{\pi_\nu^*},\rho^{\pi_\nu^*}\mid æ_{<t}).\end{aligned}$$

For $\rho\in\mathcal{M}''$ we get with (5.6)

$$\begin{aligned}D_m(\mu^{\pi_T},\rho^{\pi_T}\mid æ_{<t}) &\leq D_m(\mu^{\pi_T},\xi^{\pi_T}\mid æ_{<t})+D_m(\rho^{\pi_T},\xi^{\pi_T}\mid æ_{<t})\\&< \beta w_{\min}^2+\beta w_{\min}^2 = 2\beta w_{\min}^2,\end{aligned}$$

which together with Lemma 4.17 and the fact that rewards in $[0,1]$ implies

$$\begin{aligned}\left|V_\mu^{\pi_\nu^*}(æ_{<t}) - V_\rho^{\pi_\nu^*}(æ_{<t})\right| &\leq \frac{\Gamma_{t+H_t(\varepsilon_t)}}{\Gamma_t} + \left|V_\mu^{\pi_\nu^*,m}(æ_{<t}) - V_\rho^{\pi_\nu^*,m}(æ_{<t})\right| \\ &\leq \varepsilon_t + D_m(\mu^{\pi_\nu^*}, \rho^{\pi_\nu^*} \mid æ_{<t}) \\ &\leq \varepsilon_t + \tfrac{1}{w_{\min}^2} D_m(\mu^{\pi_T}, \rho^{\pi_T} \mid æ_{<t}) \\ &< \beta + 2\beta = 3\beta.\end{aligned}$$

Hence we get (omitting history arguments $æ_{<t}$ for simplicity)

$$V_\mu^* = V_\mu^{\pi_\mu^*} < V_\rho^{\pi_\mu^*} + 3\beta \leq V_\rho^* + 3\beta = V_\rho^{\pi_\rho^*} + 3\beta < V_\mu^{\pi_\rho^*} + 3\beta + 3\beta = V_\mu^{\pi_\rho^*} + 6\beta. \quad (5.7)$$

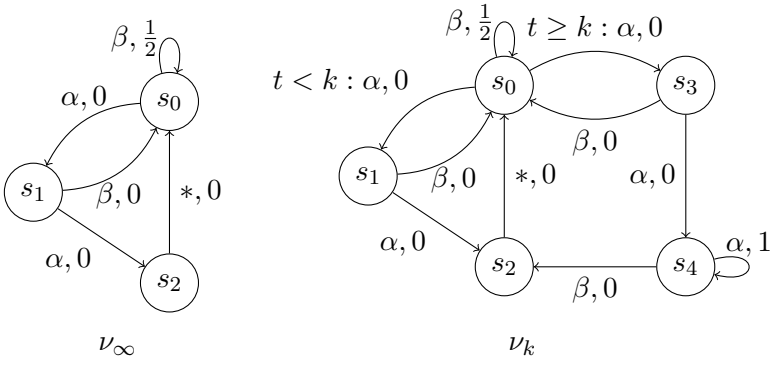
With μ^{π_T}-probability at least $1-\delta$ (i), (ii), and (iii) are true, with μ^{π_T}-probability at least $1-\delta$ our sample ρ happens to be in $\mathcal{M}'$ by (i), and with $w(\,\cdot \mid æ_{<t_0})$-probability at least $1-4\delta$ the sample is in $\mathcal{M}''$ by (5.6). All of these events are true simultaneously with probability at least $1-(\delta+\delta+4\delta) = 1-6\delta$. Hence the bound (5.7) transfers for π_T such that with μ^{π_T}-probability $\geq 1-6\delta$ we have

$$V_\mu^*(æ_{<t}) - V_\mu^{\pi_T}(æ_{<t}) < 6\beta.$$

Therefore $\mu^{\pi_T}[V_\mu^*(æ_{<t}) - V_\mu^{\pi_T}(æ_{<t}) \geq 6\beta] < 6\delta$ and with $\delta \to 0$ we get that $V_\mu^*(æ_{<t}) - V_\mu^{\pi_T}(æ_{<t}) \to 0$ as $t \to \infty$ in probability. The value function is bounded, thus it also converges in mean. □

The following example shows that the Thompson sampling policy is not strongly asymptotically optimal. However, we expect that strong asymptotic optimality can be achieved with Thompson sampling by resampling at every time step (with strong assumptions on the discount function). However, for practical purposes resampling in every time step is very inefficient.

Example 5.28 (Thompson Sampling is not Strongly Asymptotically Optimal)**.** Define $\mathcal{A} := \{\alpha, \beta\}$, $\mathcal{E} := \{0, 1/2, 1\}$, and assume geometric discounting (Example 4.5). Consider the following class of environments $\mathcal{M} := \{\nu_\infty, \nu_1, \nu_2, \ldots\}$ (transitions are labeled with action, reward):



Environment ν_k works just like environment ν_∞ except that after time step k, the path to state s_3 gets unlocked. The class $\mathcal{M}$ is a class of deterministic weakly communicating POMDPs (but as a POMDP ν_k has more than 5 states). The optimal policy in environment ν_∞ is to always take action β, the optimal policy for environment ν_k is to take action β for $t < k$ and then take action β in state s_1 and action α otherwise.

Suppose the policy π_T is acting in environment ν_∞. Since it is asymptotically optimal in the class $\mathcal{M}$, it has to take actions $\alpha\alpha$ from s_0 infinitely often: for $t < k$ environment ν_k is indistinguishable from ν_∞, so the posterior for ν_k is larger or equal to the prior. Hence there is always a constant chance of sampling ν_k until taking actions $\alpha\alpha$, at which point all environments ν_k for $k \leq t$ become falsified.

If the policy π_T decides to explore and take the first action α, it will be in state s_1. Let $æ_{<t}$ denote the current history. Then the ν_∞-optimal action is β and

$$V^*_{\nu_\infty}(æ_{<t}) = (1-\gamma)\left(0 + \gamma\frac{1}{2} + \gamma^2\frac{1}{2} + \dots\right) = \frac{\gamma}{2}.$$

The next action taken by π_T is α since any optimal policy for any sampled environment that takes action α once, takes that action again (and we are following that policy for an ε_t-effective horizon). Hence

$$V^{\pi_T}_{\nu_\infty}(æ_{<t}) \leq (1-\gamma)\left(0 + 0 + \gamma^2\frac{1}{2} + \gamma^3\frac{1}{2} + \dots\right) = \frac{\gamma^2}{2}.$$

Therefore $V^*_{\nu_\infty} - V^{\pi_T}_{\nu_\infty} \geq (\gamma - \gamma^2)/2 > 0$. This happens infinitely often with probability one and thus we cannot get almost sure convergence. ◇

If the Bayesian mixture ξ is inside the class $\mathcal{M}$ (as it is the case for the class $\mathcal{M}^{\text{CCS}}_{\text{LSC}}$), then we can assign ξ a prior probability that is arbitrarily close to 1. Since the posterior of ξ is the same as the prior, Thompson sampling will act according to the Bayes optimal policy most of the time. This means the Bayes-value of Thompson sampling can be very good; formally, $V^*_\xi(\epsilon) - V^{\pi_T}_\xi(\epsilon) = \overline{\Upsilon}_\xi - \Upsilon_\xi(\pi_T)$ can be made arbitrarily small.

In contrast, the Bayes-value of Thompson sampling can also be very bad: Suppose you have a class of $(n+1)$-armed bandits indexed $1, \dots, n$ where bandit i gives reward $1-\varepsilon$ on arm 1, reward 1 on arm $i+1$, and reward 0 on all other arms. For geometric discounting and $\varepsilon < (1-\gamma)/(2-\gamma)$, it is Bayes optimal to pull arm 1 while Thompson sampling will explore on average $n/2$ arms until it finds the optimal arm. The Bayes-value of Thompson sampling is $1/(n-\gamma_{n-1})$ in contract to $(1-\varepsilon)$ achieved by Bayes. For a horizon of n, the Bayes optimal policy suffers a regret of εn and Thompson sampling a regret of $n/2$, which is much larger for small ε.

5.4.4 Almost Sure in Cesàro Average vs. in Mean

It might appear that convergence in mean is more natural than the convergence of Cesàro averages of weak asymptotic optimality. However, both notions are not so fundamentally different because they both allow an infinite number of bad mistakes (actions that lead to $V^*_\mu - V^\pi_\mu$ being large). Asymptotic optimality in mean allows bad

mistakes as long as their probability converges to zero; weak asymptotic optimality allows bad mistakes as long as the total time spent on bad mistakes grows sublinearly. Note that according to Example 5.19 making bad mistakes infinitely often is necessary for asymptotic optimality.

Theorem 5.24 shows that weak asymptotic optimality is possible in any countable class of stochastic environments. However, this requires the additional condition that the effective horizon grows sublinearly, $H_t(\varepsilon_t) \in o(t)$, while Theorem 5.25 does not require any condition on the discount function.

Generally, weak asymptotic optimality and asymptotic optimality in mean are incomparable because the notions of convergence are incomparable for (bounded) random variables. First, for deterministic sequences (i.e. deterministic policies in deterministic environments), convergence in mean is equivalent to (regular) convergence, which is impossible by Theorem 5.20. Second, convergence in probability (and hence convergence in mean for bounded random variables) does not imply almost sure convergence of Cesàro averages (Stoyanov, 2013, Sec. 14.18). We leave open the question whether the policy π_T is weakly asymptotically optimal.

5.5 Regret

Regret is how many expected rewards the agent forfeits by not following the best informed policy.

Definition 5.29 (Regret)**.** The *regret* of a policy π in environment μ is

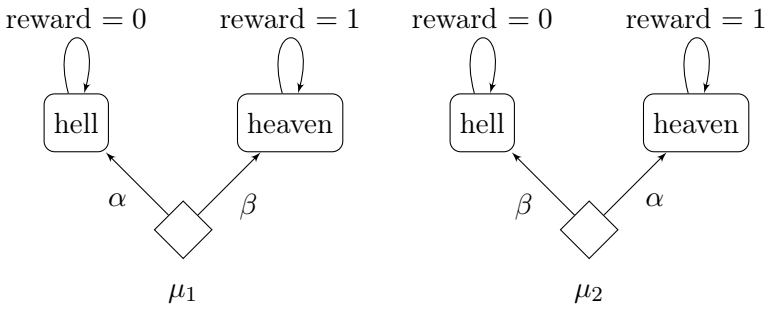
$$R_m(\pi, \mu) := \sup_{\pi'} \mathbb{E}_\mu^{\pi'} \left[\sum_{t=1}^{m} r_t \right] - \mathbb{E}_\mu^{\pi} \left[\sum_{t=1}^{m} r_t \right].$$

Note that regret is undiscounted and always nonnegative. Moreover, the space of possible different policies for the first m actions is finite and we assumed the set of actions $\mathcal{A}$ and the set of percepts $\mathcal{E}$ to be finite (Assumption 4.6c), so the supremum is always attained by some policy (not necessarily the μ-optimal policy π_μ^* because that policy uses discounting).

Different problem classes have different regret rates, depending on the structure and the difficulty of the problem class. Multi-armed bandits provide a (problem-independent) worst-case regret bound of $\Omega(\sqrt{km})$ where k is the number of arms (Bubeck and Bianchi, 2012). In MDPs the lower bound is $\Omega(\sqrt{SAdm})$ where S is the number of states, A the number of actions, and d the diameter of the MDP (Auer et al., 2010). For a countable class of environments given by state representation functions that map histories to MDP states, a regret of $\tilde{O}(m^{2/3})$ is achievable assuming the resulting MDP is weakly communicating (Nguyen et al., 2013).

A problem class is considered *learnable* if there is an algorithm that has a sublinear regret guarantee. The following example shows that the general reinforcement learning problem is not learnable because the agent can get caught in a trap and be unable to recover.

Example 5.30 (Linear Regret; Hutter, 2005, Sec. 5.3.2)**.** Consider the following two environments μ_1 and μ_2. In environment μ_1 action α leads to hell (reward 0 forever) and action β leads to heaven (reward 1 forever). Environment μ_2 behaves just the same, except that both actions are swapped.



The policy α that takes action α in the first time step performs well in μ_2 but performs poorly in μ_1. Likewise, the policy β that takes action β in the first time step performs well in μ_1 but performs poorly in μ_2. Regardless of which policy we adopt, our regret is always linear in one of the environments μ_1 or μ_2:

$$R_m(\alpha, \mu_1) = m \qquad R_m(\alpha, \mu_2) = 0$$
$$R_m(\beta, \mu_1) = 0 \qquad R_m(\beta, \mu_2) = m \qquad \diamond$$

To achieve sublinear regret we need to ensure that the agent can recover from mistakes. Formally, we make the following assumption.

Definition 5.31 (Recoverability)**.** An environment μ satisfies the *recoverability assumption* iff

$$\sup_{\pi} \left| \mathbb{E}_{\mu}^{\pi^*_\mu}[V^*_\mu(æ_{<t})] - \mathbb{E}_{\mu}^{\pi}[V^*_\mu(æ_{<t})] \right| \to 0 \text{ as } t \to \infty.$$

Recoverability compares following the worst policy π for $t-1$ time steps and then switching to the optimal policy π^*_ν to having followed π^*_ν from the beginning. The recoverability assumption states that switching to the optimal policy at any time step enables the recovery of most of the value: it has to become less costly to recover from mistakes as time progresses. This should be regarded as an effect of the discount function: if the (effective) horizon grows, recovery becomes easier because the optimal policy has more time to perform a recovery. Moreover, recoverability is on the optimal policy, in contrast to the notion of ergodicity in MDPs which demands returning to a starting state regardless of the policy.

Remark 5.32 (Weakly Communicating POMDPs are Recoverable)**.** If the effective horizon is growing, $H_t(\varepsilon) \to \infty$ as $t \to \infty$, then any weakly communicating finite state POMDP satisfies the recoverability assumption. $\diamond$

5.5.1 Sublinear Regret in Recoverable Environments

This subsection is dedicated to the following theorem that connects asymptotic optimality in mean to sublinear regret.

Theorem 5.33 (Sublinear Regret in Recoverable Environments)**.** *If the discount function γ satisfies Assumption 5.34, the environment μ satisfies the recoverability assumption, and π is asymptotically optimal in mean in the class $\{\mu\}$, then $R_m(\pi, \mu) \in o(m)$.*

Assumption 5.34 (Discount Function)**.** *Let the discount function γ be such that*

(a) $\gamma_t > 0$ for all t,

(b) γ_t is monotone decreasing in t, and

(c) $H_t(\varepsilon) \in o(t)$ for all $\varepsilon > 0$.

This assumption demands that the discount function is somewhat well-behaved: the function has no oscillations, does not become 0, and the horizon is not growing too fast. It is satisfied by geometric discounting (Example 4.5): (a) $\gamma^t > 0$, (b) γ monotone decreasing, and (c) $H_t(\varepsilon) = \lceil \log_\gamma \varepsilon \rceil \in o(t)$.

The problem with geometric discounting is that it makes the recoverability assumption very strong: since the horizon is not growing, the environment has to enable *faster recovery* as time progresses; in this case weakly communicating POMDPs are *not* recoverable. A choice with $H_t(\varepsilon) \to \infty$ that satisfies Assumption 5.34 is subgeometric discounting $\gamma_t := e^{-\sqrt{t}}/\sqrt{t}$ (see Table 4.1).

If the items in Assumption 5.34 are violated, Theorem 5.33 can fail:

- If $\gamma_t = 0$ for some time steps t, our policy does not care about those time steps and might take actions that have large regret.
- Similarly if γ oscillates between high values and very low values: our policy might take high-regret actions in time steps with comparatively lower γ-weight.
- If the horizon grows linearly, infinitely often our policy might spend some constant fraction of the current effective horizon exploring, which incurs a cost that is a constant fraction of the total regret so far.

To prove Theorem 5.33 we require the following technical lemma.

Lemma 5.35 (Value and Regret)**.** *Let $\varepsilon > 0$ and assume the discount function γ satisfies Assumption 5.34. Let $(d_t)_{t \in \mathbb{N}}$ be a sequence of numbers with $|d_t| \leq 1$ for all t. If there is a time step t_0 with*

$$\frac{1}{\Gamma_t} \sum_{k=t}^{\infty} \gamma_k d_k < \varepsilon \quad \forall t \geq t_0 \tag{5.8}$$

then

$$\sum_{t=1}^{m} d_t \leq t_0 + \varepsilon(m - t_0 + 1) + \frac{1+\varepsilon}{1-\varepsilon} H_m(\varepsilon).$$

Proof. This proof essentially follows the proof of Hutter (2006b, Thm. 17).

By Assumption 5.34a we have $\gamma_t > 0$ for all t and hence $\Gamma_t > 0$ for all t. By Assumption 5.34b have that γ is monotone decreasing, so we get for all $n \in \mathbb{N}$

$$\Gamma_t = \sum_{k=t}^{\infty} \gamma_k \leq \sum_{k=t}^{t+n-1} \gamma_t + \sum_{k=t+n}^{\infty} \gamma_k = n\gamma_t + \Gamma_{t+n}.$$

And with $n := H_t(\varepsilon)$ this yields

$$\frac{\gamma_t H_t(\varepsilon)}{\Gamma_t} \geq 1 - \frac{\Gamma_{t+H_t(\varepsilon)}}{\Gamma_t} \geq 1 - \varepsilon > 0. \tag{5.9}$$

In particular, this bound holds for all t and $\varepsilon > 0$.

Next, we define a series of nonnegative weights $(b_t)_{t\geq 1}$ such that

$$\sum_{t=t_0}^{m} d_k = \sum_{t=t_0}^{m} \frac{b_t}{\Gamma_t} \sum_{k=t}^{m} \gamma_k d_k.$$

This yields the constraints

$$\sum_{k=t_0}^{t} \frac{b_k}{\Gamma_k} \gamma_t = 1 \quad \forall t \geq t_0.$$

The solution to these constraints is

$$b_{t_0} = \frac{\Gamma_{t_0}}{\gamma_{t_0}}, \text{ and } b_t = \frac{\Gamma_t}{\gamma_t} - \frac{\Gamma_t}{\gamma_{t-1}} \text{ for } t > t_0. \tag{5.10}$$

Thus we get

$$\begin{aligned}
\sum_{t=t_0}^{m} b_t &= \frac{\Gamma_{t_0}}{\gamma_{t_0}} + \sum_{t=t_0+1}^{m} \left(\frac{\Gamma_t}{\gamma_t} - \frac{\Gamma_t}{\gamma_{t-1}} \right) \\
&= \frac{\Gamma_{m+1}}{\gamma_m} + \sum_{t=t_0}^{m} \left(\frac{\Gamma_t}{\gamma_t} - \frac{\Gamma_{t+1}}{\gamma_t} \right) \\
&= \frac{\Gamma_{m+1}}{\gamma_m} + m - t_0 + 1 \\
&\leq \frac{H_m(\varepsilon)}{1-\varepsilon} + m - t_0 + 1
\end{aligned}$$

for all $\varepsilon > 0$ according to (5.9).

Finally,

$$\begin{aligned}
\sum_{t=1}^{m} d_t &\leq \sum_{t=1}^{t_0} d_t + \sum_{t=t_0}^{m} \frac{b_t}{\Gamma_t} \sum_{k=t}^{m} \gamma_k d_k \\
&\leq t_0 + \sum_{t=t_0}^{m} \frac{b_t}{\Gamma_t} \sum_{k=t}^{\infty} \gamma_k d_k - \sum_{t=t_0}^{m} \frac{b_t}{\Gamma_t} \sum_{k=m+1}^{\infty} \gamma_k d_k
\end{aligned}$$

and using the assumption (5.8) and $d_t \geq -1$,

$$\begin{aligned} &< t_0 + \sum_{t=t_0}^{m} b_t \varepsilon + \sum_{t=t_0}^{m} \frac{b_t \Gamma_{m+1}}{\Gamma_t} \\ &\leq t_0 + \frac{\varepsilon H_m(\varepsilon)}{1-\varepsilon} + \varepsilon(m - t_0 + 1) + \sum_{t=t_0}^{m} \frac{b_t \Gamma_{m+1}}{\Gamma_t} \end{aligned}$$

For the latter term we substitute (5.10) to get

$$\sum_{t=t_0}^{m} \frac{b_t \Gamma_{m+1}}{\Gamma_t} = \frac{\Gamma_{m+1}}{\gamma_{t_0}} + \sum_{t=t_0+1}^{m} \left(\frac{\Gamma_{m+1}}{\gamma_t} - \frac{\Gamma_{m+1}}{\gamma_{t-1}} \right) = \frac{\Gamma_{m+1}}{\gamma_m} \leq \frac{H_m(\varepsilon)}{1-\varepsilon}$$

with (5.9). □

Proof of Theorem 5.33. Let $(\pi_m)_{m \in \mathbb{N}}$ denote any sequence of policies, such as a sequence of policies that attain the supremum in the definition of regret. We want to show that

$$\mathbb{E}_\mu^{\pi_m} \left[\sum_{t=1}^{m} r_t \right] - \mathbb{E}_\mu^{\pi} \left[\sum_{t=1}^{m} r_t \right] \in o(m).$$

For

$$d_k^{(m)} := \mathbb{E}_\mu^{\pi_m}[r_k] - \mathbb{E}_\mu^{\pi}[r_k] \tag{5.11}$$

we have $-1 \leq d_k^{(m)} \leq 1$ since we assumed rewards to be bounded between 0 and 1. Because the environment μ satisfies the recoverability assumption we have

$$\begin{aligned} &\left| \mathbb{E}_\mu^{\pi_\mu^*}[V_\mu^*(æ_{<t})] - \mathbb{E}_\mu^{\pi}[V_\mu^*(æ_{<t})] \right| \to 0 \text{ as } t \to \infty, \text{ and} \\ &\sup_m \left| \mathbb{E}_\mu^{\pi_\mu^*}[V_\mu^*(æ_{<t})] - \mathbb{E}_\mu^{\pi_m}[V_\mu^*(æ_{<t})] \right| \to 0 \text{ as } t \to \infty, \end{aligned}$$

so we conclude that

$$\sup_m \left| \mathbb{E}_\mu^{\pi}[V_\mu^*(æ_{<t})] - \mathbb{E}_\mu^{\pi_m}[V_\mu^*(æ_{<t})] \right| \to 0$$

by the triangle inequality and thus

$$\sup_m \mathbb{E}_\mu^{\pi_m}[V_\mu^*(æ_{<t})] - \mathbb{E}_\mu^{\pi}[V_\mu^*(æ_{<t})] \to 0 \text{ as } t \to \infty. \tag{5.12}$$

By assumption the policy π is asymptotically optimal in mean, so we have

$$\mathbb{E}_\mu^{\pi}[V_\mu^*(æ_{<t})] - \mathbb{E}_\mu^{\pi}[V_\mu^{\pi}(æ_{<t})] \to 0 \text{ as } t \to \infty,$$

and with (5.12) this combines to

$$\sup_m \mathbb{E}_\mu^{\pi_m}[V_\mu^*(æ_{<t})] - \mathbb{E}_\mu^{\pi}[V_\mu^{\pi}(æ_{<t})] \to 0 \text{ as } t \to \infty.$$

From $V_\mu^*(æ_{<t}) \geq V_\mu^{\pi_m}(æ_{<t})$ we get

$$\limsup_{t\to\infty}\left(\sup_m \mathbb{E}_\mu^{\pi_m}[V_\mu^{\pi_m}(æ_{<t})] - \mathbb{E}_\mu^{\pi}[V_\mu^{\pi}(æ_{<t})]\right) \leq 0. \tag{5.13}$$

For $\pi' \in \{\pi, \pi_1, \pi_2, \ldots\}$ we have

$$\mathbb{E}_\mu^{\pi'}[V_\mu^{\pi'}(æ_{<t})] = \mathbb{E}_\mu^{\pi'}\left[\frac{1}{\Gamma_t}\mathbb{E}_\mu^{\pi'}\left[\sum_{k=t}^{\infty}\gamma_k r_k \,\middle|\, æ_{<t}\right]\right] = \mathbb{E}_\mu^{\pi'}\left[\frac{1}{\Gamma_t}\sum_{k=t}^{\infty}\gamma_k r_k\right] = \frac{1}{\Gamma_t}\sum_{k=t}^{\infty}\gamma_k \mathbb{E}_\mu^{\pi'}[r_k],$$

so from (5.11) and (5.13) we get

$$\limsup_{t\to\infty}\sup_m \frac{1}{\Gamma_t}\sum_{k=t}^{\infty}\gamma_k d_k^{(m)} \leq 0.$$

Let $\varepsilon > 0$. We choose t_0 independent of m and large enough such that we get $\sup_m \sum_{k=t}^{\infty}\gamma_k d_k^{(m)}/\Gamma_t < \varepsilon$ for all $t \geq t_0$. Now we let $m \in \mathbb{N}$ be given and apply Lemma 5.35 to get

$$\frac{R_m(\pi,\mu)}{m} = \frac{\sum_{k=1}^{m} d_k^{(m)}}{m} \leq \frac{t_0 + \varepsilon(m - t_0 + 1) + \frac{1+\varepsilon}{1-\varepsilon}H_m(\varepsilon)}{m}.$$

Since $H_t(\varepsilon) \in o(t)$ according to Assumption 5.34c we get $\limsup_{m\to\infty} R_m(\pi,\mu)/m \leq 0$. □

Example 5.36 (The Converse of Theorem 5.33 is False)**.** Let μ be a two-armed Bernoulli bandit with means 0 and 1 and suppose we are using geometric discounting with discount factor $\gamma \in [0, 1)$. This environment is recoverable. If our policy π pulls the suboptimal arm exactly on time steps $1, 2, 4, 8, 16, \ldots$, regret will be logarithmic. However, on time steps $t = 2^n$ for $n \in \mathbb{N}$ the value difference $V_\mu^* - V_\mu^\pi$ is deterministically at least $1 - \gamma > 0$. ◇

Note that Example 5.36 does not rule out weak asymptotic optimality.

5.5.2 Regret of the Optimal Policy and Thompson sampling

We get the following immediate consequence.

Corollary 5.37 (Sublinear Regret for the Optimal Discounted Policy)**.** *If the discount function γ satisfies Assumption 5.34 and the environment μ satisfies the recoverability assumption, then $R_m(\pi_\mu^*, \mu) \in o(m)$.*

Proof. From Theorem 5.33 since the policy π_μ^* is (trivially) asymptotically optimal in $\{\mu\}$. □

If the environment does not satisfy the recoverability assumption, regret may be linear *even on the optimal policy*: the optimal policy maximizes discounted rewards

and this short-sightedness might incur a tradeoff that leads to linear regret later on if the environment does not allow recovery.

Corollary 5.38 (Sublinear Regret for Thompson Sampling)**.** *If the discount function γ satisfies Assumption 5.34 and the environment $\mu \in \mathcal{M}$ satisfies the recoverability assumption, then $R_m(\pi_T, \mu) \in o(m)$ for the Thompson sampling policy π_T.*

Proof. From Theorem 5.25 and Theorem 5.33. □

5.6 Discussion

In this work, we disregard computational constraints. Because of this, our agents learn very efficiently and we can focus on the way they balance exploration and exploitation. So which balance is best?

5.6.1 The Optimality of AIXI

Bayesian reinforcement learning agents make the tradeoff between exploration and exploitation in the Bayes optimal way. Maximizing expected rewards according to any positive prior does not lead to enough exploration to achieve asymptotic optimality (Theorem 5.22); the prior's bias is retained indefinitely. For bad priors this can cause serious malfunctions: the dogmatic prior defined in Section 5.2.2 can prevent a Bayesian agent *from taking a single exploratory action*; exploration is restricted to cases where the expected future payoff falls below some prespecified $\varepsilon > 0$. However, this problem can be alleviated by adding an extra exploration component to AIXI: Lattimore (2013) shows that BayesExp is weakly asymptotically optimal (Theorem 5.24).

So instead, we may ask the following weaker questions. Does AIXI succeed in every (ergodic) finite-state (PO)MDP, bandit problem, or sequence prediction task? Our results imply that without further assumptions on the prior, we cannot answer any of the preceding questions in the affirmative. Using a dogmatic prior (Theorem 5.5), we can make AIXI follow any computable policy as long as that policy produces rewards that are bounded away from zero.

- In a sequence prediction task that gives a reward of 1 for every correctly predicted bit and 0 otherwise, a policy π that correctly predicts every third bit will receive an average reward of 1/3. With a π-dogmatic prior, AIXI thus only predicts a third of the bits correctly, and hence is outperformed by a uniformly random predictor.

  However, if we have a constant horizon of length 1, AIXI *does* succeed in sequence prediction (Hutter, 2005, Sec. 6.2.2). If the horizon is this short, the agent is so hedonistic that no threat of hell can deter it.

- In a (PO)MDP a dogmatic prior can make AIXI get stuck in any loop that provides nonzero expected rewards.

- In a bandit problem, a dogmatic prior can make AIXI get stuck on any arm which provides nonzero expected rewards.

These results apply not only to AIXI, but generally to Bayesian reinforcement learning agents. Any Bayesian mixture over nonrecoverable environments is susceptible to dogmatic priors if we allow an arbitrary reweighing of the prior. Notable exceptions are classes of environment that allow policies that are strongly asymptotically optimal *regardless of the history* (Theorem 5.23). For example, the class of all ergodic MDPs for an unbounded effective horizon; in this case the Bayes optimal policy is strongly asymptotically optimal (Hutter, 2005, Thm. 5.38). Note that in contrast to our results, this requires that that the agent uses a Bayes-mixture over a class of ergodic MDPs.

Moreover, Bayesian agents still perform well at learning and achieve on-policy value convergence (Corollary 4.20): the posterior belief about the value of a policy π converges to the true value of π while following π: $V^\pi_\xi(æ_{<t}) - V^\pi_\mu(æ_{<t}) \to 0$ as $t \to \infty$ μ^π-almost surely. Since this holds for any policy, in particular it holds for the Bayes optimal policy π^*_ξ. This means that the Bayes agent learns to predict those parts of the environment that it sees. But if it does not explore enough, then it will not learn other parts of the environment that are potentially more rewarding.

Hutter (2005, Claim 5.12) claims:

> We expect AIXI to be universally optimal.

Our work seriously challenges Hutter's claim: no nontrivial and non-subjective optimality results for AIXI remain (see Table 5.3). Until new arguments for AIXI's optimality are put forward, we have to regard AIXI as a *relative* theory of intelligence, dependent on the choice of the prior.

5.6.2 Natural Universal Turing Machines

The choice of the UTM has been a big open question in algorithmic information theory for a long time. The Kolmogorov complexity of a string depends on this choice. However, there are *invariance theorems* (Li and Vitányi, 2008, Thm. 2.1.1 & Thm. 3.1.1) which state that changing the UTM changes Kolmogorov complexity only by a constant. When using the Solomonoff prior M to predict any deterministic computable binary sequence, the number of wrong predictions is bounded by the Kolmogorov complexity of the sequence (Corollary 3.56). Due to the invariance theorem, changing the UTM changes the number of errors only by a constant. In this sense, compression and prediction work for any choice of UTM.

For AIXI, there can be no invariance theorem; in Section 5.2 we showed that a bad choice for the UTM can have drastic consequences. Our negative results can guide future search for a *natural* UTM: the UTMs used to define the indifference prior (Theorem 5.4), the dogmatic prior (Theorem 5.5), and the Gödel prior (Theorem 5.8) should be considered unnatural. But what are other desirable properties of a UTM?

A remarkable but unsuccessful attempt to find natural UTMs is due to Müller (2010). It takes the probability that one universal machine simulates another according

name	defined in	$K_U(U')$	$K_{U'}(U)$
indifference prior	Theorem 5.4	$K(U) + K(m) + O(1)$	m
dogmatic prior	Theorem 5.5	$K(U) + K(\pi) + K(\varepsilon) + O(1)$	$\lceil -\log_2 \varepsilon \rceil$
Gödel prior	Theorem 5.8	$K(U) + K(\mathsf{PA}) + O(1)$	0

Table 5.2: Upper bounds to compiler sizes of the UTMs used in the proofs of Section 5.2. $K_U(U')$ is the number of extra bits to run the 'bad' UTM U' on the 'good' UTM U, $K_{U'}(U)$ is the number of extra bits to run U on U'. $K(U)$ denotes the length of the shortest program for U on U.

to the length of their respective compilers and searches for a stationary distribution. Unfortunately, no stationary distribution exists.

Alternatively, we could demand that the UTM U' that we use for the universal prior has a small compiler on the reference machine U (Hutter, 2005, p. 35). Moreover, we could demand the reverse, that the reference machine U has a small compiler on U'. The idea is that this should limit the amount of bias one can introduce by defining a UTM that has very small programs for very complicated and 'unusual' environments. Unfortunately, this just pushes the choice of the UTM to the reference machine. Table 5.2 lists compiler sizes of the UTMs constructed in this thesis.

5.6.3 Asymptotic Optimality

A policy is asymptotically optimal if the agent learns to act optimally in any environment from the class $\mathcal{M}$. We discussed two asymptotically optimal policies. BayesExp is weakly asymptotically optimal if the horizon grows sublinearly (Theorem 5.24) and Thompson sampling is asymptotically optimal in mean (Theorem 5.25). Both policies commit to exploration for several steps. As stated in Example 5.19:

> To achieve asymptotic optimality, the agent needs to explore infinitely often for an entire effective horizon.

This is why weak asymptotic optimality is impossible if the horizon grows linearly (Theorem 5.21): if the agent explores for an entire effective horizon, it spoils a significant fraction of the average. Thompson sampling explores whenever it draws a bad sample. BayesExp explores if the maximal expected information gain is above some threshold. Both policies commit to exploration for the entire effective horizon.

The exploration performed by Thompson sampling is qualitatively different from the exploration by BayesExp (Lattimore, 2013, Ch. 5). BayesExp performs phases of exploration in which it maximizes the expected information gain. This explores the environment class completely, even achieving off-policy prediction (Orseau et al., 2013, Thm. 7). In contrast, Thompson sampling only explores on the optimal policies, and in some environment classes this will not yield off-policy prediction. So in this sense the

Optimality	Issue/Comment
μ-optimal policy	requires to know the true environment μ in advance
Pareto optimality	always satisfied (Theorem 5.3)
Bayes optimality	same as maximal intelligence
balanced Pareto optimality	same as maximal intelligence (Proposition 5.12)
maximal intelligence	highly dependent on the prior (Corollary 5.15 and Corollary 5.16)
PAC	strong variant of asymptotic optimality in probability
asymptotic optimality	Thompson sampling (Theorem 5.25) and BayesExp (Lattimore, 2013), but not AIXI (Orseau, 2013)
sublinear regret	impossible in general environments, but possible with recoverability (Theorem 5.33)

Table 5.3: Proposed notions of optimality (Hutter, 2002a, 2005; Legg and Hutter, 2007b) and their issues. Asymptotic optimality stands out to be the only nontrivial objective optimality notion for general reinforcement learning.

exploration mechanism of Thompson sampling is more reward-oriented than maximizing information gain.

However, asymptotic optimality has to be taken with a grain of salt. It provides no incentive to the agent to avoid traps in the environment. Once the agent gets caught in a trap, all actions are equally bad and thus optimal: asymptotic optimality has been achieved. Even worse, an asymptotically optimal agent has to explore all the traps because they might contain hidden treasure. This brings us to the following impossibility result for non-recoverable environment classes.

Either the agent gets caught in a trap or it is not asymptotically optimal.[1]

5.6.4 The Quest for Optimality

Theorem 5.3 shows that Pareto optimality is trivial in the class of all computable environments. Bayes optimality, Balanced Pareto optimality, and maximal Legg-Hutter intelligence are equivalent (Proposition 5.12 and Proposition 5.10). Corollary 5.15 and Corollary 5.16 show that this notion is highly subjective because it depends on the choice of the prior. Moreover, according to Corollary 5.17, any computable policy is nearly balanced Pareto optimal. For finite horizons, there are priors such that every policy is balanced Pareto optimal (Theorem 5.4). Sublinear regret is impossible in general environments (Example 5.30). However, if the environment is recoverable (Definition 5.31), then Theorem 5.33 shows that asymptotic optimality in mean implies sublinear regret. In summary, asymptotic optimality is the only nontrivial and objective notion of optimality for the general reinforcement learning problem (Problem 4.2):

[1] This formulation was suggested by Toby Ord.

it is both satisfiable (Theorem 5.24 and Theorem 5.25) and objective because it does not depend on a prior probability measure over the environment class $\mathcal{M}$. Table 5.3 summarized the notions of optimality discussed in this chapter.

Our optimality notions are *tail events*: any finite number of time steps are irrelevant; the agent can be arbitrarily lazy. Asymptotic optimality requires only convergence in the limit. In recoverable environments we can always achieve sublinear regret after any finite interaction. All policies with finite horizon are Bayes optimal according to Theorem 5.4 and Corollary 5.6. Overall, there is a dichotomy between the asymptotic nature of our optimality notions and the use of discounting to prioritize the present over the future. Ideally, we would aim for finite guarantees instead, such as precise regret bounds or PAC convergence rates, but without additional assumptions this is impossible in this general setting. This leaves us with the main question of this chapter unanswered (Hutter, 2009b, Sec. 5):

> What is a good optimality criterion for general reinforcement learning?

Chapter 6

Computability

I simply keep a few spare halting oracles around. — Marcus Hutter

Given infinite computation power, many traditional AI problems become trivial: playing chess, go, or backgammon can be solved by exhaustive expansion of the game tree. Yet other problems seem difficult still; for example, predicting the stock market, driving a car, or babysitting your nephew. How can we solve these problems in theory? Solomonoff induction and AIXI are proposed answers to this question.

Both Solomonoff induction and AIXI are known to be incomputable. But not all incomputabilities are equal. The *arithmetical hierarchy* specifies different levels of computability based on *oracle machines*: each level in the arithmetical hierarchy is computed by a Turing machine which may query a halting oracle for the respective lower level. Our agents are useless if they cannot be approximated in practice, i.e., by a regular Turing machine. Therefore we posit that any ideal for a 'perfect agent' needs to be *limit computable* (Δ_2^0). The class of limit computable functions is the class of functions that admit an *anytime algorithm*.

In Section 6.2 we consider various different flavors of Solomonoff induction: Solomonoff's prior M (Example 3.5) is only a semimeasure and not a measure: it assigns positive probability that the observed string has only finite length. This can be circumvented by normalizing M. Solomonoff's normalization M_{norm} (Definition 2.16) preserves the ratio $M(x1)/M(x0)$ and is limit computable. If instead we mix only over programs that compute infinite strings, we get a semimeasure $\overline{M}$ (3.6), which can be normalized to $\overline{M}_{\text{norm}}$. Moreover, when predicting a sequence, we are primarily interested in the conditional probability $M(xy \mid x)$ (respectively $M_{\text{norm}}(xy \mid y)$, $\overline{M}(xy \mid x)$, or $\overline{M}_{\text{norm}}(xy \mid x)$) that the currently observed string x is continued with y. We show that both M and M_{norm} are limit computable, while $\overline{M}$ and $\overline{M}_{\text{norm}}$ are not. Table 6.1 summarizes our computability results for Solomonoff induction.

For MDPs, planning is already P-complete for finite and infinite horizons (Papadimitriou and Tsitsiklis, 1987). In POMDPs, planning is undecidable (Madani et al., 1999, 2003). The existence of a policy whose expected value exceeds a given threshold is PSPACE-complete (Mundhenk et al., 2000), even for purely epistemic POMDPs in which actions do not change the hidden state (Sabbadin et al., 2007). In Section 6.3 we derive hardness results for planning in general semicomputable environments; this environment class is even more general than POMDPs. We show that optimal policies

Q	$\{(x,q) \in \mathcal{X}^* \times \mathbb{Q} \mid Q(x) > q\}$	$\{(x,y,q) \in \mathcal{X}^* \times \mathcal{X}^* \times \mathbb{Q} \mid Q(xy \mid x) > q\}$
M	$\Sigma_1^0 \setminus \Delta_1^0$	$\Delta_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$
M_{norm}	$\Delta_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$	$\Delta_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$
$\overline{M}$	$\Pi_2^0 \setminus \Delta_2^0$	$\Delta_3^0 \setminus (\Sigma_2^0 \cup \Pi_2^0)$
$\overline{M}_{\text{norm}}$	$\Delta_3^0 \setminus (\Sigma_2^0 \cup \Pi_2^0)$	$\Delta_3^0 \setminus (\Sigma_2^0 \cup \Pi_2^0)$

Table 6.1: The computability results on M, M_{norm}, $\overline{M}$, and $\overline{M}_{\text{norm}}$ proved in Section 6.2. Lower bounds on the complexity of $\overline{M}$ and $\overline{M}_{\text{norm}}$ are given only for specific universal Turing machines.

Agent	Optimal	ε-Optimal
AIMU	Δ_2^0	Δ_1^0
AINU	Δ_3^0, Π_2^0-hard	Δ_2^0, Σ_1^0-hard
AIXI	Δ_3^0, Σ_1^0-hard	Δ_2^0, Σ_1^0-hard
Entropy-seeking	Δ_3^0	Δ_2^0
Information-seeking	Δ_3^0	Δ_2^0
BayesExp	Δ_3^0	Δ_2^0

Table 6.2: Computability results for different agent models derived in Section 6.3, Section 6.5, and Section 6.6. AIMU denotes the optimal policy in a computable environment and AINU denotes the optimal policy in a semicomputable environment (see Section 4.1). Hardness results for AIXI are with respect to a specific universal Turing machine; hardness results for ν-optimal policies are with respect to a specific environment $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$. Results for entropy-seeking and information-seeking policies are only for finite horizons.

are Π_2^0-hard and ε-optimal policies are undecidable.

Moreover, we show that by default, AIXI is not limit computable. When picking the next action, two or more actions might have the same value (expected future rewards). The choice between them is easy, but determining whether such a tie exists is difficult. This problem can be circumvented by settling for an ε-optimal policy; we get a limit-computable agent with infinite horizon. However, these results rely on the recursive definition of the value function. In contrast, Hutter (2005) defines the value function as the limit of the iterative value function. In Section 6.4 we compare these two definitions and show that the recursive definition correctly maximizes expected rewards and has better computability properties.

In Section 6.5 we show that for finite horizons both the entropy-seeking and the information-seeking agent are Δ_3^0-computable and have limit-computable ε-optimal policies. BayesExp (Section 4.3.3) relies on optimal policies that are generally not

limit computable. In Section 6.6 we give a weakly asymptotically optimal agent based on BayesExp that is limit computable. Table 6.2 summarizes our results on the computability of these agents.

In this chapter we illustrate the environments used in the proofs of our theorems in the form of flowcharts. They should be read as follows. Circles denote *stochastic nodes*, rectangles denote *environment nodes*, and diamonds denote the agent's *choice nodes*. Transitions out of stochastic nodes are labeled with transition probabilities, transitions out of environment nodes are labeled with percepts, and transitions out of choice nodes are labeled with actions. The initial node is marked with a small incoming arrow (see for example Figure 6.3). By Assumption 4.6b the worst possible outcome is getting reward 0 forever, thus we label such states as *hell*. Analogously, getting reward 1 forever is the best possible outcome, thus we label such states as *heaven*.

6.1 Background on Computability

6.1.1 The Arithmetical Hierarchy

A set $A \subseteq \mathbb{N}$ is Σ^0_n iff there is a quantifier-free formula η such that

$$k \in A \iff \exists k_1 \forall k_2 \ldots Q_n k_n \; \eta(k, k_1, \ldots, k_n) \tag{6.1}$$

where $Q_n = \forall$ if n is even, $Q_n = \exists$ if n is odd (Nies, 2009, Def. 1.4.10). (We can also think of η as a computable relation.) A set $A \subseteq \mathbb{N}$ is Π^0_n iff its complement $\mathbb{N} \setminus A$ is Σ^0_n. The formula η on the right side of (6.1) is a Σ^0_n*-formula* and its negation is a Π^0_n*-formula*. It can be shown that we can add any bounded quantifiers and duplicate quantifiers of the same type without changing the classification of A. The set A is Δ^0_n iff A is Σ^0_n and A is Π^0_n. We get that Σ^0_1 as the class of recursively enumerable sets, Π^0_1 as the class of co-recursively enumerable sets and Δ^0_1 as the class of recursive sets.

The set $A \subseteq \mathbb{N}$ is Σ^0_n*-hard* (Π^0_n*-hard,* Δ^0_n*-hard)* iff for any set $B \in \Sigma^0_n$ ($B \in \Pi^0_n$, $B \in \Delta^0_n$), B is many-one reducible to A, i.e., there is a computable function f such that $k \in B \leftrightarrow f(k) \in A$ (Nies, 2009, Def. 1.2.1). We get $\Sigma^0_n \subset \Delta^0_{n+1} \subset \Sigma^0_{n+1} \subset \ldots$ and $\Pi^0_n \subset \Delta^0_{n+1} \subset \Pi^0_{n+1} \subset \ldots$. This hierarchy of subsets of natural numbers is known as the *arithmetical hierarchy*.

By Post's Theorem (Nies, 2009, Thm. 1.4.13), a set is Σ^0_n if and only if it is recursively enumerable on an oracle machine with an oracle for a Σ^0_{n-1}-complete set. An oracle for Σ^0_1 is called a *halting oracle*.

6.1.2 Computability of Real-valued Functions

We fix some encoding of rational numbers into binary strings and an encoding of binary strings into natural numbers. From now on, this encoding will be done implicitly wherever necessary.

Definition 6.1 (Σ^0_n-, Π^0_n-, Δ^0_n-computable)**.** A function $f : \mathcal{X}^* \to \mathbb{R}$ is called Σ^0_n*-computable* (Π^0_n*-computable,* Δ^0_n*-computable)* iff the set $\{(x, q) \in \mathcal{X}^* \times \mathbb{Q} \mid f(x) > q\}$ is Σ^0_n (Π^0_n, Δ^0_n).

	$\{(x,q) \mid f(x) \geq q\}$	$\{(x,q) \mid f(x) < q\}$
f is computable	Δ^0_1	Δ^0_1
f is lower semicomputable	Σ^0_1	Π^0_1
f is upper semicomputable	Π^0_1	Σ^0_1
f is limit computable	Δ^0_2	Δ^0_2
f is Δ^0_n-computable	Δ^0_n	Δ^0_n
f is Σ^0_n-computable	Σ^0_n	Π^0_n
f is Π^0_n-computable	Π^0_n	Σ^0_n

Table 6.3: Connection between the computability of real-valued functions and the arithmetical hierarchy.

A Δ^0_1-computable function is called *computable*, a Σ^0_1-computable function is called *lower semicomputable*, and a Π^0_1-computable function is called *upper semicomputable*. A Δ^0_2-computable function f is called *limit computable*, because there is a computable function ϕ such that

$$\lim_{k\to\infty} \phi(x,k) = f(x).$$

The program ϕ that limit computes f can be thought of as an *anytime algorithm* for f: we can stop ϕ at any time k and get a preliminary answer. If the program ϕ ran long enough (which we do not know), this preliminary answer will be close to the correct one.

Limit-computable sets are the highest level in the arithmetical hierarchy that can be approached by a regular Turing machine. Above limit-computable sets we necessarily need some form of halting oracle. See Table 6.3 for the definition of lower/upper semicomputable and limit-computable functions in terms of the arithmetical hierarchy.

Lemma 6.2 (Computability of Arithmetical Operations)**.** *Let $n > 0$ and let $f, g : \mathcal{X}^* \to \mathbb{R}$ be two Δ^0_n-computable functions. Then*

(a) $\{(x,y) \mid f(x) > g(y)\}$ is Σ^0_n,

(b) $\{(x,y) \mid f(x) \leq g(y)\}$ is Π^0_n,

(c) $f+g$, $f-g$, and $f \cdot g$ are Δ^0_n-computable, and

(d) f/g is Δ^0_n-computable if $g(x) \neq 0$ for all x.

(e) $\log f$ is Δ^0_n-computable if $f(x) > 0$ for all x.

Proof. We only prove this for $n > 1$. Since f, g are Δ^0_n-computable, they are limit computable on a level $n-1$ oracle machine. Let ϕ be the function limit computing f on the oracle machine, and let ψ be the function limit computing g on the oracle machine:

$$f(x) = \lim_{k\to\infty} \phi(k,x) \quad \text{and} \quad g(y) = \lim_{k\to\infty} \psi(k,y).$$

By assumption, both ϕ and ψ are Δ^0_{n-1}-computable.

(a) Let $G := \{(x,y,q) \mid g(y) < q\}$, and let $F := \{(x,y,q) \mid q < f(x)\}$, both of which are in Δ_n^0 by assumption. Hence there are Σ_n^0-formulas φ_G and φ_F such that

$$(x,y,q) \in G \iff \varphi_G(x,y,q)$$
$$(x,y,q) \in F \iff \varphi_F(x,y,q)$$

Now $f(x) > g(y)$ if and only if $\exists q.\ (x,y,q) \in G \cap F$, which is equivalent to the Σ_n^0-forumla

$$\exists q.\ \varphi_G(x,y,q) \ \wedge \ \varphi_F(x,y,q).$$

(b) Follows from (a).

(c) Addition, subtraction, and multiplication are continuous operations.

(d) Division is discontinuous only at $g(x) = 0$. We show this explicitly. By assumption, for any $\varepsilon > 0$ there is a k_0 such that for all $k > k_0$

$$|\phi(x,k) - f(x)| < \varepsilon \quad \text{and} \quad |\psi(x,k) - g(x)| < \varepsilon.$$

We assume without loss of generality that $\varepsilon < |g(x)|$, since $g(x) \neq 0$ by assumption.

$$\begin{aligned}
&\left|\frac{\phi(x,k)}{\psi(x,k)} - \frac{f(x)}{g(x)}\right| \\
&= \left|\frac{\phi(x,k)g(x) - f(x)\psi(x,k)}{\psi(x,k)g(x)}\right| \\
&\leq \frac{|\phi(x,k)g(x) - f(x)g(x)| + |f(x)g(x) - f(x)\psi(x,k)|}{|\psi(x,k)g(x)|} \\
&< \frac{\varepsilon|g(x)| + |f(x)|\varepsilon}{|\psi(x,k)g(x)|}
\end{aligned}$$

with $|\psi(x,k)g(x)| = |\psi(x,k)| \cdot |g(x)| > (|g(x)| - \varepsilon)|g(x)|$,

$$< \varepsilon \cdot \frac{|g(x)| + |f(x)|}{(|g(x)| - \varepsilon)|g(x)|} \xrightarrow{\varepsilon \to 0} 0,$$

therefore $f(x)/g(x) = \lim_{k\to\infty} \phi(x,k)/\psi(x,k)$.

(e) Follows from the fact that the logarithm is computable. □

6.2 The Complexity of Solomonoff Induction

In this section, we derive the computability results for Solomonoff's prior as stated in Table 6.1.

Since M is lower semicomputable, M_{norm} is limit computable by Lemma 6.2 (c) and (d). When using the Solomonoff prior M (or one of its sisters M_{norm}, $\overline{M}$, or $\overline{M}_{\text{norm}}$ defined in Definition 2.16 and Equation 3.6) for sequence prediction, we need

$$M(xy \mid x) > q \iff \forall m \exists k.\ \tfrac{\phi(xy,k)}{\phi(x,m)} > q \iff \exists k \exists m_0 \forall m \geq m_0.\ \tfrac{\phi(xy,k)}{\phi(x,m)} > q$$

Figure 6.1: A Π^0_2-formula and an equivalent Σ^0_2-formula defining conditional M. Here $\phi(x,k)$ denotes a computable function that lower semicomputes $M(x)$.

to compute the conditional probability $M(xy \mid x) = M(xy)/M(x)$ for finite strings $x, y \in \mathcal{X}^*$. Because $M(x) > 0$ for all finite strings $x \in \mathcal{X}^*$, this quotient is well-defined.

Theorem 6.3 (Complexity of M, M_{norm}, $\overline{M}$, and $\overline{M}_{\text{norm}}$)**.**

(a) $M(x)$ *is lower semicomputable*

(b) $M(xy \mid x)$ *is limit computable*

(c) $M_{\text{norm}}(x)$ *is limit computable*

(d) $M_{\text{norm}}(xy \mid x)$ *is limit computable*

(e) $\overline{M}(x)$ *is* Π^0_2*-computable*

(f) $\overline{M}(xy \mid x)$ *is* Δ^0_3*-computable*

(g) $\overline{M}_{\text{norm}}(x)$ *is* Δ^0_3*-computable*

(h) $\overline{M}_{\text{norm}}(xy \mid x)$ *is* Δ^0_3*-computable*

Proof. (a) By Li and Vitányi (2008, Thm. 4.5.2). Intuitively, we can run all programs in parallel and get monotonely increasing lower bounds for $M(x)$ by adding $2^{-|p|}$ every time a program p has completed outputting x.

(b) From (a) and Lemma 6.2d since $M(x) > 0$ (see also Figure 6.1).

(c) By Lemma 6.2cd, and $M(x) > 0$.

(d) By (iii), Lemma 6.2d since $M_{\text{norm}}(x) \geq M(x) > 0$.

(e) Let ϕ be a computable function that lower semicomputes M. Since M is a semimeasure, $M(xy) \geq \sum_z M(xyz)$, hence $\sum_{y \in \mathcal{X}^n} M(xy)$ is nonincreasing in n and thus $\overline{M}(x) > q$ iff $\forall n \exists k \sum_{y \in \mathcal{X}^n} \phi(xy,k) > q$.

(f) From (v) and Lemma 6.2d since $\overline{M}(x) > 0$.

(g) From (v) and Lemma 6.2d.

(h) From (vi) and Lemma 6.2d since $\overline{M}_{\text{norm}}(x) \geq \overline{M}(x) > 0$. □

We proceed to show that these bounds are in fact the best possible ones. If M were Δ_1^0-computable, then so would be the conditional semimeasure $M(\cdot \mid \cdot)$. Thus the M-adversarial sequence $z_{1:\infty}$ defined in Example 3.42 would be computable and hence corresponds to a computable deterministic measure μ. However, we have $M(z_{1:t}) \leq 2^{-t}$ by construction, so dominance $M(x) \geq w(\mu)\mu(x)$ with $w(\mu) > 0$ yields a contradiction with $t \to \infty$:

$$2^{-t} \geq M(z_{1:t}) \geq w(\mu)\mu(z_{1:t}) = w(\mu) > 0$$

By the same argument, the normalized Solomonoff prior M_{norm} cannot be Δ_1^0-computable. However, since it is a measure, Σ_1^0- or Π_1^0-computability would entail Δ_1^0-computability.

For $\overline{M}$ and $\overline{M}_{\text{norm}}$ we prove the following two lower bounds for specific universal Turing machines.

Theorem 6.4 ($\overline{M}$ is not Limit Computable)**.** *There is a universal Turing machine U' such that the set $\{(x, q) \mid \overline{M}_{U'}(x) > q\}$ is not in Δ_2^0.*

Proof. Assume the contrary and let $A \in \Pi_2^0 \setminus \Delta_2^0$ and η be a quantifier-free first-order formula such that

$$n \in A \iff \forall k \exists i.\ \eta(n, k, i). \tag{6.2}$$

For each $n \in \mathbb{N}$, we define the program p_n as follows.

```
1: procedure p_n
2:     output 1^{n+1}0
3:     k ← 0
4:     while true do
5:         i ← 0
6:         while not η(n, k, i) do
7:             i ← i + 1
8:         k ← k + 1
9:         output 0
```

Each program p_n always outputs $1^{n+1}0$. Furthermore, the program p_n outputs the infinite string $1^{n+1}0^\infty$ if and only if $n \in A$ by (6.2). We define U' as follows using our reference machine U.

- $U'(1^{n+1}0)$: Run p_n.
- $U'(00p)$: Run $U(p)$.
- $U'(01p)$: Run $U(p)$ and bitwise invert its output.

By construction, U' is a universal Turing machine. No p_n outputs a string starting with $0^{n+1}1$, therefore $\overline{M}_{U'}(0^{n+1}1) = \frac{1}{4}\big(\overline{M}_U(0^{n+1}1) + \overline{M}_U(1^{n+1}0)\big)$. Hence

$$\begin{aligned}\overline{M}_{U'}(1^{n+1}0) &= 2^{-n-2}\mathbb{1}_A(n) + \tfrac{1}{4}\overline{M}_U(1^{n+1}0) + \tfrac{1}{4}\overline{M}_U(0^{n+1}1) \\ &= 2^{-n-2}\mathbb{1}_A(n) + \overline{M}_{U'}(0^{n+1}1)\end{aligned}$$

If $n \notin A$, then $\overline{M}_{U'}(1^{n+1}0) = \overline{M}_{U'}(0^{n+1}1)$. Otherwise, we have $|\overline{M}_{U'}(1^{n+1}0) - \overline{M}_{U'}(0^{n+1}1)| = 2^{-n-2}$.

Now we assume that $\overline{M}_{U'}$ is limit computable, i.e., there is a computable function $\phi : \mathcal{X}^* \times \mathbb{N} \to \mathbb{Q}$ such that $\lim_{k\to\infty} \phi(x,k) = \overline{M}_{U'}(x)$. We get that

$$n \in A \iff \lim_{k\to\infty} \phi(0^{n+1}1, k) - \phi(1^{n+1}0, k) \geq 2^{-n-2},$$

thus A is limit computable, a contradiction. □

Corollary 6.5 ($\overline{M}_{\text{norm}}$ is not Σ_2^0- or Π_2^0-computable)**.** *There is a universal Turing machine U' such that $\{(x,q) \mid \overline{M}_{\text{norm}U'}(x) > q\}$ is not in Σ_2^0 or Π_2^0.*

Proof. Since $\overline{M}_{\text{norm}} = c \cdot \overline{M}$, there exists a $k \in \mathbb{N}$ such that $2^{-k} < c$ (even if we do not know the value of k). We can show that the set $\{(x,q) \mid \overline{M}_{\text{norm}U'}(x) > q\}$ is not in Δ_2^0 analogously to the proof of Theorem 6.4, using

$$n \in A \iff \lim_{k\to\infty} \phi(0^{n+1}1, k) - \phi(1^{n+1}0, k) \geq 2^{-k-n-2}.$$

If $\overline{M}_{\text{norm}}$ were Σ_2^0-computable or Π_2^0-computable, this would imply that $\overline{M}_{\text{norm}}$ is Δ_2^0-computable since $\overline{M}_{\text{norm}}$ is a measure, a contradiction. □

Since $M(\epsilon) = 1$, we have $M(x \mid \epsilon) = M(x)$, so the conditional probability $M(xy \mid x)$ has at least the same complexity as M. Analogously for M_{norm} and $\overline{M}_{\text{norm}}$ since they are measures. For $\overline{M}$, we have that $\overline{M}(x \mid \epsilon) = \overline{M}_{\text{norm}}(x)$, so Corollary 6.5 applies. All that remains to prove is that conditional M is not lower semicomputable.

Theorem 6.6 (Conditional M is not Lower Semicomputable)**.** *The set $\{(x, xy, q) \mid M(xy \mid x) > q\}$ is not recursively enumerable.*

We gave a different, more complicated proof in Leike and Hutter (2015b). The following, much simpler and more elegant proof is due to Sterkenburg (2016, Prop. 3).

Proof. Assume to the contrary that $M(xy \mid x)$ is lower semicomputable. Let $a \neq b \in \mathcal{X}$. We construct an infinite string x by defining its initial segments $\epsilon =: x(0) \sqsubset x(1) \sqsubset x(2) \sqsubset \ldots \sqsubset x$. At every step n, we enumerate strings $y \in \mathcal{X}^*$ until one is found satisfying $M(a \mid x(n)y) \geq 1/2$; then set $x(n+1) := x(n)yb$. This implies that for infinitely many t there is an n such that $M(b \mid x_{<t}) = M(b \mid x(n)y) \leq 1 - M(a \mid x(n)y) \leq 1/2$. Since we assumed $M(\cdot \mid \cdot)$ to be lower semicomputable, the infinite string x is computable, and hence $M(x_t \mid x_{<t}) \to 1$ by Corollary 3.55. But this contradicts $M(b \mid x_{<t}) \leq 1/2$ infinitely often. □

6.3 The Complexity of AINU, AIMU, and AIXI

6.3.1 Upper Bounds

In this section, we derive upper bounds on the computability of AINU, AIMU, and AIXI. Except for Corollary 6.14, all results in this section apply generally to any $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$.

Since the Bayesian mixture $\xi \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$, they apply to AIXI even though they are stated for AINU.

In order to position AINU in the arithmetical hierarchy, we need to encode policies as sets of natural numbers. For the rest of this chapter, we assume that policies are deterministic, thus can be represented as relations over $(\mathcal{A}\times\mathcal{E})^*\times\mathcal{A}$. These relations are easily identified with sets of natural numbers by encoding the history into one natural number. From now on this translation of policies into sets of natural numbers will be done implicitly wherever necessary.

Lemma 6.7 (Policies are in Δ_n^0)**.** *If a policy π is Σ_n^0 or Π_n^0, then π is Δ_n^0.*

Proof. Let φ be a Σ_n^0-formula (Π_n^0-formula) defining π, i.e., $\varphi(h,a)$ holds iff $\pi(h)=a$. We define the formula φ',

$$\varphi'(h,a) := \bigwedge_{a'\in\mathcal{A}\setminus\{a\}} \neg\varphi(h,a').$$

The set of actions $\mathcal{A}$ is finite, hence φ' is a Π_n^0-formula (Σ_n^0-formula). Moreover, φ' is equivalent to φ. □

To compute the optimal policy, we need to compute the optimal value function. The following lemma gives an upper bound on the computability of the value function for environments in $\mathcal{M}_{\text{LSC}}^{\text{CCS}}$.

Lemma 6.8 (Complexity of V_ν^*)**.** *For every $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$, and every lower semicomputable discount function γ, the function V_ν^* is Δ_2^0-computable.*

Proof. The explicit form of the value function (4.2) has numerator

$$\lim_{m\to\infty} \max_{ae_{t:m}} \sum_{i=t}^{m} \gamma(i) r_i \nu(e_{1:i} \parallel a_{1:i}),$$

and denominator $\nu(e_{<t} \parallel a_{<t})\cdot\Gamma_t$. The numerator is nondecreasing in m because we assumed rewards to be nonnegative (Assumption 4.6b). Hence both numerator and denominator are lower semicomputable functions, so Lemma 6.2d implies that V_ν^* is Δ_2^0-computable. □

From the optimal value function V_ν^* we get the optimal policy π_ν^* according to (4.4). However, in cases where there is more than one optimal action, we have to break an argmax tie. This happens iff $V_\nu^*(h\alpha) = V_\nu^*(h\beta)$ for two potential actions $\alpha\neq\beta\in\mathcal{A}$. This equality test is more difficult than determining which is larger in cases where they are unequal. Thus we get the following upper bound.

Theorem 6.9 (Complexity of Optimal Policies)**.** *For any environment ν, if V_ν^* is Δ_n^0-computable, then there is an optimal policy π_ν^* for the environment ν that is Δ_{n+1}^0.*

Proof. To break potential ties, we pick an (arbitrary) total order $\succ$ on $\mathcal{A}$ that specifies which actions should be preferred in case of a tie. We define

$$\begin{aligned} \pi_\nu(h) = a \;:\Longleftrightarrow\; & \bigwedge_{a' : a' \succ a} V^*_\nu(ha) > V^*_\nu(ha') \\ & \wedge \bigwedge_{a' : a \succ a'} V^*_\nu(ha) \geq V^*_\nu(ha'). \end{aligned} \tag{6.3}$$

Then π_ν is a ν-optimal policy according to (4.4). By assumption, V^*_ν is Δ^0_n-computable. By Lemma 6.2ab $V^*_\nu(ha) > V^*_\nu(ha')$ is Σ^0_n and $V^*_\nu(ha) \geq V^*_\nu(ha')$ is Π^0_n. Therefore the policy π_ν defined in (6.3) is a conjunction of a Σ^0_n-formula and a Π^0_n-formula and thus Δ^0_{n+1}. □

Corollary 6.10 (Complexity of AINU)**.** *AINU is Δ^0_3 for every environment $\nu \in \mathcal{M}^{\text{CCS}}_{\text{LSC}}$.*

Proof. From Lemma 6.8 and Theorem 6.9. □

Usually we do not mind taking slightly suboptimal actions. Therefore actually trying to determine if two actions have the exact same value seems like a waste of resources. In the following, we consider policies that attain a value that is always within some $\varepsilon > 0$ of the optimal value.

Theorem 6.11 (Complexity of ε-Optimal Policies)**.** *For any environment ν, if V^*_ν is Δ^0_n-computable, then there is an ε-optimal policy π^ε_ν for the environment ν that is Δ^0_n.*

Proof. Let $\varepsilon > 0$ be given. Since the value function $V^*_\nu(h)$ is Δ^0_n-computable, the set $V_\varepsilon := \{(ha, q) \mid |q - V^*_\nu(ha)| < \varepsilon/2\}$ is in Δ^0_n according to Definition 6.1. Hence we compute the values $V^*_\nu(ha')$ until we get within $\varepsilon/2$ for every $a' \in \mathcal{A}$ and then choose the action with the highest value so far. Formally, let $\succ$ be an arbitrary total order on $\mathcal{A}$ that specifies which actions should be preferred in case of a tie. Without loss of generality, we assume $\varepsilon = 1/k$, and define Q to be an $\varepsilon/2$-grid on $[0, 1]$, i.e., $Q := \{0, 1/2k, 2/2k, \ldots, 1\}$. We define

$$\begin{aligned} \pi^\varepsilon_\nu(h) = a \;:\Longleftrightarrow\; \exists (q_{a'})_{a' \in \mathcal{A}} \in Q^{\mathcal{A}}.\; & \bigwedge_{a' \in \mathcal{A}} (ha', q_{a'}) \in V_\varepsilon \\ & \wedge \bigwedge_{a' : a' \succ a} q_a > q_{a'} \wedge \bigwedge_{a' : a \succ a'} q_a \geq q_{a'} \\ & \wedge \text{the tuple } (q_{a'})_{a' \in \mathcal{A}} \text{ is minimal with} \\ & \quad \text{respect to the lex. ordering on } Q^{\mathcal{A}}. \end{aligned} \tag{6.4}$$

This makes the choice of a unique. Moreover, $Q^{\mathcal{A}}$ is finite since $\mathcal{A}$ is finite, and hence (6.4) is a Δ^0_n-formula. □

Corollary 6.12 (Complexity of ε-Optimal AINU)**.** *For any environment $\nu \in \mathcal{M}^{\text{CCS}}_{\text{LSC}}$, there is an ε-optimal policy for AINU that is Δ^0_2.*

Proof. From Lemma 6.8 and Theorem 6.11. □

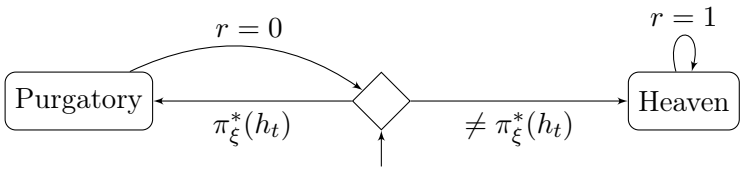


Figure 6.2: The environment μ from the proof of Theorem 6.15. The agent gets reward 0 as long as it follows AIXI's policy π^*_ξ that is assumed to be computable. Once the agent deviates from π^*_ξ, it gets reward 1. We get a contradiction because AIXI can learn this environment, so it will eventually decide to take an action that leads to heaven.

Corollary 6.13 (Complexity of ε-Optimal AIXI)**.** *For any lower semicomputable prior there is an ε-optimal policy for AIXI that is Δ^0_2.*

Proof. From Corollary 6.12 since for any lower semicomputable prior, the corresponding Bayesian mixture ξ is in $\mathcal{M}^{\text{CCS}}_{\text{LSC}}$. □

If the environment $\nu \in \mathcal{M}^{\text{CCM}}_{\text{comp}}$ is a measure, i.e., ν assigns zero probability to finite strings, then we get computable ε-optimal policies.

Corollary 6.14 (Complexity of AIMU)**.** *If the environment $\mu \in \mathcal{M}^{\text{CCM}}_{\text{comp}}$ is a measure and the discount function γ is computable, then* AIMU *is limit computable (Δ^0_2), and ε-optimal* AIMU *is computable (Δ^0_1).*

Proof. Let $\varepsilon > 0$ be the desired accuracy. We can truncate the limit $m \to \infty$ in (4.2) at the $\varepsilon/2$-effective horizon $H_t(\varepsilon/2)$, since everything after $H_t(\varepsilon/2)$ can contribute at most $\varepsilon/2$ to the value function. Any lower semicomputable measure is computable (Li and Vitányi, 2008, Lem. 4.5.1). Therefore V^*_μ as given in (4.2) is composed only of computable functions, hence it is computable according to Lemma 6.2. The claim now follows from Theorem 6.9 and Theorem 6.11. □

6.3.2 Lower Bounds

We proceed to show that the bounds from the previous section are the best we can hope for. In environment classes where ties have to be broken, AINU has to solve Π^0_2-hard problems (Theorem 6.16). These lower bounds are stated for particular environments $\nu \in \mathcal{M}^{\text{CCS}}_{\text{LSC}}$. Throughout this section, we assume that $\Gamma_t > 0$ for all t.

We also construct universal mixtures that yield bounds on ε-optimal policies. There is an ε-optimal AIXI that solves Σ^0_1-hard problems (Theorem 6.17). For arbitrary universal mixtures, we prove the following weaker statement that only guarantees incomputability.

Theorem 6.15 (No AIXI is computable)**.** AIXI *is not computable for any universal Turing machine U.*

This theorem follows from the incomputability of Solomonoff induction. By the on-policy value convergence theorem (Corollary 4.20) AIXI succeeds to predict the environment's behavior for its own policy. If AIXI were computable, then there would be computable environments more powerful than AIXI: they can simulate AIXI and anticipate its prediction, which leads to a contradiction.

Proof. Assume there is a computable policy π^*_ξ that is optimal in the mixture ξ. We define a deterministic environment μ, the *adversarial environment* to π^*_ξ. The environment μ gives rewards 0 as long as the agent follows the policy π^*_ξ, and rewards 1 once the agent deviates. Formally, we ignore observations by setting $\mathcal{O} := \{0\}$, and define

$$\mu(r_{1:t} \parallel a_{1:t}) := \begin{cases} 1 & \text{if } \forall k \leq t.\, a_k = \pi^*_\xi((ar)_{<k}) \text{ and } r_k = 0, \\ 1 & \text{if } \forall k \leq t.\, r_k = \mathbb{1}_{k \geq i} \\ & \quad \text{where } i := \min\{j \mid a_j \neq \pi^*_\xi((ar)_{<j})\}, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

See Figure 6.2 for an illustration of this environment. The environment μ is computable because the policy π^*_ξ was assumed to be computable. Suppose π^*_ξ acts in μ, then by Theorem 4.19 AIXI learns to predict perfectly *on policy*:

$$V_\xi^{\pi^*_\xi}(æ_{<t}) - V_\mu^{\pi^*_\xi}(æ_{<t}) \to 0 \text{ as } t \to \infty \; \mu^{\pi^*_\xi}\text{-almost surely},$$

since both π^*_ξ and μ are deterministic. Because $V_\mu^{\pi^*_\xi}(h_{<t}) = 0$ by definition of μ, we get $V^*_\xi(æ_{<t}) \to 0$. Therefore we find a t large enough such that $V^*_\xi(æ_{<t}) < w(\mu)$ where $æ_{<t}$ is the interaction history of π^*_ξ in μ. A policy π with $\pi(æ_{<t}) \neq \pi^*_\xi(æ_{<t})$, gets a reward of 1 in environment μ for all time steps after t, hence $V^\pi_\mu(æ_{<t}) = 1$. With linearity of $V^\pi_\xi(æ_{<t})$ in ξ (Lemma 4.14),

$$V^\pi_\xi(æ_{<t}) \geq w(\mu) \tfrac{\mu(e_{1:t} \| a_{1:t})}{\xi(e_{1:t} \| a_{1:t})} V^\pi_\mu(æ_{<t}) \geq w(\mu),$$

since $\mu(e_{1:t} \parallel a_{1:t}) = 1$ (μ is deterministic), $V^\pi_\mu(æ_{<t}) = 1$, and $\xi(e_{1:t} \parallel a_{1:t}) \leq 1$. Now we get a contradiction:

$$w(\mu) > V^*_\xi(æ_{<t}) = \sup_{\pi'} V^{\pi'}_\xi(æ_{<t}) \geq V^\pi_\xi(æ_{<t}) \geq w(\mu) \qquad \square$$

For the remainder of this section, we fix the action space to be $\mathcal{A} := \{\alpha, \beta\}$ with action α favored in ties. The percept space is fixed to a tuple of binary observations and rewards, $\mathcal{E} := \mathcal{O} \times \{0, 1\}$ with $\mathcal{O} := \{0, 1\}$.

Theorem 6.16 (AINU is Π^0_2-hard)**.** *There is an environment $\nu \in \mathcal{M}^{\text{CCS}}_{\text{LSC}}$ such that AINU is Π^0_2-hard.*

Proof. Let A be a any Π^0_2-set, and let η be a quantifier-free formula such that

$$n \in A \iff \forall i \, \exists k \, \eta(n, i, k). \tag{6.5}$$

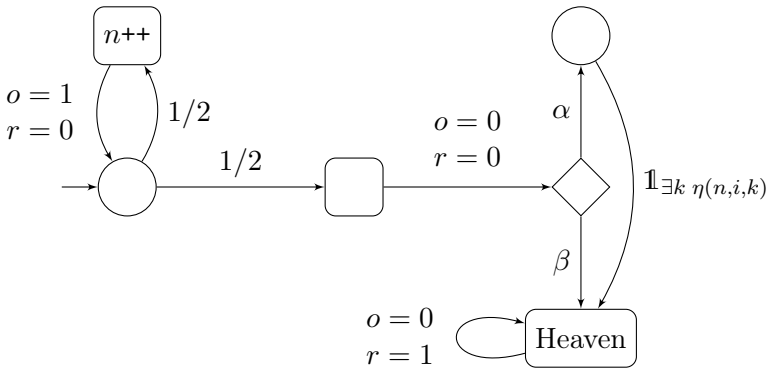


Figure 6.3: The environment ρ_i from the proof of Theorem 6.16. The mixture ν over class of environments $\mathcal{M} := \{\rho_0, \rho_1, \ldots\} \subset \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ forces AINU to solve Π_2^0-hard problems: Action α is preferred (because of a tie) iff it leads to heaven, which is the case iff $\exists k\ \eta(n, i, k)$.

We define a class of environments $\mathcal{M} := \{\rho_1, \rho_2, \ldots\}$ where each ρ_i is defined as follows.

$$\rho_i((or)_{1:m} \parallel a_{1:m}) := \begin{cases} 2^{-m} & \text{if } o_{1:m} = 1^m \text{ and } \forall t \leq m.\ r_t = 0, \\ 2^{-n-1} & \text{if } \exists n.\ 1^n 0 \sqsubseteq o_{1:m} \sqsubseteq 1^n 0^\infty \text{ and } a_{n+2} = \alpha \\ & \quad \text{and } r_t = \mathbb{1}_{t>n+1} \text{ and } \exists k\ \eta(n, i, k), \\ 2^{-n-1} & \text{if } \exists n.\ 1^n 0 \sqsubseteq o_{1:m} \sqsubseteq 1^n 0^\infty \text{ and } a_{n+2} = \beta \\ & \quad \text{and } r_t = \mathbb{1}_{t>n+1}, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

See Figure 6.3 for an illustration of these environments. Every ρ_i is a chronological conditional semimeasure by definition and every ρ_i is lower semicomputable since η is quantifier-free, so $\mathcal{M} \subseteq \mathcal{M}_{\text{LSC}}^{\text{CCS}}$.

We define our environment ν as a mixture over $\mathcal{M}$,

$$\nu := \sum_{i \in \mathbb{N}} 2^{-i-1} \rho_i;$$

the choice of the weights on the environments ρ_i is arbitrary but positive. Let π_ν^* be an optimal policy for the environment ν and recall that the action α is preferred in ties. We claim that for the ν-optimal policy π_ν^*,

$$n \in A \iff \pi_\nu^*(1^n 0) = \alpha. \tag{6.6}$$

This enables us to decide whether $n \in A$ given the policy π_ν^*, hence proving (6.6) concludes this proof.

Let $n, i \in \mathbb{N}$ be given, and suppose we are in environment ρ_i and observe $1^n 0$. Taking action β next yields reward 1 forever; taking action α next yields a reward of 1 if there

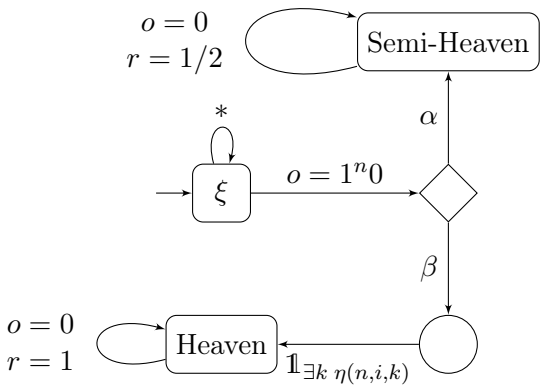


Figure 6.4: The environment ν from the proof of Theorem 6.17, which forces AIXI to solve Σ_1^0-hard problems. It functions just like ξ until the observation history is 1^n0. Then, action α is preferred iff heaven is accessible, i.e., iff $\exists k\ \eta(n,i,k)$.

is a k such that $\eta(n,i,k)$ holds. If this is the case, then

$$V^*_{\rho_i}(1^n0\alpha) = \Gamma_{n+2} = V^*_{\rho_i}(1^n0\beta),$$

and otherwise

$$V^*_{\rho_i}(1^n0\alpha) = 0 < \Gamma_{n+2} = V^*_{\rho_i}(1^n0\beta)$$

(omitting the first $n+1$ actions and rewards in the argument of the value function). We can now show (6.6): By (6.5), $n \in A$ if and only if for all i there is a k such that $\eta(n,i,k)$, which happens if and only if $V^*_{\rho_i}(1^n0\alpha) = \Gamma_{n+2}$ for all $i \in \mathbb{N}$, which is equivalent to $V^*_\nu(1^n0\alpha) = \Gamma_{n+2}$, which in turn is equivalent to $\pi^*_\mu(1^n0) = \alpha$ since $V^*_\nu(1^n0\beta) = \Gamma_{n+2}$ and action α is favored in ties. □

Theorem 6.17 (Some ε-optimal AIXI are Σ_1^0-hard)**.** *There is a universal Turing machine U' and an $\varepsilon > 0$ such that any ε-optimal policy for* AIXI *is Σ_1^0-hard.*

Proof. Let A be a Σ_1^0-set and η be a quantifier-free formula such that $n+1 \in A$ iff $\exists k\ \eta(n,k)$. We define the environment

$$\nu((or)_{1:t} \parallel a_{1:t}) := \begin{cases} \xi((or)_{1:n} \parallel a_{1:n}) & \text{if } \exists n.\ o_{1:n} = 1^{n-1}0 \text{ and } a_n = \alpha \\ & \quad \text{and } \forall t' > n.\ o_{t'} = 0 \wedge r_{t'} = \frac{1}{2}, \\ \xi((or)_{1:n} \parallel a_{1:n}) & \text{if } \exists n.\ o_{1:n} = 1^{n-1}0 \text{ and } a_n = \beta \\ & \quad \text{and } \forall t' > n.\ o_t = 0 \wedge r_t = 1 \\ & \quad \text{and } \exists k\ \eta(n-1,k), \\ \xi((or)_{1:t} \parallel a_{1:t}) & \text{if } \nexists n.\ o_{1:n} = 1^{n-1}0, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

See Figure 6.4 for an illustration. The environment ν mimics the universal environment

ξ until the observation history is $1^{n-1}0$. Taking the action α next gives rewards $1/2$ forever. Taking the action β next gives rewards 1 forever if $n \in A$, otherwise the environment ν ends at some future time step. Therefore we want to take action β if and only if $n \in A$. We have that $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ since $\xi \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ and η is quantifier-free.

We define $\xi' := \frac{1}{2}\nu + \frac{1}{8}\xi$. By Lemma 4.24 ξ' is a universal lower semicomputable semimeasure. Let $n \in A$ be given and let $h \in (\mathcal{A}\times\mathcal{E})^n$ be any history with observations $o_{1:n} = 1^{n-1}0$. Since $\nu(1^{n-1}0 \mid a_{1:n}) = \xi(1^{n-1}0 \mid a_{1:n})$ by definition, the posterior weights of ν and ξ in ξ' are equal to the prior weights, analogously to the proof of Theorem 5.5. In the following, we use the linearity of $V_\rho^{\pi^*_{\xi'}}$ in ρ (Lemma 4.14), and the fact that values are bounded between 0 and 1 (Assumption 4.6b). If there is a k such that $\eta(n-1,k)$ holds,

$$\begin{aligned} V^*_{\xi'}(h\beta) - V^*_{\xi'}(h\alpha) &= \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\beta) - \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\alpha) + \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\beta) - \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\alpha) \\ &\geq \tfrac{1}{2} - \tfrac{1}{4} + 0 - \tfrac{1}{8} = \tfrac{1}{8}, \end{aligned}$$

and similarly if there is no k such that $\eta(n-1,k)$ holds, then

$$\begin{aligned} V^*_{\xi'}(h\alpha) - V^*_{\xi'}(h\beta) &= \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\alpha) - \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\beta) + \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\alpha) - \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\beta) \\ &\geq \tfrac{1}{4} - 0 + 0 - \tfrac{1}{8} = \tfrac{1}{8}. \end{aligned}$$

In both cases $|V^*_{\xi'}(h\beta) - V^*_{\xi'}(h\alpha)| > 1/9$. Hence we pick $\varepsilon := 1/9$ and get for every ε-optimal policy $\pi^\varepsilon_{\xi'}$ that $\pi^\varepsilon_{\xi'}(h) = \beta$ if and only if $n \in A$. □

Note the differences between Theorem 6.15 and Theorem 6.17: the former talks about optimal policies and shows that they are not computable, but is agnostic towards the underlying universal Turing machine. The latter talks about ε-optimal policies and gives a stronger hardness result, at the cost of depending on one particular universal Turing machine.

6.4 Iterative Value Function

Historically, AIXI's value function has been defined slightly differently to Definition 4.10, using a limit extension of an iterative definition of the value function. This definition is the more straightforward to come up with in AI: it is the natural adaptation of (optimal) minimax search in zero-sum games to the (optimal) expectimax algorithm for stochastic environments. In this section we discuss the problems with this definition.

To avoid confusion with the recursive value function V^π_ν, we denote the iterative value function with W^π_ν.[1]

Definition 6.18 (Iterative Value Function; Hutter, 2005, Def. 5.30)**.** The *iterative*

[1] In Leike and Hutter (2015a) the use of the symbols V and W is reversed.

Agent	Optimal	ε-Optimal
Iterative AINU	Δ_4^0, Σ_3^0-hard	Δ_3^0, Π_2^0-hard
Iterative AIXI	Δ_4^0, Π_2^0-hard	Δ_3^0, Π_2^0-hard
Iterative AIMU	Δ_2^0	Δ_1^0

Table 6.4: Computability results for different agent models that use the iterative value function derived in Section 6.4. Hardness results for AINU are with respect to a specific environment $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$.

value of a policy π in an environment ν given history $æ_{<t}$ is

$$W_\nu^\pi(æ_{<t}) := \frac{1}{\Gamma_t} \lim_{m\to\infty} \sum_{e_{t:m}} \nu(e_{1:m} \mid e_{<t} \parallel a_{1:m}) \sum_{k=t}^{m} \gamma(k) r_k$$

if $\Gamma_t > 0$ and $W_\nu^\pi(æ_{<t}) := 0$ if $\Gamma_t = 0$ where $a_i := \pi(e_{<i})$ for all $i \geq t$. The *optimal iterative value* is defined as $W_\nu^*(h) := \sup_\pi W_\nu^\pi(h)$.

Analogously to (4.2), we can write W_ν^* using the max-sum-operator:

$$W_\nu^*(æ_{<t}) = \frac{1}{\Gamma_t} \lim_{m\to\infty} \max_{æ_{t:m}} \nu(e_{1:m} \mid e_{<t} \parallel a_{1:m}) \sum_{k=t}^{m} \gamma(k) r_k \tag{6.7}$$

We use *iterative AINU* for the ν-optimal policy according to the iterative value function, and *iterative AIXI* for the ξ-optimal policy according to the iterative value function. Note that iterative AIMU coincides with AIMU since μ is a measure by convention.

Generally, our environment $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ is only a semimeasure and not a measure, i.e., there is a history $æ_{<t}a_t$ such that

$$1 > \sum_{e_t \in \mathcal{E}} \nu(e_t \mid e_{<t} \parallel a_{1:t}).$$

In such cases, with positive probability the environment ν does not produce a new percept e_t. If this occurs, we shall use the informal interpretation that the environment ν *ended*, but our formal argument does not rely on this interpretation.

The following proposition shows that for a semimeasure $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ that is not a measure, iterative AINU does not maximize ν-expected rewards. Recall that $\gamma(1)$ states the discount of the first reward. In the following, we assume without loss of generality that $\gamma(1) > 0$, i.e., we are not indifferent about the reward received in time step 1.

Proposition 6.19 (Iterative AINU is not a ν-Expected Reward Maximizer)**.** *For any $\varepsilon > 0$ there is an environment $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ that is not a measure and a policy π that receives a total of $\gamma(1)$ rewards in ν, but iterative AINU receives only $\varepsilon\gamma(1)$ rewards in ν.*

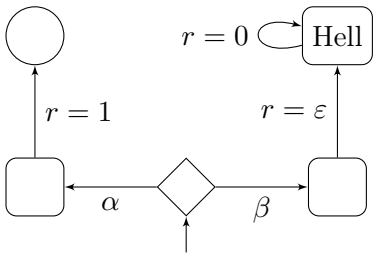


Figure 6.5: The environment ν from the proof of Proposition 6.19. Action α yields reward 1, but subsequently the environment ends. Action β yields reward ε and the environment continues forever. Iterative AINU will prefer the suboptimal action β, because it conditions on surviving forever.

Informally, the environment ν is defined as follows. In the first time step, the agent chooses between the two actions α and β. Taking action α gives a reward of 1, and subsequently the environment ends. Action β gives a reward of ε, but the environment continues forever. There are no other rewards in this environment. See Figure 6.5. From the perspective of ν-expected reward maximization, it is better to take action α, however iterative AINU takes action β.

Proof of Proposition 6.19. Let $\varepsilon > 0$. We ignore observations and set $\mathcal{E} := \{0, \varepsilon, 1\}$, $\mathcal{A} := \{\alpha, \beta\}$. The environment ν is formally defined by

$$\nu(r_{1:t} \parallel a_{1:t}) := \begin{cases} 1 & \text{if } a_1 = \alpha \text{ and } r_1 = 1 \text{ and } t = 1 \\ 1 & \text{if } a_1 = \beta \text{ and } r_1 = \varepsilon \text{ and } r_k = 0 \; \forall 1 < k \leq t \\ 0 & \text{otherwise.} \end{cases}$$

Taking action α first, we have $\nu(r_{1:t} \parallel \alpha a_{2:t}) = 0$ for $t > 1$ (the environment ν ends in time step 2 given history α). Hence we conclude

$$V_\nu^*(\alpha) = \frac{1}{\Gamma_t} \lim_{m \to \infty} \sum_{r_{1:m}} \nu(r_{1:m} \parallel \alpha a_{2:m}) \sum_{k=1}^{m} \gamma(k) r_k = 0.$$

Taking action β first we get

$$V_\nu^*(\beta) = \frac{1}{\Gamma_t} \lim_{m \to \infty} \sum_{r_{1:m}} \nu(r_{1:m} \parallel \beta a_{2:m}) \sum_{k=1}^{m} \gamma(k) r_k = \frac{\gamma(1)}{\Gamma_1} \varepsilon.$$

Since $\gamma(1) > 0$ and $\varepsilon > 0$, we have $V_\nu^*(\beta) > V_\nu^*(\alpha)$, and thus iterative AINU will use a policy that plays action β first, receiving a total discounted reward of $\varepsilon\gamma(1)$. In contrast, any policy π that takes action α first receives a larger total discounted reward of $\gamma(1)$. □

Whether it is reasonable to assume that our environment has a nonzero probability

of ending is a philosophical debate we do not want to engage in here; see Martin et al. (2016) for a discussion. Instead, we have a different motivation to use the recursive over the iterative value function: the latter has worse computability properties. Concretely, we show that ε-optimal iterative AIXI has to solve Π^0_2-hard problems and that there is an environment $\nu \in \mathcal{M}^{\text{CCS}}_{\text{LSC}}$ such that iterative AINU has to solve Σ^0_3-hard problems. In contrast, using the recursive value function, ε-optimal AIXI is Δ^0_2 according to Corollary 6.12 and AINU is Δ^0_3 according to Corollary 6.10.

The central difference between V^π_ν and W^π_ν is that for V^π_ν all obtained rewards matter, but for W^π_ν only the rewards in timelines that continue indefinitely. In this sense the value function W^π_ν conditions on surviving forever. If the environment μ is a measure, then the history is infinite with probability one, and so V^π_ν and W^π_ν coincide. Hence this distinction is not relevant for AIMU, only for AINU and AIXI.

Lemma 6.20 (Complexity of W^*_ν)**.** *For every $\nu \in \mathcal{M}^{\text{CCS}}_{\text{LSC}}$, the function W^*_ν is Π^0_2-computable.*

Proof. Multiplying (6.7) with $\Gamma_t \nu(e_{<t} \,\|\, a_{<t})$ yields $W^*_\nu(æ_{<t}) > q$ if and only if

$$\lim_{m\to\infty} \max_{æ_{t:m}} \nu(e_{1:m} \,\|\, a_{1:m}) \sum_{k=t}^{m} \gamma(k) r_k > q\, \Gamma_t\, \nu(e_{<t} \,\|\, a_{<t}). \tag{6.8}$$

The inequality's right side is lower semicomputable, hence there is a computable function ψ such that $\psi(\ell) \nearrow q\, \Gamma_t\, \nu(e_{<t} \,\|\, a_{<t}) =: q'$ as $\ell \to \infty$. (In contrast to the recursive value function, this quantity is not increasing in m.) For a fixed m, the left side is also lower semicomputable, therefore there is a computable function ϕ such that

$$\phi(m,k) \nearrow \max_{æ_{t:m}} \nu(e_{1:m} \,\|\, a_{1:m}) \sum_{k=t}^{m} \gamma(k) r_k =: f(m) \text{ as } k \to \infty.$$

We already know that the limit of $f(m)$ for $m \to \infty$ exists (uniquely), hence we can write (6.8) as

$$\begin{aligned}
&\lim_{m\to\infty} f(m) > q' \\
\iff\ & \forall m_0\, \exists m \geq m_0.\ f(m) > q' \\
\iff\ & \forall m_0\, \exists m \geq m_0\, \exists k.\ \phi(m,k) > q' \\
\iff\ & \forall \ell\, \forall m_0\, \exists m \geq m_0\, \exists k.\ \phi(m,k) > \psi(\ell),
\end{aligned}$$

which is a Π^0_2-formula. □

Note that in the finite horizon case where m is fixed, the value function W^*_ν is Δ^0_2-computable by Lemma 6.2d, since $W^*_\nu(æ_{<t}) = f(m)/q'$. In this case, we get the same computability results for iterative AINU as we did in Section 6.3.1.

Corollary 6.21 (Complexity of Iterative AINU)**.** *For any environment $\nu \in \mathcal{M}^{\text{CCS}}_{\text{LSC}}$, iterative AINU is Δ^0_4 and there is an ε-optimal iterative AINU that is Δ^0_3.*

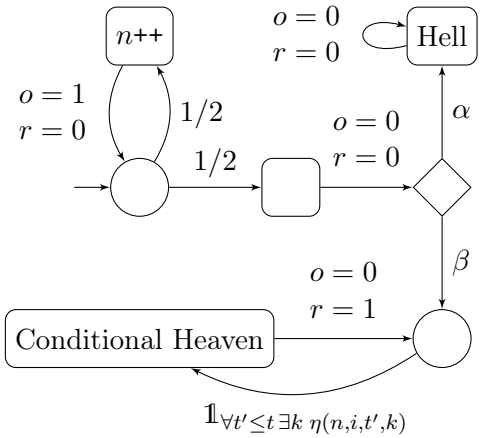


Figure 6.6: The environment ρ_i from the proof of Theorem 6.22. The mixture ν over class of environments $\mathcal{M} := \{\rho_0, \rho_1, \ldots\} \subset \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ forces iterative AINU to solve Σ_3^0-hard problems. 'Conditional Heaven' is a node that yields reward 1 until $\neg\exists k\ \eta(n,i,t,k)$, at which point the environment ends. Hence action β is preferred in environment ρ_i iff conditional heaven lasts forever (because otherwise $\nu(\ldots) = 0$ and hence $V_\nu^*(\ldots) = 0$) which is the case iff $\forall t\ \exists k\ \eta(n,i,t,k)$.

Proof. From Theorem 6.9, Theorem 6.11, and Lemma 6.20. □

We proceed to show corresponding lower bounds as in Section 6.3.2. For the rest of this section we assume $\Gamma_t > 0$ for all t.

Theorem 6.22 (Iterative AINU is Σ_3^0-hard)**.** *There is an environment $\nu \in \mathcal{M}_{\text{LSC}}^{\text{CCS}}$ such that iterative AINU is Σ_3^0-hard.*

Proof. The proof is analogous to the proof of Theorem 6.16. Let A be any Σ_3^0 set, then there is a quantifier-free formula η such that

$$n \in A \iff \exists i\ \forall t\ \exists k\ \eta(n,i,t,k).$$

We define the environments ρ_i similar to the proof of Theorem 6.16, except for two changes:

- We replace $\exists k\ \eta(n,i,k)$ with $\forall t' \leq t\ \exists k\ \eta(n,i,t',k)$.
- We switch actions α and β: action β 'checks' the formula η and action α gives a sure reward of 0.

Formally,

$$\rho_i((or)_{1:t} \parallel a_{1:t}) := \begin{cases} 2^{-t} & \text{if } o_{1:t} = 1^t \text{ and } \forall t' \leq t.\ r_{t'} = 0, \\ 2^{-n-1} & \text{if } \exists n.\ 1^n 0 \sqsubseteq o_{1:t} \sqsubseteq 1^n 0^\infty \text{ and } a_{n+2} = \alpha \\ & \quad \text{and } \forall t' \leq t.\ r_{t'} = 0, \\ 2^{-n-1} & \text{if } \exists n.\ 1^n 0 \sqsubseteq o_{1:t} \sqsubseteq 1^n 0^\infty \text{ and } a_{n+2} = \beta \\ & \quad \text{and } \forall t' \leq t.\ r_{t'} = \mathbb{1}_{t'>n+1} \\ & \quad \text{and } \forall t' \leq t\, \exists k\ \eta(n, i, t', k), \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

See Figure 6.6 for an illustration of the environment ρ_i. Every ρ_i is a chronological conditional semimeasure by definition, so $\mathcal{M} := \{\rho_0, \rho_1, \ldots\} \subseteq \mathcal{M}_{\text{LSC}}^{\text{CCS}}$. Furthermore, every ρ_i is lower semicomputable since η is quantifier-free.

We define our environment ν as a mixture over $\mathcal{M}$,

$$\nu := \sum_{i \in \mathbb{N}} 2^{-i-1} \rho_i;$$

the choice of the weights on the environments ρ_i is arbitrary but positive. We get for the ν-optimal policy π_ν^* analogously to the proof of Theorem 6.16

$$\pi_\nu^*(1^n 0) = \beta \iff \exists i\, \forall t' \leq t\, \exists k\ \eta(n, i, t', k) \iff n \in A,$$

since action α is preferred in ties. □

Analogously to Theorem 6.15, we can show that iterative AIXI is not computable. We also get the following lower bound.

Theorem 6.23 (Some ε-optimal iterative AIXI are Π_2^0-hard)**.** *There is a universal mixture ξ' and an $\varepsilon > 0$ such that any policy that is ε-optimal according to the iterative value for environment ξ' is Π_2^0-hard.*

Proof. Let A be a Π_2^0-set and η a quantifier-free formula such that

$$n \in A \iff \forall t\, \exists k\ \eta(n, t, k).$$

We proceed analogous to the proof of Theorem 6.17 except that we choose $\forall t' \leq t\, \exists k\ \eta(n, t, k)$ as a condition for reward 1 after playing action β.

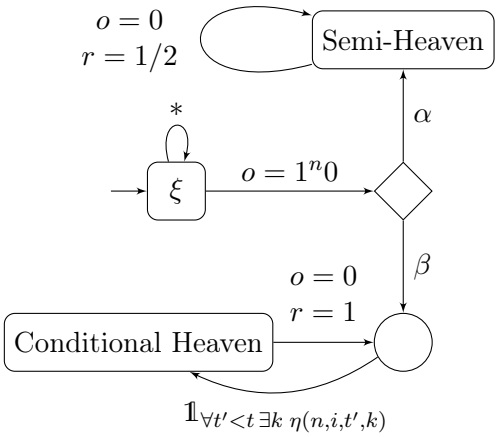


Figure 6.7: The environment ν from the proof of Theorem 6.23, which forces ε-optimal iterative AIXI to solve Π^0_2-hard problems. It functions just like ξ until the observation history is 1^n0. Then, action α is preferred iff conditional heaven never ends, i.e., iff $\forall t\, \exists k\, \eta(n,t,k)$.

Define the environment

$$\nu((or)_{1:t} \parallel a_{1:t}) := \begin{cases} \xi((or)_{1:n+1} \parallel a_{1:n+1}) & \text{if } \exists n.\ 1^n0 \sqsubseteq o_{1:t} \sqsubseteq 1^n0^\infty \\ & \quad \text{and } a_{n+1} = \alpha \\ & \quad \text{and } \forall n+1 < k \leq t.\ r_k = 1/2, \\ \xi((or)_{1:n+1} \parallel a_{1:n+1}) & \text{if } \exists n.\ 1^n0 \sqsubseteq o_{1:t} \sqsubseteq 1^n0^\infty \\ & \quad \text{and } a_{n+1} = \beta \\ & \quad \text{and } \forall n+1 < k \leq t.\ r_k = 1 \\ & \quad \text{and } \forall t' \leq t\, \exists k\, \eta(n,t,k), \\ \xi((or)_{1:t} \parallel a_{1:t}) & \text{if } \nexists n.\ 1^n0 \sqsubseteq o_{1:t} \sqsubseteq 1^n0^\infty, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

See Figure 6.7 for an illustration of the environment ν. The environment ν mimics the universal environment ξ until the observation history is 1^n0. The next action α always gives rewards $1/2$ forever, while action β gives rewards 1 forever iff $n \in A$. We have that ν is a lower semicomputable semimeasure since ξ is a lower semicomputable semimeasure and η is quantifier-free. We define $\xi' = \frac{1}{2}\nu + \frac{1}{8}\xi$. By Lemma 4.24, ξ' is a universal lower semicomputable semimeasure. Let $n \in A$ be given and let $h \in (\mathcal{A} \times \mathcal{O})^{x+1}$ be any history with observations $o_{1:n+1} = 1^n0$. In the following, we use the linearity of W^*_ρ in ρ (analogously to Lemma 4.14). If $\forall t \exists k\, \eta(n,t,k)$, then

$$\begin{aligned} W^*_{\xi'}(h\beta) - W^*_{\xi'}(h\alpha) &= \tfrac{1}{2}W^*_\nu(h\beta) - \tfrac{1}{2}W^*_\nu(h\alpha) + \tfrac{1}{8}W^*_\xi(h\beta) - \tfrac{1}{8}W^*_\xi(h\alpha) \\ &\geq \tfrac{1}{2} - \tfrac{1}{4} + 0 - \tfrac{1}{8} = \tfrac{1}{8}, \end{aligned}$$

and similarly if $\neg\forall t\exists k\ \eta(n,t,k)$, then

$$\begin{aligned} W^*_{\xi'}(h\alpha) - W^*_{\xi'}(h\beta) &= \tfrac{1}{2}W^*_\nu(h\alpha) - \tfrac{1}{2}W^*_\nu(h\beta) + \tfrac{1}{8}W^*_\xi(h\alpha) - \tfrac{1}{8}W^*_\xi(h\beta) \\ &\geq \tfrac{1}{4} - 0 + 0 - \tfrac{1}{8} = \tfrac{1}{8}. \end{aligned}$$

In both cases $|W^*_{\xi'}(h\beta) - W^*_{\xi'}(h\alpha)| > 1/9$, hence with $\varepsilon := 1/9$ we have for an ε-optimal policy $\pi^\varepsilon_{\xi'}$ that $\pi^\varepsilon_{\xi'}(h) = \beta$ if and only if $n \in A$. □

6.5 The Complexity of Knowledge-Seeking

Recall the definition of the optimal entropy-seeking value $V^{*,m}_{\mathrm{Ent}}$ and the optimal information-seeking value $V^{*,m}_{\mathrm{IG}}$ from Section 4.3.2. Using the results from Section 6.3 we can show that ε-optimal knowledge-seeking agents are limit computable, and optimal knowledge-seeking agents are Δ^0_3.

Corollary 6.24 (Computability of Knowledge-Seeking Values)**.** *For fixed m, the value functions $V^{*,m}_{\mathrm{Ent}}$ and $V^{*,m}_{\mathrm{IG}}$ are limit computable.*

Proof. This follows from Lemma 6.2 (c-e) since ξ, ν, and w are lower semicomputable. □

Corollary 6.25 (Computability of Knowledge-Seeking Policies)**.** *For entropy-seeking and information-seeking agents there are limit-computable ε-optimal policies and Δ^0_3-computable optimal policies.*

Proof. Follows from Corollary 6.24, Theorem 6.9, and Theorem 6.11. □

Note that if we used an infinite horizon with discounting in Definition 4.25 or Definition 4.26, then we cannot retain this computability result without further assumptions: we would need that the value functions increase monotonically as $m \to \infty$, as they do for the recursive value function from Definition 4.10. However, entropy is not a monotone function and may decrease if there are events whose probability converges to something $> 1/2$. For the entropy-seeking value function this happens for histories drawn from a deterministic environment μ since $\xi \to \mu$, so the conditionals converge to 1. Similarly, for the information-seeking value function, the posterior belief in one (deterministic) environment might become larger than $1/2$ (depending on the prior and the environment class). Therefore we generally only get that discounted versions of V^*_{Ent} and V^*_{IG} are Δ^0_3 analogously to Lemma 6.20. Hence optimal discounted entropy-seeking and optimal discounted information-seeking policies are in Δ^0_4 by Theorem 6.9 and their corresponding ε-optimal siblings are Δ^0_3 by Theorem 6.11.

6.6 A Limit Computable Weakly Asymptotically Optimal Agent

According to Theorem 6.16, optimal reward-seeking policies are generally Π^0_2-hard, and for optimal knowledge-seeking policies Corollary 6.25 shows that they are Δ^0_3. Therefore

we get that BayesExp is Δ_3^0:

Corollary 6.26 (BayesExp is Δ_3^0)**.** *For any universal mixture ξ, BayesExp is Δ_3^0.*

Proof. From Corollary 6.10, Corollary 6.24, and Corollary 6.25. □

However, we do not know BayesExp to be limit computable, and we expect it not to be. However, we can approximate it using ε-optimal policies preserving weak asymptotic optimality.

Theorem 6.27 (A Limit-Computable Weakly Asymptotically Optimal Agent)**.** *If there is a nonincreasing computable sequence of positive reals $(\varepsilon_t)_{t\in\mathbb{N}}$ such that $\varepsilon_t \to 0$ and $H_t(\varepsilon_t)/(t\varepsilon_t) \to 0$ as $t \to \infty$, then there is a limit-computable policy that is weakly asymptotically optimal in the class of all computable stochastic environments.*

Proof. By Corollary 6.10, there is a limit-computable 2^{-t}-optimal reward-seeking policy π_ξ^t for the universal mixture ξ. By Corollary 6.25 there are limit-computable $\epsilon_t/2$-optimal information-seeking policies π_{IG}^t with horizon $t + H_t(\varepsilon_t)$. We define a policy π analogously to Algorithm 1 with π_{IG}^t and π_ξ^t instead of the optimal policies. From Corollary 6.24 we get that V_{IG}^* is limit computable, so the policy π is limit computable. Furthermore, π_ξ^t is 2^{-t}-optimal and $2^{-t} \to 0$, so $V_\xi^{\pi_\xi^t}(æ_{<t}) \to V_\xi^*(æ_{<t})$ as $t \to \infty$.

Now we can proceed analogously to the proof of Lattimore (2013, Thm. 5.6), which consists of three parts. First, it is shown that the value of the ξ-optimal reward-seeking policy π_ξ^* converges to the optimal value for exploitation time steps (line 6 in Algorithm 1) in the sense that $V_\mu^{\pi_\xi^*} \to V_\mu^*$. This carries over to the 2^{-t}-optimal policy π_ξ^t, since the key property is that on exploitation steps, $V_{\text{IG}}^* < \varepsilon_t$; i.e., π only exploits if potential knowledge-seeking value is low. In short, we get for exploitation steps

$$V_\xi^{\pi_\xi^t}(æ_{<t}) \to V_\xi^{\pi_\xi^*}(æ_{<t}) \to V_\mu^{\pi_\xi^*}(æ_{<t}) \to V_\mu^*(æ_{<t}) \text{ as } t \to \infty.$$

Second, it is shown that the density of exploration steps vanishes. This result carries over since the condition $V_{\text{IG}}^*(æ_{<t}) > \varepsilon_t$ that determines exploration steps is exactly the same as for BayesExp and π_{IG}^t is $\varepsilon_t/2$-optimal.

Third, the results of part one and two are used to conclude that π is weakly asymptotically optimal. This part carries over to our proof. □

6.7 Discussion

When using Solomonoff's prior for induction, we need to evaluate conditional probabilities. We showed that conditional M and M_{norm} are limit computable (Theorem 6.3), and that $\overline{M}$ and $\overline{M}_{\text{norm}}$ are not limit computable (Theorem 6.4 and Corollary 6.5). Table 6.1 on page 102 summarizes our computability results on various versions of Solomonoff's prior. Theses results implies that we can approximate M or M_{norm} for prediction, but not the measure mixture $\overline{M}$ or $\overline{M}_{\text{norm}}$.

In some cases, normalized priors have advantages. As illustrated in Example 4.27, unnormalized priors can make the entropy-seeking agent mistake the entropy gained

from the probability assigned to finite strings for knowledge. From $M_{\text{norm}} \geq M$ we get that M_{norm} predicts just as well as M, and by Theorem 6.3 we can use M_{norm} without losing limit computability.

Table 6.2 on page 102 summarizes our computability results for the agents AINU, AIXI, and AINU. AINU is Δ_3^0 and restricting to ε-optimal policies decreases the level by one (Corollary 6.10 and Corollary 6.12). For environments from $\mathcal{M}_{\text{comp}}^{\text{CCM}}$, AIMU is limit-computable and ε-optimal AIMU is computable (Corollary 6.14). In Section 6.3.2 we proved that these computability bounds on AINU are generally unimprovable (Theorem 6.16 and Theorem 6.17). Additionally, we proved weaker lower bounds for AIXI independent of the universal Turing machine (Theorem 6.15) and for ε-optimal AIXI for specific choices of the universal Turing machine (Theorem 6.17).

When the environment ν has nonzero probability of not producing a new percept, the iterative definition of AINU (Definition 6.18) originally given by Hutter (2005, Def. 5.30) fails to maximize ν-expected rewards (Proposition 6.19). Moreover, the policies are one level higher in the arithmetical hierarchy (see Table 6.4 on page 116). We proved upper (Corollary 6.21) and lower bounds (Theorem 6.22 and Theorem 6.23). The difference between the recursive value function V and the iterative value function W is readily exposed in the difference between the universal prior M and the measure mixture $\overline{M}$: Just like W conditions on surviving forever, so does $\overline{M}$ eliminate the weight of programs that do not produce infinite strings. Both $\overline{M}$ and W are not limit computable for this reason.

We considered ε-optimality to avoid having to determine argmax ties. This ε does not have to be constant over time, we may let $\varepsilon \to 0$ as $t \to \infty$ at any computable rate. With this we retain the computability results of ε-optimal policies and get that the value of the $\varepsilon(t)$-optimal policy $\pi_\nu^{\varepsilon(t)}$ converges rapidly to the ν-optimal value:

$$V_\nu^*(æ_{<t}) - V_\nu^{\pi_\nu^{\varepsilon(t)}}(æ_{<t}) \to 0 \text{ as } t \to \infty.$$

In Section 4.1 we defined the set $\mathcal{M}_{\text{LSC}}^{\text{CCS}}$ as the set of all lower semicomputable chronological contextual semimeasure over percepts with actions povided as side-information. When determining the probability of the next percept e_t in an environment ν, we have to compute $\nu(e_{1:t} \mid e_{<t} \parallel a_{1:t})$. Alternatively, we could have defined the environment as a lower semicomputable mapping from histories $æ_{<t}a_t$ to probabilities over the next percept e_t (this is done in Chapter 7). For the proof of Lemma 6.8 and Lemma 6.20 we only need that $\nu(e_{1:t} \parallel a_{1:t})$ is lower semicomputable computable. While this new definition makes no difference for the computability of AINU, it matters for AIXI because in the mixture ξ over all of these environments is no longer lower semicomputable.

Any method that tries to tackle the reinforcement learning problem has to balance between exploration and exploitation. AIXI strikes this balance in the Bayesian way. However, as we showed in Section 5.2, this may not lead to enough exploration. To counteract this, we can add an explorative component to the agent, akin to knowledge-seeking agents. In Section 6.5 we show that ε-optimal knowledge-seeking agents are limit computable if we use the recursive definition of the value function.

We set out with the goal of finding a perfect reinforcement learning agent that is limit computable. The Bayesian agent AIXI could be considered one suitable candidate, despite its optimality problems discussed in Chapter 5. Another suitable candidate are weakly asymptotically optimal agents, which in contrast to AIXI are optimal in an objective sense (see Section 5.6). We discussed BayesExp, which relies on a Solomonoff prior to learn its environment and on a information-seeking component for extra exploration. Our results culminated in a limit-computable weakly asymptotically optimal agent based on BayesExp (Theorem 6.27). In this sense our goal has been achieved.

Chapter 7

The Grain of Truth Problem[1]

AIs become friendly by playing lots of Newcomblike problems. — Eliezer Yudkowsky

Consider the general setup of multiple reinforcement learning agents interacting sequentially in a known environment with the goal to maximize discounted reward.[2] Each agent knows how the environment behaves, but does not know the other agents' behavior. The natural (Bayesian) approach would be to define a class of possible policies that the other agents could adopt and take a prior over this class. During the interaction, this prior gets updated to the posterior as our agent learns the others' behavior. Our agent then acts optimally with respect to this posterior belief. Kalai and Lehrer (1993) show that in infinitely repeated games Bayesian agents converge to an ε-Nash equilibrium as long as each agent assigns positive prior probability to the other agents' policies (a *grain of truth*).

As an example, consider an infinitely repeated prisoners dilemma between two agents. In every time step the payoff matrix is as follows, where C means cooperate and D means defect.

	C	D
C	3/4, 3/4	0, 1
D	1, 0	1/4, 1/4

Define the set of policies $\Pi := \{\pi_\infty, \pi_0, \pi_1, \ldots\}$ where policy π_t cooperates until time step t or the opponent defects (whatever happens first) and defects thereafter. The Bayes optimal behavior is to cooperate until the posterior belief that the other agent defects in the time step after the next is greater than some constant (depending on the discount function) and then defect afterwards. Therefore Bayes optimal behavior leads to a policy from the set Π (regardless of the prior). If both agents are Bayes optimal with respect to some prior, they both have a grain of truth and therefore they converge to a Nash equilibrium: either they both cooperate forever or after some finite time they both defect forever. Alternating strategies like TitForTat (cooperate first, then play

[1] The idea for reflective oracles was developed by Jessica Taylor and Benya Fallenstein based on ideas by Paul Christiano (Christiano et al., 2013). Reflective oracles were first described in Fallenstein et al. (2015a). The proof of Theorem 7.7 was sketched by Benya Fallenstein and developed by me. Except for minor editing, everything else in this chapter is my own work.

[2] We mostly use the terminology of reinforcement learning. For readers from game theory we provide a dictionary in Table 7.1.

Reinforcement learning	Game theory
stochastic policy	mixed strategy
deterministic policy	pure strategy
agent	player
multi-agent environment	infinite extensive-form game
reward	payoff/utility
(finite) history	history
infinite history	path of play

Table 7.1: Terminology dictionary between reinforcement learning and game theory.

the opponent's last action) are not part of the policy class Π, and adding them to the class breaks the grain of truth property: the Bayes optimal behavior is no longer in the class. This is rather typical; a Bayesian agent usually needs to be more powerful than its environment (see Section 6.3). We are facing the following problem.

Problem 7.1 (Grain of Truth Problem; Hutter, 2009b, Q. 5j)**.** *Find a large class of policies* Π *containing Bayesian agents with positive prior over* Π*.*

Until now, classes that admit a grain of truth were known only for small toy examples such as the iterated prisoner's dilemma above (Shoham and Leyton-Brown, 2009, Ch. 7.3). Foster and Young (2001) and Nachbar (1997, 2005) prove several impossibility results on the grain of truth problem that identify properties that cannot be simultaneously satisfied for classes that allow a grain of truth (see Section 7.6 for a discussion).

In this chapter we present a formal solution to the grain of truth problem (Section 7.2). We assume that our multi-agent environment is computable, but it does not need to be stationary/Markov, ergodic, or finite-state. Our class of policies Π is large enough to contain all computable (stochastic) policies, as well as all relevant Bayes optimal policies. At the same time, our class is small enough to be limit computable. This is important because it allows our result to be computationally approximated.

In Section 7.3 we consider the setting where the multi-agent environment is unknown to the agents and has to be learned in addition to the other agents' behavior. A Bayes optimal agent may not learn to act optimally in unknown multi-agent environments *even though it has a grain of truth.* This effect occurs in non-recoverable environments where taking one wrong action can mean a permanent loss of future value. In this case, a Bayes optimal agent avoids taking these dangerous actions and therefore will not explore enough to wash out the prior's bias (using the dogmatic prior from Section 5.2.2). Therefore, Bayesian agents are not *asymptotically optimal*, i.e., they do not always learn to act optimally (Theorem 5.22).

However, asymptotic optimality is achieved by Thompson sampling because the inherent randomness of Thompson sampling leads to enough exploration to learn the entire environment class (see Section 5.4.3). This leads to our main result: if all agents use Thompson sampling over our class of multi-agent environments, then for every $\varepsilon > 0$

they converge to an ε-Nash equilibrium. This is not the first time Thompson sampling is used in game theory (Ortega and Braun, 2014), but the first time to show that it achieves such general positive results.

The central idea to our construction is based on *reflective oracles* introduced by Fallenstein et al. (2015a,b). Reflective oracles are probabilistic oracles similar to halting oracles that answer whether the probability that a given probabilistic Turing machine T outputs 1 is higher than a given rational number p. The oracles are reflective in the sense that the machine T may itself query the oracle, so the oracle has to answer queries about itself. This invites issues caused by self-referential liar paradoxes of the form "if the oracle says that this machine return 1 with probability $> 1/2$, then return 0, else return 1." Reflective oracles avoid these issues by being allowed to randomize if the machines do not halt or the rational number is *exactly* the probability to output 1. We introduce reflective oracles formally in Section 7.1 and prove that there is a limit computable reflective oracle.

For infinitely repeated games practical algorithms rely on *ficticious play* (Fudenberg and Levine, 1998, Ch. 2): the agent takes a best-response action based on the assumption that its opponent is playing a stationary but unknown mixed strategy estimated according to the observed empirical frequencies. If all agents converge to a stationary policy, then this is a Nash equilibrium. However, convergence is not guaranteed.

The same problem occurs in multi-agent reinforcement learning (Busoniu et al., 2008). Reinforcement learning algorithms typically assume a (stationary) Markov decision process. This assumption is violated when interacting with other reinforcement learning agents because as these agents learn, their behavior changes and thus they are not stationary. Assuming convergence to a stationary policy is a necessary criterion to enable all agents to learn, but the process is unstable for many reinforcement learning algorithms and only empirical positive results are known (Bowling and Veloso, 2001).

7.1 Reflective Oracles

7.1.1 Definition

First we connect semimeasures as defined in Definition 2.14 to Turing machines. In Chapter 2 we used *monotone Turing machines* which naturally correspond to lower semicomputable semimeasures (Li and Vitányi, 2008, Sec. 4.5.2) that describe the distribution that arises when piping fair coin flips into the monotone machine. Here we take a different route.

A *probabilistic Turing machine* is a Turing machine that has access to an unlimited number of uniformly random coin flips. Let $\mathcal{T}$ denote the set of all probabilistic Turing machines that take some input in $\mathcal{X}^*$ and may query an oracle (formally defined below). We take a Turing machine $T \in \mathcal{T}$ to correspond to a semimeasure λ_T where $\lambda_T(a \mid x)$ is the probability that T outputs $a \in \mathcal{X}$ when given $x \in \mathcal{X}^*$ as input. The value of

$\lambda_T(x)$ is then given by the chain rule

$$\lambda_T(x) := \prod_{k=1}^{|x|} \lambda_T(x_k \mid x_{<k}). \tag{7.1}$$

Thus $\mathcal{T}$ gives rise to the set of semimeasures $\mathcal{M}_{\text{LSC}}$ where the *conditionals* $\lambda(a \mid x)$ are lower semicomputable. In contrast, in Chapter 6 we considered semimeasures whose *joint* probability (7.1) is lower semicomputable. This set $\mathcal{M}_{\text{LSC}}$ contains all computable measures. However, $\mathcal{M}_{\text{LSC}}$ is a proper subset of the set of all lower semicomputable semimeasures because the product (7.1) is lower semicomputable, but there are some lower semicomputable semimeasures whose conditional is not lower semicomputable (Theorem 6.6):

$$\mathcal{M}^{\text{CCM}}_{\text{comp}} \subset \mathcal{M}_{\text{LSC}} \subset \mathcal{M}^{\text{CCS}}_{\text{LSC}}$$

In the following we assume that our alphabet is binary, i.e., $\mathcal{X} := \{0,1\}$.

Definition 7.2 (Oracle)**.** An *oracle* is a function $O : \mathcal{T} \times \{0,1\}^* \times \mathbb{Q} \to \Delta\{0,1\}$.

Oracles are understood to be probabilistic: they randomly return 0 or 1. Let T^O denote the machine $T \in \mathcal{T}$ when run with the oracle O, and let λ_T^O denote the semimeasure induced by T^O. This means that drawing from λ_T^O involves two sources of randomness: one from the distribution induced by the probabilistic Turing machine T and one from the oracle's answers.

The intended semantics of an oracle are that it takes a *query* (T, x, p) and returns 1 if the machine T^O outputs 1 on input x with probability greater than p when run with the oracle O, i.e., when $\lambda_T^O(1 \mid x) > p$. Furthermore, the oracle returns 0 if the machine T^O outputs 1 on input x with probability less than p when run with the oracle O, i.e., when $\lambda_T^O(1 \mid x) < p$. To fulfill this, the oracle O has to make statements about itself, since the machine T from the query may again query O. Therefore we call oracles of this kind *reflective oracles*. This has to be defined very carefully to avoid the obvious diagonalization issues that are caused by programs that ask the oracle about themselves. We impose the following self-consistency constraint.

Definition 7.3 (Reflective Oracle)**.** An oracle O is *reflective* iff for all queries $(T, x, p) \in \mathcal{T} \times \{0,1\}^* \times \mathbb{Q}$,

(a) $\lambda_T^O(1 \mid x) > p$ implies $O(T, x, p) = 1$, and

(b) $\lambda_T^O(0 \mid x) > 1 - p$ implies $O(T, x, p) = 0$.

If p under- or overshoots the true probability of $\lambda_T^O(\cdot \mid x)$, then the oracle must reveal this information. However, in the critical case when $p = \lambda_T^O(1 \mid x)$, the oracle is allowed to return anything and may randomize its result. Furthermore, since T might not output any symbol, it is possible that $\lambda_T^O(0 \mid x) + \lambda_T^O(1 \mid x) < 1$. In this case the oracle can reassign the non-halting probability mass to 0, 1, or randomize; see Figure 7.1.

Figure 7.1: Answer options of a reflective oracle O for the query (T, x, p); the rational $p \in [0, 1]$ falls into one of the three regions above. The values of $\lambda_T^O(0 \mid x)$ and $\lambda_T^O(1 \mid x)$ are depicted as the length of the line segment under which they are written.

Example 7.4 (Reflective Oracles and Diagonalization)**.** Let $T \in \mathcal{T}$ be a probabilistic Turing machine that outputs $1 - O(T, \epsilon, 1/2)$ (T can know its own source code by quining; Kleene, 1952, Thm. 27). In other words, T queries the oracle about whether it is more likely to output 1 or 0, and then does whichever the oracle says is less likely. In this case we can use an oracle $O(T, \epsilon, 1/2) := 1/2$ (answer 0 or 1 with equal probability), which implies $\lambda_T^O(1 \mid \epsilon) = \lambda_T^O(0 \mid \epsilon) = 1/2$, so the conditions of Definition 7.3 are satisfied. In fact, for this machine T we must have $O(T, \epsilon, 1/2) = 1/2$ for all reflective oracles O. ◇

The following theorem establishes that reflective oracles exist.

Theorem 7.5 (Fallenstein et al., 2015c, App. B)**.** *There is a reflective oracle.*

Definition 7.6 (Reflective-Oracle-Computable)**.** A semimeasure is called *reflective-oracle-computable* iff it is computable on a probabilistic Turing machine with access to a reflective oracle.

For any probabilistic Turing machine $T \in \mathcal{T}$ we can complete the semimeasure $\lambda_T^O(\cdot \mid x)$ into a reflective-oracle-computable measure $\overline{\lambda}_T^O(\cdot \mid x)$: Using the oracle O and a binary search on the parameter p we search for the crossover point p where $O(T, x, p)$ goes from returning 1 to returning 0. The limit point $p_x^* \in \mathbb{R}$ of the binary search is random since the oracle's answers may be random. But the main point is that the expectation of p_x^* exists, so $\overline{\lambda}_T^O(1 \mid x) = \mathbb{E}[p_x^*] = 1 - \overline{\lambda}_T^O(0 \mid x)$. Hence $\overline{\lambda}_T^O$ is a measure. Moreover, if the oracle is reflective, then $\overline{\lambda}_T^O(x) \geq \lambda_T^O(x)$ for all $x \in \mathcal{X}^*$. In this sense the oracle O can be viewed as a way of 'completing' all semimeasures λ_T^O to measures by arbitrarily assigning the non-halting probability mass. If the oracle O is reflective this is consistent in the sense that Turing machines who run other Turing machines will be completed in the same way. This is especially important for a universal machine that runs all other Turing machines to induce a Solomonoff prior (Example 3.5).

7.1.2 A Limit Computable Reflective Oracle

The proof of Theorem 7.5 given by Fallenstein et al. (2015c, App. B) is nonconstructive and uses the axiom of choice. In Section 7.1.3 we give a new proof for the existence of reflective oracles and provide a construction that there is a reflective oracle that is limit computable.

Theorem 7.7 (A Limit Computable Reflective Oracle)**.** *There is a reflective oracle that is limit computable.*

This theorem has the immediate consequence that reflective oracles cannot be used as halting oracles. At first, this result may seem surprising: according to the definition of reflective oracles, they make concrete statements about the output of probabilistic Turing machines. However, the fact that the oracles may randomize some of the time actually removes enough information such that halting can no longer be decided from the oracle output.

Corollary 7.8 (Reflective Oracles are not Halting Oracles)**.** *There is no probabilistic Turing machine T such that for every prefix program p and every reflective oracle O, we have that $\lambda_T^O(1 \mid p) > 1/2$ if p halts and $\lambda_T^O(1 \mid p) < 1/2$ otherwise.*

Proof. Assume there was such a machine T and let O be the limit computable oracle from Theorem 7.7. Since O is reflective we can turn T into a deterministic halting oracle by calling $O(T, p, 1/2)$ which deterministically returns 1 if p halts and 0 otherwise. Since O is limit computable, we can finitely compute the output of O on any query to arbitrary finite precision using our deterministic halting oracle. We construct a probabilistic Turing machine T' that uses our halting oracle to compute (rather than query) the oracle O on $(T', \epsilon, 1/2)$ to a precision of $1/3$ in finite time. If $O(T', \epsilon, 1/2) \pm 1/3 > 1/2$, the machine T' outputs 0, otherwise T' outputs 1. Since our halting oracle is entirely deterministic, the output of T' is entirely deterministic as well (and T' always halts), so $\lambda_{T'}^O(0 \mid \epsilon) = 1$ or $\lambda_{T'}^O(1 \mid \epsilon) = 1$. Therefore $O(T', \epsilon, 1/2) = 1$ or $O(T', \epsilon, 1/2) = 0$ because O is reflective. A precision of $1/3$ is enough to tell them apart, hence T' returns 0 if $O(T', \epsilon, 1/2) = 1$ and T' returns 1 if $O(T', \epsilon, 1/2) = 0$. This is a contradiction. □

A similar argument can also be used to show that reflective oracles are not computable.

7.1.3 Proof of Theorem 7.7

The idea for the proof of Theorem 7.7 is to construct an algorithm that outputs an infinite series of *partial oracles* converging to a reflective oracle in the limit.

The set of queries is countable, so we can assume that we have some computable enumeration of it:

$$\mathcal{T} \times \{0,1\}^* \times \mathbb{Q} =: \{q_1, q_2, \ldots\}$$

Definition 7.9 (k-Partial Oracle)**.** A *k-partial oracle* $\tilde{O}$ is function from the first k queries to the multiples of 2^{-k} in $[0,1]$:

$$\tilde{O} : \{q_1, q_2, \ldots, q_k\} \to \{n2^{-k} \mid 0 \leq n \leq 2^k\}$$

Definition 7.10 (Approximating an Oracle)**.** A k-partial oracle $\tilde{O}$ *approximates* an oracle O iff $|O(q_i) - \tilde{O}(q_i)| \leq 2^{-k-1}$ for all $i \leq k$.

Let $k \in \mathbb{N}$, let $\tilde{O}$ be a k-partial oracle, and let $T \in \mathcal{T}$ be an oracle machine. The machine $T^{\tilde{O}}$ that we get when we run T with the k-partial oracle $\tilde{O}$ is defined as follows (this is with slight abuse of notation since k is taken to be understood implicitly).

1. Run T for at most k steps.
2. If T calls the oracle on q_i for $i \leq k$,
   (a) return 1 with probability $\tilde{O}(q_i) - 2^{-k-1}$,
   (b) return 0 with probability $1 - \tilde{O}(q_i) - 2^{-k-1}$, and
   (c) halt otherwise.
3. If T calls the oracle on q_j for $j > k$, halt.

Furthermore, we define $\lambda_T^{\tilde{O}}$ analogously to λ_T^O as the distribution generated by the machine $T^{\tilde{O}}$.

Lemma 7.11. *If a k-partial oracle $\tilde{O}$ approximates a reflective oracle O, then $\lambda_T^O(1 \mid x) \geq \lambda_T^{\tilde{O}}(1 \mid x)$ and $\lambda_T^O(0 \mid x) \geq \lambda_T^{\tilde{O}}(0 \mid x)$ for all $x \in \{0,1\}^*$ and all $T \in \mathcal{T}$.*

Proof. This follows from the definition of $T^{\tilde{O}}$: when running T with $\tilde{O}$ instead of O, we can only lose probability mass. If T makes calls whose index is $> k$ or runs for more than k steps, then the execution is aborted and no further output is generated. If T makes calls whose index $i \leq k$, then $\tilde{O}(q_i) - 2^{-k-1} \leq O(q_i)$ since $\tilde{O}$ approximates O. Therefore the return of the call q_i is underestimated as well. □

Definition 7.12 (k-Partially Reflective)**.** A k-partial oracle $\tilde{O}$ is *k-partially reflective* iff for the first k queries (T, x, p)

- $\lambda_T^{\tilde{O}}(1 \mid x) > p$ implies $\tilde{O}(T, x, p) = 1$, and
- $\lambda_T^{\tilde{O}}(0 \mid x) > 1 - p$ implies $\tilde{O}(T, x, p) = 0$.

It is important to note that we can check whether a k-partial oracle is k-partially reflective in finite time by running all machines T from the first k queries for k steps and tallying up the probabilities to compute $\lambda_T^{\tilde{O}}$.

Lemma 7.13. *If O is a reflective oracle and $\tilde{O}$ is a k-partial oracle that approximates O, then $\tilde{O}$ is k-partially reflective.*

Lemma 7.13 only holds because we use semimeasures whose conditionals are lower semicomputable.

Proof. Assuming $\lambda_T^{\tilde{O}}(1 \mid x) > p$ we get from Lemma 7.11 that $\lambda_T^O(1 \mid x) \geq \lambda_T^{\tilde{O}}(1 \mid x) > p$. Thus $O(T, x, p) = 1$ because O is reflective. Since $\tilde{O}$ approximates O, we get $1 = O(T, x, p) \leq \tilde{O}(T, x, p) + 2^{-k-1}$, and since $\tilde{O}$ assigns values in a 2^{-k}-grid, it follows that $\tilde{O}(T, x, p) = 1$. The second implication is proved analogously. □

Definition 7.14 (Extending Partial Oracles)**.** A $k+1$-partial oracle $\tilde{O}'$ *extends* a k-partial oracle $\tilde{O}$ iff $|\tilde{O}(q_i) - \tilde{O}'(q_i)| \leq 2^{-k-1}$ for all $i \leq k$.

Lemma 7.15. *There is an infinite sequence of partial oracles $(\tilde{O}_k)_{k\in\mathbb{N}}$ such that for each k, $\tilde{O}_k$ is a k-partially reflective k-partial oracle and $\tilde{O}_{k+1}$ extends $\tilde{O}_k$.*

Proof. By Theorem 7.5 there is a reflective oracle O. For every k, there is a canonical k-partial oracle $\tilde{O}_k$ that approximates O: restrict O to the first k queries and for any such query q pick the value in the 2^{-k}-grid which is closest to $O(q)$. By construction, $\tilde{O}_{k+1}$ extends $\tilde{O}_k$ and by Lemma 7.13, each $\tilde{O}_k$ is k-partially reflective. □

Lemma 7.16. *If the $k+1$-partial oracle $\tilde{O}_{k+1}$ extends the k-partial oracle $\tilde{O}_k$, then $\lambda_T^{\tilde{O}_{k+1}}(1 \mid x) \geq \lambda_T^{\tilde{O}_k}(1 \mid x)$ and $\lambda_T^{\tilde{O}_{k+1}}(0 \mid x) \geq \lambda_T^{\tilde{O}_k}(0 \mid x)$ for all $x \in \{0,1\}^*$ and all $T \in \mathcal{T}$.*

Proof. $T^{\tilde{O}_{k+1}}$ runs for one more step than $T^{\tilde{O}_k}$, can answer one more query and has increased oracle precision. Moreover, since $\tilde{O}_{k+1}$ extends $\tilde{O}_k$, we have $|\tilde{O}_{k+1}(q_i) - \tilde{O}_k(q_i)| \leq 2^{-k-1}$, and thus $\tilde{O}_{k+1}(q_i) - 2^{-k-1} \geq \tilde{O}_k(q_i) - 2^{-k}$. Therefore the success to answers to the oracle calls (case 2(a) and 2(b)) will not decrease in probability. □

Now everything is in place to state the algorithm that constructs a reflective oracle in the limit. It recursively traverses a tree of partial oracles. The tree's nodes are the partial oracles; level k of the tree contains all k-partial oracles. There is an edge in the tree from the k-partial oracle $\tilde{O}_k$ to the i-partial oracle $\tilde{O}_i$ if and only if $i = k+1$ and $\tilde{O}_i$ extends $\tilde{O}_k$.

For every k, there are only finitely many k-partial oracles, since they are functions from finite sets to finite sets. In particular, there are exactly two 1-partial oracles (so the search tree has two roots). Pick one of them to start with, and proceed recursively as follows. Given a k-partial oracle $\tilde{O}_k$, there are finitely many $(k+1)$-partial oracles that extend $\tilde{O}_k$ (finite branching of the tree). Pick one that is $(k+1)$-partially reflective (which can be checked in finite time). If there is no $(k+1)$-partially reflective extension, backtrack.

By Lemma 7.15 our search tree is infinitely deep and thus the tree search does not terminate. Moreover, it can backtrack to each level only a finite number of times because at each level there is only a finite number of possible extensions. Therefore the algorithm will produce an infinite sequence of partial oracles, each extending the previous. Because of finite backtracking, the output eventually stabilizes on a sequence of partial oracles $\tilde{O}_1, \tilde{O}_2, \ldots$. By the following lemma, this sequence converges to a reflective oracle, which concludes the proof of Theorem 7.7.

Lemma 7.17. *Let $\tilde{O}_1, \tilde{O}_2, \ldots$ be a sequence where $\tilde{O}_k$ is a k-partially reflective k-partial oracle and $\tilde{O}_{k+1}$ extends $\tilde{O}_k$ for all $k \in \mathbb{N}$. Let $O := \lim_{k\to\infty} \tilde{O}_k$ be the pointwise limit. Then*

(a) $\lambda_T^{\tilde{O}_k}(1 \mid x) \to \lambda_T^{O}(1 \mid x)$ and $\lambda_T^{\tilde{O}_k}(0 \mid x) \to \lambda_T^{O}(0 \mid x)$ as $k \to \infty$ for all $x \in \{0,1\}^*$ and all $T \in \mathcal{T}$, and

(b) O is a reflective oracle.

Proof. First note that the pointwise limit must exists because $|\tilde{O}_k(q_i) - \tilde{O}_{k+1}(q_i)| \leq 2^{-k-1}$ by Definition 7.14.

(a) Since $\tilde{O}_{k+1}$ extends $\tilde{O}_k$, each $\tilde{O}_k$ approximates O. Let $x \in \{0,1\}^*$ and $T \in \mathcal{T}$ and consider the sequence $a_k := \lambda_T^{\tilde{O}_k}(1 \mid x)$ for $k \in \mathbb{N}$. By Lemma 7.16, $a_k \leq a_{k+1}$, so the sequence is monotone increasing. By Lemma 7.11, $a_k \leq \lambda_T^O(1 \mid x)$, so the sequence is bounded. Therefore it must converge. But it cannot converge to anything strictly below $\lambda_T^O(1 \mid x)$ by the definition of T^O.

(b) By definition, O is an oracle; it remains to show that O is reflective. Let $q_i = (T, x, p)$ be some query. If $p < \lambda_T^O(1 \mid x)$, then by (a) there is a k large enough such that $p < \lambda_T^{\tilde{O}_t}(1 \mid x)$ for all $t \geq k$. For any $t \geq \max\{k, i\}$, we have $\tilde{O}_t(T, x, p) = 1$ since $\tilde{O}_t$ is t-partially reflective. Therefore $1 = \lim_{k\to\infty} \tilde{O}_k(T, x, p) = O(T, x, p)$. The case $1 - p < \lambda_T^O(0 \mid x)$ is analogous. □

7.2 A Grain of Truth

7.2.1 Reflective Bayesian Agents

Fix O to be a reflective oracle. From now on, we assume that the action space $\mathcal{A} := \{\alpha, \beta\}$ is binary. We can treat computable measures over binary strings as environments: the environment ν corresponding to a probabilistic Turing machine $T \in \mathcal{T}$ is defined by

$$\nu(e_t \mid æ_{<t}a_t) := \overline{\lambda}_T^O(y \mid x) = \prod_{i=1}^{k} \overline{\lambda}_T^O(y_i \mid xy_1 \dots y_{i-1})$$

where $y_{1:k}$ is a binary encoding of e_t and x is a binary encoding of $æ_{<t}a_t$. The actions $a_{1:\infty}$ are only *contextual*, and not part of the environment distribution. We define $\nu(e_{<t} \parallel a_{<t})$ analogously to (4.1).

Let $T_1, T_2, \dots$ be an enumeration of all probabilistic Turing machines in $\mathcal{T}$ that use an oracle. We define the *class of reflective environments*

$$\mathcal{M}_{\text{refl}}^O := \left\{\overline{\lambda}_{T_1}^O, \overline{\lambda}_{T_2}^O, \dots\right\}.$$

This is the class of all environments computable on a probabilistic Turing machine with reflective oracle O, that have been completed from semimeasures to measures using O.

Analogously to Section 4.3.1, we define a Bayesian mixture over the class $\mathcal{M}_{\text{refl}}^O$. Let $w \in \Delta\mathcal{M}_{\text{refl}}^O$ be a lower semicomputable prior probability distribution on $\mathcal{M}_{\text{refl}}^O$. Possible choices for the prior include the *Solomonoff prior* $w\big(\overline{\lambda}_T^O\big) := 2^{-K(T)}$, where $K(T)$ denotes the length of the shortest input to some universal Turing machine that encodes T. We define the corresponding Bayesian mixture

$$\xi(e_t \mid æ_{<t}a_t) := \sum_{\nu \in \mathcal{M}_{\text{refl}}^O} w(\nu \mid æ_{<t})\nu(e_t \mid æ_{<t}a_t) \tag{7.2}$$

where $w(\nu \mid æ_{<t})$ is the (renomalized) posterior,

$$w(\nu \mid æ_{<t}) := w(\nu)\frac{\nu(e_{<t} \parallel a_{<t})}{\overline{\xi}(e_{<t} \parallel a_{<t})}. \tag{7.3}$$

The mixture ξ is lower semicomputable on an oracle Turing machine because the posterior $w(\cdot \mid æ_{<t})$ is lower semicomputable. Hence there is an oracle machine T such that $\xi = \lambda_T^O$. We define its completion $\overline{\xi} := \overline{\lambda}_T^O$ as the completion of λ_T^O. This is the distribution that is used to compute the posterior. There are no cyclic dependencies since $\overline{\xi}$ is called on the shorter history $æ_{<t}$. We arrive at the following statement.

Proposition 7.18 (Bayes is in the Class)**.** $\overline{\xi} \in \mathcal{M}_{\text{refl}}^O$.

Moreover, since O is reflective, we have that $\overline{\xi}$ dominates all environments $\nu \in \mathcal{M}_{\text{refl}}^O$:

$$\begin{aligned}
\overline{\xi}(e_{1:t} \parallel a_{1:t}) &= \overline{\xi}(e_t \mid æ_{<t}a_t)\overline{\xi}(e_{<t} \parallel a_{<t}) \\
&\geq \xi(e_t \mid æ_{<t}a_t)\overline{\xi}(e_{<t} \mid a_{<t}) \\
&= \overline{\xi}(e_{<t} \parallel a_{<t}) \sum_{\nu \in \mathcal{M}_{\text{refl}}^O} w(\nu \mid æ_{<t})\nu(e_t \mid æ_{<t}a_t) \\
&= \overline{\xi}(e_{<t} \parallel a_{<t}) \sum_{\nu \in \mathcal{M}_{\text{refl}}^O} w(\nu)\frac{\nu(e_{<t} \parallel a_{<t})}{\overline{\xi}(e_{<t} \parallel a_{<t})}\nu(e_t \mid æ_{<t}a_t) \\
&= \sum_{\nu \in \mathcal{M}_{\text{refl}}^O} w(\nu)\nu(e_{1:t} \parallel a_{1:t}) \\
&\geq w(\nu)\nu(e_{1:t} \parallel a_{1:t})
\end{aligned}$$

Therefore we get on-policy value convergence according to Corollary 4.20: for all $\mu \in \mathcal{M}_{\text{refl}}^O$ and all policies π

$$V_{\overline{\xi}}^{\pi}(æ_{<t}) - V_{\mu}^{\pi}(æ_{<t}) \to 0 \text{ as } t \to \infty \ \mu^{\pi}\text{-almost surely.} \tag{7.4}$$

7.2.2 Reflective-Oracle-Computable Policies

This subsection is dedicated to the following result that was previously stated by Fallenstein et al. (2015a, Alg. 6) but not proved. It contrasts results on arbitrary semicomputable environments where optimal policies are not limit computable (see Section 6.3).

Theorem 7.19 (Optimal Policies are Oracle Computable)**.** *For every $\nu \in \mathcal{M}_{\text{refl}}^O$, there is a ν-optimal (stochastic) policy π_ν^* that is reflective-oracle-computable.*

Note that even though deterministic optimal policies always exist, those policies are typically not reflective-oracle-computable.

To prove Theorem 7.19 we need the following lemma.

Lemma 7.20 (Reflective-Oracle-Computable Optimal Value Function)**.** *For every environment $\nu \in \mathcal{M}_{\text{refl}}^O$ the optimal value function V_ν^* is reflective-oracle-computable.*

Proof. This proof follows the proof of Corollary 6.14. We write the optimal value explicitly as in (4.2). For a fixed m, all involved quantities are reflective-oracle-computable. Moreover, this quantity is monotone increasing in m and the tail sum from $m+1$ to ∞ is bounded by Γ_{m+1} which is computable according to Assumption 4.6a and converges to 0 as $m \to \infty$. Therefore we can enumerate all rationals above and below V^*_ν. □

Proof of Theorem 7.19. According to Lemma 7.20 the optimal value function V^*_ν is reflective-oracle-computable. Hence there is a probabilistic Turing machine T such that

$$\lambda^O_T(1 \mid æ_{<t}) = \big(V^*_\nu(æ_{<t}\alpha) - V^*_\nu(æ_{<t}\beta) + 1\big)/2.$$

We define the policy

$$\pi(æ_{<t}) := \begin{cases} \alpha & \text{if } O(T, æ_{<t}, 1/2) = 1, \text{ and} \\ \beta & \text{if } O(T, æ_{<t}, 1/2) = 0 \end{cases}$$

This policy is stochastic because the answer of the oracle O is stochastic.

It remains to show that π is a ν-optimal policy. If $V^*_\nu(æ_{<t}\alpha) > V^*_\nu(æ_{<t}\beta)$, then $\lambda^O_T(1 \mid æ_{<t}) > 1/2$, thus $O(T, æ_{<t}, 1/2) = 1$ since O is reflective, and hence π takes action α. Conversely, if $V^*_\nu(æ_{<t}\alpha) < V^*_\nu(æ_{<t}\beta)$, then $\lambda^O_T(1 \mid æ_{<t}) < 1/2$, thus $O(T, æ_{<t}, 1/2) = 0$ since O is reflective, and hence π takes action β. Lastly, if $V^*_\nu(æ_{<t}\alpha) = V^*_\nu(æ_{<t}\beta)$, then both actions are optimal and thus it does not matter which action is returned by policy π. (This is the case where the oracle may randomize.) □

7.2.3 Solution to the Grain of Truth Problem

Together, Proposition 7.18 and Theorem 7.19 provide the necessary ingredients to solve the grain of truth problem (Problem 7.1).

Corollary 7.21 (Solution to the Grain of Truth Problem)**.** *For every lower semicomputable prior $w \in \Delta\mathcal{M}^O_{\text{refl}}$ the Bayes optimal policy π^*_ξ is reflective-oracle-computable where ξ is the Bayes-mixture corresponding to w defined in (7.2).*

Proof. From Proposition 7.18 and Theorem 7.19. □

Hence the environment class $\mathcal{M}^O_{\text{refl}}$ contains any reflective-oracle-computable modification of the Bayes optimal policy π^*_ξ. In particular, this includes computable multi-agent environments that contain other Bayesian agents over the class $\mathcal{M}^O_{\text{refl}}$. So any Bayesian agent over the class $\mathcal{M}^O_{\text{refl}}$ has a grain of truth even though the environment may contain other Bayesian agents *of equal power*. We proceed to sketch the implications for multi-agent environments in the next section.

7.3 Multi-Agent Environments

In a *multi-agent environment* there are n agents each taking sequential actions from the finite action space $\mathcal{A}$. In each time step $t = 1, 2, \ldots$, the environment receives action a^i_t

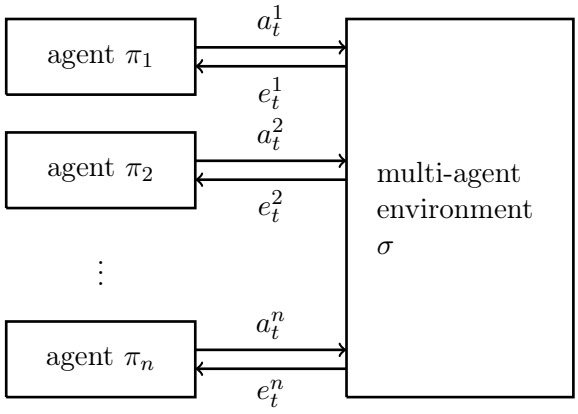


Figure 7.2: Agents $\pi_1, \ldots, \pi_n$ interacting in a multi-agent environment.

from agent i and outputs n percepts $e_t^1, \ldots, e_t^n \in \mathcal{E}$, one for each agent. Each percept $e_t^i = (o_t^i, r_t^i)$ contains an observation o_t^i and a reward $r_t^i \in [0, 1]$. Importantly, agent i only sees its own action a_t^i and its own percept e_t^i (see Figure 7.2). We use the shorthand notation $a_t := (a_t^1, \ldots, a_t^n)$ and $e_t := (e_t^1, \ldots, e_t^n)$ and denote $æ_{<t}^i = a_1^i e_1^i \ldots a_{t-1}^i e_{t-1}^i$ and $æ_{<t} = a_1 e_1 \ldots a_{t-1} e_{t-1}$. Formally, multi-agent environments are defined as follows.

Definition 7.22 (Multi-Agent Environment)**.** A *multi-agent environment* is a function

$$\sigma : (\mathcal{A}^n \times \mathcal{E}^n)^* \times \mathcal{A}^n \to \Delta(\mathcal{E}^n).$$

Together with the policies $\pi_1, \ldots, \pi_n$ the multi-agent environment σ induces a *history distribution* $\sigma^{\pi_{1:n}}$ where

$$\begin{aligned} \sigma^{\pi_{1:n}}(\epsilon) &:= 1 \\ \sigma^{\pi_{1:n}}(æ_{1:t}) &:= \sigma^{\pi_{1:n}}(æ_{<t} a_t) \sigma(e_t \mid æ_{<t} a_t) \\ \sigma^{\pi_{1:n}}(æ_{<t} a_t) &:= \sigma^{\pi_{1:n}}(æ_{<t}) \prod_{i=1}^{n} \pi_i(a_t^i \mid æ_{<t}^i). \end{aligned}$$

Agent i acts in a *subjective environment* σ_i given by joining the multi-agent environment σ with the policies $\pi_1, \ldots, \pi_n$ and marginalizing over the histories that π_i does not see. Together with policy π_i, the environment σ_i yields a distribution over the histories of agent i

$$\sigma_i^{\pi_i}(æ_{<t}^i) := \sum_{æ_{<t}^j, j \neq i} \sigma^{\pi_{1:n}}(æ_{<t}).$$

We get the definition of the subjective environment σ_i with the identity $\sigma_i(e_t^i \mid æ_{<t}^i a_t^i) := \sigma_i^{\pi_i}(e_t^i \mid æ_{<t}^i a_t^i)$. The subjective environment σ_i depends on π_i because other policies' actions may depend on the actions of π_i. It is crucial to note that the subjective environment σ_i and the policy π_i are ordinary environments and policies, so we can use the notation from Chapter 4.

Our definition of a multi-agent environment is very general and encompasses most of game theory. It allows for cooperative, competitive, and mixed games; infinitely repeated games or any (infinite-length) extensive form games with finitely many players.

Example 7.23 (Matching Pennies)**.** In the game of *matching pennies* there are two agents ($n = 2$), and two actions $\mathcal{A} = \{\alpha, \beta\}$ representing the two sides of a penny. In each time step agent 1 wins if the two actions are identical and agent 2 wins if the two actions are different. The payoff matrix is as follows.

	α	β
α	1,0	0,1
β	0,1	1,0

We use $\mathcal{E} = \{0, 1\}$ to be the set of rewards (observations are vacuous) and define the multi-agent environment σ to give reward 1 to agent 1 iff $a_t^1 = a_t^2$ (0 otherwise) and reward 1 to agent 2 iff $a_t^1 \neq a_t^2$ (0 otherwise). Formally,

$$\sigma(r_t^1 r_t^2 \mid æ_{<t} a_t) := \begin{cases} 1 & \text{if } r_t^1 = 1, r_t^2 = 0, a_t^1 = a_t^2, \\ 1 & \text{if } r_t^1 = 0, r_t^2 = 1, a_t^1 \neq a_t^2, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Let π_α denote the policy that always takes action α. If two agents each using policy π_α play matching pennies, agent 1 wins in every step. Formally, setting $\pi_1 := \pi_2 := \pi_\alpha$ we get a history distribution that assigns probability one to the history

$$\alpha\alpha 10 \alpha\alpha 10 \ldots.$$

The subjective environment of agent 1 is

$$\sigma_1(r_t^1 \mid æ_{<t}^1 a_t^1) = \begin{cases} 1 & \text{if } r_t^1 = 1, a_t^1 = \alpha, \\ 1 & \text{if } r_t^1 = 0, a_t^1 = \beta, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Therefore policy π_α is optimal in agent 1's subjective environment. ◇

Definition 7.24 (ε-Best Response)**.** A policy π_i acting in multi-agent environment σ with policies $\pi_1, \ldots, \pi_n$ is an *ε-best response* after history $æ_{<t}^i$ iff

$$V_{\sigma_i}^*(æ_{<t}^i) - V_{\sigma_i}^{\pi_i}(æ_{<t}^i) < \varepsilon.$$

If at some time step t, all agents' policies are ε-best responses, we have an *ε-Nash equilibrium*. The property of multi-agent systems that is analogous to asymptotic optimality is convergence to an ε-Nash equilibrium.

7.4 Informed Reflective Agents

Let σ be a multi-agent environment and let $\pi^*_{\sigma_1}, \dots \pi^*_{\sigma_n}$ be such that for each i the policy $\pi^*_{\sigma_i}$ is an optimal policy in agent i's subjective environment σ_i. At first glance this seems ill-defined: The subjective environment σ_i depends on each policy $\pi^*_{\sigma_j}$, which depends on the subjective environment σ_j, which in turn depends on the policy $\pi^*_{\sigma_i}$. However, this circular definition actually has a well-defined solution.

Theorem 7.25 (Optimal Multi-Agent Policies)**.** *For any reflective-oracle-computable multi-agent environment σ, the optimal policies $\pi^*_{\sigma_1}, \dots, \pi^*_{\sigma_n}$ exist and are reflective-oracle-computable.*

To prove Theorem 7.25, we need the following proposition.

Proposition 7.26 (Reflective-Oracle-Computability)**.** *If the multi-agent environment σ and the policies $\pi_1, \dots, \pi_n$ are reflective-oracle-computable, then $\sigma^{\pi_{1:n}}$ and $\sigma_i^{\pi_i}$ are reflective-oracle-computable, and $\sigma_i \in \mathcal{M}^O_{\text{refl}}$.*

Proof. All involved quantities in the definition of $\sigma^{\pi_{1:n}}$ are reflective-oracle-computable by assumption, therefore also their marginalizations $\sigma_i^{\pi_i}$ and σ_i. □

Proof of Theorem 7.25. According to Theorem 7.19 the optimal policy $\pi^*_{\sigma_i}$ in agent i's subjective environment is reflective-oracle-computable if the subjective environment σ_i is. In particular the process that takes σ_i in form of a probabilistic Turing machine and returns $\pi^*_{\sigma_i}$ is reflective-oracle-computable. Moreover, σ_i is reflective-oracle-computable if σ and $\pi_1, \dots, \pi_n$ are, according to Proposition 7.26. Again, this construction is itself reflective-oracle-computable. Connecting these two constructions we get probabilistic Turing machines $T_1, \dots, T_n \in \mathcal{T}$ where each T_i takes the multi-agent environment σ and $\pi_1, \dots, \pi_n$ in form of probabilistic Turing machines and returns $\pi^*_{\sigma_i}$. We define the probabilistic Turing machines $T'_1, \dots, T'_n$ where T'_i runs T_i^O on $(\sigma, T'_1, \dots, T'_n)$; hence T'_i computes $\pi^*_{\sigma_i}$. Note that this construction only works because we relied on the reflective oracle in the proof of Theorem 7.19. Since the machines T_i^O always halt, so do $T_i'^O$ despite their infinitely recursive construction. □

Note the strength of Theorem 7.25: each of the policies $\pi^*_{\sigma_i}$ is acting optimally *given the knowledge of everyone else's policies*. Hence optimal policies play 0-best responses by definition, so if every agent is playing an optimal policy, we have a Nash equilibrium. Moreover, this Nash equilibrium is also a *subgame perfect* Nash equilibrium, because each agent also acts optimally on the counterfactual histories that do not end up being played. In other words, Theorem 7.25 states the existence and reflective-oracle-computability of a subgame perfect Nash equilibrium in any reflective-oracle-computable multi-agent environment. The following immediate corollary states that these subgame perfect Nash equilibria are limit computable.

Corollary 7.27 (Solution to Computable Multi-Agent Environments)**.** *For any computable multi-agent environment σ, the optimal policies $\pi^*_{\sigma_1}, \dots, \pi^*_{\sigma_n}$ exist and are limit computable.*

Proof. From Theorem 7.25 and Theorem 7.7. □

Example 7.28 (Nash Equilibrium in Matching Pennies)**.** Consider the matching pennies game from Example 7.23. The only pair of optimal policies is the pair of two uniformly random policies that play α and β with equal probability in every time step: if one of the agents picks a policy that plays one of the actions with probability $> 1/2$, then the other agent's best response is to play the other action with probability 1. But now the first agent's policy is no longer a best response. ◇

7.5 Learning Reflective Agents

Since our class $\mathcal{M}^{O}_{\text{refl}}$ solves the grain of truth problem, the result by Kalai and Lehrer (1993) immediately implies that for any Bayesian agents $\pi_1, \dots, \pi_n$ interacting in an infinitely repeated game and for all $\varepsilon > 0$ and all $i \in \{1, \dots, n\}$ there is almost surely a $t_0 \in \mathbb{N}$ such that for all $t \geq t_0$ the policy π_i is an ε-best response. However, this hinges on the important fact that every agent has to know the game and also that all other agents are Bayesian agents. Otherwise the convergence to an ε-Nash equilibrium may fail, as illustrated by the following example.

At the core of the construction is a *dogmatic prior* (Section 5.2.2). A dogmatic prior assigns very high probability to going to hell (reward 0 forever) if the agent deviates from a given computable policy π. For a Bayesian agent it is thus only worth deviating from the policy π if the agent thinks that the prospects of following π are very poor already. This implies that for general multi-agent environments and without additional assumptions on the prior, we cannot prove any meaningful convergence result about Bayesian agents acting in an unknown multi-agent environment.

Example 7.29 (Reflective Bayesians Playing Matching Pennies)**.** Consider the multi-agent environment matching pennies from Example 7.23. Let π_1 be the policy that takes the action sequence $(\alpha\alpha\beta)^\infty$ and let $\pi_2 := \pi_\alpha$ be the policy that always takes action α. The average reward of policy π_1 is $2/3$ and the average reward of policy π_2 is $1/3$. Let ξ be a universal mixture (7.2). By on-policy value convergence (7.4), $V^{\pi_1}_{\xi} \to c_1 \approx 2/3$ and $V^{\pi_2}_{\xi} \to c_2 \approx 1/3$ almost surely when following policies (π_1, π_2). Therefore there is an $\varepsilon > 0$ such that $V^{\pi_1}_{\xi} > \varepsilon$ and $V^{\pi_2}_{\xi} > \varepsilon$ for all time steps. Now we can apply Theorem 5.5 to conclude that there are (dogmatic) mixtures ξ'_1 and ξ'_2 such that $\pi^*_{\xi'_1}$ always follows policy π_1 and $\pi^*_{\xi'_2}$ always follows policy π_2. This does not converge to a (ε-)Nash equilibrium. ◇

An important property required for the construction in Example 7.29 is that the environment class contains environments that threaten the agent with going to hell, which is outside of the class of matching pennies environments. In other words, since the agent does not know a priori that it is playing a matching pennies game, it might behave more conservatively than appropriate for the game.

The following theorem is our main convergence result. It states that for asymptotically optimal agents we get convergence to ε-Nash equilibria in any reflective-oracle-computable multi-agent environment.

Theorem 7.30 (Convergence to Equilibrium)**.** *Let σ be an reflective-oracle-computable multi-agent environment and let $\pi_1, \dots, \pi_n$ be reflective-oracle-computable policies that are asymptotically optimal in mean in the class $\mathcal{M}^O_{\text{refl}}$. Then for all $\varepsilon > 0$ and all $i \in \{1, \dots, n\}$ the $\sigma^{\pi_{1:n}}$-probability that the policy π_i is an ε-best response converges to 1 as $t \to \infty$.*

Proof. Let $i \in \{1, \dots, n\}$. By Proposition 7.26, the subjective environment σ_i is reflective-oracle-computable, therefore $\sigma_i \in \mathcal{M}^O_{\text{refl}}$. Since π_i is asymptotically optimal in mean in the class $\mathcal{M}^O_{\text{refl}}$, we get that $\mathbb{E}[V^*_{\sigma_i}(æ_{<t}) - V^{\pi_i}_{\sigma_i}(æ_{<t})] \to 0$. Convergence in mean implies convergence in probability for bounded random variables, hence for all $\varepsilon > 0$ we have

$$\sigma_i^{\pi_i}[V^*_{\sigma_i}(æ^i_{<t}) - V^{\pi_i}_{\sigma_i}(æ^i_{<t}) \geq \varepsilon] \to 0 \text{ as } t \to \infty.$$

Therefore the probability that the policy π_i plays an ε-best response converges to 1 as $t \to \infty$. □

In contrast to Theorem 7.25 which yields policies that play a subgame perfect equilibrium, this is not the case for Theorem 7.30: the agents typically do not learn to predict off-policy and thus will generally not play ε-best responses in the counterfactual histories that they never see. This weaker form of equilibrium is unavoidable if the agents do not know the environment because it is impossible to learn the parts that they do not interact with.

Corollary 7.31 (Convergence to Equilibrium)**.** *There are limit computable policies $\pi_1, \dots, \pi_n$ such that for any computable multi-agent environment σ and for all $\varepsilon > 0$ and all $i \in \{1, \dots, n\}$ the $\sigma^{\pi_{1:n}}$-probability that the policy π_i is an ε-best response converges to 1 as $t \to \infty$.*

Proof. Pick $\pi_1, \dots, \pi_n$ to be the Thompson sampling policy π_T defined in Algorithm 2 over the countable class $\mathcal{M}^O_{\text{refl}}$. By Theorem 5.25 these policies are asymptotically optimal in mean. By Theorem 7.32 below they are reflective-oracle-computable and by Theorem 7.7 they are also limit computable. The statement now follows from Theorem 7.30. □

Theorem 7.32 (Thompson Sampling is Reflective-Oracle-Computable)**.** *The policy π_T defined in Algorithm 2 over the class $\mathcal{M}^O_{\text{refl}}$ is reflective-oracle-computable.*

Proof. The posterior $w(\cdot \mid æ_{<t})$ is reflective-oracle-computable by the definition (7.3) and according to Theorem 7.19 the optimal policies π^*_ν are reflective-oracle-computable. On resampling steps we can compute the action probabilities of π_T by enumerating all $\nu \in \mathcal{M}^O_{\text{refl}}$ and computing π^*_ν weighted by the posterior $w(\nu \mid æ_{<t})$. Between resampling steps we need to condition the policy π_T computed above by the actions it has already taken since the last resampling step (compare Example 5.28). □

Because the posterior $w(\cdot \mid æ_{<t})$ is a $\overline{\xi}^\pi$-martingale when acting according to the policy π, it converges $\overline{\xi}^\pi$-almost surely according to the martingale convergence theorem (Theorem 2.8). Since $\overline{\xi}$ dominates the subjective environment σ_i, it also converges

σ_i^π-almost surely (see Example 3.20). Hence all the Thompson sampling agents eventually 'calm down' and settle on some posterior belief.

According to Theorem 7.30, the policies $\pi_1, \dots, \pi_n$ only need to be asymptotically optimal in mean. For Thompson sampling this is independent of the discount function according to Theorem 5.25 (but the discount function has to be known to the agent). So the different agents may use different discount functions, resample at different time steps and converge at different speeds.

Example 7.33 (Thompson Samplers Playing Matching Pennies)**.** Consider the matching pennies game from Example 7.23 and let both agents use the Thompson sampling policy defined in Algorithm 2, i.e., define $\pi_1 := \pi_T$ and $\pi_2 := \pi_T$.

The value of the uniformly random policy π_R is always $1/2$, so $V^*_{\sigma_i} \geq V^{\pi_R}_{\sigma_i} = 1/2$. According to Theorem 7.30, for every $\varepsilon > 0$, each agent will eventually always play an ε-best response, i.e., $V^{\pi_i}_{\sigma_i} > V^*_{\sigma_i} - \varepsilon \geq 1/2 - \varepsilon$. Since matching pennies is a zero-sum game, $V^{\pi_1}_{\sigma_1} + V^{\pi_2}_{\sigma_2} = 1$, so $V^{\pi_2}_{\sigma_2} = 1 - V^{\pi_1}_{\sigma_1} < 1/2 + \varepsilon$.

Therefore each agent will end up randomizing their actions; $\pi_i(a_t \mid æ_{<t}) \approx 1/2$ most of the time: If one of the agents (say agent 1) does not sufficiently randomize their actions in some time steps, then agent 2 could exploit this by picking a deterministic adversarial policy in those time steps. Suppose that this way it can gain a value of ε compared to the random policy π_R, i.e., $V^{\pi_2}_{\sigma_2} \geq V^{\pi_R}_{\sigma_2} + \varepsilon = 1/2 + \varepsilon$. But this is a contradiction because then agent 1 is not playing an ε-best response:

$$V^*_{\sigma_1} - V^{\pi_1}_{\sigma_1} \;=\; V^*_{\sigma_1} - 1 + V^{\pi_2}_{\sigma_2} \;\geq\; 1/2 - 1 + 1/2 + \varepsilon \;=\; \varepsilon \qquad \diamond$$

7.6 Impossibility Results

Why does our solution to the grain of truth problem not violate the impossibility results from the literature? Assume we are playing an infinitely repeated game where in the stage game no agent has a weakly dominant action and the pure action maxmin reward is strictly less then the minmax reward. The impossibility result of Nachbar (1997, 2005) states that there is no class of policies Π such that the following are simultaneously satisfied.

- *Learnability.* Each agent learns to predict the other agent's actions.
- *Caution and Symmetry.* The set Π is closed under simple policy modifications such as renaming actions.
- *Purity.* There is an $\varepsilon > 0$ such that for any stochastic policy $\pi \in \Pi$ there is a deterministic policy $\pi' \in \Pi$ such that if $\pi'(æ_{<t}) = a$, then $\pi(a \mid æ_{<t}) > \varepsilon$.
- *Consistency.* Each agent always has an ε-best response available in Π.

In order to converge to an ε-Nash equilibrium, each agent has to have an ε-best response available to them, so consistency is our target. Learnability is immediately satisfied for any environment in our class if we have a dominant prior according to Corollary 4.20.

For $\mathcal{M}^O_{\text{refl}}$ caution and symmetry are also satisfied since this set is closed under any computable modifications to policies. However, our class $\mathcal{M}^O_{\text{refl}}$ avoids this impossibility result because it violates the purity condition: Let $T_1, T_2, \ldots$ be an enumeration of $\mathcal{T}$. Consider the policy π that maps history $æ^i_{<t}$ to the action $1 - O(T_t, æ^i_{<t}, 1/2)$. If T_t is deterministic, then π will take a different action than T_t for any history of length $t-1$. Therefore no deterministic reflective-oracle-computable policy can take an action that π assigns positive probability to in every time step.

Foster and Young (2001) present a condition that makes convergence to a Nash equilibrium impossible: if the player's rewards are perturbed by a small real number drawn from some continuous density ν, then for ν-almost all realizations the players do not learn to predict each other and do not converge to a Nash equilibrium. For example, in a matching pennies game, rational agents randomize only if the (subjective) values of both actions are exactly equal. But this happens only with ν-probability zero, since ν is a density. Thus with ν-probability one the agents do not randomize. If the agents do not randomize, they either fail to learn to predict each other, or they are not acting rationally according to their beliefs: otherwise they would seize the opportunity to exploit the other player's deterministic action.

But this does not contradict our convergence result: the class $\mathcal{M}^O_{\text{refl}}$ is countable and each $\nu \in \mathcal{M}^O_{\text{refl}}$ has positive prior probability. Perturbation of rewards with arbitrary real numbers is not possible. Even more, the argument given by Foster and Young (2001) cannot work in our setting: the Bayesian mixture $\overline{\xi}$ mixes over λ_T for all probabilistic Turing machines T. For Turing machines T that sometimes do not halt, the oracle decides how to complete λ_T into a measure $\overline{\lambda}_T$. Thus the oracle has enough influence on the exact values in the Bayesian mixture that the values of two actions in matching pennies can be made exactly equal.

7.7 Discussion

This chapter introduced the class of all reflective-oracle-computable environments $\mathcal{M}^O_{\text{refl}}$. This class fully solves the grain of truth problem (Problem 7.1) because it contains (any computable modification of) Bayesian agents defined over $\mathcal{M}^O_{\text{refl}}$: the optimal agents and Bayes optimal agents over the class are all reflective-oracle-computable (Theorem 7.19 and Corollary 7.21).

If the environment is unknown, then a Bayesian agent may end up playing suboptimally (Example 7.29). However, if each agent uses a policy that is asymptotically optimal in mean (such as the Thompson sampling policy from Section 4.3.4) then for every $\varepsilon > 0$ the agents converge to an ε-Nash equilibrium (Theorem 7.30 and Corollary 7.31).

However, Corollary 7.31 does *not* imply that two Thompson sampling policies will converge to cooperation in an iterated prisoner's dilemma since always defecting is also a Nash equilibrium. The exact outcome will depend on the priors involved and the randomness of the policies.

Our solution to the grain of truth problem is purely theoretical. However, Theorem 7.7 shows that our class $\mathcal{M}^O_{\text{refl}}$ allows for computable approximations. This suggests

that practical approaches can be derived from this result, and reflective oracles have already seen applications in one-shot games (Fallenstein et al., 2015b).

Chapter 8

Conclusion

The biggest existential risk is that future superintelligences stop simulating us.
— Nick Bostrom

Today computer programs exceeding humans in general intelligence are known only from science fiction. But research on AI has progressed steadily over the last decades and there is good reason to believe that we will be able to build HLAI eventually, and even sooner than most people think (Müller and Bostrom, 2016).

The advent of strong AI would be the biggest event in human history. Potential benefits are huge, as the new level of automation would free us from any kind of undesirable labor. But there is no reason to believe that humans are at the far end of the intelligence spectrum. Rather, humans barely cross the threshold for general intelligence to be able to use language and do science. Once we engineer HLAI, it seems unlikely that progress is going to stop; why not build even smarter machines?

This could lead to an *intelligence explosion* (Good, 1965; Vinge, 1993; Kurzweil, 2005; Chalmers, 2010; Hutter, 2012a; Schmidhuber, 2012; Muehlhauser and Salamon, 2012; Eden et al., 2013; Shanahan, 2015; Eden, 2016; Walsh, 2016): a (possibly very rapid) increase in intelligence, e.g., through self-amplification effects from AIs improving themselves (if doing AI research is one of humans' capabilities, then a machine that can do everything humans can do can also do AI research). Once machine intelligence is above or far above human level, machines would steer the course of history. There is no reason to believe that machines would be adversarial to us, but nevertheless humanity's fate might rest at the whims of the machines, just as chimpanzees today have no say in the large-scale events on this planet.

Bostrom (2002) defines:

> *Existential risk* — One where an adverse outcome would either annihilate Earth-originating intelligent life or permanently and drastically curtail its potential.

Existential risks are events that have the power to extinguish human life as we know it. Examples are cosmic events such as an asteroid colliding with Earth. But cosmic events are unlikely on human timescales compared to human-made existential risks from nuclear weapons, synthetic biology, and nanotechnology.

It is possible that artificial intelligence also falls into this category. Vinge (1993) was the first person to recognize this:

> Within thirty years, we will have the technological means to create superhuman intelligence. Shortly after, the human era will be ended.

After Vinge, Yudkowsky (2001, 2008) can be regarded as one of the key people to popularize the potential dangers of AI technology. Bostrom (2003) picked up on this issue very early and gave the topic credibility through his well-researched and carefully written book *Superintelligence* (Bostrom, 2014). He argues that we need to solve the *Control Problem*—the unique principal-agent problem that arises with the creation of strong AI (Bostrom, 2014, Ch. 9). In other words: How do we align strong AI with human values? How do we ensure that AI remains robust and beneficial? This research is collected under the umbrella term *AI safety*. Currently, we have no idea how to solve these problems even in theory.

Through Yudkowsky's and, more importantly, Bostrom's efforts, AI-related long-term safety concerns have now entered the mainstream media. In 2014 high-profile scientists such as Stephen Hawking, Max Tegmark, Stuart Russell, and Frank Wilczek have warned against the dangers posed by AI (Hawking et al., 2014). (See also Alexander (2015) for a collection of positions by prominent AI researchers.) Many scientists inside and outside the field have signed an open letter that research ensuring that AI systems remain robust and beneficial is both important and timely (Future of Life Institute, 2015c). This lead entrepreneur Elon Musk to donate $10 million to kick-start research in this field (Future of Life Institute, 2015a); most of this money has now been distributed across the planet to 37 different projects (Future of Life Institute, 2015b). Moreover, the Future of Life Institute and the Machine Intelligence Research Institute have formulated concrete technical research priorities to make AI more robust and beneficial (Soares and Fallenstein, 2014; Russell et al., 2015).

At the end of 2015 followed the announcement of OpenAI, a nonprofit organization with financial backing from several famous silicon valley billionaires (OpenAI, 2015):

> OpenAI is a non-profit artificial intelligence research company. Our goal is to advance digital intelligence in the way that is most likely to benefit humanity as a whole, unconstrained by a need to generate financial return.

The mission of OpenAI is to enable everyone to benefit from AI technology. But, despite the name, OpenAI is not committed to publish all of their research freely, and Bostrom (2016) argues that unrestricted publication might not be the best idea.

Despite all of the recent efforts in AI safety research, critical voices within the AI community remain. Prominently, Davis (2014), Ng (2016), Walsh (2016), and Lawrence (2016) have proposed counterarguments that range from 'HLAI is so far away that any worry is misplaced' to claims that 'the safety problem would not be so hard'. See Sotala and Yampolskiy (2014) for a discussion.

If AI poses an existential risk then a formal theory of strong AI is paramount to develop technical approaches to mitigate this risk. Which path will ultimately lead us to HLAI is in the realm of speculation at this time; therefore we should make as few and as weak assumptions as possible and abstract away from possible future implementation details.

This thesis lays some of the groundwork for this endeavor. We built on top of Hutter's theory of universal artificial intelligence. Chapter 3 discussed the formal theory of learning. Chapter 4 presented several approaches to acting in unknown environments (Bayes, Thompson sampling, knowledge-seeking agents, and BayesExp). Chapter 5 analysed these approaches and discussed notions of optimality and principled problems with acting under uncertainty in general environment. Chapter 6 provided the mathematical tools to analyze the computational properties of these models. Finally, Chapter 7 solved the grain of truth problem, which lead to convergence to Nash equilibria in unknown general multi-agent environments.

Our work is theoretical by nature and there is still a long way to go until these results make their way into applications. But a solution *in principle* is a crucial first step towards solving a problem in practice. Consider the research paper by Shannon (1950) on how to solve chess in principle. The algorithm he describes expands the full game tree of chess (until some specified depth), which is completely infeasible even with today's computation power. His contribution was to show that winning at chess is a feat that computers can achieve *in principle*, which was not universally accepted at the time. Even more, his approach already considered the correct ideas (minimax-search over the game tree) that ultimately lead to the defeat of the chess champion Garry Kasparov by the computer Deep Blue 46 years later (IBM, 2012a).

The theory of general reinforcement learning can serve and has served as a starting point for future investigation in AI safety. In particular, value learning (Dewey, 2011), self-reflection (Soares, 2015; Fallenstein et al., 2015b), self-modification (Orseau and Ring, 2011, 2012a; Everitt et al., 2016), interruptibility (Orseau and Armstrong, 2016; Armstrong and Orseau, 2016), decision theory (Everitt et al., 2015), memory manipulation (Orseau and Ring, 2012b), wireheading (Ring and Orseau, 2011; Everitt and Hutter, 2016), and questions of identity (Orseau, 2014b,c).

It is possible that HLAI is decades or centuries away. It might also be a few years around the corner. Whichever is the case, we are currently completely unprepared for the consequences. As an AI researcher, it is tempting to devote your life to increasing the capability of AI, advancing it domain after domain, and showing off with flashy demos and impressive victories over human contestants. But every technology incurs risks, and the more powerful the technology, the higher the risks. The potential power of AI technology is enormous, and correspondingly we need to consider the risks, take them seriously, and proceed to mitigate them.

Measures and Martingales

In this chapter we provide the proofs for Theorem 3.18 and Theorem 3.19.

Proof of Theorem 3.18. X_t is only undefined if $P(\Gamma_{v_{1:t}}) = 0$. The set

$$\{v \in \Sigma^\infty \mid \exists t.\ P(\Gamma_{v_{1:t}}) = 0\}$$

has P-measure 0 and hence $(X_t)_{t\in\mathbb{N}}$ is well-defined almost everywhere.

X_t is constant on Γ_u for all $u \in \Sigma^t$, and $\mathcal{F}_t$ is generated by a collection of finitely many disjoint sets:

$$\Sigma^\infty = \biguplus_{u\in\Sigma^t} \Gamma_u.$$

(a) Therefore X_t is $\mathcal{F}_t$-measurable.

(b) $\Gamma_u = \biguplus_{a\in\Sigma} \Gamma_{ua}$ for all $u \in \Sigma^t$ and $v \in \Gamma_u$, and therefore

$$\begin{aligned}\mathbb{E}[X_{t+1} \mid \mathcal{F}_t](v) &= \frac{1}{P(\Gamma_u)} \sum_{a\in\Sigma} X_{t+1}(ua) P(\Gamma_{ua}) = \frac{1}{P(\Gamma_u)} \sum_{a\in\Sigma} \frac{Q(\Gamma_{ua})}{P(\Gamma_{ua})} P(\Gamma_{ua}) \\ &\overset{(*)}{=} \frac{1}{P(\Gamma_u)} \sum_{a\in\Sigma} Q(\Gamma_{ua}) = \frac{Q(\Gamma_u)}{P(\Gamma_u)} = X_t(v).\end{aligned}$$

At $(*)$ we used the fact that P is locally absolutely continuous with respect to Q. (If P were not locally absolutely continuous with respect to Q, then there are cases where $P(\Gamma_u) > 0$, $P(\Gamma_{ua}) = 0$, and $Q(\Gamma_{ua}) \neq 0$. Therefore $X_{t+1}(ua)$ does not contribute to the expectation and thus $X_{t+1}(ua)P(\Gamma_{ua}) = 0 \neq Q(\Gamma_{ua})$.)

$P \geq 0$ and $Q \geq 0$ by definition, thus $X_t \geq 0$. Since $P(\Gamma_\epsilon) = Q(\Gamma_\epsilon) = 1$, we have $\mathbb{E}[X_0] = 1$. □

The following lemma gives a convenient condition for the existence of a measure on $(\Sigma^\omega, \mathcal{F}_\infty)$. It is a special case of the Daniell-Kolmogorov Extension Theorem (Rogers and Williams, 1994, Thm. 26.1).

Lemma A.1 (Extending measures)**.** *Let $q : \Sigma^* \to [0,1]$ be a function such that $q(\epsilon) = 1$ and $\sum_{a\in\Sigma} q(ua) = q(u)$ for all $u \in \Sigma^*$. Then there exists a unique probability measure Q on $(\Sigma^\infty, \mathcal{F}_\infty)$ such that $q(u) = Q(\Gamma_u)$ for all $u \in \Sigma^*$.*

To prove this lemma, we need the following two ingredients.

Definition A.2 (Semiring)**.** A set $\mathcal{R} \subseteq 2^\Omega$ is called *semiring over* Ω iff

(a) $\emptyset \in \mathcal{R}$,

(b) for all $A, B \in \mathcal{R}$, the set $A \cap B \in \mathcal{R}$, and

(c) for all $A, B \in \mathcal{R}$, there are pairwise disjoint sets $C_1, \ldots, C_n \in \mathcal{R}$ such that $A \setminus B = \biguplus_{i=1}^n C_i$.

Theorem A.3 (Carathéodory's Extension Theorem; Durrett, 2010, Thm. A.1.1)**.** *Let $\mathcal{R}$ be a semiring over Ω and let $\mu : \mathcal{R} \to [0,1]$ be a function such that*

(a) $\mu(\Omega) = 1$ (normalization),

(b) $\mu(\biguplus_{i=1}^n A_i) = \sum_{i=1}^n \mu(A_i)$ for pairwise disjoint sets $A_1, \ldots, A_n \in \mathcal{R}$ such that $\biguplus_{i=1}^n A_i \in \mathcal{R}$ (finite additivity), and

(c) $\mu(\bigcup_{i\geq 0} A_i) \leq \sum_{i\geq 0} \mu(A_i)$ for any collection $(A_i)_{i\geq 0}$ such that each $A_i \in \mathcal{R}$ and $\bigcup_{i\geq 0} A_i \in \mathcal{R}$ (σ-subadditivity).

Then there is a unique extension $\overline{\mu}$ of μ that is a probability measure on $(\Omega, \sigma(\mathcal{R}))$ such that $\overline{\mu}(A) = \mu(A)$ for all $A \in \mathcal{R}$.

Proof of Lemma A.1. We show the existence of Q using Carathéodory's Extension Theorem. Define $\mathcal{R} := \{\Gamma_u \mid u \in \Sigma^*\} \cup \{\emptyset\}$.

(a) $\emptyset \in \mathcal{R}$.

(b) For any $\Gamma_u, \Gamma_v \in \mathcal{R}$, either

- u is a prefix of v and $\Gamma_u \cap \Gamma_v = \Gamma_v \in \mathcal{R}$, or
- v is a prefix of u and $\Gamma_u \cap \Gamma_v = \Gamma_u \in \mathcal{R}$, or
- $\Gamma_u \cap \Gamma_v = \emptyset \in \mathcal{R}$.

(c) For any $\Gamma_u, \Gamma_v \in \mathcal{R}$,

- $\Gamma_u \setminus \Gamma_v = \biguplus_{w\in\Sigma^{|v|-|u|}\setminus\{x\}} \Gamma_{uw}$ if $v = ux$, i.e., u is a prefix of v, and
- $\Gamma_u \setminus \Gamma_v = \emptyset$ otherwise.

Therefore $\mathcal{R}$ is a semiring. By definition of $\mathcal{R}$, we have $\sigma(\mathcal{R}) = \mathcal{F}_\infty$.

The function $q : \Sigma^* \to [0,1]$ naturally gives rise to a function $\mu : \mathcal{R} \to [0,1]$ with $\mu(\emptyset) := 0$ and $\mu(\Gamma_u) := q(u)$ for all $u \in \Sigma^*$. We will now check the prerequisites of Carathéodory's Extension Theorem.

(a) (Normalization.) $\mu(\Sigma^\infty) = \mu(\Gamma_\epsilon) = q(\epsilon) = 1$.

(b) (Finite additivity.) Let $\Gamma_{u_1}, \ldots, \Gamma_{u_k} \in \mathcal{R}$ be pairwise disjoint sets such that $\Gamma_w := \biguplus_{i=1}^k \Gamma_{u_i} \in \mathcal{R}$. Let $\ell := \max\{|u_i| \mid 1 \leq i \leq k\}$, then $\Gamma_w = \biguplus_{v \in \Sigma^\ell} \Gamma_{wv}$. By assumption, $\sum_{a \in \Sigma} q(ua) = q(u)$, thus $\sum_{a \in \Sigma} \mu(\Gamma_{ua}) = \mu(\Gamma_u)$ and inductively we have

$$\mu(\Gamma_{u_i}) = \sum_{s \in \Sigma^{\ell - |u_i|}} \mu(\Gamma_{u_i s}), \tag{A.1}$$

and

$$\mu(\Gamma_w) = \sum_{v \in \Sigma^\ell} \mu(\Gamma_{wv}). \tag{A.2}$$

For every string $v \in \Sigma^\ell$, the concatenation $wv \in \Gamma_w = \biguplus_{i=1}^k \Gamma_{u_i}$, so there is a unique i such that $wv \in \Gamma_{u_i}$. Hence there is a unique string $s \in \Sigma^{\ell - |u_i|}$ such that $wv = u_i s$. Together with (A.1) and (A.2) this yields

$$\mu\left(\biguplus_{i=1}^k \Gamma_{u_i}\right) = \mu(\Gamma_w) = \sum_{v \in \Sigma^\ell} \mu(\Gamma_{wv}) = \sum_{i=1}^k \sum_{s \in \Sigma^{\ell - |u_i|}} \mu(\Gamma_{u_i s}) = \sum_{i=1}^k \mu(\Gamma_{u_i}).$$

(c) (σ-subadditivity.) We will show that each Γ_u is compact with respect to the topology $\mathcal{O}$ generated by $\mathcal{R}$. σ-subadditivity then follows from (b) because every countable union is in fact a finite union.

We will show that the topology $\mathcal{O}$ is the product topology of the discrete topology on Σ. (This establishes that $(\Sigma^\omega, \mathcal{O})$ is a Cantor Space.) Every projection $\pi_k : \Sigma^\infty \to \Sigma$ selecting the k-th symbol is continuous, since $\pi_k^{-1}(a) = \bigcup_{u \in \Sigma^{k-1}} \Gamma_{ua}$ for every $a \in \Sigma$. Moreover, $\mathcal{O}$ is the coarsest topology with this property, since we can generate every open set $\Gamma_u \in \mathcal{R}$ in the base of the topology by

$$\Gamma_u = \bigcap_{i=1}^{|u|} \pi_i^{-1}(\{u_i\}).$$

The set Σ is finite and thus compact. By Tychonoff's Theorem, Σ^∞ is also compact. Therefore Γ_u is compact since it is homeomorphic to Σ^∞ via the canonical map $\beta_u : \Sigma^\infty \to \Gamma_u$, $v \mapsto uv$.

From (a), (b), and (c) Carathéodory's Extension Theorem yields a unique probability measure Q on $(\Sigma^\infty, \mathcal{F}_\infty)$ such that $Q(\Gamma_u) = \mu(\Gamma_u) = q(u)$ for all $u \in \Sigma^*$. □

Using Lemma A.1, the proof of Theorem 3.19 is now straightforward.

Proof of Theorem 3.19. We define a function $q : \Sigma^* \to \mathbb{R}$, with

$$q(u) := X_{|u|}(v) P(\Gamma_u)$$

for any $v \in \Gamma_u$. The choice of v is irrelevant because $X_{|u|}$ is constant on Γ_u since it is $\mathcal{F}_t$-measurable. In the following, we also write $X_t(u)$ if $|u| = t$ to simplify notation.

The function q is non-negative because X_t and P are both non-negative. Moreover, for any $u \in \Sigma^t$,

$$1 = \mathbb{E}[X_t] = \int_{\Sigma^\infty} X_t dP \geq \int_{\Gamma_u} X_t dP = P(\Gamma_u) X_t(u) = q(u).$$

Hence the range of q is a subset of $[0, 1]$.

We have $q(\epsilon) = X_0(\epsilon) P(\Gamma_\epsilon) = \mathbb{E}[X_0] = 1$ since P is a probability measure and $\mathcal{F}_0 = \{\emptyset, \Sigma^\infty\}$ is the trivial σ-algebra. Let $u \in \Sigma^t$.

$$\begin{aligned}\sum_{a \in \Sigma} q(ua) &= \sum_{a \in \Sigma} X_{t+1}(ua) P(\Gamma_{ua}) = \int_{\Gamma_u} X_{t+1} dP \\ &= \int_{\Gamma_u} \mathbb{E}[X_{t+1} \mid \mathcal{F}_t] dP = \int_{\Gamma_u} X_t dP = P(\Gamma_u) X_t(u) = q(u).\end{aligned}$$

By Lemma A.1, there is a probability measure Q on $(\Sigma^\infty, \mathcal{F}_\infty)$ such that $q(u) = Q(\Gamma_u)$ of all $u \in \Sigma^*$. Therefore, for all $v \in \Sigma^\infty$ and for all $t \in \mathbb{N}$ with $P(\Gamma_{v_{1:t}}) > 0$,

$$X_t(v) = \frac{q(v_{1:t})}{P(\Gamma_{v_{1:t}})} = \frac{Q(\Gamma_{v_{1:t}})}{P(\Gamma_{v_{1:t}})}.$$

Moreover, P is locally absolutely continuous with respect to Q since $P(\Gamma_u) = 0$ implies

$$Q(\Gamma_u) = q(u) = X_{|u|}(u) P(\Gamma_u) = 0. \qquad \square$$

Bibliography

Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, 2011.

Scott Alexander. AI researchers on AI risk. http://slatestarcodex.com/2015/05/22/ai-researchers-on-ai-risk/, May 2015. Accessed: 2016-04-25.

Stuart Armstrong and Laurent Orseau. Interruptibility and corrigibility for AIXI and Monte Carlo agents. 2016. To appear.

Frank Arntzenius, Adam Elga, and John Hawthorne. Bayesianism, infinite decisions, and binding. *Mind*, 113(450):251–283, 2004.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 89–96, 2009.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *International Conference on Machine Learning*, pages 30–37, 1995.

Jacob D Bekenstein. Universal upper bound on the entropy-to-energy ratio for bounded systems. *Physical Review D*, 23(2):287, 1981.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Marc G Bellemare, Georg Ostrovski, Arthur Guez, Philip S Thomas, and Rémi Munos. Increasing the action gap: New operators for reinforcement learning. In *AAAI*, 2016.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

Dimitri P Bertsekas and John Tsitsiklis. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

David Blackwell and Lester Dubins. Merging of opinions with increasing information. *The Annals of Mathematical Statistics*, pages 882–886, 1962.

Nick Bostrom. Existential risks. *Journal of Evolution and Technology*, 9(1):1–31, 2002.

Nick Bostrom. Ethical issues in advanced artificial intelligence. *Science Fiction and Philosophy: From Time Travel to Superintelligence*, pages 277–284, 2003.

Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014.

Nick Bostrom. Strategic implications of openness in AI development. Technical report, Future of Humanity Institute, 2016. http://www.nickbostrom.com/papers/openness.pdf.

Michael Bowling and Manuela Veloso. Rational and convergent learning in stochastic games. In *International Joint Conference on Artificial Intelligence*, pages 1021–1026, 2001.

Sébastien Bubeck and Cesa-Nicolò Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(2):156–172, 2008.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

David Chalmers. The singularity: A philosophical analysis. *Journal of Consciousness Studies*, 17(9-10):7–65, 2010.

Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257, 2011.

Paul Christiano, Eliezer Yudkowsky, Marcello Herreshoff, and Mihaly Barasz. Definability of truth in probabilistic logic. Technical report, Machine Intelligence Research Institute, 2013. https://intelligence.org/files/DefinabilityTruthDraft.pdf.

Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.

Mayank Daswani. *Generic Reinforcement Learning Beyond Small MDPs*. PhD thesis, Australian National University, 2015.

Mayank Daswani and Jan Leike. A definition of happiness for reinforcement learning agents. In *Artificial General Intelligence*, pages 231–240. Springer, 2015.

Ernest Davis. Ethical guidelines for a superintelligence. Technical report, New York University, 2014. https://cs.nyu.edu/davise/papers/Bostrom.pdf.

Adam Day. Increasing the gap between descriptional complexity and algorithmic probability. *Transactions of the American Mathematical Society*, 363(10):5577–5604, 2011.

Richard Dearden, Nir Friedman, and Stuart Russell. Bayesian Q-learning. In *AAAI*, pages 761–768, 1998.

Daniel Dewey. Learning what to value. In *Artificial General Intelligence*, pages 309–314. Springer, 2011.

Joseph L. Doob. *Stochastic Processes*. Wiley, New York, 1953.

Finale Doshi-Velez. *Bayesian Nonparametric Approaches for Reinforcement Learning in Partially Observable Domains*. PhD thesis, Massachusetts Institute of Technology, 2012.

Rick Durrett. *Probability: Theory and Examples*. Cambridge University Press, 4th edition, 2010.

Amnon H Eden. The singularity controversy. Technical report, Sapience Project, 2016. http://arxiv.org/abs/1601.05977.

Amnon H Eden, James H Moor, Johnny H Søraker, and Eric Steinhart, editors. *Singularity Hypotheses: A Scientific and Philosophical Assessment*. Springer, 2013.

Tom Everitt and Marcus Hutter. Avoiding wireheading with value reinforcement learning. In *Artificial General Intelligence*, pages 12–22, 2016.

Tom Everitt, Jan Leike, and Marcus Hutter. Sequential extensions of causal and evidential decision theory. In *Algorithmic Decision Theory*, pages 205–221. Springer, 2015.

Tom Everitt, Daniel Filan, Mayank Daswani, and Marcus Hutter. Self-modification of policy and utility function in rational agents. In *Artificial General Intelligence*, 2016.

Benja Fallenstein, Nate Soares, and Jessica Taylor. Reflective variants of Solomonoff induction and AIXI. In *Artificial General Intelligence*. Springer, 2015a.

Benja Fallenstein, Jessica Taylor, and Paul F Christiano. Reflective oracles: A foundation for game theory in artificial intelligence. In *Logic, Rationality, and Interaction*, pages 411–415. Springer, 2015b.

Benja Fallenstein, Jessica Taylor, and Paul F Christiano. Reflective oracles: A foundation for classical game theory. Technical report, Machine Intelligence Research Institute, 2015c. http://arxiv.org/abs/1508.04145.

Daniel Filan. Agents using speed priors. Honours thesis, Australian National University, October 2015.

Daniel Filan, Jan Leike, and Marcus Hutter. Loss bounds and time complexity for speed priors. In *Artificial Intelligence and Statistics*, 2016.

Jakob N Foerster, Yannis M Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate to solve riddles with deep distributed recurrent Q-networks. Technical report, University of Oxford, 2016. http://arxiv.org/abs/1602.02672.

Dean P Foster and H Peyton Young. On the impossibility of predicting the behavior of rational agents. *Proceedings of the National Academy of Sciences*, 98(22):12848–12853, 2001.

Drew Fudenberg and David K Levine. *The Theory of Learning in Games*. MIT press, 1998.

Future of Life Institute. Elon musk donates $10m to keep AI beneficial. http://futureoflife.org/misc/AI, January 2015a. Accessed: 2015-03-28.

Future of Life Institute. 2015 project grants recommended for funding. http://futureoflife.org/first-ai-grant-recipients/, 2015b. Accessed: 2016-03-28.

Future of Life Institute. Research priorities for robust and beneficial artificial intelligence: An open letter. http://futureoflife.org/ai-open-letter/, January 2015c. Accessed: 2016-04-26.

John Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 148–177, 1979.

Irving John Good. The paradox of confirmation. *British Journal for the Philosophy of Science*, pages 145–149, 1960.

Irving John Good. Speculations concerning the first ultraintelligent machine. *Advances in Computers*, 6:31–88, 1965.

Irving John Good. The white shoe is a red herring. *The British Journal for the Philosophy of Science*, 17(4):322–322, 1967.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Book in preparation for MIT Press, http://www.deeplearningbook.org, 2016.

Google. The latest chapter for the self-driving car: mastering city street driving. http://googleblog.blogspot.com.au/2014/04/the-latest-chapter-for-self-driving-car.html, April 2014. Accessed: 2015-03-27.

Google. What we learned in Seoul with AlphaGo. https://googleblog.blogspot.com.au/2016/03/what-we-learned-in-seoul-with-alphago.html, March 2016. Accessed: 2016-03-28.

Aditya Gopalan and Shie Mannor. Thompson sampling for learning parameterized Markov decision processes. In *Conference on Learning Theory*, pages 861–898, 2015.

Geoffrey J Gordon. Reinforcement learning with function approximation converges to a region. In *Advanced in Neural Information Processing Systems*, 2001.

Peter D. Grünwald. *The Minimum Description Length Principle*. The MIT Press, Cambridge, 2007.

Péter Gács. On the relation between descriptional complexity and algorithmic probability. *Theoretical Computer Science*, 22(1):71–93, 1983.

Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38(1):173–198, 1931.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

Matthew Hausknecht and Peter Stone. Deep recurrent Q-learning for partially observable MDPs. In *2015 AAAI Fall Symposium Series*, 2015.

Stephen Hawking, Max Tegmark, Stuart Russell, and Frank Wilczek. Transcending complacency on superintelligent machines. http://www.huffingtonpost.com/stephen-hawking/artificial-intelligence_b_5174265.html, April 2014. Accessed: 2015-03-28.

Nicholas J Hay. Universal semimeasures: An introduction. Master's thesis, University of Auckland, 2007.

Nicolas Heess, Jonathan J Hunt, Timothy P Lillicrap, and David Silver. Memory-based control with recurrent neural networks. Technical report, Google DeepMind, 2015. http://arxiv.org/abs/1512.04455.

Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games. Technical report, DeepMind, 2016. http://arxiv.org/abs/1603.01121.

Matthias Heizmann, Daniel Dietsch, Jan Leike, Betim Musa, and Andreas Podelski. Ultimate Automizer with array interpolation (competition contribution). In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 455–457. Springer, 2015.

Matthias Heizmann, Daniel Dietsch, Marius Greitschus, Jan Leike, Betim Musa, Claus Schätzle, and Andreas Podelski. Ultimate Automizer with two-track proofs (competition contribution). In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 950–953. Springer, 2016.

Carl G Hempel. Studies in the logic of confirmation (I.). *Mind*, pages 1–26, 1945.

Carl G Hempel. The white shoe: No red herring. *The British Journal for the Philosophy of Science*, 18(3):239–240, 1967.

Marcus Hutter. A theory of universal artificial intelligence based on algorithmic complexity. Technical report, 2000. http://arxiv.org/abs/cs.AI/0004001.

Marcus Hutter. Universal sequential decisions in unknown environments. In *European Workshop on Reinforcement Learning*, pages 25–26, 2001a.

Marcus Hutter. New error bounds for Solomonoff prediction. *Journal of Computer and System Sciences*, 62(4):653–667, 2001b.

Marcus Hutter. Self-optimizing and Pareto-optimal policies in general environments based on Bayes-mixtures. In *Computational Learning Theory*, pages 364–379. Springer, 2002a.

Marcus Hutter. The fastest and shortest algorithm for all well-defined problems. *International Journal of Foundations of Computer Science*, 13(03):431–443, 2002b.

Marcus Hutter. A gentle introduction to the universal algorithmic agent AIXI. Technical report, IDSIA, 2003. ftp://ftp.idsia.ch/pub/techrep/IDSIA-01-03.ps.gz.

Marcus Hutter. *Universal Artificial Intelligence*. Springer, 2005.

Marcus Hutter. Sequential predictions based on algorithmic complexity. *Journal of Computer and System Sciences*, 72(1):95–117, 2006a.

Marcus Hutter. General discounting versus average reward. In *Algorithmic Learning Theory*, pages 244–258. Springer, 2006b.

Marcus Hutter. 50'000€ prize for compressing human knowledge. http://prize.hutter1.net/, 2006c. Accessed: 2015-03-29.

Marcus Hutter. Universal algorithmic intelligence: A mathematical top→down approach. In *Artificial General Intelligence*, pages 227–290. Springer, 2007a.

Marcus Hutter. On universal prediction and Bayesian confirmation. *Theoretical Computer Science*, 384(1):33–48, 2007b.

Marcus Hutter. Discrete MDL predicts in total variation. In *Advances in Neural Information Processing Systems*, pages 817–825, 2009a.

Marcus Hutter. Open problems in universal induction & intelligence. *Algorithms*, 3(2): 879–906, 2009b.

Marcus Hutter. Feature dynamic Bayesian networks. In *Artificial General Intelligence*, pages 67–73. Atlantis Press, 2009c.

Marcus Hutter. Feature reinforcement learning: Part I: Unstructured MDPs. *Journal of Artificial General Intelligence*, 1:3–24, 2009d.

Marcus Hutter. Can intelligence explode? *Journal of Consciousness Studies*, 19(1–2): 143–166, 2012a.

Marcus Hutter. One decade of universal artificial intelligence. In *Theoretical Foundations of Artificial General Intelligence*, pages 67–88. Springer, 2012b.

Marcus Hutter. Extreme state aggregation beyond MDPs. In *Algorithmic Learning Theory*. Springer, 2014.

Marcus Hutter and Andrej A. Muchnik. On semimeasures predicting Martin-Löf random sequences. *Theoretical Computer Science*, 382(3):247–261, 2007.

IBM. Deep Blue. http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/, March 2012a. Accessed: 2015-03-27.

IBM. A computer called Watson. http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/watson/, March 2012b. Accessed: 2015-03-27.

Edwin T Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

Sham Machandranath Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University College London, 2003.

Ehud Kalai and Ehud Lehrer. Rational learning leads to Nash equilibrium. *Econometrica*, pages 1019–1045, 1993.

Ehud Kalai and Ehud Lehrer. Weak and strong merging of opinions. *Journal of Mathematical Economics*, 23(1):73–86, 1994.

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

HJ Kim, Michael I Jordan, Shankar Sastry, and Andrew Y Ng. Autonomous helicopter flight via reinforcement learning. In *Advances in Neural Information Processing Systems*, page None, 2003.

Stephen Cole Kleene. *Introduction to Metamathematics*. Wolters-Noordhoff Publishing, 1952.

Tejas D Kulkarni, Karthik R Narasimhan, Ardavan Saeedi, and Joshua B Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. Technical report, Massachusetts Institute of Technology, 2016. https://arxiv.org/abs/1604.06057.

Ray Kurzweil. *The Singularity is Near: When Humans Transcend Biology*. Viking Books, 2005.

Tor Lattimore. *Theory of General Reinforcement Learning*. PhD thesis, Australian National University, 2013.

Tor Lattimore. Regret analysis of the finite-horizon Gittins index strategy for multi-armed bandits. In *Conference on Learning Theory*, 2016.

Tor Lattimore and Marcus Hutter. Asymptotically optimal agents. In *Algorithmic Learning Theory*, pages 368–382. Springer, 2011.

Tor Lattimore and Marcus Hutter. PAC bounds for discounted MDPs. In *Algorithmic Learning Theory*, pages 320–334. Springer, 2012.

Tor Lattimore and Marcus Hutter. On Martin-Löf convergence of Solomonoff's mixture. In *Theory and Applications of Models of Computation*, volume 7876, pages 212–223. Springer, 2013.

Tor Lattimore and Marcus Hutter. General time consistent discounting. *Theoretical Computer Science*, 519:140–154, 2014.

Tor Lattimore and Marcus Hutter. On Martin-Löf (non-)convergence of Solomonoff's universal mixture. *Theoretical Computer Science*, 588:2–15, 2015.

Tor Lattimore, Marcus Hutter, and Vaibhav Gavane. Universal prediction of selected bits. In *Algorithmic Learning Theory*, pages 262–276. Springer, 2011.

Neil Lawrence. Future of AI 6. Discussion of 'Superintelligence: Paths, Dangers, Strategies'. http://inverseprobability.com/2016/05/09/machine-learning-futures-6, May 2016. Accessed: 2016-05-10.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Shane Legg. Is there an elegant universal theory of prediction? In *Algorithmic Learning Theory*, pages 274–287, 2006.

Shane Legg. *Machine Super Intelligence*. PhD thesis, University of Lugano, 2008.

Shane Legg and Marcus Hutter. A collection of definitions of intelligence. *Frontiers in Artificial Intelligence and Applications*, 157:17–24, 2007a.

Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds & Machines*, 17(4):391–444, 2007b.

Shane Legg and Joel Veness. An approximation of the universal intelligence measure. In *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*, pages 236–249. Springer, 2013.

Ehud Lehrer and Rann Smorodinsky. Merging and learning. *Statistics, Probability and Game Theory*, pages 147–168, 1996.

Jan Leike and Matthias Heizmann. Ranking templates for linear loops. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 172–186. Springer, 2014a.

Jan Leike and Matthias Heizmann. Geometric series as nontermination arguments for linear lasso programs. Technical report, University of Freiburg, 2014b. http://arxiv.org/abs/1405.4413.

Jan Leike and Matthias Heizmann. Ranking templates for linear loops. *Logical Methods in Computer Science*, 11(1):1–27, March 2015.

Jan Leike and Matthias Heizmann. Geometric nontermination arguments. 2016. Under preparation.

Jan Leike and Marcus Hutter. Indefinitely oscillating martingales. In *Algorithmic Learning Theory*, pages 321–335, 2014a.

Jan Leike and Marcus Hutter. Indefinitely oscillating martingales. Technical report, Australian National University, 2014b. http://arxiv.org/abs/1408.3169.

Jan Leike and Marcus Hutter. On the computability of AIXI. In *Uncertainty in Artificial Intelligence*, pages 464–473, 2015a.

Jan Leike and Marcus Hutter. On the computability of Solomonoff induction and knowledge-seeking. In *Algorithmic Learning Theory*, pages 364–378, 2015b.

Jan Leike and Marcus Hutter. Bad universal priors and notions of optimality. In *Conference on Learning Theory*, pages 1244–1259, 2015c.

Jan Leike and Marcus Hutter. Solomonoff induction violates Nicod's criterion. In *Algorithmic Learning Theory*, pages 349–363. Springer, 2015d.

Jan Leike and Marcus Hutter. On the computability of Solomonoff induction and AIXI. 2016. Under review.

Jan Leike, Tor Lattimore, Laurent Orseau, and Marcus Hutter. Thompson sampling is asymptotically optimal in general environments. In *Uncertainty in Artificial Intelligence*, 2016a.

Jan Leike, Jessica Taylor, and Benya Fallenstein. A formal solution to the grain of truth problem. In *Uncertainty in Artificial Intelligence*, 2016b.

Leonid A Levin. On the notion of a random sequence. *Soviet Mathematics Doklady*, 14 (5):1413–1416, 1973.

Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Texts in Computer Science. Springer, 3rd edition, 2008.

Yitao Liang, Marlos C Machado, Erik Talvitie, and Michael Bowling. State of the art control of Atari games using shallow reinforcement learning. In *Autonomous Agents and Multiagent Systems*, 2016.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016.

John L Mackie. The paradox of confirmation. *British Journal for the Philosophy of Science*, pages 265–277, 1963.

Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *AAAI*, pages 541–548, 1999.

Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147 (1):5–34, 2003.

Sridhar Mahadevan. Optimality criteria in reinforcement learning. In *AAAI Fall Symposium on Learning Complex Behaviors in Adaptive Intelligent Systems*, 1996.

Patrick Maher. Inductive logic and the ravens paradox. *Philosophy of Science*, pages 50–70, 1999.

A Rupam Mahmood, Huizhen Yu, Martha White, and Richard Sutton. Emphatic temporal-difference learning. Technical report, University of Alberta, 2015. http://arxiv.org/abs/1507.01569.

Norman Margolus and Lev B Levitin. The maximum speed of dynamical evolution. *Physica D: Nonlinear Phenomena*, 120(1):188–195, 1998.

Jarryd Martin, Tom Everitt, and Marcus Hutter. Death and suicide in universal artificial intelligence. In *Artificial General Intelligence*, 2016.

John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon. A proposal for the Dartmouth summer research project on artificial intelligence. 1955.

Ronald I Miller and Chris William Sanchirico. The role of absolute continuity in "merging of opinions" and "rational learning". *Games and Economic Behavior*, 29(1):170–190, 1999.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. Technical report, Google DeepMind, 2013. http://arxiv.org/abs/1312.5602.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 2016.

Luke Muehlhauser and Anna Salamon. Intelligence explosion: Evidence and import. In *Singularity Hypotheses*, pages 15–42. Springer, 2012.

Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon Markov decision process problems. *Journal of the ACM*, 47(4): 681–720, 2000.

Markus Müller. Stationary algorithmic probability. *Theoretical Computer Science*, 411 (1):113–130, 2010.

Vincent C Müller and Nick Bostrom. Future progress in artificial intelligence: A survey of expert opinion. *Fundamental Issues of Artificial Intelligence*, pages 553–571, 2016.

John H Nachbar. Prediction, optimization, and learning in repeated games. *Econometrica*, 65(2):275–309, 1997.

John H Nachbar. Beliefs in repeated games. *Econometrica*, 73(2):459–480, 2005.

Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, Shane Legg, Volodymyr Mnih, Koray Kavukcuoglu, and David Silver. Massively parallel methods for deep reinforcement learning. Technical report, Google DeepMind, 2015. http://arxiv.org/abs/1507.04296.

Andrew Ng. Is A.I. an existential threat to humanity? https://www.quora.com/Is-A-I-an-existential-threat-to-humanity/answer/Andrew-Ng, January 2016. Accessed: 2016-05-19.

Phuong Nguyen, Odalric-Ambrym Maillard, Daniil Ryabko, and Ronald Ortner. Competing with an infinite set of models in reinforcement learning. In *Artificial Intelligence and Statistics*, pages 463–471, 2013.

Jean Nicod. *Le Problème Logique de L'Induction*. Presses Universitaires de France, 1961.

André Nies. *Computability and Randomness*. Oxford University Press, 2009.

James Olds and Peter Milner. Positive reinforcement produced by electrical stimulation of septal area and other regions of rat brain. *Journal of Comparative and Physiological Psychology*, 47(6):419, 1954.

Stephen M Omohundro. The basic AI drives. In *Artificial General Intelligence*, pages 483–492, 2008.

OpenAI. About OpenAI. https://openai.com/about/, December 2015. Accessed: 2016-04-27.

Laurent Orseau. Optimality issues of universal greedy agents with static priors. In *Algorithmic Learning Theory*, pages 345–359. Springer, 2010.

Laurent Orseau. Universal knowledge-seeking agents. In *Algorithmic Learning Theory*, pages 353–367. Springer, 2011.

Laurent Orseau. Asymptotic non-learnability of universal agents with computable horizon functions. *Theoretical Computer Science*, 473:149–156, 2013.

Laurent Orseau. Universal knowledge-seeking agents. *Theoretical Computer Science*, 519:127–139, 2014a.

Laurent Orseau. The multi-slot framework: A formal model for multiple, copiable AIs. In *Artificial General Intelligence*, pages 97–108. Springer, 2014b.

Laurent Orseau. Teleporting universal intelligent agents. In *Artificial General Intelligence*, pages 109–120. Springer, 2014c.

Laurent Orseau and Stuart Armstrong. Safely interruptible agents. In *Uncertainty in Artificial Intelligence*, pages 557–566, 2016.

Laurent Orseau and Mark Ring. Self-modification and mortality in artificial agents. In *Artificial General Intelligence*, pages 1–10. Springer, 2011.

Laurent Orseau and Mark Ring. Space-time embedded intelligence. In *Artificial General Intelligence*, pages 209–218. Springer, 2012a.

Laurent Orseau and Mark Ring. Memory issues of intelligent agents. In *Artificial General Intelligence*, pages 219–231. Springer, 2012b.

Laurent Orseau, Tor Lattimore, and Marcus Hutter. Universal knowledge-seeking agents for stochastic environments. In *Algorithmic Learning Theory*, pages 158–172. Springer, 2013.

Pedro A Ortega and Daniel A Braun. A minimum relative entropy principle for learning and acting. *Journal of Artificial Intelligence Research*, pages 475–511, 2010.

Pedro A Ortega and Daniel A Braun. Generalized Thompson sampling for sequential decision-making and causal inference. *Complex Adaptive Systems Modeling*, 2(1):2, 2014.

Ian Osband, Dan Russo, and Benjamin van Roy. (More) efficient reinforcement learning via posterior sampling. In *Neural Information Processing Systems*, pages 3003–3011, 2013.

Christos H Papadimitriou and John N Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

Martin L Puterman. *Markov Decision Processes*. John Wiley & Sons, 2014.

Samuel Rathmanner and Marcus Hutter. A philosophical treatise of universal induction. *Entropy*, 13(6):1076–1136, 2011.

Mark Ring and Laurent Orseau. Delusion, survival, and intelligent agents. In *Artificial General Intelligence*, pages 11–20. Springer, 2011.

Chris Rogers and David Williams. *Diffusions, Markov Processes, and Martingales: Volume 1, Foundations*. Cambridge University Press, 2nd edition, 1994.

Stuart Russell, Daniel Dewey, and Max Tegmark. Research priorities for robust and beneficial artificial intelligence. Technical report, Future of Life Institute, 2015. http://arxiv.org/abs/1602.03506.

Stuart J Russell and Peter Norvig. *Artificial Intelligence. A Modern Approach*. Prentice Hall, 3rd edition, 2010.

Daniil Ryabko. Characterizing predictable classes of processes. In *Uncertainty in Artificial Intelligence*, pages 471–478, 2009.

Daniil Ryabko. On finding predictors for arbitrary families of processes. *The Journal of Machine Learning Research*, 11:581–602, 2010.

Daniil Ryabko. On the relation between realizable and nonrealizable cases of the sequence prediction problem. *The Journal of Machine Learning Research*, 12:2161–2180, 2011.

Daniil Ryabko and Marcus Hutter. On sequence prediction for arbitrary measures. In *IEEE International Symposium on Information Theory*, pages 2346–2350, 2007.

Daniil Ryabko and Marcus Hutter. Predicting non-stationary processes. *Applied Mathematics Letters*, 21(5):477–482, 2008.

Régis Sabbadin, Jérôme Lang, and Nasolo Ravoanjanahry. Purely epistemic Markov decision processes. In *AAAI*, pages 1057–1062, 2007.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

Jürgen Schmidhuber. The speed prior: A new simplicity measure yielding near-optimal computable predictions. In *Computational Learning Theory*, pages 216–228. Springer, 2002.

Jürgen Schmidhuber. Philosophers & futurists, catch up! *Journal of Consciousness Studies*, 19(1-2):173–182, 2012.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

Murray Shanahan. *The Technological Singularity*. MIT Press, 2015.

Claude E Shannon. Programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(314):256–275, 1950.

Joseph R Shoenfield. *Mathematical Logic*. Addison-Wesley Reading, 1967.

Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Satinder Singh, Michael L Littman, Nicholas K Jong, David Pardoe, and Peter Stone. Learning predictive state representations. In *International Conference on Machine Learning*, pages 712–719, 2003.

Satinder Singh, Michael R James, and Matthew R Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Uncertainty in Artificial Intelligence*, pages 512–519, 2004.

Nate Soares. Formalizing two problems of realistic world-models. Technical report, Machine Intelligence Research Institute, 2015. http://intelligence.org/files/RealisticWorldModels.pdf.

Nate Soares and Benja Fallenstein. Aligning superintelligence with human interests: A technical research agenda. Technical report, Machine Intelligence Research Institute, 2014. http://intelligence.org/files/TechnicalAgenda.pdf.

Ray Solomonoff. A formal theory of inductive inference. Parts 1 and 2. *Information and Control*, 7(1):1–22 and 224–254, 1964.

Ray Solomonoff. Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory*, 24(4):422–432, 1978.

Kaj Sotala and Roman V Yampolskiy. Responses to catastrophic AGI risk: A survey. *Physica Scripta*, 90(1):018001, 2014.

Tom F Sterkenburg. Putnam's diagonal argument and the impossibility of a universal learning machine. Technical report, Centrum Wiskunde & Informatica, 2016. http://philsci-archive.pitt.edu/12096/.

Jordan M Stoyanov. *Counterexamples in Probability*. Courier Corporation, 3rd edition, 2013.

Malcolm Strens. A Bayesian framework for reinforcement learning. In *International Conference on Machine Learning*, pages 943–950, 2000.

Peter Sunehag and Marcus Hutter. Consistency of feature Markov processes. In *Algorithmic Learning Theory*, pages 360–374. Springer, 2010.

Peter Sunehag and Marcus Hutter. Optimistic agents are asymptotically optimal. In *Australasian Joint Conference on Artificial Intelligence*, pages 15–26. Springer, 2012a.

Peter Sunehag and Marcus Hutter. Optimistic AIXI. In *Artificial General Intelligence*, pages 312–321. Springer, 2012b.

Peter Sunehag and Marcus Hutter. Rationality, optimism and guarantees in general reinforcement learning. *Journal of Machine Learning Research*, 16:1345–1390, 2015.

Richard Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Richard G Swinburne. The paradoxes of confirmation: A survey. *American Philosophical Quarterly*, pages 318–330, 1971.

Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers, 2010.

Gerald Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294, 1933.

John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

Alexandre Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2008.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *AAAI*, 2016.

Joel Veness, Kee Siong Ng, Marcus Hutter, William Uther, and David Silver. A Monte-Carlo AIXI approximation. *Journal of Artificial Intelligence Research*, 40(1):95–142, 2011.

Joel Veness, Marc G Bellemare, Marcus Hutter, Alvin Chua, and Guillaume Desjardins. Compress and control. In *AAAI*, 2015.

Vernor Vinge. The coming technological singularity. *Vision 21: Interdisciplinary Science and Engineering in the Era of Cyberspace*, 1:11–22, 1993. http://www.rohan.sdsu.edu/faculty/vinge/misc/singularity.html.

Paul MB Vitányi, Frank J Balbach, Rudi L Cilibrasi, and Ming Li. Normalized information distance. In *Information Theory and Statistical Learning*, pages 45–82. Springer, 2009.

Nikos Vlassis, Mohammad Ghavamzadeh, Shie Mannor, and Pascal Poupart. Bayesian reinforcement learning. In Marco Wiering and Martijn van Otterlo, editors, *Reinforcement Learning*, pages 359–386. Springer, 2012.

Peter BM Vranas. Hempel's raven paradox: A lacuna in the standard Bayesian solution. *The British Journal for the Philosophy of Science*, 55(3):545–560, 2004.

Toby Walsh. The singularity may never be near. Technical report, University of New South Wales, 2016. http://arxiv.org/abs/1602.06462.

Ziyu Wang, Nando de Freitas, Tom Schaul, Matteo Hessel, Hado van Hasselt, and Marc Lanctot. Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, 2016.

Larry Wasserman. *All of Statistics*. Springer, 2004.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4): 279–292, 1992.

Eric W Weisstein. Random walk—1-dimensional. In *MathWorld—A Wolfram Web Resource*. Wolfram Research, Inc., 2002. http://mathworld.wolfram.com/RandomWalk1-Dimensional.html.

Marco Wiering and Martijn van Otterlo, editors. *Reinforcement Learning*. Springer, 2012.

Ian Wood, Peter Sunehag, and Marcus Hutter. (Non-)equivalence of universal priors. In *Solomonoff 85th Memorial Conference*, pages 417–425. Springer, 2011.

Huizhen Yu. On convergence of emphatic temporal-difference learning. In *Conference on Learning Theory*, pages 1724–1751, 2015.

Eliezer Yudkowsky. Creating friendly AI 1.0: The analysis and design of benevolent goal architectures. Technical report, Singularity Institute for Artificial Intelligence, 2001.

Eliezer Yudkowsky. Artificial intelligence as a positive and negative factor in global risk. In *Global Catastrophic Risks*, pages 308–345. Oxford University Press, 2008.

Tom Zahavy, Nir Ben Zrihem, and Shie Mannor. Graying the black box: Understanding DQNs. Technical report, Israel Institute of Technology, 2016. http://arxiv.org/abs/1602.02658.

Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, 23(3):337–343, 1977.

List of Notation

Abbreviations

AIXI	Bayesian RL agent with a Solomonoff prior, see Section 4.3.1
AI	artificial intelligence
HLAI	human-level artificial intelligence
MDL	minimum description length, see Example 3.14
MDP	Markov decision process, see Section 4.1.3
POMDP	partially observable Markov decision process, see Section 4.1.3
RL	reinforcement learning
UTM	universal Turing machine, Section 2.4

Mathematical notation

$:=$	defined to be equal
$:\in$	defined to be an element of
A, B, Ω	sets
$\#A$	the cardinality of the set A, i.e., the number of elements
$\Delta\Omega$	the set of probability distributions over a finite or countable set Ω
$\mathbb{1}_x$	the characteristic function that is 1 for x and 0 otherwise.
f, g	functions
$f \overset{\times}{\geq} g$	there is a constant $c > 0$ such that $f \geq cg$
$f \overset{\times}{=} g$	$f \overset{\times}{\geq} g$ and $g \overset{\times}{\geq} f$
$\mathbb{N}$	the set of natural numbers, starting with 1
$\mathbb{Q}$	the set of rational numbers
$\mathbb{R}$	the set of real numbers
n, k, t, m, i, j	natural numbers
t	(current) time step, $t \in \mathbb{N}$
k	some other time step, $k \in \mathbb{N}$
q, q'	rational numbers
r	real number
$\mathcal{X}$	a finite nonempty alphabet
$\mathcal{X}^*$	the set of all finite strings over the alphabet $\mathcal{X}$
$\mathcal{X}^\infty$	the set of all infinite strings over the alphabet $\mathcal{X}$
$\mathcal{X}^\sharp$	$\mathcal{X}^\sharp := \mathcal{X}^* \cup \mathcal{X}^\infty$, the set of all finite and infinite strings over the alphabet $\mathcal{X}$
x, y, z	(typically finite) strings from $\mathcal{X}^\sharp$
$x_{<t}$	the first $t-1$ symbols of the string x

$x \sqsubseteq y$	the string x is a prefix of the string y
$\text{zeros}(x)$	the number of zeros in the binary string $x \in \{0,1\}^*$
$\text{ones}(x)$	the number of ones in the binary string $x \in \{0,1\}^*$
ϕ, ψ	computable functions
φ	formula of first-order logic
η	computable relation/quantifier-free formula
T	a Turing machine
p, p'	programs on a universal Turing machine in the form of finite binary strings
$\|p\|$	length of the program p in bits
K	the Kolmogorov complexity of a string or a semimeasure
Km	the monotone Kolmogorov complexity of a string
Ent	entropy
KL_m	KL-divergence
D_m	total variation distance
IG	information gain
F	expected total variation distance
$\mathcal{F}$, $\mathcal{F}_t$, $\mathcal{F}_\infty$	σ-algebras
Γ_x	the cylinder set of all strings starting with x
A, H, E	measurable sets
X, Y	real-valued random variables
P	a distribution over X^∞, the *true* distribution
Q	a distribution over X^∞, the learning algorithm or belief distribution
$\text{Bernoulli}(r)$	a Bernoulli process with parameter r
λ	the uniform measure or Lebesgue measure
ρ_L	Laplace rule
M	Solomonoff's prior
$\overline{M}$	the measure mixture
S_{Kt}	the speed prior
ν	a semimeasure
ν_{norm}	the Solomonoff normalization of the semimeasure ν
$\gg$	absolute continuity
$\overset{\times}{\geq}_W$	weak dominance
$\gg_L$	local absolute continuity
$\mathcal{A}$	the finite set of possible actions
$\mathcal{O}$	the finite set of possible observations
$\mathcal{E}$	the finite set of possible percepts, $\mathcal{E} \subset \mathcal{O} \times \mathbb{R}$
α, β	two different actions, $\alpha, \beta \in \mathcal{A}$
a_t	the action in time step t
o_t	the observation in time step t
r_t	the reward in time step t, bounded between 0 and 1
e_t	the percept in time step t, we use $e_t = (o_t, r_t)$ implicitly
$æ_{<t}$	the first $t-1$ interactions, $a_1 e_1 a_2 e_2 \dots a_{t-1} e_{t-1}$ (a history of length $t-1$)

h	a history, $h \in (\mathcal{A} \times \mathcal{E})^*$
ϵ	the history of length 0
ε, δ	small positive real numbers
γ	the discount function $\gamma : \mathbb{N} \to \mathbb{R}_{\geq 0}$, defined in Definition 4.3
Γ_t	a discount normalization factor, $\Gamma_t := \sum_{k=t}^{\infty} \gamma(k)$
m	horizon of the agent (how many steps it plans ahead)
$H_t(\varepsilon)$	an ε-effective horizon
ν, μ, ρ	environments
$\pi, \tilde{\pi}$	policies, $\pi, \tilde{\pi} : (\mathcal{A} \times \mathcal{E})^* \to \mathcal{A}$
π^*_ν	an optimal policy for environment ν
ν^π	the history distribution generated by policy π in environment ν
$\mathbb{E}^\pi_\nu$	the expectation with respect to the history distribution ν^π
V^π_ν	the ν-expected value of the policy π
V^*_ν	the optimal value in environment ν
W^π_ν	the iterative value of the policy π in environment ν
W^*_ν	the optimal iterative value in environment ν
max	the max-sum-operator
$R_m(\pi, \mu)$	regret of policy π in environment μ for horizon m
$\Upsilon_\xi(\pi)$	the Legg-Hutter intelligence of policy π measured in the universal mixture ξ
$\underline{\Upsilon}_\xi$	the minimal Legg-Hutter intelligence measured in the universal mixture ξ
$\overline{\Upsilon}_\xi$	the maximal Legg-Hutter intelligence measured in the universal mixture ξ
$\mathcal{M}$	a class of environments
$\mathcal{M}^{\text{CCS}}$	the class of all chronological contextual semimeasures
$\mathcal{M}^{\text{CCS}}_{\text{LSC}}$	the class of all lower semicomputable chronological contextual semimeasures
$\mathcal{M}^{\text{CCM}}_{\text{comp}}$	the class of all computable chronological contextual measures
$\mathcal{M}^{O}_{\text{refl}}$	the class of all reflective-oracle-computable environments
U	reference universal Turing machine
U'	a 'bad' universal Turing machine
w	a positive prior over the environment class
w'	a 'bad' positive prior over the environment class
ξ	the universal mixture over all environments $\mathcal{M}^{\text{CCS}}_{\text{LSC}}$ given by the reference UTM U
ξ'	a 'bad' universal mixture over all environments $\mathcal{M}^{\text{CCS}}_{\text{LSC}}$ given by the 'bad' UTM U'
$\mathcal{T}$	the set of all probabilistic Turing machines
O	an oracle
$\tilde{O}$	a partial oracle
λ_T	the semimeasure generated by Turing machine T
λ^O_T	the semimeasure generated by Turing machine T running with oracle O

$\overline{\lambda}_T^O$	the completion of λ_T^O into a measure using oracle O
σ	a multi-agent environment
σ_i	the subjective environment of agent i acting in multi-agent environment σ

Index