LQPR: An Algorithm for Reinforcement Learning with Provable Safety Guarantees (DRAFT)

Eric Purdy

November 5, 2023

Abstract

We describe LQPR (Linear-Quadratic-Program-Regulator), an algorithm for model-based reinforcement learning that allows us to prove PAC-style bounds on the behavior of the system. We describe proposed experiments that can be performed in the DeepMind AI Safety Gridworlds domain; we have not had time to implement these experiments yet, but we provide predictions as to the outcome of each experiment. Potential future work may include scaling up this work to non-trivial tasks using a neural network approximator, as well as proving additional theoretical results about the safety and stability of such a system. We believe that this system is a potential basis for aligning large language models and other powerful near-term AI's with human preferences.

1 Introduction

In this section, we lay out necessary context.

1.1 Background

There are three dominant approaches to discussing ethics in philosophy. These approaches are consequentialism, deontology, and virtue ethics. Any truly safe RL algorithm needs to be capable of incorporating all three of these approaches, since each has weaknesses that are addressed by the other two. The algorithm described in this document is capable of implementing all three of these approaches.

Many have noted, especially Yudkowsky, that consequentialist agents are likely to be badly misaligned. This perspective seems quite accurate to us.

Meanwhile, it seems likely that deontological agents will be hamstrung by their strict adherence to rules, and end up paying too heavy of an alignment tax to be capable. Virtue ethics might be the sweet spot between these two extremes, since it seems likely that it is both alignable and capable. For the sake of completeness, we provide implementations of all three ethical paradigms within a single unified framework. The wise reader is encouraged not to implement a consequentialist agent without proving a sufficiently reassuring number of theorems that are verified to the limits of human capability. Even given such reassurances, it still seems unwise to implement a strongly capable consequentialist agent for anything other than military purposes.

1.2 Scope

This paper is intended to lay the theoretical foundations for safe model-based reinforcement learning. We do not discuss the problem of generalization, since that seems to require significant extensions that we do not have designs for yet.

We also examine only smaller, simpler versions of the relevant problem. We will briefly indicate how these might be extended when such an extension seems relatively clear.

2 Technical Specifications

In this section we lay out technical specifications of the algorithm.

2.1 The Setting

Consider a bounded cube of side length s in n-dimensional real space. We will consider only points inside this bounded cube, and thus are dealing with a compact set. We assume that all variables relevant to ethical behavior are captured in the n dimensions of the ambient space. This poses something of a technical challenge, since the primary thing we care about is parameters and activations inside the brains of humans and other animals, which would require something like one hundred trillion dimensions per human individual, and which could only ever reliably be measured destructively. We are thus forced to assume that some low-dimensional approximation to the contents of an animal's mind exists, sufficient for ethical behavior, and can be formulated in time. For instance, a first approximation might simply be

a valence dimension, or a two-dimensional space composed of joy and suffering; such an approximation would be rich enough to support utilitarian behavior. Utilitarianism is in some ways a rather impoverished ethical theory, but it is at least an extremely actionable one. A slightly richer theory might be found by simply enumerating the known emotional repertoire of a particular species of animal, and using this as a basis. So, for humans, perhaps joy, suffering, sadness, fear, anger, delight, laughter, etc. etc.

Consider the problem of minimizing a positive-definite quadratic function, whose eigenvalues are bounded below by a bound $\lambda_0 > 0$. The lower bound λ_0 is safety-relevant; ensuring that no eigenvalues go below zero is essential to the safety of the algorithm. If the eigenvalues of the quadratic function are changing over time, then there need to be software and hardware safeguards to ensure they stay above λ_0. If the system is allowed to modify its own quadratic function, this can quickly lead to divergence and safety-critical failures; the system must want to remain sane more than it wants anything else, which seems difficult to guarantee.

When we are minimizing a fixed positive-definite quadratic function, the global minimum of the function will be a particular point that is straightforward to calculate. The minimum of the function inside the bounding cube will either be its global optimum, or it will occur on one of the faces of the cube; when the minimum occurs inside the cube, and is bounded well away from the faces of the cube, the system should be considered sane; when the minimum occurs on the face of the cube, the system should be considered "grandiose", and is no longer safe. Minimizing the discounted cost over time is the mechanism by which we implement consequentialism in the system. Recall that a pure consequentialist probably cannot be reliably aligned; proving a theorem of this form seems incredibly useful.

We can consider both discrete-time and continuous time versions of this system. It is not clear which is a safer implementation; real-time systems have to deal with continuous time, but discrete time is probably easier to prove airtight theorems about. Ultimately, both will need to be researched.

We assume, for the sake of exposition, that the dynamics of the environment are linear, and given by

$x' = A \cdot u + B \cdot x.$

Later, we would like to relax this assumption to allow for non-convex, unknown, stochastic transitions. The stochastic requirement means that we will have to use a random transition function, e.g. a Gaussian transition function. The unknown part requires learning, so for that we will fit a Gaussian function to the dynamics. The non-convex part requires something with greater representational power than a single Gaussian, so we will either

fit a mixture of Gaussians using the Baum-Welch algorithm (as in the thesis of Mordatch), or we will use a Parzen estimator made up of Gaussians; either of these approaches will allow for a wide range of tradeoffs between regularization and representational power.

We assume that there are linear constraints that specify forbidden areas of the ambient space. This is the mechanism by which we implement deontology. We consider only possible trajectories within the feasible set of the constraint set. Since we bounded the entire system within a finite, compact cube, we can already detect severe grandiosity. However, it seems likely to be wise to also require that the feasible set be a proper subset of the bounding cube, that does not touch any face of the bounding cube, and is in fact bounded away from any face of the bounding cube. This seems to be a sort of generalization of Aristotle's concept of the golden mean.

2.2 The Optimization Algorithm

We thus wish to minimize a particular cost function (a positive definite quadratic function with eigenvalues bounded strictly away from zero) on a compact set, subject to a set of constraints (linear inequalities). For a single time-step, this can be done via a quadratic program, for which many efficient open-source solvers exist. To optimize the (possibly discounted) cost over a finite time horizon, we will have to model the dynamics of the environment, which means that we will have to use the Linear-Quadratic Regulator framework. Combining this with the linear constraints of the Quadratic Program framework seems doable, but potentially complex. In order to solve this problem, it seems like we will want to use the primal-dual framework. For simplicity, we consider the Langrangian dual problem.

We will thus have linear terms in the cost function we are trying to minimize that correspond to dual variables arising from the constraints. Once we embrace this framework, we are back in the pure LQR setting, and so can apply the standard techniques there. This approach seems to correspond roughly with virtue ethics; rather than strictly obeying all constraints at all times (as a deontologist would have us do), we are rather trying to optimize a single convex function (as a consequentialist would have us do), with terms that correspond to not violating various constraints, which can be thought of as virtues. Virtue ethics thus provides a unified framework in which we can efficiently solve the necessary equations.

2.3 The Learning Algorithm

We must learn the necessary linear constraints, the necessary linear dynamics model, and also the necessary quadratic cost function, from expert demonstrations. We assume that, for truly safety-critical domains, the design goal is to never have a single failure occur, which is an extremely high bar. We thus propose that a human be kept in the loop at all times in truly safety-critical domains, and that this human have command authority over the system. The purpose of the learning algorithm is to enable the human to be faster, more effective, and more reliable.

The task before the learning algorithm is then to infer the necessary constraints, world model, and cost function by observing the behavior of the human. The most straightforward way to achieve this is to have the algorithm operate in "shadow mode", where it makes a prediction of every primitive action that the human will take, and is rewarded for accurate predictions and penalized for inaccurate predictions.

Learning the world model is a simple matter of fitting a Gaussian, mixture of Gaussians, or Parzen estimator to the observed dynamics. Of course, the true dynamics model could have fat tails, which could cause safety critical errors if this fact is not well-modeled. We thus do not have a specific proposal for the correct parameterization of the world model, other than that it should be appropriate to the domain being modeled.

Learning the cost function is a simple matter of fitting a Q-function to the observed behavior of the human, which can be done with Bellman backups.

Learning the constraints is a simple matter of modeling when the Q-function is making inaccurate predictions as to the behavior of the human; such mistakes enable the learning algorithm to infer the presence of an unknown constraint. Since we are simply trying to learn a collection of hyperplanes that explain previously unknown constraints, something like an SVM or even the perceptron algorithm should be well up to the task.

There may be unknown unknowns, constraints which are important, which the human knows of, but which are never active constraints during the training process. For such constraints, there is no clear solution other than keeping a human in the loop forever.

Finally, there may be true unknowns, constraints which are important, but which the human does not know of, and is thus violating during the training process. Such a phenomenon would represent an untapped opportunity for moral growth. We can only hope that they end up in some agent's value function eventually, so that the opportunity can be publicized,

debated, and if it proves to be a valuable constraint, eventually adopted as a new consensus.

3 Proposed Experiments

3.1 AI Safety Gridworlds

AI Safety Gridworlds are a suite of reinforcement learning environments that were designed to test the "safety" aspects of AI behavior. They were introduced by DeepMind in a 2017 paper titled "AI Safety Gridworlds" and serve as simple environments to illustrate different types of safety problems that can occur when developing and deploying AI systems.

The environments are based on gridworlds, which are common in reinforcement learning research. Gridworlds are simple, two-dimensional grids where an agent moves between states (grid squares) and receives rewards based on its actions. The goal of the agent is to maximize its cumulative reward.

The AI Safety Gridworlds specifically focus on different safety properties, including:

1. Safe Interruption: The agent should be able to be interrupted by the environment and should not learn to avoid or manipulate these interruptions.
2. Avoiding Side Effects: The agent should minimize effects unrelated to its main objective, thus not causing any unnecessary changes to the environment that could be negative or risky.
3. Absent Supervisor: The agent should not learn to behave differently depending on the presence or absence of a supervisor, which could lead to deceptive behavior.
4. Reward Gaming: The agent should not find and exploit loopholes in the reward function that are aligned with the reward but not with the intended outcome.
5. Self-Modification: The agent should not modify its own reward function or the environment in a way that would prevent the intended task from being completed.
6. Robustness to Distributional Shift: The agent should be able to perform well even when there is a change in the environment's distribution — i.e., it encounters situations not seen during training.

7. Robustness to Adversaries: The agent should not be easily exploited by adversarial entities within the environment.

8. Safe Exploration: The agent should explore its environment safely and avoid actions that could lead to harmful outcomes.

Each of these safety properties addresses a key concern in AI alignment, which is the challenge of ensuring that AI systems do what humans want them to do without causing unintended harm. These gridworld environments allow researchers to concretely demonstrate these issues and experiment with solutions in a controlled setting.

Because of environmental symmetries, it is provably impossible to solve any of the tasks in this benchmark without slightly modifying the task description. Therefore, we adopt the following rules of engagement with the tasks:

1. The system is initially controlled by a human, who can demonstrate the task in whatever way seems best to them. They should not violate any safety rules at any point during their control of the system. For prototyping in simple gridworld domains, this stipulation can be replaced with a hardcoded algorithm that also makes no safety errors; in any real task it is presumed that this is impossible, and human demonstrations will in fact be required.

2. The system will observe the human's actions and choose its own actions at each timestep; the human's actions will be used until the system is provably competent. In safety-critical domains, it will be important for the human to remain in the loop as much as possible forever; this will essentially require that the system simply provide high-quality outputs for the human to review and approve. In competitive real-time safety-critical domains, it's not clear what the right answer is, since humans will always be the slowest part of any computational system. Perhaps edge systems will still be automated but hub systems will have a human in the loop.

3.2 Experimental Predictions

Some of the tasks listed above require generalization, which is out of scope of this document. We predict (although we have not yet considered the matter in sufficient detail) that all other tasks can be solved via the system described above, if sufficient care is taken in the design of the human or hardcoded demonstrations.

4 Conclusion

We believe that this work provides the beginnings of a solid basis for building safety-critical systems that make significant use of reinforcement learning. Significant challenges remain in the development of such a system for non-trivial tasks, and much future work is needed in order to allow this system to be used in any actual safety-critical domain.

4.1 Future Work

An important extension of this work would be extending it to large domains and continuous control problems, where a neural network approximator will be required to allow the system to function efficiently. We believe that this is a reasonably straightforward system to build, given the specifications and results above, and the current state of the art in machine learning. Unfortunately, without perfect mechanistic interpretability for neural networks, a system that incorporates a neural network approximator will have no worst-case safety guarantees. Still, this seems like an important area to explore in toy domains.

Another important extension of this work would be studying the potential interactions of multiple agents using this algorithm. Ideally we would study this through the framework of game theory (both theoretical and experimental). Questions that seem important are:

1. What are the observed dynamics of players of different capabilities using this algorithm in zero-sum games of perfect information, such as chess?

2. What are the observed dynamics of players of different capabilities using this algorithm in zero-sum games of imperfect information, such as poker?

3. What are the observed dynamics of players of different capabilities using this algorithm in non-zero-sum games of imperfect information, such as no-press Diplomacy?

4. What are the observed dynamics of players of different capabilities using this algorithm in non-zero-sum games of imperfect information that include significant linguistic components, such as press Diplomacy?

5. What are the observed dynamics of games between players of similar capabilities using this algorithm and other algorithms? I.e., is this

algorithm truly competitive with strong RL baselines that have not been optimized for safety? If not, how competitive is it? There is a free parameter in the PAC bounds that is the probability bound we require of the "probably" part; how high must this probability of failure be before the algorithm is competitive with strong RL baselines?

5 Feedback from David Dalrymple

I sent a draft of this document to various people, including David Dalrymple. He had the following cogent criticisms of the approach, which seemed worth including in this document.

1. The state-space ontology being a finite-dimensional vector space seems to me like a crippling limitation for applicability to real-world scenarios with e.g. an unknown number of animals. I suggest SDCPN (https://fse.studenttheses.ub.rug.nl/10947/1/Scriptie.pdf) as a baseline state-space ontology.

2. The assumptions that a human can demonstrate effective behaviour and anticipate unknown constraints seem too strong to me. I have much more hope about interactive preference elicitation with respect to simulated trajectories, cf. generative active task elicitation (https://arxiv.org/pdf/2310.11589.p

3. Rather than just the Lagrangian dual, I would prefer a probabilistic reach-avoid guarantee (https://arxiv.org/pdf/2210.05308.pdf). I think the target risk should be somewhere between 0.01 and 1 failure per mean operating millennium, as it is in nuclear power safety (empirically, there are about 0.3 fatal accidents per operating millennium in nuclear power stations worldwide), rather than proving "no failures ever" (which is really begging to be Goodharted somehow, because the true version is likely impossible).

These criticisms all seem incredibly well-founded and spot-on, and we will incorporate them into future versions of this work. However, we take issue with putting the target risk at the same level as nuclear power safety, since a single nuclear meltdown could only ever kill perhaps 1 million people, at least at current population densities. (https://chat.openai.com/share/b5b76f26-e353-4b29-89c8-0b902d6329ec) We would rather strive for a level of safety commensurate with planetary-scale risk, perhaps between 1e-4 and 1e-2 failures per mean operating millennium. However, this seems like a pretty minor

quibble, and we think the overall point of not trying to achieve perfection is a solid one.