Risk-Sensitive Decision Making in the Presence of Model Uncertainty

Anonymous Authors[1]

Abstract

This is the abstract.

1. Introduction

Autonomous decision making and planning require models of the environment. However, in many recent applications of robotics, exact models of the environment are rarely available. For example, in human-robot interaction scenarios, the autonomous agent must reason about the impact of its actions on the behavior of other humans. However, we cannot derive an exact model for human behavior, and instead must learn a model from data, a process which introduces uncertainty over the model [JH: vague?]. Furthermore, it is often the case that multiple distinct models are possible: for example, in the context of autonomous driving, a distracted driver may behave according to a different model than an attentive driver. Thus, planning in the presence of model ambiguity is an important challenge in many real world settings.

The Bayesian approach to dealing with this model ambiguity is to maintain a *belief distribution* over possible models. This belief distribution can easily be updated as the agent observes the true environment, simply employing Bayes' rule. However, the problem of acting optimally given a certain belief over models is a challenging one. A natural approach might be to optimize the expected performance if the model was sampled from the belief. In this case, however, if the true underlying model has high expected cost but a low belief probability, catastrophic failure may occur. We wish to enable an autonomous agent to disambiguate between possible models of the environment, while simultaneously acting robustly with respect to its subjective belief over these models. In this work, we present Robust Adaptive Monte Carlo Planning (RAMCP), a risk-sensitive planning algorithm based on Bayes-Adaptive Monte Carlo Planning (Guez et al., 2013) that allows safe model disambiguation while ensuring robustness to adversarially perturbed beliefs.

Related Work The notion of robust policy selection subject to model uncertainty in Markov Decision Processes is captured by the *Robust MDP* framework, and there is a substantial body of literature on this problem. Typically, these problems are concerned with uncertainty in the the transition probabilities (Nilim & El Ghaoui, 2005). More specifically, work in this area aims to learn policies that maximize expected reward subject to the worst-case disturbances in some uncertainty set, with no consideration for the associated probabilities of these disturbances. In the online system identification case, where it is typically undesirable to set the belief in any one model to zero, this would result in always acting with respect to the worst-case model. This framework typically leads to extremely conservative policies, and computing robust policies for general uncertainty sets is NP-hard (Bagnell et al., 2001). Moreover even for rectangular uncertainty sets, computing the policy for the static case (in which the perturbed model is the same every time the state-action pair (s, a) is visited) is NP-hard (Iyengar, 2005).

While Robust MDPs consider the worst-case expected reward, the *Risk-Sensitive MDP* framework instead considers replacing the expected reward of standard MDPs with a risk measure (Howard & Matheson, 1972). This allows a naturally tune-able notion of conservatism, and maximizing some risk measures has been shown to be equivalent to maximizing the expected reward under the worst-case perturbations, where the perturbations have a multiplicative budget (Chow et al., 2015). However, both the risk-sensitive and robust MDP frameworks do not consider that the uncertainty over dynamics models may be reduced as the agent interacts with the environment, still leading to unnecessary conservative policies that do not consider the value of information gathering actions.

The field of adaptive control generally aims to estimate parameters in a model while safely controlling the system (Aström & Wittenmark, 2013). However, guarantees are typically only available for specific types of systems (e.g. linear with Gaussian noise) and estimation methods (e.g. linear parametric models), and robustness is typically in the worst-case sense (Ioannou & Sun, 1996). Again, these approaches typically do not consider the value of information in their action selection. Blackmore & Williams (2006) develop an algorithm for model discrimination with constraints on task

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

satisfaction for aircraft, but they do not simultaneously consider reward maximization. This exploration-exploitation trade-off was addressed in Bai et al. (2013), in which the authors model the problem as a POMDP and generates offline plans, but the work does not consider robustness.

Statement of Contributions In this work we present, RAMCP a risk-sensitive algorithm for control of discrete MDPs for which we are uncertain of the underlying model. For each model in this set, the transition dynamics are known. We present theoretical results on the asymptotic convergence of this algorithm. This approach safely disambiguates between models while simultaneously maximizing performance. We demonstrate this algorithm on a simple bandit problem, in which we can compare to exact solutions, and on a challenging autonomous driving problem.

Organization This paper is structured as follows. In section 2 we briefly review material relevant to risk-sensitive planning and learning, and we formally present the problem of safe planning under model uncertainty. In section 3, we present the RAMCP algorithm, and state theoretical results on its convergence. In section 4, we evaluate its performance on a bandit problem and a more challenging autonomous driving example. Finally, in section 5, we offer concluding remarks and discuss avenues for future work.

2. Problem Formulation

[AS: make all risk definitions consistent with rewards rather than costs] We wish to control an agent in a system defined by the MDP $M = (S, A, T, R, \gamma)$, where S is the state space, A is the action space, $T(s'|s, a)$ is the transition function, $R(s, a, s')$ is the stage-wise reward function, and γ is the discount factor. We assume that the exact transition dynamics T depend on a parameter θ, and denote this dependence as T_θ. Let Θ represent the random variable for which θ are realizations. We further assume that we have determined a discrete distribution over this parameter, $b_{\text{prior}}(i) = p(\Theta = \theta_i),\ i = 1, \dots, N$ which represents our belief over the true transition dynamics. [AS: might want to add a line justifying the discrete distribution over models here]

While acting within this MDP, an optimal policy will aim to simultaneously *explore*, to try to disambiguate between possible underlying models, and *exploit*, to decrease cumulative cost. As observed state transitions enable computing a posterior distribution over environments, an optimal policy will necessarily be history dependent. Writing the history in an environment at time t as $h_t = (s_0, a_0, \dots, s_t)$ and given a prior distribution b_{prior}, we can define optimal behavior in the Bayesian setting. A history-dependent policy is said to be Bayes-optimal with respect to the prior over models if its associated value function $V^*(s) = \sup_\pi V(s, \pi)$ (Martin, 1967). [JH: should double check this def] [AS: should this be $V(h)$?] Duff & Barto (2002) introduced the *Bayes-Adaptive Markov Decision Process* (BAMDP) as a framework for planning optimal actions under model uncertainty. The framework considers an MDP over hyperstates which augment the original state space of the problem with a continuous belief space over model parameters. The optimal policy in this framework is the Bayes-optimal policy (Martin, 1967). In general, optimizing these policies is computationally difficult. Offline global value approximation approaches, based on typical offline POMDP solution methods, scale poorly to large state spaces (Ghavamzadeh et al., 2015). Online approaches (Wang et al., 2005; Guez et al., 2013; Chen et al., 2016) use tree search with heuristics to either simplify the problem or guide the search.

Most previous work in solving BAMDPs has focused on optimizing performance in expectation, and thus does not offer the notion of safety offered by the robust MDP formulation. Directly applying robust MDP tools to the Bayesian setting is overly conservative, because their worst-case analysis does not consider the probability associated with each model. Moreover, because it is typically not desirable to collapse the belief over any model to zero in practice, a robust formulation would likely always act with respect to the worst performing model. In short, the issue arises because the safety guarantees of robust MDPs are based on *sets* of models, but the Bayesian approach instead keeps *distributions* over models. Tools from risk theory can be used to achieve tuneable, distribution-dependent conservatism. A key concept in risk theory is that of a *coherent risk metric*. Given a random cost variable Z, a *risk metric* is a function $\rho(Z)$ that maps the uncertain cost to a real scalar, which encodes a preference model over uncertain outcomes where lower values of $\rho(Z)$ are preferred. Coherent risk metrics are defined as follows:

Definition 1 (Coherent Risk Metrics)**.** *Let* $(\Omega, \mathcal{F}, \mathbb{P})$ *be a probability space and let* $\mathcal{Z}$ *be the space of random variables on* Ω*. A coherent risk metric (CRM) is a mapping* $\rho : \mathcal{Z} \to \mathbb{R}$ *that obeys the following four axioms. For all* $Z, Z' \in \mathcal{Z}$*:*

A1. Monotonicity: $Z \le Z' \implies \rho(Z) \le \rho(Z')$

A2. Translation invariance: $\forall a \in \mathbb{R}, \rho(Z + a) = \rho(Z) + a$

A3. Positive homogeneity: $\forall \lambda \ge 0, \rho(\lambda Z) = \lambda \rho(Z)$

A4. Subadditivity: $\rho(Z + Z') \le \rho(Z) + \rho(Z')$

Note that this definition is stated in terms of costs, as opposed to rewards. These axioms, originally proposed in (Artzner et al., 1999), ensure a notion of rationality in risk assessments. We refer the reader to (Majumdar & Pavone, 2017) for a more thorough discussion of why coherent risk metrics are a useful tool in decision making. While expectation and worst-case are two possible coherent risk metric, the set of coherent risk measures is a rich class of metrics including the Conditional Value at Risk (CVaR) metric popular in mathematical finance (Majumdar & Pavone, 2017). Recently, the connections between risk sensitivity and robustness have been examined. Chow et al. (2015) showed that optimizing the CVaR metric on the total cost in an MDP can be thought of as robust planning while allowing multiplicative disturbances δ_i to the transition probabilities at every step, with the constraint that the product of the disturbances over the planning period is bounded.

We aim to compute a history dependent policy $\pi(h_t)$ which is optimal according to a coherent risk metric over model uncertainty. To make this objective more concrete, let $\xi = (s_0, a_0, \ldots, s_{T-1}, a_{T-1}, s_T)$ be particular trajectory realization, and let Ξ represent the corresponding random variable. Notice that the probability of a given trajectory depends on both the choice of policy π and the transition dynamics of the MDP T_θ. Let $q^\pi_\theta(\xi) = p(\Xi = \xi | \Theta = \theta_i; \pi)$ represent this conditional distribution over trajectories. The cumulative reward of a given trajectory $J(\xi)$ can be calculated by summing the stage-wise rewards:

$$J(\xi) = \sum_{t=0}^{T-1} R(s_t, a_t, s_{t+1}) \tag{1}$$

Note that the total cost J is a random variable whose randomness stems from both the uncertainty over model realizations, as well as potential stochasticity in dynamics induced by the policy in any given model realization.

In this work, we focus our attention to risk-sensitivity to the randomness from model uncertainty only. Concretely, we can write the objective as

$$\pi^* = \arg\max_\pi \quad \rho\left(\mathbb{E}_{\Xi \sim q^\pi_\theta}\left[J(\xi)\right]\right). \tag{2}$$

In a similar fashion, Robust MDPs consider the worst-case model realization, but only consider the expectation over stochasticity present in a given MDP. This work can be thought of as a natural extension, replacing the "worst-case over a set of models" analysis with one that is "risk-sensitive to a distribution over models". [JH: also need to argue about the reasoning for this problem formulation]

3. Robust Adaptive Monte Carlo Planning

3.1. Reformulation as a Zero-Sum, Two Player Game

Several recent works have viewed risk-sensitive policy optimization as a two-player game. In Robust Adversarial Reinforcement Learning (Pinto et al., 2017), the CVaR metric is used to give philosophical motivation for their proposed optimization scheme, which trains the agent while simultaneously training an adversary which can apply disturbances to hinder the performance of the agent. Chen & Bowling (2012) consider the k-of-N measures, an approximation of the CVaR metric for continuous distributions, and mathematically formulate the optimization of such a metric as a two-player zero-sum game. In this work, we derive a similar game-theoretic formulation for the general class of coherent risk measures on discrete distributions.

Our reformulation of the objective stems from a universal representation theorem which all coherent risk metrics satisfy.

Theorem 1 (Representation Theorem for Coherent Risk Metrics (Artzner et al., 1999))**.** *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, where Ω is a finite set with cardinality $|\Omega|$, $\mathcal{F}$ is a $\sigma-$algebra over subsets (i.e., $\mathcal{F} = 2^\Omega$), probabilities are assigned according to $\mathbb{P} = (p(1), \ldots, p(|\Omega|))$, and $\mathcal{Z}$ is the space of random variables on Ω. Denote by $\mathcal{C}$ the set of valid probability densities:*

$$\mathcal{C} := \left\{ \zeta \in \mathbb{R}^{|\Omega|} \mid \sum_{i=1}^{|\Omega|} p(i)\zeta(i) = 1, \zeta \geq 0 \right\} \tag{3}$$

Define $q_\zeta \in \mathbb{R}^{|\Omega|}$ where $q_\zeta(i) = p(i)\zeta(i), i = 1, \ldots, |\Omega|$. A risk metric $\rho : \mathcal{Z} \to \mathbb{R}$ with respect to the space $(\Omega, \mathcal{F}, \mathbb{P})$ is a CRM if and only if there exists a compact convex set $\mathcal{B} \subset \mathcal{C}$ such that for any $Z \in \mathcal{Z}$:

$$\rho(Z) = \max_{\zeta \in \mathcal{B}} \mathbb{E}_{q_\zeta}[Z] = \max_{\zeta \in \mathcal{B}} \sum_{i=1}^{|\Omega|} p(i)\zeta(i)Z(i). \tag{4}$$

Again, note that the above is stated in terms of cost as opposed to reward. This theorem offers an interpretation of CRMs as a worst-case expectation over a set of densities $\mathcal{B}$, often referred to as the *risk envelope*. In this work we focus on *polytopic risk metrics*, for which the envelope $\mathcal{B}$ is a polytope $\mathcal{P}$ (Chow & Pavone, 2014; Majumdar et al., 2017). For this class of metrics, the constraints on the maximization in equation 4 become linear in the optimization variable ζ, and thus solving for the value of the risk metric becomes a tractable linear programming problem. Polytopic risk metrics constitute a broad class of risk metrics, encompassing risk neutrality, worst-case assessments, as well as the CVaR metric often used in financial applications.

Through the representation theorem for coherent risk metrics (equation 4), we can understand equation 2 as applying an adversarial reweighting ζ to the distribution over models b_{prior}. Let $b_{\text{adv}}(i) = b_{\text{prior}}(i)\zeta(i),\ i = 1, \ldots, N$ represent this reweighted distribution. Thus, the optimization problem we wish to solve may be written

$$\pi^* = \arg\max_\pi \quad \min_{\zeta \in \mathcal{B}} \mathbb{E}_{\Theta \sim b_{\text{adv}}}\left[\mathbb{E}_{\Xi \sim q^\pi_\theta}\left[J(\xi)\right]\right]. \tag{5}$$

Note that this takes the form of a two player zero-sum game between the agent (the maximizer) and an adversary (the minimizer). One play of this game corresponds to the following three step sequence:

1. The adversary chooses b_{adv} from the risk envelope.
2. Chance chooses $\theta \sim b_{\text{adv}}$.
3. The agent acts according to its strategy, or policy, $\pi(h)$ in the MDP with dynamics T_θ.

The adversary's strategy is how to pick b_{adv} while the agent's strategy is $\pi(h)$. Optimality in this scenario means coming up with b^*_{adv} and $\pi^*(h)$ such that the adversary picks an altered *adversarial distribution* b_{adv} over models from the risk envelope, chance picks a model from the altered distribution, and the agent picks a policy to execute in this model without knowing which model was chosen. Determining the optimal agent policy in this setting requires also computing the optimal adversarial distribution. Optimality thus implies that this game is in a Nash equilibrium.

3.2. Algorithm Outline

Having formulated our problem statement as a two-player zero-sum game with imperfect information, we leverage algorithms developed in computational game theory to efficiently compute a Nash equilibrium. Chen & Bowling (2012) propose using Counterfactual Regret Minimization (CFR) (Zinkevich et al., 2008), an algorithm that has been successfully used to compute optimal strategies for large-scale games such as no-limit poker. However, this algorithm requires traversing the full game tree at each iteration, which becomes intractable when considering history-dependent policies in stochastic MDPs. Fictitious Play (Brown, 1949) is another method for computing Nash equilibria, in which players repeatedly play a game and update their strategies toward the best-response to the average strategy of their opponents. Generalized weakened fictitious play (GWFP) (Leslie & Collins, 2006) allows approximate best-response computation but maintains convergence bounds, and thus has worked well for large-scale extensive form games (Heinrich et al., 2015). GWFP converges to Nash equilibria in several classes of games, including two-player zero-sum games like that presented in problem (5).

Algorithm 1 Robust Adaptive Monte Carlo Planning

Require: Hyperparameters $K, \eta, d_{\max}, \kappa$
Require: Rollout policy π_{ro}

```
function SEARCH($s_0$, $b_{\text{prior}}$)
    $\bar{R}(i) \leftarrow 0$ for all $i = 1, \ldots, N$
    $N_{\text{iter}} \leftarrow 0$
    $\bar{b}_{\text{adv}} \leftarrow b_{\text{prior}}$
    while within computational budget do
        for $k = 1$ to $K$ do
            for $i = 1$ to $N$ do
                $w \leftarrow N\bar{b}_{\text{adv}}(i)$
                $R \leftarrow$ SIMULATE($s_0$, $\theta_i$, 0, $w$)
                $N(\theta_i) \leftarrow N(\theta_i) + 1$
                $\bar{R}(i) \leftarrow \frac{R - \bar{R}(i)}{N(\theta_i)}$
            end for
        end for
        $N_{\text{iter}} \leftarrow N_{\text{iter}} + 1$
        $b^*_{\text{adv}} \leftarrow$ solution to linear program (6)
        $\bar{b}_{\text{adv}} \leftarrow \bar{b}_{\text{adv}} + (b^*_{\text{adv}} - \bar{b}_{\text{adv}})/N_{\text{iter}}$
    end while
    return $\pi_{\text{tree}}$
end function
```

The key challenge in applying GWFP to this problem is determining how each player computes its best response to the average opponent policy. For the adversary, this is relatively straightforward thanks to the linear structure of polytopic risk measures. If the adversary maintains monte-carlo estimates of the performance of the agent under each discrete model θ_i in the vector $\bar{R} \in \mathbb{R}^N$, then computing the best-response selection of the adversarial belief b_{adv} becomes the linear program:

$$\begin{aligned} \text{minimize}_b \quad & \bar{R}^T b \\ \text{subject to} \quad & b(i) = \zeta(i) b_{\text{prior}}(i) \;\; \forall\, i = 1, \ldots, N \\ & \zeta \in \mathcal{P} \end{aligned} \tag{6}$$

where $\mathcal{P}$ is the polytopic risk envelope. Note that for stochastic environments, the estimate of average agent performances $\bar{R}$ will be an approximation for a finite number of samples in the running average, and thus the solution of this LP may be incorrect. However, GWFP allows ϵ_k errors in the best response at iteration k of the self-play, as long as $\epsilon_k \to 0$ as $k \to \infty$, which we can ensure by sampling performance of the agent on each model at each iteration of self-play.

The agent, on the other hand, must compute as its best response an optimal policy $\pi(h)$ under the average adversarial belief over models $\bar{b}_{adv}$. Notice that this problem of choosing a history dependent policy that is optimal for a given belief over models is identical to that of solving a BAMDP. This is a challenging task, and computing the optimal solution exactly at every iteration would be intractable. Instead, we can make use of two simplifying factors. The first is that GWFP allows suboptimality, meaning we do not need to exactly solve the BAMDP at every step. Second is that the best-response to the average adversarial belief changes smoothly across iterations of self-play due to smooth changes to the adversary's policy. This suggests if the solution algorithm for the BAMDP can transfer progress across iterations of self play, only a finite amount of progress toward the optimal solution may be necessary at every step. This insight was used in the development of Fictitious Self-Play (Heinrich et al., 2015), which applies machine learning methods to GWFP, and has played a role in developing super-human artificial intelligence for Go without expert knowledge (Silver et al., 2017).

Monte Carlo Tree Search (MCTS) planning algorithms are well suited to self-play (Heinrich & Silver, 2015), and thus we turn to the MCTS based Bayes-Adaptive Monte-Carlo Planning (BAMCP) algorithm (Guez et al., 2013). BAMCP operates by sampling trajectories, and building a search tree over these trajectories. At the root, a specific model θ_i is sampled from the distribution over models, in our case from $\bar{b}_{\text{adv}}$. As the tree is constructed, each node maintains an estimate of the average reward gathered after visiting that node. By choosing actions based on the Upper Confidence Bound heuristic, these value estimates converge toward the optimal values. The error in these estimates converges as $\mathcal{O}(\log(n(h_t))/n(h_t))$.

Note that the adversary's strategy updates require unbiased estimates of the performance of the average agent strategy on each model $\bar{R}_i$, while the standard BAMCP algorithm requires sampling θ_i from the adversary's current strategy $\bar{b}_{\text{adv}}$, which may put zero weight on some models. By introducing an weighting scheme to the MCTS tree updates, we can manage these competing interests. We sample under each model deterministically, essentially drawing θ_i from a discrete uniform distribution over the set of models. However, we replace the standard MCTS update with an weighted update:

$$\begin{aligned} N(ha) &\leftarrow N(ha) + 1 \\ Q(ha) &\leftarrow Q(ha) + (R - Q(ha))/N(ha) \\ &\downarrow \\ W(ha) &\leftarrow W(ha) + w \\ Q(ha) &\leftarrow Q(ha) + w(R - Q(ha))/W(ha) \end{aligned}$$

where $w = \bar{b}_{\text{adv}}(i)/p_{\text{unif}}(i) = N\bar{b}_{\text{adv}}(i)$ when the update corresponds to a simulation under model θ_i. In this way, the tree is updated as if θ_i had been sampled from $\bar{b}_{\text{adv}}$, but we are able to ensure that the adversary gets an equal number of samples of the agents performance under each model. We can also interpret this formulation as a re-framed game, where the role of the adversary is not to change the model distribution, but rather to determine how much a simulation from each model can impact the agent's tree. [JH: could state convergence of this weighted MCTS framework as a theorem]

The linear programming scheme can be played against the modified BAMCP algorithm to create a self-play system to solve the desired optimization problem (5). Algorithm 1 details the overall procedure, and the algorithmic details (other functions used) are available in the supplementary material. To compute a robust plan from state s_0 and belief over model parameters b_{prior}, the agent calls the SEARCH function. For each model θ_i, the algorithm computes the weighting w_i as a function of the current adversarial belief, which corresponds to the adversary's play, then calls SIMULATE which corresponds to a combination of the agent's plays and strategy updates. Finally, the procedure updates the adversary's best response according to the LP (6). The remaining functions are identical to the BAMCP algorithm, with the exception of the SELECT function, which is adapted from the Smooth UCT algorithm to mix the average and approximate-best-response strategies.

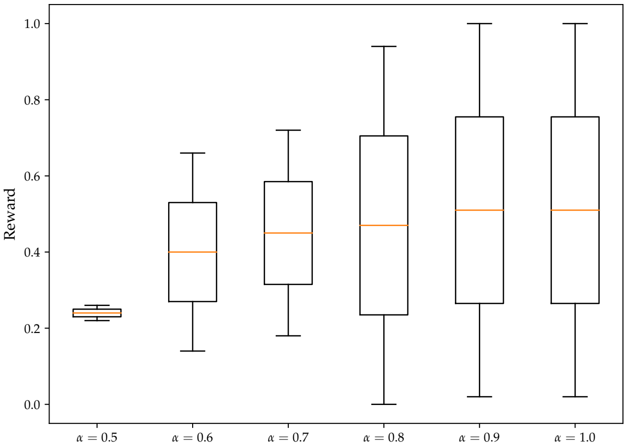

Figure 1. [JH: axis labels should be fixed to match fig 2, should be x label] Box plot showing performance of the RAMCP algorithm on a simple bandit problem for varying CVaR quantiles (α). For each quantile, 500 trials were performed. In this plot, the inner boxes denote the 25% – 75% range, and the full lines extending from the boxes denote the full 0% – 100% range.

[JH: Proof of convergence/error bounds and associated discussion go here]

4. Experiments

We evaluate this algorithm on two domains, a small illustrative bandit problem, and a more complex sequential decision making problem.

4.1. Multi-Armed Bandit

First, we consider a simple multi-armed bandit scenario to illustrate several properties of the RAMCP algorithm. Additionally, because we can compute true risk-sensitive optimal policies, we can examine the convergence rate in practice. In this problem, there are two possible rewards, $R_1 = 0$ and $R_2 = 1$. There are four possible actions, $\{1, 2, 3, 4\}$.

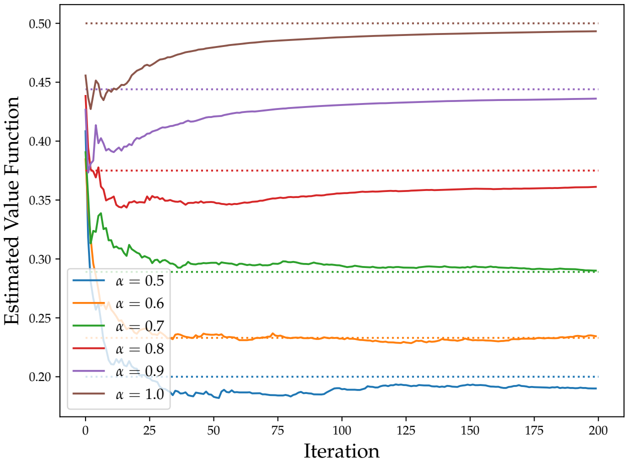

Figure 2. Convergence of the RAMCP algorithm for different CVaR quantiles (α). The doted line denotes the true optimal value function for each quantile. Rapid convergence can be observed for most values of α.

Table 1. Bandit Model. Each cell lists $P(R_1 \mid a_i)$, where a_i is the column heading, under the associated model θ_i (row heading).

	a_1	a_2	a_3	a_4
θ_1	0.20	0.18	0.14	0.00
θ_2	0.20	0.50	0.76	1.00

There are two possible models, each with different transition probabilities, and these transition probabilities are known in the planning problem. Table 1 shows the probability of receiving R_1 given each action. Thus, for each model θ_i, the probability of receiving reward R_2 for action a_i is one minus the table entry.

In this scenario, we consider ambiguity over two models, θ_1 and θ_2. This bandit is designed such that action 1 has high expected cost consistently across models. Subsequent action choices offer slightly higher expected cost under θ_1, but lower expected cost under θ_2. Action 4 is deterministically high cost under θ_1, while being zero cost under θ_2. Note that this structure means that an adversarial choice of belief over models would always assign as much weight as the risk envelope allows to θ_1. Therefore, the risk of each action can be calculated directly for various risk metrics. Furthermore, the agent's policy should switch from action 4 towards action 1 as the risk-sensitivity is increased from neutral ($\text{CVaR}_{1.0}$) to worst-case ($\lim_{\alpha \to 0} \text{CVaR}_\alpha$).

Figure 1 shows a box plot of the performance for the CVaR_α risk metric for 500 iterations of RAMCP, under various values of the CVaR quantile, α. In this plot, the inner boxes denote the 25% – 75% range, and the full lines extending from the boxes denote the full 0% – 100% range. Note that the as the CVaR quantile decreases, the expected reward and the variance decrease. Note that with $\alpha = 1.0$, the algorithm essentially simplifies to the original BAMCP algorithm, with the Smooth UCT and weighted tree update being the only differences. Figure 2 shows convergence of the RAMCP algorithm. The dotted line for each value of α denotes the true optimal value for that CVaR quantile. It can be seen that in most cases, approximate convergence is reached quickly, in approximately 25 iterations. This is particularly true for lower values of α, which is likely more useful than values of α close to one, which is essentially risk-neutral.

[JH: need to add experiment that shows exploration]

4.2. Autonomous Driving with Driver Model Ambiguity

5. Discussion and Conclusions

5.1. Conclusions

Planning with uncertain models is a pervasive problem in robotics, from human robot interactions to planning with models learned from data. There have traditionally been two approaches to dealing with this uncertainty. The Robust MDP approach prioritizes safety by planning for the worst-case among a set of likely models, while the BAMDP formulation expresses uncertainty over models through a belief distribution, and factors in exploration for model disambiguation into the planning to achieve adaptive policies that are optimal in expectation. In this work, we have presented an approach to uses coherent risk theory to enable safe planning in BAMDPs, which had previously not been addressed. The proposed algorithm RAMCP combines two insights: (1) a coherent risk metric on model uncertainty induces a two-player zero-sum game, and (2) the BAMCP algorithm can be adapted to approach a Nash equilibrium in this game by using tools from algorithmic game theory. [JH: should we have stronger claims about contributions here?] We have evaluated the algorithm on two small examples, which have shown promising results.

5.2. Future Extensions

[JH: Experiments to do still: all of the car example, performance curves for different hyperparams, a bandit example that includes exploration, exploitability experiments]

There are several interesting avenues for future work. First, it may be the case that this adversarial training scheme can be extended to achieve risk sensitivity to environment stochasticity as well. Also, it may be possible to extend this work to continuous state and action spaces by replacing the tree search agent with an RL agent optimizing a recurrent, history dependent policy. Finally, this work could be extended to continuous beliefs over model parameters. This would complicate the adversarial belief computation, but a combination of Gaussian Process regression to maintain the $\bar{J}(\theta)$ estimates and particle filter style sampling of the continuous belief could be used to maintain the LP structure of the adversarial optimization. In all, these extensions would serve to bring safe and adaptive planning under model uncertainty to large-scale problems in robotics.

References

Artzner, Philippe, Delbaen, Freddy, Eber, Jean-Marc, and Heath, David. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

Aström, Karl J and Wittenmark, Björn. *Adaptive control*. Courier Corporation, 2013.

Bagnell, J Andrew, Ng, Andrew Y, and Schneider, Jeff G. Solving uncertain markov decision processes. 2001.

Bai, Haoyu, Hsu, David, and Lee, Wee Sun. Planning how to learn. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 2853–2859. IEEE, 2013.

Blackmore, Lars and Williams, Brian. Finite horizon control design for optimal discrimination between several models. In *Decision and Control, 2006 45th IEEE Conference on*, pp. 1147–1152. IEEE, 2006.

Brown, George W. Some notes on computation of games solutions. Technical report, RAND CORP SANTA MONICA CA, 1949.

Chen, Katherine and Bowling, Michael. Tractable objectives for robust policy optimization. In *Advances in Neural Information Processing Systems*, pp. 2069–2077, 2012.

Chen, Min, Frazzoli, Emilio, Hsu, David, and Lee, Wee Sun. Pomdp-lite for robust robot planning under uncertainty. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 5427–5433. IEEE, 2016.

Chow, Yin-Lam and Pavone, Marco. A framework for time-consistent, risk-averse model predictive control: Theory and algorithms. In *American Control Conference (ACC), 2014*, pp. 4204–4211. IEEE, 2014.

Chow, Yinlam, Tamar, Aviv, Mannor, Shie, and Pavone, Marco. Risk-sensitive and robust decision-making: a cvar optimization approach. In *Advances in Neural Information Processing Systems*, pp. 1522–1530, 2015.

Duff, Michael O'Gordon and Barto, Andrew. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts at Amherst, 2002.

Ghavamzadeh, Mohammad, Mannor, Shie, Pineau, Joelle, Tamar, Aviv, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

Guez, Arthur, Silver, David, and Dayan, Peter. Scalable and efficient bayes-adaptive reinforcement learning based on monte-carlo tree search. *Journal of Artificial Intelligence Research*, 48:841–883, 2013.

Heinrich, Johannes and Silver, David. Smooth uct search in computer poker. In *IJCAI*, pp. 554–560, 2015.

Heinrich, Johannes, Lanctot, Marc, and Silver, David. Fictitious self-play in extensive-form games. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 805–813, 2015.

Howard, Ronald A and Matheson, James E. Risk-sensitive markov decision processes. *Management science*, 18(7): 356–369, 1972.

Ioannou, Petros A and Sun, Jing. *Robust adaptive control*, volume 1. PTR Prentice-Hall Upper Saddle River, NJ, 1996.

Iyengar, Garud N. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

Leslie, David S and Collins, Edmund J. Generalised weakened fictitious play. *Games and Economic Behavior*, 56 (2):285–298, 2006.

Majumdar, Anirudha and Pavone, Marco. How should a robot assess risk? towards an axiomatic theory of risk in robotics. *arXiv preprint arXiv:1710.11040*, 2017.

Majumdar, Anirudha, Singh, Sumeet, Mandlekar, Ajay, and Pavone, Marco. Risk-sensitive inverse reinforcement learning via coherent risk models. In *Robotics: Science and Systems*, 2017.

Martin, James John. *Bayesian decision problems and Markov chains*. Wiley, 1967.

Nilim, Arnab and El Ghaoui, Laurent. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Pinto, Lerrel, Davidson, James, Sukthankar, Rahul, and Gupta, Abhinav. Robust adversarial reinforcement learning. 2017. URL https://arxiv.org/abs/1703.02702.

Silver, David, Schrittwieser, Julian, Simonyan, Karen, Antonoglou, Ioannis, Huang, Aja, Guez, Arthur, Hubert, Thomas, Baker, Lucas, Lai, Matthew, Bolton, Adrian, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

Wang, Tao, Lizotte, Daniel, Bowling, Michael, and Schuurmans, Dale. Bayesian sparse sampling for on-line reward optimization. In *Proceedings of the 22nd international conference on Machine learning*, pp. 956–963. ACM, 2005.

Zinkevich, Martin, Johanson, Michael, Bowling, Michael, and Piccione, Carmelo. Regret minimization in games with incomplete information. In *Advances in neural information processing systems*, pp. 1729–1736, 2008.

A. Algorithmic Details of RAMCP

Algorithm 2 RAMCP Functions

```
function Rollout($h$, $\theta$, $d$)
    if $d > d_{\max}$ or $h$ is terminal then
        return $0$
    end if
    $a \sim \pi_{\mathrm{ro}}(\cdot|h)$
    $s' \sim T_\theta(\cdot|s, a)$
    $c \leftarrow C(s, a, s')$
    return $c + \gamma$ Rollout($has'$, $\theta$, $d + 1$)
end function

function UpdateTree($h$, $a$, $J$)
    $N(h) \leftarrow N(h) + 1$
    $N(ha) \leftarrow N(ha) + 1$
    $Q(ha) \leftarrow Q(ha) + \frac{J - Q(ha)}{N(ha)}$
end function
function Simulate($h$, $\theta$, d)
    if $d > d_{\max}$ or $h$ is terminal then
        return $0$
    end if
    if $N(h) = 0$ then
        for all $a \in A$ do
            $N(ha) \leftarrow 0$, $Q(ha) \leftarrow 0$
        end for
        $s \leftarrow$ end of $h$
        $a \sim \pi_{\mathrm{ro}}(\cdot|h)$
        $s' \sim T_\theta(\cdot|s, a)$
        $c \leftarrow C(s, a, s')$
        $J \leftarrow c + \gamma$ Rollout($has'$, $\theta$, $d + 1$)
        UpdateTree($h$, $a$, $J$)
        return $J$
    end if
    $a \leftarrow$ Select($h$)
    $s' \leftarrow T_\theta(\cdot|s, a)$
    $c \leftarrow C(s, a, s')$
    $J \leftarrow c + \gamma$ Simulate($has'$, $\theta$, $d + 1$)
    UpdateTree($h$, $a$, $J$)
    return $J$
end function

function Select($h$)
    $z \sim \mathrm{Uniform}(0, 1)$
    if $z < \eta$ then
        return $\arg\min_a Q(ha) - \kappa\sqrt{\frac{\log N(h)}{N(ha)}}$
    else
        $\forall a \in A: \quad p(a) \leftarrow \frac{N(ha)}{N(h)}$
        return $a \sim p$
    end if
end function
```