Policy-Conditioned Uncertainty Sets for Robust Markov Decision Processes

Andrea Tirinzoni
Politecnico di Milano
andrea.tirinzoni@polimi.it

Xiangli Chen
Amazon Robotics
cxiangli@amazon.com

Marek Petrik
University of New Hampshire
mpetrik@cs.unh.edu

Brian D. Ziebart
University of Illinois at Chicago
bziebart@uic.edu

Abstract

What policy should be employed in a Markov decision process with uncertain parameters? Robust optimization's answer to this question is to use rectangular uncertainty sets, which independently reflect available knowledge about each state, and then to obtain a decision policy that maximizes the expected reward for the worst-case decision process parameters from these uncertainty sets. While this rectangularity is convenient computationally and leads to tractable solutions, it often produces policies that are too conservative in practice, and does not facilitate knowledge transfer between portions of the state space or across related decision processes. In this work, we propose non-rectangular uncertainty sets that bound marginal moments of state-action features defined over entire trajectories through a decision process. This enables generalization to different portions of the state space while retaining appropriate uncertainty of the decision process. We develop algorithms for solving the resulting robust decision problems, which reduce to finding an optimal policy for a mixture of decision processes, and demonstrate the benefits of our approach experimentally.

1 Introduction

Policies with high expected reward are often desired for uncertain decision processes with which little experience exists. Specifically, we consider the setting in which only a limited number of trajectories from a sub-optimal control policy through a decision process are available. Robust control approaches for this task [1, 2, 3] define uncertainty sets for the decision process based on the limited outcome samples and seek the policy that maximizes this expected reward for the worst possible choice of decision process parameters in these sets.

When the uncertainty sets relating to different decision process states are jointly constrained in seemingly natural ways, the robust control problem becomes NP-hard (e.g., [4]). To avoid these computationally intractable robust control problems, uncertainty sets have often been independently constructed for parameters associated with a particular state-action pair or particular state—s, a-rectangularity or s-rectangularity [5, 6, 3], respectively. Unfortunately, independently assuming the worst-case in every encountered state is often too conservative in practice to be useful [7].

Leveraging ideas from distributionally robust optimization [8, 9, 10], we construct *policy-conditioned marginal uncertainty sets* for robustly learning a decision policy that optimizes the reward given trajectory samples produced by a sub-optimal policy. State transition dynamics under our formulation are estimated based on two competing objectives. First, the estimated dynamics must (approximately)

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

match measured properties observed under the sub-optimal reference policy. Second, the estimated dynamics must be the worst case for the simultaneously-hypothesized optimal policy.

This formulation has three main benefits: (1) *Non-rectangularity:* Our uncertainty sets are defined by feature-based statistics of distributions over entire trajectories, enabling generalization across states; (2) *Off-policy robustness:* We define our performance objective using the desired control policy and the uncertainty set using the sub-optimal data generation policy; and (3) *Convex parameter optimization:* We avoid the nonconvex parameter optimization pitfalls of other nonrectangular formulations by shifting the main computational difficulties to parameterized prediction/control problems (which can be efficiently approximated). Together, these properties aid in addressing a number of existing concerns for robust control, including settings in which the state definition violates the Markov assumption [11] or the transition probabilities are derived from limited data sets [2, 7].

In the remainder of this paper, we review existing robust control methods and directed information theory concepts in Section 2. Using these concepts, we formulate the robust control task using feature-based marginal constraints in Section 3. We reformulate this problem and present algorithms for solving it using a combination of convex optimization and dynamic programming to optimize a non-Markovian mixed decision process optimal control problem that arises from the formulation. We evaluate our approach in Section 4 to demonstrate its comparative benefits over rectangular robust control methods. Lastly, we provide concluding thoughts and discuss future work in Section 5.

2 Background and Related Work

2.1 Robust control

The Markov Decision Process (MDP) with state set $\mathcal{S}$ and action set $\mathcal{A}$ povides a common formulation of discrete control problems. In the MDP, the transition probabilities are given by $\tau(s_{t+1} \mid s_t, a_t)$ and the reward is $R(s_t, a_t, s_{t+1})$. Though consideration is often restricted to deterministic Markovian policies, $\pi : \mathcal{S} \to \mathcal{A}$, the generalization to *randomized* Markovian policies $\Pi_M = \{\pi : \mathcal{S} \to \Delta_{\mathcal{A}}\}$ provides stochastic mappings from the current state to actions. Even more generally, we will consider non-Markovian, history-dependent, randomized policies $\Pi_H = \{\pi : \mathcal{S}^t \times \mathcal{A}^t \to \Delta_{\mathcal{A}}\}$ in this work.

The expected sum of rewards or return ρ of a policy π applied to an MDP with dynamics τ and reward function R is: $\rho_R(\pi, \tau) = \mathbb{E}_{\tau,\pi}[\sum_{t=1}^{T-1} R(S_t, A_t, S_{t+1})]$. For decision problems, the standard objective is to choose a policy that maximizes the expected sum of rewards: $\max_\pi \rho_R(\pi, \tau)$. Since a Markovian and deterministic policy always exists that maximizes this quantity, one with those characteristics is typically sought when solving this optimization problem by many well-known algorithms, such as value iteration or policy iteration [12].

Unfortunately, in many settings the dynamics τ are not entirely known. Control policies are needed that can perform well despite this uncertainty about the decision process. One option is to formally define the uncertainty as a set of possibilities and assume the worst case (Definiton 1).

Definition 1. *The robust control problem is to find a control policy $\pi \in \Pi$ that performs best for the worst-case choice of state transition dynamics, $\tau \in \Xi$:*

$$\max_{\pi \in \Pi} \min_{\tau \in \Xi} \rho(\pi, \tau) = \max_{\pi \in \Pi} \min_{\tau \in \Xi} \mathbb{E}_{\tau,\pi} \left[\sum_{t=1}^{T-1} R(S_t, A_t, S_{t+1}) \right]. \tag{1}$$

The specification of the uncertainty set(s), Ξ, has significant implications for the tractability of this problem. Robust MDPs [5] are typically used to represent uncertainty in transition probabilities and rewards in regular MDPs. When the state-transition probabilities for different states are jointly constrained in arbitrary ways, the robust control problem becomes NP-hard [4]. Two common forms of constraints that enable efficient solutions are *s,a*-rectangular and *s*-rectangular [3] constraint sets. This form arises when transition probabilities are not known precisely, but are known to be bounded in terms of an L_1 norm. A corresponding robust MDP has uncertain transition probabilities:

$$\Xi = \{\tau \ : \ \forall s, a \in \mathcal{S} \times \mathcal{A}, \ \|\tau(\cdot|s, a) - p(\cdot \mid s, a)\|_1 \leq c\}.$$

This is an s, a-rectangular set. It employs independent constraints for each state-action pair or state (s-rectangular set). A convenient way to model a robust MDP is to introduce a set of outcomes $\mathcal{B}$ to represent the uncertainty in transitions and rewards. The transition probabilities are then defined

as $p(s_{t+1} \mid s_t, a_t, b_t)$ and rewards become $r(s_t, a_t, b_t, s_{t+1})$, while $\xi(b_t|s_t, a_t)$ denotes the nature's policy, i.e., a distribution over outcomes.

The optimal value function $v^\star$ in a robust MDP with s-rectangular and s, a-rectangular uncertainty sets (and discount factor γ) satisfies the Bellman optimality equation for each $s \in \mathcal{S}$ as follows:

$$v^\star(s) = \max_{\pi \in \Pi} \min_{\xi \in \Xi} \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}} \pi(a|s)\xi(b|s,a)\Big(r(s,a,b,s') + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a,b)v^\star(s')\Big). \quad (2)$$

In our formulation, we consider state-action feature-based constraints over the marginals of state-action sequences to define our uncertainty sets. When the sum of rewards and the constraints are defined in terms of different policies, this naturally induces a "belief state" that is similar to the augmenting set of outcomes $\mathcal{B}$ previously described. In our case, this augmenting information tracks the relative significance of the policies for providing robustness based on the sum of rewards versus matching feature-based measurements from training trajectories.

2.2 Directed information theory for processes

We make extensive use of ideas and notation from directed information theory [13, 14, 15, 16, 17]. Under this theory, processes—the products of T conditional probabilities over a sequence of T variables—are treated as first-order objects. The causally conditioned probability distribution [18], $p(\mathbf{y}_{1:T}||\mathbf{x}_{1:T}) \triangleq \prod_{t=1}^{T} p(y_t|\mathbf{y}_{1:t-1}, \mathbf{x}_{1:t})$, illustrates the notation for this process of generating the sequence of $\mathbf{y}_{1:T}$ variables given the sequence of $\mathbf{x}_{1:T}$ variables. It differs from the conditional probability distribution, $p(\mathbf{y}_{1:T}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} p(y_t|\mathbf{x}_{1:T}, \mathbf{y}_{1:t-1})$, in the limited history of x variables each y_t variable is conditioned upon.

Both (stochastic) control policies, $\pi(\mathbf{a}_{1:T}||\mathbf{s}_{1:T}) \triangleq \prod_{t=1}^{T} \pi(a_t|\mathbf{a}_{1:t-1}, \mathbf{s}_{1:t})$, and (stochastic) state transition dynamics, $\tau(\mathbf{s}_{1:T}||\mathbf{a}_{1:T-1}) \triangleq \prod_{t=1}^{T} \tau(s_t|\mathbf{s}_{1:t-1}, \mathbf{a}_{1:t-1})$, can be expressed using this notation. The joint probability distribution over states and actions is then $p(\mathbf{a}_{1:T}, \mathbf{s}_{1:T}) = \pi(\mathbf{a}_{1:T}||\mathbf{s}_{1:T})\tau(\mathbf{s}_{1:T}||\mathbf{a}_{1:T-1})$, and the expected reward can be expressed as an affine combination of bilinear functions of these processes:

$$\rho_R(\pi, \tau) = \sum_{\mathbf{a}_{1:T}} \sum_{\mathbf{s}_{1:T}} \pi(\mathbf{a}_{1:T}||\mathbf{s}_{1:T})\tau(\mathbf{s}_{1:T}||\mathbf{a}_{1:T-1}) \sum_{t=1}^{T-1} R(s_t, a_t, s_{t+1}). \quad (3)$$

Additionally, the uncertainty of state sequence outcomes can be quantified using the causally conditioned entropy:

$$H_{\tau,\pi}(S_{1:T}||A_{1:T-1}) = -\sum_{\mathbf{a}_{1:T}, \mathbf{s}_{1:T}} \pi(\mathbf{a}_{1:T}||\mathbf{s}_{1:T})\tau(\mathbf{s}_{1:T}||\mathbf{a}_{1:T-1}) \log \tau(\mathbf{s}_{1:T}||\mathbf{a}_{1:T-1}). \quad (4)$$

Of crucial importance for optimization purposes, the set of causally conditioned probability distributions is convex and the causal entropy is a convex function of those probabilities [19].

3 Marginally-Constrained Robust Control Processes

We define constraints on uncertainties about a decision process based on its interactions with a reference policy. In other words, state-action trajectories through the decision process are available that were produced from a policy that may be quite different from the optimal one. Similarly to previous works [4, 20], we propose practical algorithms for this problem by augmenting the state space.

3.1 Defining Uncertainty Sets with Marginal Features

We consider a feature function $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ characterizing the relationships between states and actions to restrict the set of possible realizations of uncertain MDP parameters. We denote the first moment of the occupancy frequencies with respect to $\boldsymbol{\phi}$ (also known as feature expectations in the inverse reinforcement learning literature [21, 22]) as $\boldsymbol{\kappa}_{\boldsymbol{\phi}}(\pi, \tau) := \mathbb{E}_{\tau,\pi}\left[\sum_{t=1}^{T-1} \boldsymbol{\phi}(S_t, A_t, S_{t+1})\right]$, while we denote the empirical sample statistics, which are measured from N sample trajectories,

as $\widehat{\boldsymbol{\kappa}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T-1}\boldsymbol{\phi}(s_t^{(i)}, a_t^{(i)}, s_{t+1}^{(i)})$. Based on these quantities, we can now define the robust control problem with constraints using marginal statistics of the state-action sequence to define the uncertainty set Ξ.

Definition 2. *The marginally-constrained robust control problem given reference policy $\tilde{\pi}$ is:*

$$\max_{\pi\in\Pi}\min_{\tau\in\Xi}\rho(\pi,\tau) - \frac{1}{\lambda}H_{\tau,\bar{\pi}}(S_{1:T}||A_{1:T-1}), \tag{5}$$

where Ξ is the set of all transition probabilities whose feature expectations match the empirical sample statistics, i.e., $\Xi = \{\tau \mid \boldsymbol{\kappa}_{\boldsymbol{\phi}}(\tilde{\pi},\tau) = \widehat{\boldsymbol{\kappa}}\}$. In general, and of practical significance, slack can also be added to the constraints, leading to a relaxed uncertainty set $\widetilde{\Xi} = \{\tau \mid ||\boldsymbol{\kappa}_{\boldsymbol{\phi}}(\tilde{\pi},\tau) - \widehat{\boldsymbol{\kappa}}|| \leq \beta\}$[1]. We include an optional causal entropy (Equation 4) regularization penalty term, $\frac{1}{\lambda}H_{\tau,\bar{\pi}}(S_{1:T}||A_{1:T-1})$, where $\lambda \in (0,\infty)$ is a provided parameter and $\bar{\pi}(\mathbf{a}_{1:T}||\mathbf{s}_{1:T})$ is an arbitrary distribution.

Intuitively, our formulation allows constraints for whole trajectories rather single state-action pairs, as with rectangular constraints. Furthermore, features $\boldsymbol{\phi}$ allow us to specify properties of the unknown transition dynamics that generalize globally across the state-action space, which is not possible using local constraints, such as rectangular ones. When limited data is available and generalization is therefore required to achieve good performance, this constitutes a significant advantage. Finally, our optional entropy regularization term leads to smoother solutions, where the smoothness is controlled by parameter λ. Many previous works have shown the benefits of having entropy-based smoothing [1, 23].

In practice, the design of the feature function ϕ is fundamental for properly constraining the estimated transition probabilities. Although a specific choice is highly application dependent, the features should in general encode known properties of the underlying MDP. Since our solution reduces to finding dynamics that induce a behavior on the reference policy, specified through $\boldsymbol{\kappa}_{\boldsymbol{\phi}}$, that approximately matches the one observed from the given trajectories, many analogies exist with feature design in the IRL literature (see, e.g., chapter 6 of [24]). Common choices thus include indicator functions over important properties/events, such as reaching certain goal states, entering dangerous zones, taking very likely (or unlikely) transitions, and so on. The key consequence of adding these kinds of features is that the probability of these events occurring under the estimated dynamics will be (approximately) the same as the one observed in the given trajectories. Consider, for instance, an MDP where s, a, s' triples with some known property $\mathcal{P}(s,a,s')$ have zero probability (e.g., in a gridworld or a chain-walk domain, a transition is impossible if s and s' are not adjacent). Then, using a feature $\phi(s,a,s') = \mathbb{1}[\mathcal{P}(s,a,s')]$, i.e., an indicator function over $\mathcal{P}$, will constrain the estimated transition probabilities to be zero for all triples where such property holds. In fact, $\kappa_\phi(\tau,\tilde{\pi}) = 0$ and $\hat{\kappa}_\phi = 0$ for any reference policy. More generally, most MDPs of practical interest have properties that couple the transition probabilities of several state-action pairs. Capturing these global properties using moment-based constraints is typically much better than focusing on single states or state-action pairs, which is more prone to overfitting the given trajectories. In the limiting case, one could consider a separate feature (e.g., an indicator) over each s, a, s' triple. However, similarly to rectangular solutions, having separate constraints for different state-action pairs is likely to lead to very conservative solutions in the presence of limited data. Finally, notice that using an indicator function over each s, a, s' triple is equivalent to matching the (empirical) joint distribution $p(S_t, A_t, S_{t+1})$ induced by the reference policy and the true dynamics. Thus, even when we consider a different constraint for each triple, our solution implicitly couples the transition probabilities of different state-action pairs and differs from a rectangular formulation which focuses on matching the conditional distribution $p(S_{t+1}|S_t, A_t)$.

A key characteristic of this formulation is the difference in control policies: the expected reward is defined in terms of π, while the constraints are defined in terms of $\tilde{\pi}$. Unfortunately, treating the marginally-constrained robust control problem (Definition 1) as an optimization problem over the individual state transition probabilities, $\tau(s_{t+1}|s_t, a_t)$, appears daunting. This is because the constraints in Equation (5) are not convex functions of those transition probabilities. We instead consider optimizing the control policy and state transition dynamics as causally conditioned probability distributions in the following section. Though the solution for this formulation does not naturally have a Markovian property, our process estimation leads to an augmented-Markovian representation in Section 3.3.

[1]Notice that τ must also belong to the set of valid probability distributions. We omit the corresponding constraints for the sake of clarity.

3.2 Reformulation as Process Estimation

We re-express the optimization problem of Definition 2 using processes—the causally conditioned probabilities of Section 2.2—for the control policy $\pi(a_{1:T-1}||s_{1:T-1})$ and state transition dynamics $\tau(s_{1:T}||a_{1:T-1})$, which conveniently combine the individual conditional probabilities over the state-action sequence. Notice that we consider stochastic processes ending with a state at time T and an action at time $T-1$. Using this new notation, we now reformulate our main optimization problem in a more convenient manner.

Theorem 1. *The marginally-constrained robust control problem of Definition 2 can be solved by posing it as an unconstrained zero-sum game parameterized by a vector of Lagrange multipliers, $\boldsymbol{\omega}$:*

$$\max_{\boldsymbol{\omega}\in\mathbb{R}^d}\max_{\pi\in\Pi}\operatorname*{softmin}_{\tau\in\Xi}\left(\mathbb{E}_{\tau,\pi}\left[\sum_{t=1}^{T-1}R(S_t,A_t,S_{t+1})\right]+\mathbb{E}_{\tau,\widetilde{\pi}}\left[\sum_{t=1}^{T-1}\boldsymbol{\omega}\cdot\boldsymbol{\phi}(S_t,A_t,S_{t+1})\right]\right)-\boldsymbol{\omega}\cdot\widehat{\boldsymbol{\kappa}},\quad(6)$$

where $\operatorname{softmin}_{x\in\mathcal{X}} f(x) = -\frac{1}{\lambda}\log\sum_{x\in\mathcal{X}} e^{-\lambda f(x)}$ *and $\cdot$ denotes the dot product.*

The proof is given in Appendix A. Notice that Theorem 1 holds for the slack-free uncertainty set Ξ of Definition 2. Using the slack-based version leads to regularization of the dual parameters $\boldsymbol{\omega}$. As shown by [25], adding l_1 regularization $-\beta||\boldsymbol{\omega}||_1$ to the dual objective is equivalent to a constraint $||\boldsymbol{\kappa}_{\boldsymbol{\phi}}(\widetilde{\pi},\tau)-\widehat{\boldsymbol{\kappa}}||_1\leq\beta$ in the primal, while adding l_2^2 regularization $-\frac{\alpha}{2}||\boldsymbol{\omega}||_2^2$ is equivalent to an l_2^2 potential on the constraint values in the primal. In practice, it is important to add l_1 and/or l_2^2 regularization to ensure proper convergence of the algorithm. Both types of regularization enjoy similar theoretical guarantees [26].

We now address the inner minimax game for choosing τ and π in Section 3.3 and the outer optimization of $\boldsymbol{\omega}$ from Equation (6) in Section 3.4.

3.3 Mixed Objective Minimax Optimal Control

Choosing state transition dynamics to optimize a mixture of expected returns under different control policies, π and $\tilde{\pi}$ (Definition 3)[2] is an important subproblem arising from our formulation of robust control as a process estimation task with robustness properties and uncertainty sets defined by different control policies. To the best of our knowledge, this problem has not been previously investigated in the literature.

Definition 3. *Given two control policies π and $\tilde{\pi}$, and two reward functions R and $\tilde{R}$, the mixed objective optimization problem seeks state transition dynamics τ that minimizes a mixture of these weighted by $\theta\geq 0$: $\min_\tau\{\theta\rho_R(\pi,\tau)+(1-\theta)\rho_{\tilde{R}}(\tilde{\pi},\tau)\}$.*

Notice that the inner minimization of Equation (6) is an entropy-regularized instance of this problem. In fact, we can set $\widetilde{R}(s_t,a_t,s_{t+1})\leftarrow\boldsymbol{\omega}\cdot\boldsymbol{\phi}(s_t,a_t,s_{t+1})$ and $\theta=\frac{1}{2}$ (provided that rewards are properly rescaled). As we already know from Theorem 1, the entropy leads to a softmin solution and does not pose any additional complication in solving the optimization problem of Definition 3. Furthermore, in the inner zero-sum game of Equation (6), π is chosen as the maximizer of $\rho(\pi,\tau)$. Thus, we can see Definition 3 as a special case where π is fixed rather than chosen dynamically.

An important observation for this problem is that the optimal transition dynamics are not Markovian. Indeed, the influence of ρ_R and $\rho_{\tilde{R}}$ on choosing the next-state distribution at some decision point depends on how probable it is for that decision point to be realized under π and under $\tilde{\pi}$. This, in turn, depends on the entire history of states and actions leading to the current decision point. However, we establish that this non-Markovian problem can be Markovianized by augmenting the current state-action pair with a continuous "belief state" as follows:

$$b(\mathbf{a}_{1:t}||\mathbf{s}_{1:t})\triangleq\frac{\prod_{i=1}^t\pi(a_i|\mathbf{a}_{1:i-1},\mathbf{s}_{1:i})}{\prod_{i=1}^t\pi(a_i|\mathbf{a}_{1:i-1},\mathbf{s}_{1:i})+\prod_{i=1}^t\widetilde{\pi}(a_i|\mathbf{a}_{1:i-1},\mathbf{s}_{1:i})}=\frac{\pi(\mathbf{a}_{1:t}||\mathbf{s}_{1:t})}{\pi(\mathbf{a}_{1:t}||\mathbf{s}_{1:t})+\widetilde{\pi}(\mathbf{a}_{1:t}||\mathbf{s}_{1:t})}.\quad(7)$$

The belief state tracks the relative probability of the decision point under π and $\tilde{\pi}$. Defining it in this manner is convenient because it limits the domain for b to $[0,1]$. It can also be updated to incorporate

[2]Without any loss of generality, this problem could be equivalently posed as finding the control policy π that maximizes a mixture of rewards $\theta\rho_R(\pi,\tau)+(1-\theta)\rho_{\tilde{R}}(\pi,\tilde{\tau})$ for two different decision processes with dynamics/reward (τ,R) and $(\tilde{\tau},\tilde{R})$.

a new action a_{t+1} in state s_{t+1} as:

$$b(\mathbf{a}_{1:t+1}||\mathbf{s}_{1:t+1}) = \frac{b(\mathbf{a}_{1:t}||\mathbf{s}_{1:t})\pi(a_{t+1}|\mathbf{a}_{1:t},\mathbf{s}_{1:t+1})}{b(\mathbf{a}_{1:t}||\mathbf{s}_{1:t})\pi(a_{t+1}|\mathbf{a}_{1:t},\mathbf{s}_{1:t+1}) + (1-b(\mathbf{a}_{1:t}||\mathbf{s}_{1:t}))\widetilde{\pi}(a_{t+1}|\mathbf{a}_{1:t},\mathbf{s}_{1:t+1})}. \quad (8)$$

Augmenting with the belief state of Equation (7), we prove that it is possible to compute a Markovian solution to the inner zero-sum game of Equation (6) and, thus, to the optimization problem of Definition 3.

Theorem 2. *Let $\tilde{\pi}$ be a given randomized Markovian policy and $Z(s_t, a_t, b_{t-1}) = b_{t-1} + (1 - b_{t-1})\tilde{\pi}(a_t|s_t)$, where b_t is the belief state defined in Equation* (7)*. Then, a solution (π^*, τ^*) to the inner zero-sum game of Equation* (6) *is:*

$$\tau^*(s_{t+1}|s_t, a_t, b_t) = \frac{e^{-\lambda Q(s_t,a_t,b_t,s_{t+1})}}{\sum_{s'} e^{-\lambda Q(s_t,a_t,b_t,s')}};\ \pi^*(s_t, b_{t-1}) = \operatorname*{argmax}_{a_t} Q_R\left(s_t, a_t, \frac{b_{t-1}}{Z(s_t,a_t,b_{t-1})}\right), \quad (9)$$

with Q as the value of a transition to state s_{t+1}, V as the value of state s_t and belief state b_{t-1}, and Q_R as the expected return from R obtained by taking action a_t in state s_t and belief state b_t:

$$Q(s_t, a_t, b_t, s_{t+1}) = b_t R(s_t, a_t, s_{t+1}) + (1 - b_t)\widetilde{R}(s_t, a_t, s_{t+1}) + V(s_{t+1}, b_t), \quad (10)$$

$$V(s_t, b_{t-1}) = Z'(s_t, b_{t-1}) \operatorname*{softmin}_{s_{t+1}} Q\left(s_t, \pi^*(s_t, b_{t-1}), \frac{b_{t-1}}{Z'(s_t, b_{t-1})}, s_{t+1}\right), \quad (11)$$

$$Q_R(s_t, a_t, b_t) = \sum_{s_{t+1}} \tau^*(s_{t+1}|s_t, a_t, b_t)\left(R(s_t, a_t, s_{t+1}) + Q_R\left(s_{t+1}, \pi^*(s_{t+1}, b_t), \frac{b_t}{Z'(s_{t+1}, b_t)}\right)\right) \quad (12)$$

where $Z'(s_t, b_{t-1}) = Z(s_t, \pi^*(s_t, b_{t-1}), b_{t-1})$.

The proof is given in Appendix A.

Since we have a maximum causal entropy estimation problem, τ^* (Equation 9) takes the form of a Boltzmann distribution with temperature λ^{-1} and energy given by $Q(s_t, a_t, b_t, s_{t+1})$. Function Q (Equation 10) specifies the value of a transition from s_t, a_t, b_t to state s_{t+1}. Intuitively, it is a sum of (i) the immediate return, which in turn is a mixture of rewards from R and $\tilde{R}$ weighted by the current belief state, and (ii) the value of the next state s_{t+1} given that the current belief is b_t. We have the additional complication that π is chosen dynamically as the maximizer of $\rho_R(\pi, \tau)$ rather than statically. Given τ^*, the optimal policy π^* (Equation 9) aims at maximizing the expected future return from R defined in (12). Notice that since the optimal policy π^* is deterministic and $\tilde{\pi}$ is Markovian, the belief state update rule of (8) can be written in the more concise form: $b_{t+1} = \frac{b_t}{Z'(s_{t+1},b_t)}$. Finally, given τ^* and π^*, we can compute the optimal value V obtained from state s_t and belief state b_{t-1} as defined in (11). Algorithm 1 summarizes our Markovian dynamic program.

In contrast to typical value iteration in discrete MDPs, the belief states are continuous variables in Algorithm 1. In practice, we discretize them by considering a set $\mathcal{B}$ of values in the range $[0, 1]$ and then interpolate between these points. Notice that since π^* is deterministic, values in $(0, 0.5)$ are not possible and can be safely neglected. This discretization allows for a compact tabular representation of all functions defined in Theorem 2. The asymptotic complexity of this procedure (Algorithm 1) is then $\mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|\ |\mathcal{B}|\ T)$.

Algorithm 1 Min-max Dynamic Programming

Require: Reference policy $\tilde{\pi}$, reward function $R(s_t, a_t, s_{t+1})$, feature function $\boldsymbol{\phi}(s_t, a_t, s_{t+1})$, Lagrange multiplier $\boldsymbol{\omega}$, entropy regularization weight λ
Ensure: Robust dynamics τ^*, optimal policy π^*

$V(s_T, b_{T-1}) \leftarrow 0$; $\widetilde{R}(s_t, a_t, s_{t+1}) \leftarrow \omega \cdot \phi(s_t, a_t, s_{t+1})$
for $t = T - 1$ to 1 **do**
  Set $Q(s_t, a_t, b_t, s_{t+1})$ from V using (10)
  Set $\tau^*(\cdot|s_t, a_t, b_t) \propto e^{-\lambda Q(s_t,a_t,b_t,\cdot)}$
  Set $Q_R(s_t, a_t, b_t)$ from τ^* and Q_R using (12)
  Set $\pi^*(s_t, b_{t-1}) = \operatorname{argmax}_{a_t} Q_R(s_t, a_t, b_t)$
  Set $V(s_t, b_{t-1})$ from Q and π^* using (11)
end for

The robust policy π^* returned by Algorithm 1 is, for each time-step t, a function $\pi_t^* : \mathcal{S} \times \mathcal{B} \to \mathcal{A}$ mapping state-belief state couples to actions. For the sake of completeness, we show how such a policy can be used in a regular MDP with dynamics τ. Notice that, since belief states are updated according to Equation (8), we need to keep track of the reference policy $\tilde{\pi}$. At the first time-step, state s_1 is drawn from the MDP's initial state distribution, while the initial belief state b_0 is set to 0.5, as can be seen from Equation (7). Then, action $a_1 = \pi_1^*(s_1, b_0)$ is taken, and the system transitions to the next state $s_2 \sim \tau(\cdot|s_1, a_1)$. Finally, the belief state is updated to account for the choice of action a_1: $b_1 = b_0 \ / \ N(s_1, b_0)$. Then, this process is repeated until the maximum time-step is reached.

3.4 Parameter Optimization

Standard gradient-based methods can be used to optimize the choice of model parameters $\boldsymbol{\omega}$, since the unconstrained dual objective function is a concave function of $\boldsymbol{\omega}$. Any such method is required to repeatedly solve the inner minimax problem of Equation (6) as specified in the previous section, obtaining (π^*, τ^*), compute the feature expectations of the reference policy $\widetilde{\pi}$ under τ^*, and use these to update $\boldsymbol{\omega}$. Conceptually, model parameters $\boldsymbol{\omega}$ are chosen to motivate the adversary's dynamics to satisfy the constraints from the reference policy—(approximately) matching the state-action feature statistics of the training trajectories. Hence, under the assumption that matching features is feasible, following the gradient update rule, $\boldsymbol{\omega}_{i+1} \leftarrow \boldsymbol{\omega}_i + \eta_i(\boldsymbol{\kappa}_{\boldsymbol{\phi}}(\widetilde{\pi}, \tau^*) - \widehat{\boldsymbol{\kappa}})$, converges when the statistics match, i.e., when $\boldsymbol{\kappa}_{\boldsymbol{\phi}}(\widetilde{\pi}, \tau^*) = \widehat{\boldsymbol{\kappa}}$[3].

Computing the expected features under the adversary's non-Markovian dynamics, τ^*, requires an extension of the dynamic programming algorithm used to obtain τ^* itself. The next result follows almost straightforwardly from Theorem 2. For the sake of completeness, we include a proof in Appendix A.

Corollary 1. *Let (π^*, τ^*) be the belief-augmented solution of Theorem 2, $p(s_1)$ be the initial state distribution of the given MDP, and $\widetilde{\pi}$ be a randomized Markovian policy. Then:*

$$\boldsymbol{\kappa}_{\boldsymbol{\phi}}(\widetilde{\pi}, \tau^*) = \sum_{s_1} p(s_1)\boldsymbol{\Psi}(s_1, b_0), \tag{13}$$

where $\boldsymbol{\Psi}$ is defined recursively for $t = 1, \ldots, T-1$ as:

$$\boldsymbol{\Psi}(s_t, b_{t-1}) = \sum_{a_t} \widetilde{\pi}(a_t|s_t) \sum_{s_{t+1}} \tau^*(s_{t+1}|s_t, a_t, b_t) \left[\boldsymbol{\phi}(s_t, a_t, s_{t+1}) + \boldsymbol{\Psi}(s_{t+1}, b_t)\right], \tag{14}$$

with $\boldsymbol{\Psi}(s_T, b_{T-1}) = \mathbf{0}$ and $b_t = \frac{b_{t-1}}{Z(s_t, a_t, b_t - 1)} \mathbb{1}\left[a_t = \pi^*(s_t, b_{t-1})\right]$.

Notice that the computation of $\boldsymbol{\kappa}_{\boldsymbol{\phi}}(\widetilde{\pi}, \tau^*)$, as given by Corollary 1, can be efficiently included in the dynamic program of Algorithm 1 by updating $\boldsymbol{\Psi}$ as the last step of each iteration according to (14).

4 Experiments

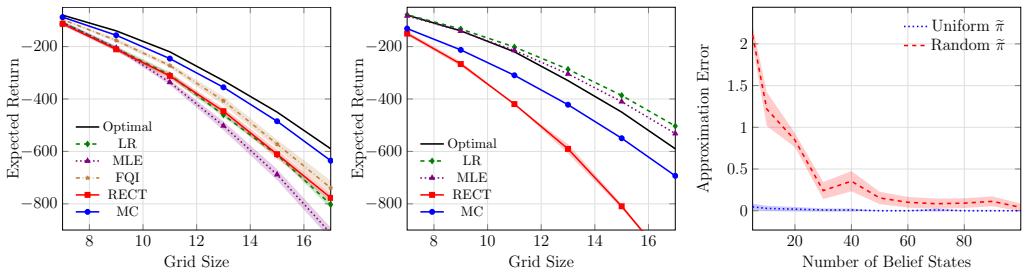
In this section, we empirically evaluate our robust approach for control using uncertainty sets defined by marginal state-action statistics. We consider two different experiments. The first one is a classic grid navigation problem and the second one is a more challenging domain in which the goal is to control the population change of an invasive species. In all experiments, we compare our marginally-constrained approach (MC) to three other methods for estimating the state-transition dynamics: (1) a supervised approach using logistic regression (LR); (2) a robust MDP with *s,a*-rectangular uncertainty sets (RECT); and (3) a simple maximum likelihood estimation (MLE) of the conditional transition probabilities for all state-action pairs. Furthermore, due to the similarity between our settings and batch reinforcement learning, we also compare to fitted Q-iteration (FQI) [27].

4.1 Gridworld

We consider an agent navigating through an $N \times N$ grid in order to reach a goal position. The agent's location is described by its horizontal and vertical coordinates (x, y). At each time-step, the agent can attempt to move in each of the four cardinal directions. With probability $p = 0.3$, the action fails and the agent moves in a random direction instead. Attempts to move off the grid have no effect. The agent's initial position is $(1, 1)$, while the goal is to reach state (N, N). The horizon is set to $T = 2N$, while the reward function is the negative l_1 distance between the next state and the goal.

In this experiment, we prove the generalization capabilities of our approach. We consider a sequence of gridworlds with increasing size. For each of them, we collect 50 trajectories under a uniform reference policy and we run all algorithms on such data. Intuitively, for small grids, such trajectories provide enough exploration to allow all methods to accurately approximate the state-transition

[3]When l_1 or l_2^2 regularization of $\boldsymbol{\omega}$ is used, this procedure converges when the feature expectations are close to the sample statistics, where the closeness depends on the amount of regularization used (see Section 3.2).



(a) (b) (c)

Figure 1: Results of the gridworld experiments, each with 95% confidence intervals. (a) Expected return under the true dynamics as a function of the grid size. (b) Expected return under the estimated (robust) dynamics as function of the grid size. (c) Approximation error incurred by our algorithm due to the discretization of the belief space.

dynamics. However, as the grid grows larger, only a small portion of the state-space is observed in the training data. Thus, generalization is required to achieve good performance. Additional details on the adopted parameters are given in Appendix C.1.

Figure 1a shows the expected return achieved by all algorithms as a function of the grid size N. Results are averaged over 20 runs. As expected, for small grids (e.g., $N \leq 7$) all approaches obtain nearly-optimal performance. However, as the grid size increases, only our method is able to estimate dynamics that generalize across unseen regions of the state-space, thus maintaining nearly-optimal performance. FQI is also able to generalize and achieves a significant improvement over the other alternatives, but is not able to compete with our method due to the small number of trajectories available. LR is likely to estimate very optimistic dynamics, thus leading to worse performance. Finally, RECT obtains results comparable to LR even without generalizing. However, rectangular uncertainty sets are too conservative to compete with our method. To better demonstrate this fact, Figure 1b shows the performance achieved by the optimal policy computed by each algorithm under their own estimated dynamics (except for FQI, which is model-free). We clearly notice that the worst-case expected return obtained by the rectangular solution is, as claimed, very conservative. Our approach, on the other hand, shows robust performance comparable to the true ones of the other methods. Due to their optimistic estimates, both LR and MLE obtain an expected return even larger than the optimal one.

Finally, we analyze the approximation error incurred from discretizing the belief states in our approach. We consider a 5×5 gridworld with the same parameters as before and run the dynamic program of Algorithm 1 for 50 random values of $\boldsymbol{w}$ using two different reference policies: the uniform one and a random one. Figure 1c shows the average absolute deviation of the objective function from its true value as a function of the number of discrete belief states N_b. Since, as we can observe from (7), the total number of belief states that are reachable in a finite horizon depends on the number of different probability values assigned by $\widetilde{\pi}$, the uniform reference policy achieves a very small approximation error even with few belief states. Interestingly, the approximation error for a random reference policy, which can be regarded as a 'worst-case' scenario, can also be reduced using a relatively small number of belief states.

4.2 Invasive Species

We next consider modeling the population change of an invasive species in an ecosystem with a single action available for mitigating its spread (e.g., introducing a predator). Our starting point is a state-space model with exponential dynamics adapted from Chapter 5 of [28]. Each state captures the current abundance of the invasive species, which we denote as N_t at time t. The population evolves according to exponential dynamics, so that $N_{t+1} = \min\{\nu_t N_t, K\}$, where K is the maximum carrying capacity. The growth rate ν depends on (i) whether the control action a_t has been applied, (ii) the current population level N_t, and (iii) random noise. When the control action is not applied ($a_t = 0$), the growth rate is: $\nu_t = \max\{0, \bar{\nu} + \mathcal{N}(0, \sigma_\nu^2)\}$, where $\bar{\nu}$ is the mean growth rate. In this case, the growth rate is independent of the current population level. When the control action is applied ($a_t = 1$), the growth rate is: $\nu_t = \bar{\nu} - \beta_1 N_t - \beta_2 \max\{0, N_t - \hat{N}\}^2 + \mathcal{N}(0, \sigma_\nu^2)$, where

Table 1: Negative expected return for different numbers of trajectories M and reference policy's control probabilities p in the invasive species experiment. Each value is the average of 20 independent runs. 95% confidence intervals are shown. The best algorithms are highlighted in bold.

Alg.	M	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
MLE	50	121.74 ± 0.82	128.34 ± 2.06	140.36 ± 1.28	147.189 ± 1.78	149.82 ± 2.12
LR	50	152.95 ± 13.5	106.77 ± 2.21	117.43 ± 5.09	122.756 ± 5.94	$\mathbf{123.28} \pm 4.82$
MC	50	$\mathbf{99.37} \pm 0.96$	$\mathbf{102.38} \pm 1.82$	$\mathbf{98.36} \pm 0.78$	$\mathbf{107.39} \pm 3.44$	$\mathbf{124.47} \pm 1.81$
RECT	50	111.91 ± 5.33	$\mathbf{107.71} \pm 4.13$	117.15 ± 6.76	123.55 ± 7.95	142.26 ± 8.28
FQI	50	140.85 ± 6.11	133.08 ± 5.36	133.77 ± 4.70	134.05 ± 6.22	140.25 ± 5.04
MLE	100	120.91 ± 0.63	125.21 ± 1.25	134.23 ± 1.33	140.96 ± 1.76	145.42 ± 1.72
LR	100	169.27 ± 8.72	$\mathbf{104.70} \pm 3.43$	110.09 ± 2.57	114.23 ± 2.49	$\mathbf{124.53} \pm 4.98$
MC	100	$\mathbf{98.25} \pm 0.88$	$\mathbf{103.66} \pm 1.05$	$\mathbf{96.20} \pm 0.95$	$\mathbf{105.17} \pm 1.95$	$\mathbf{115.04} \pm 6.18$
RECT	100	$\mathbf{100.98} \pm 3.33$	$\mathbf{103.80} \pm 3.22$	108.69 ± 4.95	$\mathbf{106.18} \pm 4.02$	136.24 ± 8.41
FQI	100	126.66 ± 5.84	121.93 ± 6.27	119.85 ± 4.30	125.65 ± 5.08	131.51 ± 4.92

β_1 and β_2 are the coefficients of effectiveness and $\hat{N}$ is the population at which the effectiveness peaks. That is, the effectiveness of the control method may increase or decrease depending on the population of the invasive species. This dependence is modeled using a simplified quadratic spline. The precise population N_t of the species cannot be directly observed. Instead, one can observe a noisy estimate $y_t = N_t + \mathcal{N}(0, \sigma_y^2)$. The exact values of the parameters used in this experiment are $K = 500$, $T = 100$, $\hat{K} = 300$, $\bar{\nu} = 1.02$, $\beta_1 = 0.001$, $\beta_2 = -0.0000021$, $\sigma_\nu^2 = 0.02$, $\sigma_y^2 = 20$. Notice that due to its highly unstable dynamics and noisy observations, this domain represents a very challenging control problem.

In this experiment, we analyze the behavior of all algorithms when given different amounts of trajectories collected under different reference policies. In particular, we consider five reference policies, where each chooses to apply the control action with a fixed probability $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. For each reference policy, we generate two datasets of $M_1 = 50$ trajectories and $M_2 = 100$ trajectories, respectively. Additional details are given in Appendix C.2.

Results of our experiments in these settings are reported in Table 1. Each datapoint is obtained as the result of an average over 20 runs, We notice that MC outperforms all alternatives when $p < 0.5$ and $M = 50$. As before, this is due to its generalization capabilities. When considering $M = 100$ trajectories, all other approaches significantly improve their performance. However, MC is still able to achieve better results for most values of p. The rectangular solution (RECT) also achieves good performance, but shows a much higher variability. Finally, we note that all algorithms suffer from the very limited exploration provided by a reference policy with $p = 0.5$. In such cases, the performance of the feature-based approaches are superior.

5 Conclusion & Future Work

In this paper, we have proposed a new approach to robust control based on causally conditioned probability distribution estimation that defines uncertainty sets using features of the interaction with the decision process with a different policy. Though the solution to the corresponding robust control problem is non-Markovian, we show that it can be closely approximated by augmenting the typical Markovian robust MDP formulation [29, 4] with a continuous-valued "belief state" that can then be discretized. We have empirically tested our approach on a synthetic experiment and a real-world control problem, highlighting its advantages over methods that form rectangular uncertainty sets.

We plan to extend our formulation to incorporate constraints that are obtained from multiple separate reference control policies. This could also allow episodic reinforcement learning [30] where the robust optimal control policy is employed and then updated based on the trajectories that are observed from its application. Incorporating more sophisticated ideas for solving POMDPs using belief state compression will likely be required, since discretizing the belief space scales poorly with the number of different reference policies.

Acknowledgments

We thank the anonymous reviewers whose comments helped to improve the paper significantly. This work was supported, in part, by the National Science Foundation under Grant No. 1652530 and Grant No. 1717368, and by the Future of Life Institute (futureoflife.org) FLI-RFP-AI1 program.

References

[1] A. Nilim and L. El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[2] Grani Adiwena Hanasusanto and Daniel Kuhn. Robust data-driven dynamic programming. In *Advances in Neural Information Processing Systems*, pages 827–835, 2013.

[3] Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

[4] Shie Mannor, Ofir Mebel, and Huan Xu. Lightning does not strike twice: Robust mdps with coupled uncertainty. *arXiv preprint arXiv:1206.4643*, 2012.

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

[6] Yann Le Tallec. *Robust, Risk-Sensitive, and Data-driven Control of Markov Decision Processes*. PhD thesis, MIT, 2007.

[7] Marek Petrik, Mohammad Ghavamzadeh, and Yinlam Chow. Safe Policy Improvement by Minimizing Robust Baseline Regret. In *Advances in Neural Information Processing Systems*, 2016.

[8] Erick Delage. Distributionally Robust Optimization under Moment Uncertainty with Application to Data-Driven Problems. 00(0):1–26, 2000.

[9] Huan Xu and Shie Mannor. Distributionally robust markov decision processes. In *Advances in Neural Information Processing Systems*, pages 2505–2513, 2010.

[10] Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

[11] Marek Petrik and Dharmashankar Subramanian. RAAM : The benefits of robustness in approximating aggregated MDPs in reinforcement learning. In *Neural Information Processing Systems (NIPS)*, 2014.

[12] Martin L Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc., 2005.

[13] G. Kramer. *Directed Information for Channels with Feedback*. PhD thesis, Swiss Federal Institute of Technology (ETH) Zurich, 1998.

[14] Hans Marko. The bidirectional communication theory – a generalization of information theory. In *IEEE Transactions on Communications*, pages 1345–1351, 1973.

[15] James L. Massey. Causality, feedback and directed information. In *Proc. IEEE International Symposium on Information Theory and Its Applications*, pages 27–30, 1990.

[16] Haim H. Permuter, Young-Han Kim, and Tsachy Weissman. On directed information and gambling. In *Proc. IEEE International Symposium on Information Theory*, pages 1403–1407, 2008.

[17] S. Tatikonda. *Control under Communication Constraints*. PhD thesis, Massachusetts Institute of Technology, 2000.

[18] G. Kramer. Capacity results for the discrete memoryless network. *Proc. IEEE Transactions on Information Theory*, 49(1):4–21, Jan 2003.

[19] Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. Modeling interaction via the principle of maximum causal entropy. In *Proc. International Conference on Machine Learning*, pages 1255–1262, 2010.

[20] Bita Analui and Georg Ch Pflug. On distributionally robust multiperiod stochastic optimization. *Computational Management Science*, 11(3):197–220, 2014.

[21] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proc. International Conference on Machine Learning*, pages 1–8, 2004.

[22] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proc. AAAI Conference on Artificial Intelligence*, pages 1433–1438, 2008.

[23] Zhaolin Hu and L Jeff Hong. Kullback-leibler divergence constrained distributionally robust optimization.

[24] B. D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010.

[25] Jun'ichi Kazama and Jun'ichi Tsujii. Evaluation and extension of maximum entropy models with inequality constraints. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 137–144. Association for Computational Linguistics, 2003.

[26] Miroslav Dudík, Steven J. Phillips, and Robert E. Schapire. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *J. Mach. Learn. Res.*, 8:1217–1260, 2007.

[27] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research*, 6(1):503–556, 2005.

[28] M. Kery and M. Schaub. *Bayesian Population Analysis using WinBUGS: A Hierarchical Perspective*. Elsevier Science, 2011.

[29] A B Philpott, V de Matos, and V L De Matos. Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. *European Journal of Operations Research*, 218(2):470–483, 2012.

[30] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1.

[31] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

A Proofs

A.1 Proof of Theorem 1

Proof of Theorem 1. Lagrangianizing the constraints of the optimization problem of Definition 2, we obtain:

$$\max_{\pi} \min_{\tau} \max_{\boldsymbol{\omega}} \left(\rho(\pi, \tau) - \frac{1}{\lambda} H_{\tau,\bar{\pi}}(S_{1:T}||A_{1:T-1}) + \boldsymbol{\omega} \cdot (\boldsymbol{\kappa}_{\boldsymbol{\phi}}(\widetilde{\pi}, \tau) - \widehat{\boldsymbol{\kappa}}) \right).$$

Under the assumption that there exists τ that matches the features while strictly satisfying the probabilistic inequality constraints (which we kept implicit for the sake of clarity), Slater's condition and, thus, strong duality, holds [31]. Then, we can write:

$$\max_{\boldsymbol{\omega}} \max_{\pi} \min_{\tau} \left(\rho(\pi, \tau) + \boldsymbol{\omega} \cdot \boldsymbol{\kappa}_{\boldsymbol{\phi}}(\widetilde{\pi}, \tau) - \frac{1}{\lambda} H_{\tau,\bar{\pi}}(S_{1:T}||A_{1:T-1}) \right) - \boldsymbol{\omega} \cdot \widehat{\boldsymbol{\kappa}}.$$

The inner minimization problem in this expression is equivalent to a maximum causal entropy estimation problem scaled by λ [19] and has a *softmin* Boltzmann distribution over state outcomes,

$$\operatorname*{softmin}_{x} f(x) = -\frac{1}{\lambda} \log \sum_{x} e^{-\lambda f(x)},$$

which is a smooth relaxation of the *min* function. Thus, we can safely remove the entropy term, while replacing *min* with *softmin*:

$$\max_{\boldsymbol{\omega}} \max_{\pi} \operatorname*{softmin}_{\tau} \left(\rho(\pi, \tau) + \boldsymbol{\omega} \cdot \boldsymbol{\kappa}_{\boldsymbol{\phi}}(\widetilde{\pi}, \tau) \right) - \boldsymbol{\omega} \cdot \widehat{\boldsymbol{\kappa}}.$$

□

A.2 Proof of Theorem 2

Proof of Theorem 2. For the sake of completeness, we derive the theorem for the zero-sum game where the entropy regularizer is explicitly stated, although we already know from Theorem 1 that it will lead to a softmin solution. Thus, we want to solve the problem:

$$\max_{\pi} \min_{\tau} \left\{ \rho_R(\pi, \tau) + \rho_{\widetilde{R}}(\widetilde{\pi}, \tau) - \frac{1}{\lambda} H_{\tau,\bar{\pi}}(S_{1:T}||A_{1:T-1}) \right\}.$$

Consider the computation of the distribution over final states s_T after state-action sequence $\mathbf{s}_{1:T-1}$, $\mathbf{a}_{1:T-1}$, namely $\tau(\cdot|\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1})$. Specifically, we want to solve the optimization problem:

$$\min_{\tau(\cdot|\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1})} L(\tau, \mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}),$$

where:

$$\begin{aligned} L(\tau, \mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}) = \tau(\mathbf{s}_{1:T-1}||\mathbf{a}_{1:T-2}) \sum_{s_T} \tau(s_T|\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}) \\ \Bigg(\pi(\mathbf{a}_{1:T-1}||\mathbf{s}_{1:T-1}) R(s_{T-1}, a_{T-1}, s_T) \\ + \widetilde{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1}) \widetilde{R}(s_{T-1}, a_{T-1}, s_T) \\ + \frac{1}{\lambda} \bar{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1}) \log \tau\left(s_T|\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}\right) \Bigg). \end{aligned} \tag{15}$$

Notice that we kept implicit the positivity constraints, $\tau(s_T|\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}) >= 0$, and the normalization ones, $\sum_{s_T} \tau(s_T|\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}) = 1$. Differentiating with respect to $\tau(s_T|\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1})$ we obtain:

$$\begin{aligned} \frac{\partial L}{\partial \tau} = \tau(\mathbf{s}_{1:T-1}||\mathbf{a}_{1:T-2}) \Big((\pi(\mathbf{a}_{1:T-1}||\mathbf{s}_{1:T-1}) R(s_{T-1}, a_{T-1}, s_T) \\ + \widetilde{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1}) \widetilde{R}(s_{T-1}, a_{T-1}, s_T) \\ + \lambda^{-1} \bar{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1}) \log \tau(s_T|\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}) \\ + \lambda^{-1} \bar{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1}) \Big). \end{aligned}$$

Equating this last term to zero and solving for $\tau(s_T|\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1})$, we obtain:

$$\tau(s_T|\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}) \propto e^{-\lambda Q(\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}, s_T)}$$

where we set:

$$Q(\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}, s_T) = \frac{\pi(\mathbf{a}_{1:T-1}||\mathbf{s}_{1:T-1})R(s_{T-1}, a_{T-1}, s_T) + \widetilde{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1})\widetilde{R}(s_{T-1}, a_{T-1}, s_T)}{\bar{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1})}$$

Notice that we neglect the term $\lambda\bar{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1})$ since it is constant over next states. Although convenient, this form requires the knowledge of the whole state-action history in order to compute τ. However, since $\bar{\pi}$ is arbitrary and do not affect our solution, we can set it to $(\pi + \tilde{\pi})/2$, thus obtaining:

$$\begin{aligned} Q(\mathbf{s}_{1:T-1}, \mathbf{a}_{1:T-1}, s_T) &= \frac{\pi(\mathbf{a}_{1:T-1}||\mathbf{s}_{1:T-1})R(s_{T-1}, a_{T-1}, s_T) + \widetilde{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1})\widetilde{R}(s_{T-1}, a_{T-1}, s_T)}{\pi(\mathbf{a}_{1:T-1}||\mathbf{s}_{1:T-1}) + \widetilde{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1})} \\ &= b_{T-1}R(s_{T-1}, a_{T-1}, s_T) + (1 - b_{T-1})\widetilde{R}(s_{T-1}, a_{T-1}, s_T), \end{aligned}$$

where we neglect the constant term 2 since it can easily be embedded into λ. Here, we introduce a continuous "belief state" summarizing the history of states and actions:

$$b_t \triangleq \frac{\pi(\mathbf{a}_{1:t}||\mathbf{s}_{1:t})}{\pi(\mathbf{a}_{1:t}||\mathbf{s}_{1:t}) + \widetilde{\pi}(\mathbf{a}_{1:t}||\mathbf{s}_{1:t})}.$$

Notice that now $Q(s_{T-1}, a_{T-1}, b_{T-1}, s_T)$ depends only on variables at time $T-1$, and so does τ. Then, our final next-state distribution is obtained after normalization:

$$\tau^*(s_T|s_{T-1}, a_{T-1}, b_{T-1}) = \frac{e^{-\lambda Q(s_{T-1}, a_{T-1}, b_{T-1}, s_T)}}{\sum_{s'_T} e^{-\lambda Q(s_{T-1}, a_{T-1}, b_{T-1}, s'_T)}}. \tag{16}$$

Continuing backwards, the optimal action for π to take in state s_{T-1} and belief state b_{T-2} is:

$$\pi^*(s_{T-1}, b_{T-2}) = \operatorname*{argmax}_{a_{T-1}} Q_R\left(s_{T-1}, a_{T-1}, \frac{b_{T-2}}{Z(s_{T-1}, a_{T-1}, b_{T-2})}\right), \tag{17}$$

where we define $Z(s_t, a_t, b_{t-1}) \triangleq b_{t-1} + (1 - b_{t-1})\tilde{\pi}(a_t|s_t)$ and Q_R is the expected reward under τ^*:

$$Q_R(s_{T-1}, a_{T-1}, b_{T-1}) = \sum_{s_T} \tau^*(s_T|s_{T-1}, a_{T-1}, b_{T-1})R(s_{T-1}, a_{T-1}, s_T).$$

Equation (17) can be verified by noticing that, after taking an action a_{T-1}, the belief state b_{T-2} must be updated according to (8). Furthermore, it is easy to verify that the optimal policy is deterministic by writing the objective as a function of the whole history rather than belief states.

Given the distribution over final states s_T and the final action a_{T-1}, we can compute the objective value at time $T-1$ by substituting Equation (16) and 17 into Equation (15):

$$\begin{aligned} V(s_{T-1}, b_{T-2}) = \sum_{s_T} \tau^* &\left(s_T|s_{T-1}, \pi^*(s_{T-1}, b_{T-2}), \frac{b_{T-2}}{Z'(s_{T-1}, b_{T-2})}\right) \\ &\Bigg(\pi(\mathbf{a}_{1:T-1}||\mathbf{s}_{1:T-1})R(s_{T-1}, \pi^*(s_{T-1}, b_{T-2}), s_T) \\ &+ \widetilde{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1})\widetilde{R}(s_{T-1}, \pi^*(s_{T-1}, b_{T-2}), s_T) \\ &- \bar{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1})Q\left(s_{T-1}, \pi^*(s_{T-1}, b_{T-2}), \frac{b_{T-2}}{Z'(s_{T-1}, b_{T-2})}, s_T\right) \\ &- \lambda^{-1}\bar{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1}) \log \sum_{s'_T} e^{-\lambda Q\left(s_{T-1}, \pi^*(s_{T-1}, b_{T-2}), \frac{b_{T-2}}{Z'(s_{T-1}, b_{T-2})}, s'_T\right)}\Bigg) \\ = &- \lambda^{-1}\bar{\pi}(\mathbf{a}_{1:T-1}|\mathbf{s}_{1:T-1}) \log \sum_{s'_T} e^{-\lambda Q\left(s_{T-1}, \pi^*(s_{T-1}, b_{T-2}), \frac{b_{T-2}}{Z'(s_{T-1}, b_{T-2})}, s'_T\right)} \\ = &\, Z'(s_{T-1}, b_{T-2}) \operatorname*{softmin}_{s_T} Q\left(s_{T-1}, \pi^*(s_{T-1}, b_{T-2}), \frac{b_{T-2}}{Z'(s_{T-1}, b_{T-2})}, s_T\right), \end{aligned}$$

where we defined $Z'(s_t, b_{t-1}) \triangleq b_{t-1} + (1 - b_{t-1})\tilde{\pi}(\pi^*(s_t, b_{t-1})|s_t) = Z(s_t, \pi^*(s_t, b_{t-1}), b_{t-1})$.
We can now move to the preceding time-step to choose $\tau(\cdot|s_{T-2}, a_{T-2}, b_{T-2})$. Similarly as before, we want to compute:

$$\min_{\tau(\cdot|s_{T-2},a_{T-2},b_{T-2})} L(\tau, s_{T-2}, a_{T-2}, b_{T-3}),$$

where we replace the dependency on the full history with the previously-defined belief state:

$$\begin{aligned} L(\tau, s_{T-2}, a_{T-2}, b_{T-3}) = \sum_{s_{T-1}} \tau(s_{T-1}|s_{T-2}, a_{T-2}, b_{T-2}) Z(s_{T-2}, a_{T-2}, b_{T-3}) \\ \Big(b_{T-2} R(s_{T-2}, a_{T-2}, s_{T-1}) \\ + \ (1 - b_{T-2}) \widetilde{R}(s_{T-2}, a_{T-2}, s_{T-1}) \\ + \ V(s_{T-1}, b_{T-2}) \\ + \ \lambda^{-1} \log \tau(s_{T-1}|s_{T-2}, a_{T-2}, b_{T-2}) \Big). \end{aligned}$$

By differentiating, equating to zero, and solving for τ, we obtain:

$$\tau^*(s_{T-1}|s_{T-2}, a_{T-2}, b_{T-2}) = \frac{e^{-\lambda Q(s_{T-2}, a_{T-2}, b_{T-2}, s_{T-1})}}{\sum_{s'_{T-1}} e^{-\lambda Q(s_{T-2}, a_{T-2}, b_{T-2}, s'_{T-1})}},$$

where:

$$\begin{aligned} Q(s_{T-2}, a_{T-2}, b_{T-2}, s_{T-1}) = b_{T-2} R(s_{T-2}, a_{T-2}, s_{T-1}) \\ + (1 - b_{T-2}) \widetilde{R}(s_{T-2}, a_{T-2}, s_{T-1}) \\ + V(s_{T-1}, b_{T-2}). \end{aligned}$$

Similarly as before, we can compute the optimal action a_{T-2} as:

$$\pi^*(s_{T-2}, b_{T-3}) = \operatorname*{argmax}_{a_{T-2}} Q_R \left(s_{T-2}, a_{T-2}, \frac{b_{T-3}}{Z(s_{T-2}, a_{T-2}, b_{T-3})} \right),$$

where Q_R is, similarly to Q, augmented with the future expectation:

$$\begin{aligned} Q_R(s_{T-2}, a_{T-2}, b_{T-2}) = \sum_{s_{T-1}} \tau^*(s_{T-1}|s_{T-2}, a_{T-2}, b_{T-2}) \Big(R(s_{T-2}, a_{T-2}, s_{T-1}) \\ + Q_R(s_{T-1}, \pi^*(s_{T-1}, b_{T-2}), \frac{b_{T-2}}{Z'(s_{T-1}, b_{T-2})}) \Big). \end{aligned}$$

Finally, the objective value is:

$$\begin{aligned} V(s_{T-2}, b_{T-3}) = \sum_{s_{T-1}} \tau^* \left(s_{T-1}|s_{T-2}, \pi^*(s_{T-2}, b_{T-3}), \frac{b_{T-3}}{Z'(s_{T-2}, b_{T-3})} \right) Z'(s_{T-2}, b_{T-3}) \\ \Bigg(\frac{b_{T-3}}{Z'(s_{T-2}, b_{T-3})} R(s_{T-2}, \pi^*(s_{T-2}, b_{T-3}), s_{T-1}) \\ + \ \left(1 - \frac{b_{T-3}}{Z'(s_{T-2}, b_{T-3})} \right) \widetilde{R}(s_{T-2}, \pi^*(s_{T-2}, b_{T-3}), s_{T-1}) \\ + \ V \left(s_{T-1}, \frac{b_{T-3}}{Z'(s_{T-2}, b_{T-3})} \right) \\ - \ Q \left(s_{T-2}, \pi^*(s_{T-2}, b_{T-3}), \frac{b_{T-3}}{Z'(s_{T-2}, b_{T-3})}, s_{T-1} \right) \\ - \ \lambda^{-1} \log \sum_{s'_{T-1}} e^{-\lambda Q \left(s_{T-2}, \pi^*(s_{T-2}, b_{T-3}), \frac{b_{T-3}}{Z'(s_{T-2}, b_{T-3})}, s'_{T-1} \right)} \Bigg) \\ = Z'(s_{T-2}, b_{T-3}) \operatorname*{softmin}_{s_{T-1}} Q \left(s_{T-2}, \pi^*(s_{T-2}, b_{T-3}), \frac{b_{T-3}}{Z'(s_{T-2}, b_{T-3})}, s_{T-1} \right). \end{aligned}$$

Continuing this procedure backwards, alternating the computation of transitions and optimal actions, completes the proof. □

A.3 Proof of Corollary 1

Proof of Corollary 1. We start by showing that:

$$\boldsymbol{\Psi}(s_t, b_{t-1}) = \mathbb{E}_{\widetilde{\pi},\tau^*}\left[\sum_{i=t}^{T-1} \boldsymbol{\phi}(s_i, a_i, s_{i+1}) | s_t, b_{t-1}\right]. \tag{18}$$

This can be easily proven by induction using similar steps as in the proof of Theorem 2. Clearly, at time step $T-1$:

$$\boldsymbol{\Psi}(s_{T-1}, b_{T-2}) = \sum_{a_{T-1}} \widetilde{\pi}(a_{T-1}|s_{T-1}) \sum_{s_T} \tau^*(s_T|s_{T-1}, a_{T-1}, b_{T-1}) \boldsymbol{\phi}(s_{T-1}, a_{T-1}, s_T).$$

Recalling the belief update rule (8), we have:

$$b_{T-1} = \frac{b_{T-2}\mathbb{1}\left[a_{T-1} = \pi^*(s_{T-1}, b_{T-2})\right]}{b_{T-2}\mathbb{1}\left[a_{T-1} = \pi^*(s_{T-1}, b_{T-2})\right] + (1 - b_{T-2})\widetilde{\pi}(a_{T-1}|s_{T-1})},$$

where the indicator is due to the fact that π^* is deterministic. Notice that this is equivalent to the more concise update rule given in the main statement. Thus, (18) holds at time $T-1$. Assume now that (18) holds at time $t+1$. Let us show that this implies (18) holds at time t as well. We have:

$$\begin{aligned}
\boldsymbol{\Psi}(s_t, b_{t-1}) &= \sum_{a_t} \widetilde{\pi}(a_t|s_t) \sum_{s_{t+1}} \tau^*(s_{t+1}|s_t, a_t, b_t) \left[\boldsymbol{\phi}(s_t, a_t, s_{t+1}) + \boldsymbol{\Psi}(s_{t+1}, b_t)\right] \\
&= \sum_{a_t} \widetilde{\pi}(a_t|s_t) \sum_{s_{t+1}} \tau^*(s_{t+1}|s_t, a_t, b_t) \left(\boldsymbol{\phi}(s_t, a_t, s_{t+1}) + \mathbb{E}_{\widetilde{\pi},\tau^*}\left[\sum_{i=t+1}^{T-1} \boldsymbol{\phi}(s_i, a_i, s_{i+1}) | s_{t+1}, b_t\right]\right) \\
&= \mathbb{E}_{\widetilde{\pi},\tau^*}\left[\sum_{i=t}^{T-1} \boldsymbol{\phi}(s_i, a_i, s_{i+1}) | s_t, b_{t-1}\right],
\end{aligned}$$

where the last equation holds since b_{t-1} is again correctly updated as given in the main statement. Finally, since $b_0 = 0.5$ is the only possible initial belief state:

$$\begin{aligned}
\boldsymbol{\kappa}_{\boldsymbol{\phi}}(\widetilde{\pi}, \tau^*) &= \mathbb{E}_{\widetilde{\pi},\tau^*}\left[\sum_{t=1}^{T-1} \boldsymbol{\phi}(s_t, a_t, s_{t+1})\right] \\
&= \sum_{s_1} p(s_1) \mathbb{E}_{\widetilde{\pi},\tau^*}\left[\sum_{t=1}^{T-1} \boldsymbol{\phi}(s_t, a_t, s_{t+1}) | s_1\right] \\
&= \sum_{s_1} p(s_1) \boldsymbol{\Psi}(s_1, b_0).
\end{aligned}$$

This concludes the proof. □

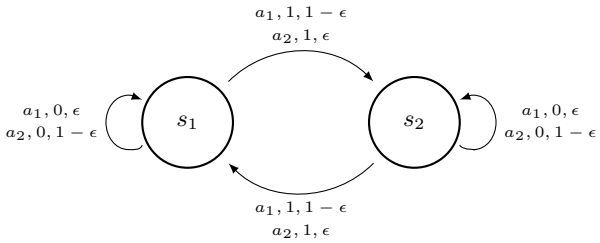


Figure 2: The two-state MDP used throughout this section. The notation a, r, p on the arcs denotes a stochastic transition performed by action a with probability p and reward r. The initial state is s_1.

B A Simple Two-State MDP

In order to provide a better understanding of the proposed approach, we begin by considering a simple example where we can clearly appreciate the effect of all elements involved in our algorithm.

Consider the simple MDP of Figure 2. The MDP has two states (s_1 and s_2) and two actions (a_1 and a_2). Action a_1 forces the system to change state, while a_2 stays in the current state. All actions fail with probability $\epsilon \in [0, 0.5]$ and succeed with probability $1 - \epsilon$. The reward is 1 whenever the state is changed, 0 otherwise. The system starts deterministically in state s_1 and runs for $T = 4$ time steps. Clearly, the optimal policy is to execute a_1 in all states for $\epsilon < 0.5$, while any policy is optimal for $\epsilon = 0.5$.

We now establish some common parameters that will be used throughout this section. Then, we analyze the effect of each of them separately. First, we need to specify features that allow us to define the marginal constraints of Section 3.1. In order to better visualize the results, we adopt only one feature (so that our objective is a function defined on the real line). A useful property we would like to capture is the fact that action a_1 frequently changes the current state. Thus, we define our (scalar) feature function as:

$$\phi(s, a, s') = \begin{cases} 1 & \text{if } s \neq s' \text{ and } a = a_1 \\ 0 & \text{otherwise} \end{cases}$$

Notice that the feature expectations $\kappa_\phi(\pi, \tau)$ of some policy π under some transition probabilities τ are equivalent to the expected number of times in which a_1 changes state. Thus, if we assume π to choose a_1 with a fixed probability p (independently of the state) and that the horizon is T, we have:

$$\kappa_\phi(\pi, \tau_\epsilon) = p(1 - \epsilon)(T - 1),$$

where τ_ϵ denotes the transition probabilities of our two-state MDP with probability of failure ϵ. Unless otherwise stated, we adopt the true feature expectations instead of the sample statistics $\hat{\kappa}_\phi$.

We consider a uniform reference policy $\tilde{\pi}$ (hence $p = 0.5$). Notice that, under uniform $\tilde{\pi}$, the set of all reachable belief states in a fixed horizon T scales linearly with T rather than exponentially[4]:

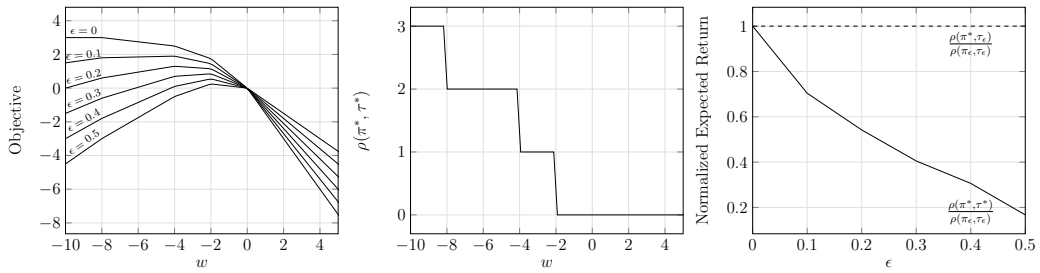
$$\mathcal{B} = \{0\} \cup \left\{ \frac{0.5}{0.5 + 0.5^t} \mid t = 1, \ldots, T \right\}$$

This fact can be better verified from the belief update rule (8). Thus, in this small domain we can adopt the full belief set so that no approximation error will be incurred. With $T = 4$, we have $\mathcal{B} = \{0, 0.5, 0.67, 0.8, 0.89\}$.

Finally, we use no entropy, l_1, or l_2^2 regularization.

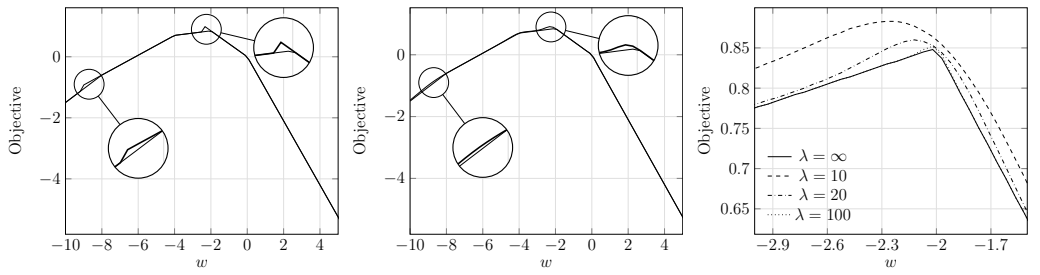
The effect of stochasticity We start by analyzing how the objective function (6) and the corresponding solution change for different values of ϵ, i.e., for different levels of stochasticity in the underlying system. From Figure 3a, we can notice that the optimal solution lies in an interval when the system is deterministic, while, as we increase the stochasticity through ϵ, it moves on an increasingly peaked corner. Furthermore, the algorithm becomes more conservative for larger ϵ, as can be noticed from the expected return achieved by the min-max solution for different weights

[4]Whatever action is chosen, $\tilde{\pi}(a|s) = 0.5$, and, thus, we need not enumerate all possible trajectories.



(a) (b) (c)

Figure 3: (a) The objective function for different values of ϵ. The higher is the stochasticity, the more peaked is the global optimum. (b) The robust expected return for different values of the Lagrange multiplier w. (c) The normalized worst-case performance (solid line) and the true expected return (dashed line) achieved by the optimal solution for different values of ϵ. Here π_ϵ denotes the optimal policy under dynamics τ_ϵ, while (π^*, τ^*) denote the min-max solution at the corresponding optimum.



(a) (b) (c)

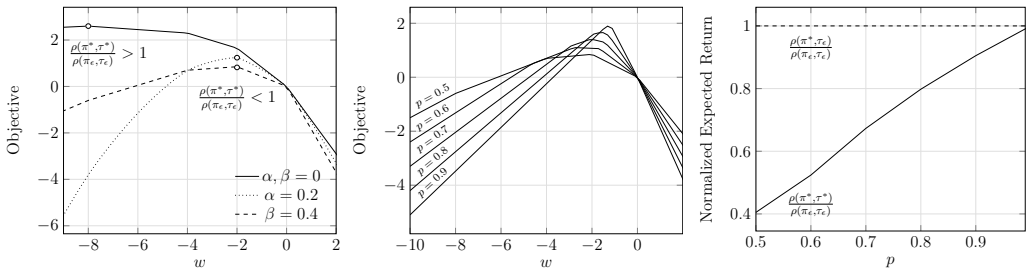
Figure 4: (a) Approximation error due to missing belief states, and (b) how it can be alleviated by adding entropy regularization ($\lambda = 20$). (c) The objective function for different entropy regularizers.

w (Figure 3b). This can be better observed from Figure 3c, which shows the true and worst-case expected returns achieved by the global maximizer for each ϵ, normalized by the performance of the corresponding optimal policy. This decrease in the worst-case performance is expected since one feature is not sufficient for constraining the solution with high stochasticity. Interestingly, the performance under the true dynamics τ_ϵ remains optimal for all ϵ, which implies that the chosen feature is sufficient for characterizing the optimal policy.

The effect of misspecified belief states When a non-uniform reference policy is used, the reachable belief states cannot be efficiently enumerated as we did before. Thus, in practice we approximate them with a smaller set. We now investigate the consequences of such approximation on the objective function. We consider discretizing the belief space uniformly with 0.1 step, thus obtaining the set $\widetilde{\mathcal{B}} = \{0, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Notice that values in $(0, 0.5)$ cannot occur and are safely removed. Figure 4a shows the resulting approximate objective function (from now on we use $\epsilon = 0.3$). As expected, the two missing belief states, which are now approximated with the closest one in $\widetilde{\mathcal{B}}$, result in two small deviations from the ideal objective. These can be easily alleviated by using a more fine-grained discretization or, as we shall see, by smoothing the objective.

The effect of entropy regularization Although subgradient methods are guaranteed to converge under general assumptions, optimizing an almost non-differentiable objective function like the one of Figure 3a can be very slow for high-dimensional problems[5]. Adding a small amount of entropy regularization can dramatically improve the smoothness of the objective function and the corresponding gradient, thus simplifying its optimization. Figure 4c shows an example. Interestingly, the entropy regularizer can also be adopted to alleviate the approximation error due to discretizing the belief space (Figure 4b), which can be much more efficient than increasing the size of $\widetilde{\mathcal{B}}$.

[5]Recall that our objective is to find a point with zero gradient, so that the feature expectations correctly match the sample statistics.



(a) (b) (c)

Figure 5: (a) The objective function using l_1 and l_2 regularization. (b) The objective functions of different reference policies. (c) Expected returns for increasingly good reference policies.

The effect of estimating feature expectations In practice, the feature expectations cannot be computed in closed-form as we did so far but must be estimated from given trajectories. Consequently, l_1 or l_2^2 regularization of the dual variables is typically required to enforce softened constraints, as the estimation error might make the exact optimization problem of Definition 2 infeasible. Even if the problem remained feasible, a deviation from the ideal feature expectations could void our robustness guarantees since the true transition probabilities might not be part of the uncertainty set anymore. We now show an example demonstrating this fact. Assume that the previously introduced feature expectations κ_ϕ are now estimated from a small amount of data, obtaining a value $\hat{\kappa}_\phi$ that deviates from the true one by 0.4, i.e., $\hat{\kappa}_\phi = \kappa_\phi + 0.4$. The solid line in Figure 5a shows the resulting objective (once again, we fix $\epsilon = 0.3$). As we can notice, the optimum has been considerably shifted from its ideal value of Figure 3a. Furthermore, the worst-case expected return achieved at such point, $\rho(\pi^*, \tau^*)$, is now larger than the one achieved by the optimal policy $\rho(\pi_\epsilon, \tau_\epsilon)$, which implies that our solution is not robust anymore. However, we can easily solve this issue by regularizing the Lagrange multiplier w. Figure 5a shows the results when adding l_1 regularization with parameter β (which enforces the soft constraint $|\kappa_\phi - \hat{\kappa}_\phi| \leq \beta$) and l_2^2 regularization with parameter α. As we can see, for properly chosen parameters, the optimal weight coincides with the original one and so does the worst-case expected return.

The effect of the reference policy So far, we have been using a uniform (non-informative) reference policy. Let us now investigate what happens when the reference policy gets closer to the optimal one. We set $\tilde{\pi}(a_1|s) = p$ and $\tilde{\pi}(a_2|s) = 1 - p$ for all states s. Figure 5b shows how the objective function varies when increasing p above 0.5. Furthermore, Figure 5c shows that the worst-case expected return gets closer to the optimal one as p increases (i.e., as the reference policy becomes optimal). This is again expected since, when the two policies involved in our min-max problem are equivalent, the solution becomes Markovian. Although the single feature we have been using so far is not enough for constraining history-dependent transition dynamics, it suffices for Markovian ones, thus resulting in an optimal worst-case expected return.

C Additional Details on the Experiments

C.1 Gridworld

We provide additional details on the gridworld experiments.

MC In order to define our marginal constraints, we consider 6 features:

$$\phi_1(s,a,s') = \mathbb{1}\left[a = \text{UP} \wedge (s'_y > s_y \wedge s_y < N \vee s'_y = s_y \wedge s_y = N)\right]$$
$$\phi_2(s,a,s') = \mathbb{1}\left[a = \text{RIGHT} \wedge (s'_x > s_x \wedge s_x < N \vee s'_x = s_x \wedge s_x = N)\right]$$
$$\phi_3(s,a,s') = \mathbb{1}\left[a = \text{DOWN} \wedge (s'_y < s_y \wedge s_y > 0 \vee s'_y = s_y \wedge s_y = 0)\right]$$
$$\phi_4(s,a,s') = \mathbb{1}\left[a = \text{LEFT} \wedge (s'_x < s_x \wedge s_x > 0 \vee s'_x = s_x \wedge s_x = 0)\right]$$
$$\phi_5(s,a,s') = \mathbb{1}\left[|s_x - s'_x| + |s_y - s'_y| > 1\right]$$
$$\phi_6(s,a,s') = \mathbb{1}\left[s = s' \wedge s_x \in (1,N) \wedge s_y \in (1,N)\right]$$

Intuitively, the first 4 features are enabled by successful transitions. For instance, feature ϕ_1 is 1 whenever action UP correctly increases the y coordinate of the current state (or stays at the upper border). The last two features encode deterministic properties of the environment. ϕ_5 is enabled whenever a transition to a non-adjacent state is performed. This is impossible in the true MDP, so the corresponding feature expectation must be zero. Similarly, ϕ_6 is enabled whenever an action stays in the current state while not at the borders. This is, again, impossible in the true MDP.

For solving the dynamic program of Algorithm 1, we enumerate all possible belief states reachable in a horizon $T = 6$[6]. Notice that, for longer horizons, all belief values are above 0.999 and can be neglected by introducing a negligible approximation error.

We set $\lambda = 100$, while we add l_2^2 entropy regularization with parameter $\alpha = 0.01$ to the dual objective. We optimize the latter using standard gradient ascent, stopping when the l_∞ norm of the gradient goes below 10^{-5} or 200 iterations are reached.

LR We adopt the same features as for MC. Given the set of input trajectories, we learn a multi-class logistic model to predict the next state s' from a state-action pair s, a. That is, we learn the conditional probabilities:

$$P(s'|s,a) \propto e^{\boldsymbol{w}^\intercal \boldsymbol{\phi}(s,a,s')},$$

with $\boldsymbol{w}$ as the parameter computed by the logistic model.

RECT We use l_1 constrained uncertainty sets computed using Hoeffding's inequality with 95% confidence level, as described in Appendix A in [7].

FQI We use extra-trees with an ensemble of 50 approximators, with a minimum number of 2 samples to split a node. We run the algorithm for 50 iterations and evaluate the greedy policy with respect to the resulting Q-function.

MLE We estimate the transition probabilities using maximum likelihood:

$$P(s'|s,a) = \frac{\#[s,a,s']}{\#[s,a]},$$

where $\#[s,a]$ denotes the number of times action a was taken from state s in the given trajectories, and similarly for $\#[s,a,s']$.

Additional results We now investigate the sensitivity of our approach to the choice of the entropy regularization parameter λ. In order to address this point, we repeat the experiments for a fixed gridworld of size 11×11 using different values of λ, with all other parameters as before. Figure 6 shows how the expected return under the true and estimated dynamics vary as a function of λ. As expected, we notice a clear performance drop under the estimated dynamics as λ increases, i.e.,

[6] As described in Appendix B, when using a uniform reference policy we can efficiently enumerate all reachable belief states in a horizon T.

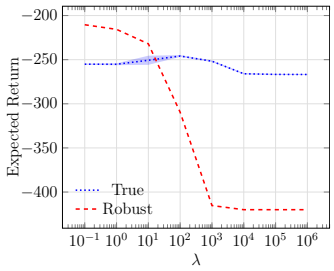


Figure 6: Expected returns as a function of the entropy regularizer λ

as the regularization is removed. Interestingly, the performance under the true dynamics decreases by a much smaller margin. Furthermore, the robust expected return becomes higher than the true one only for very small values of λ, i.e., when keeping high entropy becomes more important than minimizing the reward. This intuitively suggests that, using intermediate values of λ, we are able to find less conservative solutions while preserving robustness guarantees under certain assumptions on the unknown MDP. Proving this statement from a theoretical perspective would be of great practical interest and is left for future work.

C.2 Invasive Species

We provide additional details on the invasive species experiment.

MC We consider 14 different features. The first 10 are binary and discretize the population in bins of width 50. Thus,

$$\phi_j(s, a, s') = \mathbb{1}\left[s \in (50(j-1), 50j]\right] \quad \forall j = 1, \dots, 10.$$

The next 2 features encode population intervals in which the two available actions transition with very high probability according to our dynamics:

$$\phi_{11}(s, a, s') = \mathbb{1}\left[a \neq \mathrm{C} \wedge (s' < s - 30 \vee s' > s + 100)\right]$$
$$\phi_{12}(s, a, s') = \mathbb{1}\left[a = \mathrm{C} \wedge (s' > s + 30 \vee s' < s - 100)\right],$$

where 'C' represents the control action. Thus, these features are enabled whenever an action fails at transitioning to an interval containing the true next state with high probability. Finally, the last 2 features encode the (absolute) population change after an action has been applied:

$$\phi_{13}(s, a, s') = |s - s'| \mathbb{1}\left[a \neq \mathrm{C}\right]$$
$$\phi_{14}(s, a, s') = |s - s'| \mathbb{1}\left[a = \mathrm{C}\right]$$

We discretize belief states in 18 different values in $[0.1, 0.9]$ and we add entropy regularization with $\lambda = 100$. For the experiment with 50 trajectories, we add l_1 regularization with $\beta = 0.01$ for all feature components except ϕ_{11} and ϕ_{12} (for which $\beta = 0$). As usual when given more data, we decrease β to 0.001 for the experiment with 100 trajectories. We run MC for at most 200 iterations, stopping when the l_∞-norm of the gradient goes below 10^{-4}.

Other algorithms We keep the same configuration as in the gridworld experiments for all other algorithms. For LR, we use the new feature functions.