An Experimental Study of Advice in Sequential Decision-Making under Uncertainty

Florian Benavent and **Bruno Zanuttini**
Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

Abstract

We consider sequential decision making problems under uncertainty, in which a user has a general idea of the task to achieve, and gives advice to an agent in charge of computing an optimal policy. Many different notions of advice have been proposed in somewhat different settings, especially in the field of inverse reinforcement learning and for resolution of Markov Decision Problems with Imprecise Rewards.

Two key questions are whether the advice required by a specific method is natural for the user to give, and how much advice is needed for the agent to compute a good policy, as evaluated by the user. We give a unified view of a number of proposals made in the literature, and propose a new notion of advice, which corresponds to a user telling why she would take a given action in a given state. For all these notions, we discuss their naturalness for a user and the integration of advice. We then report on an experimental study of the amount of advice needed for the agent to compute a good policy. Our study shows in particular that continual interaction between the user and the agent is worthwhile, and sheds light on the pros and cons of each type of advice.

Introduction

We consider sequential decision making under uncertainty, by an autonomous agent solving tasks on behalf of a user. Such situations arise naturally in practice, for instance, GPS navigation devices compute optimal routes for the driver. We are especially interested in applications where the agent has information about the dynamics of the task to achieve, but lacks information about the precise goal, or preferences, of the user. In the case of a GPS, this corresponds to the device having all information about the roads and traffic (dynamics) and some information about the goal (starting place and destination), but lacking information about whether the driver wants to, *e.g.*, avoid going though such place, or keep close to others (like rest areas when driving with young kids). We refer the reader to Azaria et al. (2016) for more examples.

We assume that the task to be solved is modelled as a Markov Decision Problem (MDP), thus taking into account uncertainty in the outcome of actions at execution time. In that setting, natural solutions are *policies* rather than plans. In the GPS example, this means that the device computes a contingent plan taking into account, not only the intended, optimal path to destination, but also alternative paths (because of, *e.g.*, unforeseen traffic jams, blocked roads, etc.).

It is well-known that the optimal policy for a given MDP is highly dependent on the precise reward function. However, it is typically difficult for a user to give precise numerical values for rewarding states or transitions between states. Hence many authors have studied MDPs with *Imprecise Reward functions*, that is, with a set of candidate reward functions rather than a single one. Approaches encompass inverse reinforcement learning and learning from demonstrations, preference-based reinforcement learning, ordinal reward MDPs, minimax regret policies, etc. (see related work).

At the heart of most approaches is some notion of information communicated by the user to the agent: demonstrated (portions of) trajectories, preferences between trajectories, etc. Such information can be obtained by observing the user, or by actively querying her whenever more information is needed for solving the task. Orthogonally, the information can consist of (near-)optimal actions to take at given states, to information about the real reward function, etc.

We are interested in what information is given by the user to the agent, which we call *advice*. We give a unified view of the most important notions in the literature, focusing on notions which, we argue, are natural for a user to give. We introduce a new, natural notion, formalising a user giving the action to take in a given state and telling what outcome made her choose this action. We extend known algorithms for incorporating such advice in the computation of a minimax regret policy. We finally report on an experimental study on random MDPs with different structures, in which we compare the various notions with different interaction scenarios.

Related Work

It has long been observed that a drawback of MDPs, as a framework for modelling sequential decision making tasks, is that a numerical reward function must be defined, and that the optimal policies are quite sensitive to its precise values. A typical setting where this is problematic is in medical decision support, where one would need to give a numerical cost to the death of a patient, implying possible tradeoffs with many other being cured of a simple cold, for instance. The point is that under very reasonable postulates, there is indeed a numerical reward function which captures exactly the task which the user has in mind (Weng 2012), but that it is not realistic to expect the user to know it.

To cope with this, a first line of approaches considers *or-*

dinal (Weng 2012), or *preference-based* MDPs (Fürnkranz et al. 2012), in which rewards or trajectories are ordered but have no numerical values. This results in MDPs with several nondominated policies. For choosing one, methods have been proposed that ask the user the minimum amount of additional information about how rewards compare and can be traded off with each other, either in an offline planning (Weng and Zanuttini 2013) or in a reinforcement learning setting (Busa-Fekete et al. 2013). This information is typically obtained as answers to *queries* asked to the user.

Another group of approaches consists of considering the problem as one of *inverse reinforcement learning* (Ng and Russell 2000). In this setting, the dynamics of the problem is known to the agent but the reward function is considered as an unknown, the value of which (or enough information about this value) can be discovered by the agent by observing optimal or near-optimal policies. Some approaches, called *model-based*, aim at estimating the underlying reward function first and then deducing an optimal policy, while other, called *direct* or *model-free*, aim at computing an optimal policy directly from the demonstrations. We refer the reader to (Piot, Geist, and Pietquin 2013) for a recent survey.

Finally, another (closely related) group of approaches takes a decision-theoretic point of view, and defines an *Imprecise Reward MDP* (IRMDP) to be a set of MDPs, all with a different reward function, hence modelling uncertainty about the real one. Different solution concepts can be given, but the one most studied is minimax regret (Bell 1982; Xu and Mannor 2009; Regan and Boutilier 2009). With this view, ignorance of the real reward function is directly coped with, and an optimal policy is one which copes best with it: in the case of minimax regret, which is most robust to adversaries choosing the real reward function. This line of work also studies methods for reducing the uncertainty using queries to the user, as far as needed for reducing regret (Regan and Boutilier 2009; 2011; Alizadeh, Chevaleyre, and Zucker 2015; Ahmed et al. 2017). In this paper, we heavily build on this view and these techniques.

Preliminaries

A *Markov Decision Problem*, or MDP (Puterman 2005), is a tuple $\langle S, A, T, R, \gamma \rangle$, where: S, A are finite sets of *states* and *actions*; $T : S \times A \times S' \to [0, 1]$ is a *transition function*, with $T(s, a, s')$ the probability that when action a is taken in state s, the system evolves to state s'; $R : S \to \mathbb{R}$ is the *reward function*, with $R(s)$ the immediate reward (or penalty, if negative) obtained when the current state is s; and $\gamma \in [0, 1[$ is the *discount factor*.[1]

Write $\Delta(X)$ for the set of all probability distributions over a set X. The transition function T is made of one probability distribution for each state s and action a, written $T_{s,a} \in \Delta(S)$. A (stationary) *deterministic* policy is a function $\pi : S \to A$, which for each state s prescribes an action a to take, and more generally a (stationary) *stochastic* policy is a function $\tilde{\pi} : S \to \Delta(A)$; we write $\tilde{\pi}(s, a)$ for $\tilde{\pi}(s)(a)$, that is, for the probability that a is chosen in state s according to $\tilde{\pi}$. We write $f^{\alpha}_{\tilde{\pi}}$ for the *occupation frequency* of $\tilde{\pi}$ wrt an initial distribution α on S: $f^{\alpha}_{\tilde{\pi}}(s, a) = \mathbf{E}(\sum_{t=0}^{\infty} \gamma^t \Pr(S^t = s, A^t = a))$, where expectation is taken over trajectories defined by $S_0 \sim \alpha$, $A_t \sim \tilde{\pi}(S_t)$, and $S_{t+1} \sim T_{S_t, A_t}$.

In this paper, we define the quality of policies respective to the *infinite horizon discounted criterion*: the *value* of $\tilde{\pi}$ at a state s is defined to be the expectation of cumulative discounted rewards obtained when following $\tilde{\pi}$ starting in s:

$$V^{\tilde{\pi}}(s) = \mathbf{E}(\sum_{t=0}^{\infty} \gamma^t R(S_t))$$

where S_t is a random variable on S for $t = 0, 1, \ldots$, and expectation is taken over trajectories defined by $S_0 = s, A_t \sim \tilde{\pi}(S_t), S_{t+1} \sim T_{S_t, A_t}$. The *quality function* is given by

$$Q^{\tilde{\pi}}(s, a) = R(s) + \gamma \sum_{s' \in S} T(s, a, s') V^{\tilde{\pi}}(s')$$

It is well-known that an MDP M always has an optimal *deterministic, stationary* policy π which maximises $V^{\pi}(s)$ at all states s. We denote by π^*_M such a policy, by $V^*_M : S \to \mathbb{R}$ its value function, and by $Q^*_M : S \times A \to \mathbb{R}$ its Q-function.

Imprecise Reward MDPs We adopt the setting of *Imprecise Reward MDPs* (IRMDPs). Formally, an IRMDP (Regan and Boutilier 2009) is defined to be a tuple $\langle S, A, T, \tilde{R}, \gamma \rangle$, where S, A, T, γ are as for an MDP, and $\tilde{R}$ is a (possibly infinite) set of reward functions on S; $\tilde{R}$ models the uncertainty about the *real* reward function which would precisely define the task at hand. To avoid confusion, we use notation $\tilde{M}$ for IRMDPs, keeping M for (standard) MDPs.

We are interested in the case when $\tilde{R}$ is given as a polytope. Precisely, writing $S = \{s_1, \ldots, s_n\}$ for the set of states, we assume that $\tilde{R}$ is the set of all solutions of a linear system of the form $C \cdot \vec{r} \geq \vec{d}$, where C is a $k \times n$-matrix, $\vec{d}$ is a k-dimensional column vector, and $\vec{r}$ is the column vector $(R(s_1), \ldots, R(s_n))^T$. We call such IRMDPs *linear*.

The restriction to linear IRMDPs is very common in the literature (Regan and Boutilier 2009; 2010; Weng and Zanuttini 2013; Alizadeh, Chevaleyre, and Zucker 2015; Ahmed et al. 2017). It is indeed very natural; in particular, it encompasses all cases where the modeller of the task specifies an interval instead of a precise value for some $R(s)$'s.

Solving IRMDPs Xu and Mannor (2009) and Regan and Boutilier (2009) propose to assess the value of a stochastic policy for an IRMDP using max regret (MR). The intuition is that a good policy should not bet too much on one of the reward functions, and should rather try to minimise its regret if an adversary chooses the actual one.

Definition 1 (max regret) *Let $\tilde{M} = \langle S, A, T, \tilde{R}, \gamma \rangle$ be an IRMDP, $\tilde{\pi}$ be a stochastic policy, and α be a distribution on S. For $R \in \tilde{R}$, the* regret *of $\tilde{\pi}$ wrt $\tilde{M}$ and R is defined by $\rho^{\alpha}_{\tilde{M}}(\tilde{\pi}, R) = \max_{g \in \mathcal{F}_{\alpha}} (R \cdot g - R \cdot f^{\alpha}_{\tilde{\pi}})$, where $\mathcal{F}_{\alpha}$ is the set of all valid action occupation frequencies for $\tilde{M}$ wrt α. The* max regret *of $\tilde{\pi}$ is defined by $MR^{\alpha}_{\tilde{M}}(\tilde{\pi}) = \max_{R \in \tilde{R}} \rho^{\alpha}_{\tilde{M}}(\tilde{\pi}, R)$.*[2]

[1] Our study can be easily generalised to $R : S \times A \times S \to \mathbb{R}$

[2] It is easy to see that $R \cdot f^{\alpha}_{\tilde{\pi}}$ is equal to $\sum_s \alpha(s) V^{\tilde{\pi}}(s)$.

$\min_{\tilde{\pi},\rho}$	ρ	
s.t.	$R \cdot g_R - R \cdot f \leq \rho$	$(\forall \langle g_R, R\rangle \in \underline{\tilde{R}})$
	$\gamma E^T f + \alpha = 0$	

$\max_{Q,V,I,r}$	$\alpha \cdot V - r \cdot f$	
s.t.	$Q_a = r_a + \gamma P_a V$	$(\forall a \in A)$
	$V \geq Q_a$	$(\forall a \in A)$
	$V \leq (1 - I_a)M_a + Q_a$	$(\forall a \in A)$
	$Cr \leq d$	
	$\sum_a I_a = 1$	
	$I_a \in \{0, 1\}$	
	$M_a \in M_\top - M_a^\perp$	

Figure 1: Master problem (top) and subproblem (bottom) for $\tilde{M} = \langle S, A, T, \tilde{R}, \gamma\rangle$ with $\tilde{R}$ given by $C \cdot \vec{r} \geq \vec{d}$.

When unambiguous, we simply write $\rho(\tilde{\pi}, R)$, $MR(\tilde{\pi})$. A policy $\tilde{\pi}^*$ is said to be ***MMR-optimal*** (for $\tilde{M}$) if its max regret is minimum: $MR^\alpha_{\tilde{M}}(\tilde{\pi}^*) = \min_{\tilde{\pi}\in(\Delta(A))^S} MR^\alpha_{\tilde{M}}(\tilde{\pi})$.

Several algorithms have been proposed for computing an MMR-optimal policy (Regan and Boutilier 2009; da Silva and Costa 2011; Alizadeh, Chevaleyre, and Lévy 2016), either exactly or approximately. Among these, we will build on Regan and Boutilier's approach (2009). Given an IRMDP $\tilde{M} = \langle S, A, T, \tilde{R}, \gamma\rangle$ with $\tilde{R}$ given by a linear system $C \cdot \vec{r} \geq \vec{d}$, the idea is to use Bender's Decomposition (Benders 1962). A *master problem* computes a policy which minimises the maximal regret with respect to the reward functions in a finite subset $\underline{\tilde{R}}$ of $\tilde{R}$. Given such a candidate policy $\tilde{\pi}$, a *subproblem* searches for a reward function $R \in \tilde{R}$ which maximises the regret of $\tilde{\pi}$, R is added to $\underline{\tilde{R}}$, and the process repeats until a fix-point is reached.

It turns out that the master problem can be formulated as a linear program over variables $f(s, a) \in [0, 1]$ ($s \in S, a \in A$, representing the occupation frequency of an MMR policy) and ρ (representing the min max regret). Figure 1 (top) gives this program. The first constraint occurs once per couple $\langle g_R, R\rangle$ which has been found by the subproblem so far, where R is a regret maximising reward function as found by the subproblem, and g_R is the occupation frequency of the corresponding adversary policy. The second constraint ensures that f is a valid occupation frequency (for α). A solution *policy* can be retrieved from f through $\tilde{\pi}(s, a) = f(s, a)/\sum_{a'} f(s, a')$. Now, the subproblem can be formulated as a mixed 0/1 linear program as shown on Figure 1 (bottom). There, f is the occupation of the current MMR solution $\tilde{\pi}$ found by the master problem, r is a vector of real-valued variables representing a reward function for which $\rho(\tilde{\pi}, r)$ is maximal, Q, V are vectors of real-valued variables representing the quality function and value function of a deterministic policy π_g maximising $r \cdot g - r \cdot f$ (with g the occupation function of π_g wrt α) or, equivalently, $\alpha \cdot V - r \cdot f$, and I consists of 0/1 variables such that $I_a(s)$ is 1 iff $\pi_g(s)$ is a (cf. Regan and Boutilier (2009) for more details).

Max Regret under Advice

We consider agents who know the dynamics (S, A, T, γ) of the target MDP, but know a set $\tilde{R}$ instead of the reward function R. However, we consider situations where there is a precise task to be solved, as formalised by a precise numerical reward function R. This function must be understood as the one in the user's mind, but we insist that we do *not* assume that the user knows it precisely. Note that, under reasonable postulates, there is necessarily such a function which models the user's preferences (Weng 2011, Theorem 2).

We call *advice context* a tuple $\langle S, A, T, \tilde{R}, \gamma, R\rangle$, where $\langle S, A, T, \tilde{R}, \gamma\rangle$ is an IRMDP and $R \in \tilde{R}$ is a reward function over S: $\langle S, A, T, \tilde{R}, \gamma\rangle$ is the information available to the agent, and the MDP $M = \langle S, A, T, R, \gamma\rangle$ is the *target* task.

Advice In general, by advice we formalise the hints a user can give to the agent about the target reward function. Hence we view a piece of advice as a predicate on reward functions.

A natural manner for a user to give advice about the precise task at hand, is to point at decisions which she would take in specific states. This is the point of view taken by learning from demonstrations (Maclin and Shavlik 1996; Judah et al. 2010; Azaria et al. 2016; Cederborg et al. 2015). Importantly, giving such advice does not require the user to know anything about the components of the target task. Rather, the agent can simply run its MMR policy (for $\tilde{M}$) until the user says "here, you should have done that".

Definition 2 (action advice) *Let $\langle S, A, T, \tilde{R}, \gamma, R\rangle$ be an advice context, write π^* for an optimal policy for $M = \langle S, A, T, R, \gamma\rangle$, and $\tilde{\pi}^*$ for an MMR policy for $\tilde{M} = \langle S, A, T, \tilde{R}, \gamma\rangle$. An* action advice *is a pair $(s, a) \in S \times A$ satisfying $Q^M_{\pi^*}(s, a) \geq Q^M_{\pi^*}(s, \tilde{\pi}^*(s))$.*

We emphasise that in Definition 2, the Q-values are all evaluated in the target model M and assuming that the policy executed after them is the target policy π^*. Hence the advice indeed makes sense, in that it does not require the user to know the agent's model.

Another natural type of advice is about the *optimal* action in a given state, still according to the user's point of view.

Definition 3 (optimal action advice) *Let $\langle S, A, T, R, \gamma\rangle$ be an MDP, and π^* be an optimal policy for it. An* optimal action advice *is a pair $(s, a) \in S \times A$ satisfying $a = \pi^*(s)$.*

Again, such advice does not require the user to know the agent's model. From the point of view of the agent, the informative content of such advice is

$$\forall a' \in A, \sum_{s'\in S} T(s, a, s')V^M_{\pi^*}(s') \geq \sum_{s'\in S} T(s, a', s')V^M_{\pi^*}(s')$$

Because the inequality is large, such advice may be non-informative, which is reminiscent of the fact that the basic inverse reinforcement learning problem is ill-posed (Ng and Russell 2000). However, contrary to (nonoptimal) action advice, such advice does not require the user to know the agent's policy. Dually, nonoptimal action advice requires this, but does not require her to know *the* optimal action in

the target MDP. Hence it makes sense for complex tasks for which the optimal action is not clear to the user.

Action advice is a direct notion of advice. However, in many natural situations we can expect a human user to consider such decisions with a precise outcome in mind. For instance, when deciding to play or not to play a lottery, we can expect a user to focus on the outcomes of winning the jackpot, of not winning anything, but less likely to take into account the probability of second-rank gains or the probability of being hit by a car when walking for buying a ticket.

For that reason, we introduce new notions of advice, in which the user specifies a successor $s' \in T(s,a)$ of (s,a) in addition to the action advice (s,a). This is obviously always at least as demanding to her as giving only an advice (s,a), but it is natural in many situations, as we argue below.

In general, (optimal) successor advice will consist of a triple (s,a,s') with (s,a) an (optimal) action advice and $T(s,a,s') \neq 0$. A natural choice for s' is a most probable outcome ($s' \in \operatorname{argmax}_{s''} T(s,a,s'')$), but on the one hand, this is noninformative to the agent (it knows T and hence can compute s' from (s,a)), and even so, this fails to capture natural situations: for instance, if a lottery is such that the user has 1 % chance to win 101 and 99 % chances to win 0, with a cost of 1 in all cases, then the optimal policy is to play, but this is not supported by the most plausible outcome.

Given that we take expected utility as the criterion for decision making, a more natural choice is the successor s' of s,a which maximises $T(s,a,s')V_M^{\pi^*}(s')$, that is, which is both likely enough and rewarding enough for supporting the decision. Considering the lottery example again, the agent would learn $1\,\% \times (V_M^{\pi^*}(win) - 1) > 99\,\% \times (V_M^{\pi^*}(lose) - 1)$, which is indeed informative (this is closely related to von Neumann's elicitation of the value of winning *vs* losing).

However, this notion has the drawback of being sensitive to affine transformations of the reward function, while the decision process itself is not. For instance, if the reward function in the lottery example is modified by adding 101 to $R(s)$ for all s, which does *not* change the optimal policies, then the previous inequality would become $1\,\% \times (V_{\pi^*}^M(win) - 1) = 2.01 < 99\,\% \times (V_{\pi^*}^M(lose) - 1) = 99$, hence the advice would be with successor $lose$ instead of win while the decision problem is the same. Hence at least, giving such advice requires the user and the agent to refer to a common, absolute numerical scale.

To fix this, we add an extra term to the inequality, which acts as a normalisation term while not requiring more insight of the user into the ramifications of her decision.

Definition 4 (gain-risk successor advice) *Let $M = \langle S, A, T, R, \gamma \rangle$ be an MDP, π^* be an optimal policy for it, and write $\underline{V} = \min_{s \in S} V_M^{\pi^*}(s)$. An* (optimal) gain-risk successor advice *is a triple $(s,a,s') \in S \times A \times S$ such that (s,a) is an (optimal) action advice and satisfying*

$$s' \in \arg\max_{s''} \left(T(s,a,s'') V_{\pi^*}^M(s'') + (1 - T(s,a,s''))\underline{V} \right)$$

This criterion is similar to maximal weighted utility, and requires exactly the same effort for the user to give advice. It is in the *interpretation* of the advice by the agent that normalisation occurs. Intuitively, a gain-risk successor advice formalises the user saying "in that state, I would choose such action, expecting to reach such state, while aware of the risk to reach such other unwanted state".

Example 5 *Consider a user who wants to go from City A to City B. There are two roads, X and Y, and one can easily switch road. Assume a GPS chooses X because it is shorter. Assume the user wants to take Y because she likes contemplating the canyon from there, but it may rain and the canyon would not be visible. An action advice would be "from A take Y", and would make the GPS take Y then switch to X asap. A gain-risk successor advice would be "from A take Y, with the aim of reaching a state where the canyon can be contemplated". The GPS would then consider taking Y until the canyon, then switch to X. Indeed, from the advice and the probability of rain, the GPS would infer information about the value of contemplating the canyon according to the user.*

Example 6 *As another example, consider a lottery with 1/4 chance to win 100 and 1/2 chance to win 10. The hope (when playing) is to get 100, since this is probable enough and more rewarding. But if the first reward were 15, one would mainly hope to get 10 (which now contributes $10 \times 1/2 = 5$ to the expected value,* vs 3.75*), while still having "play the lottery" as the optimal action, so that action advice would be poorly informative. Indeed, it can be seen that the successor s' indirectly provides information about the rewards, which is then useful for computing actions in other states.*

Computing MMR Policies with Advice

In the setting of IRMDPs, once the agent has received an advice $\mathcal{A}$ of any type, all the information which it has is that the target reward function R is in the imprecise set $\tilde{R}$ and is such that the MDP $\langle S, A, T, R, \gamma \rangle$ satisfies the advice. Write $\tilde{R} \oplus \mathcal{A}$ for the set of all such reward functions, and $\tilde{M} \oplus \mathcal{A}$ for the IRMDP $\langle S, A, T, \tilde{R} \oplus \mathcal{A}, \gamma \rangle$. From $\tilde{M} \oplus \mathcal{A}$, the agent can compute a new MMR policy. In an iterative setting, as we consider in our experiments, the agent can display this new policy to the user, who can give new action advice, etc.

Definition 7 (min max regret with advice) *Let $\tilde{M} = \langle S, A, T, \tilde{R}, \gamma \rangle$ be an IRMDP, and let $\mathcal{A}$ be an advice for $\tilde{M}$. A policy $\tilde{\pi} : S \to \Delta(A)$ is said to be* MMR-optimal knowing $\mathcal{A}$ *if it satisfies $\tilde{\pi} \in \arg\min_{\tilde{\pi}' \in \Delta(A)^S} MR_\alpha^{\tilde{M} \oplus \mathcal{A}}(\tilde{\pi}')$.*

There is no reason in general for $\tilde{R} \oplus \mathcal{A}$ to be a polytope. Still, we now give efficient ways to compute MMR-optimal policies with respect to $\tilde{R} \oplus \mathcal{A}$.

It is easy to modify Regan and Boutilier's approach (2009) to take optimal *action* advice into account. Precisely, referring to Figure 1, and keeping in mind that $I_a(s)$ is a variable which is true if and only if the adversary policy π_g computed by the subproblem satisfies $\pi_g(s) = a$, it is easily seen that an optimal action advice (s,a) can be integrated to the subproblem through the additional linear constraint $I_a(s) = 1$ (or , equivalently, $Q_a(s) \geq Q_{a'}(s)$

$(\forall a' \in A)$). Similarly, a (nonoptimal) action advice can be integrated by adding the constraint $Q_a(s) \geq Q_{\tilde{\pi}^*(s)}(s)$.

The modification is more involved for gain-risk successor advice (s, a, s'), for which we need to refer to $\underline{V}$ inside the subproblem. For this, we use an extra (real-valued) variable $\underline{V}$, meant to hold this value, and $|S|$ extra 0/1 variables, written $M(s)$, encoding the state $\underline{s}$ at which V is minimum.

Definition 8 (subproblem for gain-risk successors) *Let $\tilde{M} = \langle S, A, T, \tilde{R}, \gamma \rangle$ be an IRMDP, let $\mathcal{A} = (s, a, s')$ be a (optimal) gain-risk successor advice for it, and write $\underline{R}$ for $\min_{R \in \tilde{R}} \min_{s \in S} R(s)$. The* subproblem for $\tilde{M}$ with advice $\mathcal{A}$ *is defined to be the subproblem of Figure 1 augmented with the constraint for the (optimal) action advice (s, a), variables $\underline{V}$ and $M(s)$ $(s \in S)$, and the constraints:*

$$
\begin{array}{lll}
(1) & T(s,a,s') \cdot V(s') + (1 - T(s,a,s'))\underline{V} & \\
\geq & T(s,a,s'') \cdot V(s'') + (1 - T(s,a,s''))\underline{V} & (\forall s'') \\
(2) & \underline{V} \leq V(s'') & (\forall s'') \\
(3) & \sum_{s'' \in S} M(s'') = 1 & \\
(4) & M(s'') \in \{0, 1\} & (\forall s'') \\
(5) & \underline{V} \geq M(s'')V(s'') + (1 - M(s'))\frac{1}{1-\gamma}\underline{R} & (\forall s'')
\end{array}
$$

Constraints (2) and (5) realise a standard trick for forcing $\underline{V} = \min_{s''} V(s'')$ in any optimal solution, so that it is easily seen that this program is correct.

Proposition 9 *Let $\tilde{M} = \langle S, A, T, \tilde{R}, \gamma \rangle$ be an IRMDP, let $\mathcal{A} = (s, a, s')$ be a gain-risk successor advice for it, and let $\tilde{\pi}$ be a policy (as computed by the master problem). The optimal solutions of the subproblem for $\tilde{M}$ and $\tilde{\pi}$ with advice $\mathcal{A}$ are exactly at those reward functions R which satisfy $\mathcal{A}$.*

Experimental Results

We now report on experiments on synthetic MDPs, aimed at evaluating advice along the following dimensions:

- the kind of advice given to the user, optimal action advice *vs* optimal gain-risk successor advice,
- the kind of interactions between the user and the agent, one-shot interaction *vs* iterative interaction (see below),
- for iterative interaction, the impact of the number of interaction steps on the quality of the agent's policy,
- for iterative interaction, the impact of the method used for selecting the advice (see below).

Scenarios of Interaction We distinguish two types of scenarios. In the first one, which we call *one-shot interaction*, the agent equipped with an IRMDP first shows its MMR-optimal policy $\tilde{\pi}^*$ to the user, the user computes all pieces of advice (either optimal actions or optimal gain-risk successors) $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$ relative to this policy, and communicates them all to the agent. No further interaction occurs, and the agent computes an MMR-optimal policy knowing $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$ as in Definition 7. On plots, we refer to this policy by "OptAction-all" (resp. "GainRisk-all").

In the second scenario, which we call *iterative interaction*, the agent again shows its MMR-optimal policy to the user, but the user selects a single piece of advice $\mathcal{A}_1$ and gives it to the agent. The agent then computes an MMR-optimal policy $\tilde{\pi}_1^*$ knowing $\mathcal{A}_1$ and shows it to the user, who selects a piece of advice $\mathcal{A}_2$ relative to this new policy, gives it to the agent, and so on. We plot the quality of policies $\tilde{\pi}_1^*, \tilde{\pi}_2^*, \ldots, \tilde{\pi}_i^*$, etc., as a function of i, where the ith iteration corresponds to an MMR-optimal policy knowing $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_i$. On the one hand, we expect advice to be more informative in this iterative scenario, but on the other hand, a number of iterations is required before the agent gets as much advice in this scenario as it got with one-shot interaction, where *all* advice is given to it at the first step.

Finally, we studied different methods for the user to select a single piece of advice at each iteration. We report results for two methods proposed by Regan and Boutilier (2009): "Halve Largest Gap" (HLG) and "Current Solution" (CS).

Formally, given $\tilde{R}$, HLG selects the state s for which the "gap" $\delta(s) = \max_{R \in \tilde{R}} R(s) - \min_{R \in \tilde{R}} R(s)$ is maximal. Regan and Boutilier (2009) then ask the user a "bound query" of the form "is $R(s) \geq b$?", where b is the midpoint of the gap. We adapt it to our setting by giving the optimal (action or successor) advice at s.

Now, CS can be seen as a weighted version of HLG. Intuitively, this accounts for the fact that being uncertain about s is not so serious if s has little influence on the value and on the regret of our current best policy. Precisely, CS selects the state s for which the quantity

$$\max\{\sum_{a \in A} f(s,a)\delta(s), \sum_{a \in A} g(s,a)\delta(s)\}$$

is maximal, where $f(s, a)$ (resp. $g(s, a)$) is the occupation measure of s, a according to the MMR-optimal policy $\tilde{\pi}^*$ of $\tilde{M}$, as shown by the agent to the user (resp. according to the policy incurring the maximal regret to $\tilde{\pi}^*$).

We wish to emphasise than CS and HLG were *not* designed to cope with advice contexts as we investigate in this paper, in the sense that they were designed for reducing maximum regret (Regan and Boutilier 2009), without the advice referring to any target, concrete reward function whatsoever. Those methods however proved efficient in our context.

All in all, combining optimal action and optimal gain-risk successor advice, on the one hand, with HLG or CS, on the other hand, gives four different iterative interaction scenarios, each of which we analyse along the number of iterations. On plots, the scenarios are referred to by "OptAction-cs", "OptAction-hlg", "GainRisk-cs", and "GainRisk-hlg".

Finally, for all experiments we plot the quality of the MMR-optimal policy, computed without any advice. For the first experiments, we also plot the curves for the iterative scenario and bound queries ("mmr-cs" and "mmr-hlg").

We also want to mention that we ran experiments with a number of other definitions of advice (*e.g.*, most probable successor, optimal one under various criteria) and with a number of other selection methods for a single piece of advice (*e.g.*, restricting to advice on a most probable trajectory, to advice on a state as "close" as possible to a state

with unknown reward, etc.). However, some combinations gave results quite uniformly worse than the methods listed above, and most of them gave similar results, which is why we stick to a few, previously investigated, methods.

Evaluation of Policies Our aim is to measure the quality of policies computed with advice of a certain type, selected with a certain method. For this, given that we consider advice meant to give information about the target reward function R (or about the optimal, target policy), our essential measure is the regret of the computed policies $\tilde{\pi}_i^*$ with respect to R (Definition 1). To stick to natural scenarios (like GPS navigation devices), we computed the regret with a singleton distribution for α, at a so-called "starting state", hence measuring the regret of the policy for a specific task. We refer to this measure by "Regret in the user MDP".

As a reference, for the first experiments we also measure the maximum regret of the computed policy $\tilde{\pi}_i^*$ with respect to the set of rewards $\tilde{R} \oplus \mathcal{A}_1 \oplus \cdots \oplus \mathcal{A}_k$, to which we refer by "Regret in the agent MDP"; there, to be consistent with Regan and Boutilier's work, we use a uniform distribution over all states for α. We however want to mention that results are essentially the same with a uniform or singleton distributions, as confirmed by our preliminary experiments.

Regret in the agent MDP hence measures the extent to which the advice helps reducing the agent's uncertainty, while regret in the user MDP measures the extent to which it helped the agent to correctly identify the target reward function. This may be significantly different as soon as the target function is far from the "middle" of the set $\tilde{R}$.

Generic MDPs We first ran experiments on generic MDPs, randomly generated using the same procedure as Regan and Boutilier (2009). Precisely, we generated random MDPs with 20 to 50 states and with 2 to 4 different actions available at each state. The transition function was generated by drawing, for each pair (s, a), $\log(|S| \times |A|)$ reachable states, and the probability of each was generated from a Gaussian. To model a generic goal-based task, we generated the target reward function by choosing a goal state g uniformly at random as well as a value $R(g)$ drawn uniformly from $[750, 1000]$. Now, to consider important levels of impreciseness, we built a set U ("unknowns") consisting of 2/3 of the states drawn at random; each of them got the same reward, drawn from $[-600, +600]$, in the target R, and $\tilde{R}$ was defined to be $\Pi_{s \in U}[-1000, +1000]$ (the agent only knows that for each $s \in U$, $R(s)$ is between -1000 and +1000). The reward for all other states was fixed to 0 in both R and $\tilde{R}$.

We emphasise that with such settings, the agent is not even sure in general that g is the goal of the problem, since according to its uncertainty about R, it may be the case that some other state (in U) has a higher reward.

We ran 50 simulations with different settings, each one for 10 iterations in iterative scenarios. The results, averaged over all simulations, are depicted on Figures 2, 3, and 4.

Figure 2 (evaluation in agent MDP) clearly confirms that CS and HLG with bound queries are efficient methods for reducing the global uncertainty of the agent, a task they

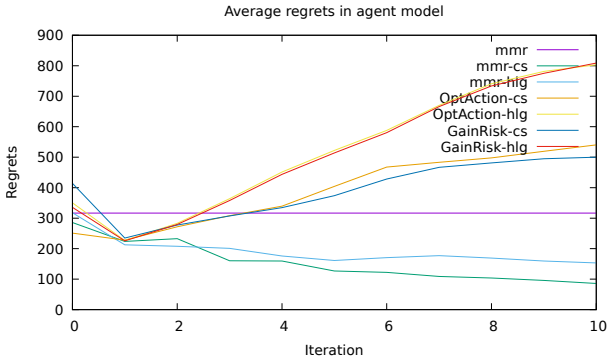


Figure 2: Average regret in agent IRMDP (Generic).

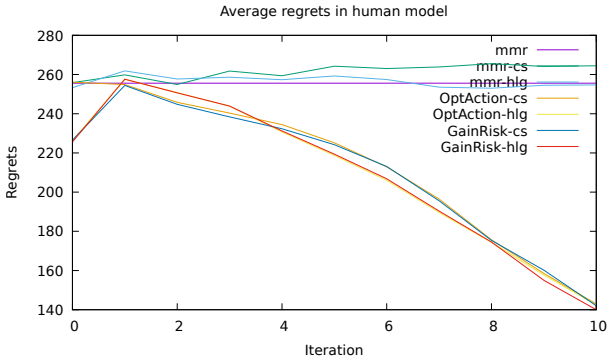


Figure 3: Average regret in user MDP (Generic).



Figure 4: Average regret in user MDP (Generic).

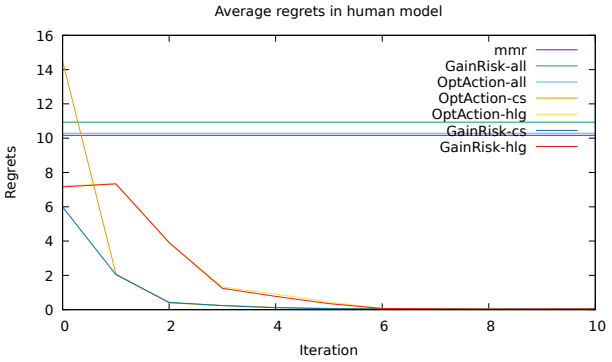


Figure 5: Average regret in user model (Grid).

were designed for, while our advice, which is targetted at a specific reward function, performs badly (even worse than MMR *without* advice). The situation is reversed on Figure 3 (evaluation in user MDP), which shows that the target policy can be approached very quickly with advice dedicated to this task. However, in these dynamic scenarios, all methods using HLG or CS, and optimal action or gain-risk successor advice, essentially perform the same, despite the fact that they do not *a priori* have the same informative content.

The situation is different when we consider one-shot interaction. Figure 4 clearly shows that optimal action advice is useful (compared to the baseline MMR), but that optimal gain-risk successor advice is much more so.

Finally, Figure 4 also shows that iterative interaction is worthwhile: on average, less that 10 iterative communications of advice led to better results than one-shot interaction with optimal action advice, despite the fact that one-shot interaction provided no less that 8 to 24 pieces of advice at once, with an average of 17.

Grid MDPs We next generated random MDPs simulating a grid world in which an agent must go from one square to another one. Such "grid MDPs" are per design much more regular than generic MDPs, and allow us to observe the utility of advice in environments were a policy can easily "come back" and "correct a mistake" (said otherwise, in which there are few or no dead-ends).

We considered grids of size 5×5 to 7×7 with one action per direction, including diagonals (plus "stay"). We also added a low probability for each action to fail (lead to a state adjacent to the intended destination). We drew one state g as the goal, and a set U of 10 to 15 other states as states with unknown reward. The reward $R(g)$ for the goal was drawn from $[80, 100]$, and that for states in U from $[35, 50]$. All other states got a reward of 50. The IRMDP was defined by $\tilde{R} = \Pi_{s \in U}[0, 200]$. By this setting, we wanted to investigate whether the notions of advice would behave the same in environments where all states are rewarding.

Fig. 5 shows the results, averaged over 10 runs. Here, the iterative approaches perform still much better than for generic MDPs, as compared to one-shot interaction. This can be explained by little advice being available in one-shot

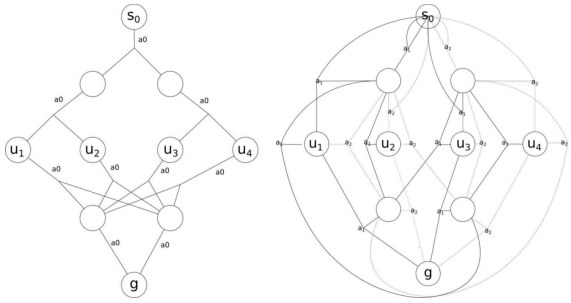


Figure 6: Diamond MDP: actions a_0 (left), a_1, a_2 (right).

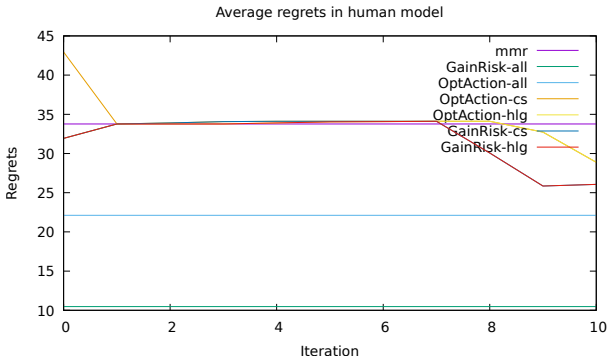


Figure 7: Average regret in user model (Diamond).

interaction (between 2 and 16, with an average of 8). Indeed, since such MDPs are quite regular, min-max regret policies tend to perform rather well (to be close to most "good" policies), hence being subject to not so much advice. On the other hand, we can see that there is essentially no difference between GainRisk-all and OptAction-all. This can be explained by the fact that optimal actions have a clearly defined "preferred" successor, namely, the next step towards the goal, which is the most probable successor at the same time (otherwise, one of the other 8 actions would be better). Hence successor advice adds no significant information.

Diamond MDPs We finally designed a family of problems for which the reward of a few states suffices to generate a lot of uncertainty about the optimal policy. The construction is illustrated on Figure 6. Action a_0 has probability 0.5 to reach each child, a_1, probability 0.3 to reach the left child and reaches a parent otherwise, and a_2 is dual. We generated R by letting $R(s) = 0 | \forall s \in S \setminus (\{g\} \bigcup U)$ and drawing $R(g), R(s)(\forall s \in U)$ from $[600, 1000], [-600, 600]$, resp., where U denotes the middle horizontal line. The agent only knows that states in U have a reward in $[-1000, 1000]$. Hence each of the 2^n shortest trajectories from s_0 to g goes through one unknown-reward state, which makes this family a test case different from the previous ones.

Figure 7 shows the result. Iterative approaches perform badly, which we explain by the fact that CS and HLG

have difficulties finding advice outside of the current trajectory. On the other hand, optimal gain-risk successors advice perform very well in one-shot interaction, like on generic MDPs. Here, clearly, the direction aimed at by the user, rather than the action alone, really helps the agent to learn.

Conclusion and future work

We have considered a generic notion of advice for Markov Decision Problems, allowing a user to give information to an agent about the reward function of the task at hand, and of minimax regret policies knowing advice. We have proposed a new type of advice and shown how to take it into account when computing minimax regret policies. We have shown its usefulness in some scenarios, through experiments. More generally, our experiments shed light on the pros and cons of various interaction scenarios and types of advice.

A short-term perspective of this work is to generalise it to a formal model of interaction with a user who has an Imprecise Reward MDP just as the agent. This is natural but leads to difficulties, in particular because the value function of a policy is not well-defined for Imprecise Reward MDPs. Another perspective is to generalise our study to MDP with imprecision on the transition function.

References

Ahmed, A.; Varakantham, P.; Lowalekar, M.; Adulyasak, Y.; and Jaillet, P. 2017. Sampling based approaches for minimizing regret in uncertain Markov decision processes (MDPs). *J. Artificial Intelligence Research* 59:229–264.

Alizadeh, P.; Chevaleyre, Y.; and Lévy, F. 2016. Solving MDPs with unknown rewards using nondominated vector-valued functions. In *Proc. 8th European Starting AI Researcher Symposium (STAIRS 2016)*, 15–26.

Alizadeh, P.; Chevaleyre, Y.; and Zucker, J.-D. 2015. Approximate regret based elicitation in Markov decision process. In *Proc. 11th International Conference on Computing & Communication Technologies-Research, Innovation, and Vision for the Future (IEEE RIVF 2015)*, 47–52.

Azaria, A.; Gal, Y.; Kraus, S.; and Goldman, C. V. 2016. Strategic advice provision in repeated human-agent interactions. *Autonomous Agents and Multi-Agent Systems* 30(1):4–29.

Bell, D. E. 1982. Regret in decision making under uncertainty. *Operations research* 30(5):961–981.

Benders, J. F. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik* 4(1):238–252.

Busa-Fekete, R.; Szörényi, B.; Weng, P.; Cheng, W.; and Hüllermeier, E. 2013. Preference-based evolutionary direct policy search. In *ICRA Workshop on Autonomous Learning*.

Cederborg, T.; Grover, I.; Isbell, C. L.; and Thomaz, A. L. 2015. Policy shaping with human teachers. In *Proc. 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, 3366–3372.

da Silva, V. F., and Costa, A. H. R. 2011. A geometric approach to find nondominated policies to imprecise reward MDPs. In *Proc. 8th International Conference on Computing & Communication Technologies-Research, Innovation, and Vision for the Future (IEEE RIVF 2011)*, 439–454.

Fürnkranz, J.; Hüllermeier, E.; Cheng, W.; and Park, S.-H. 2012. Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. *Machine learning* 89(1-2):123–156.

Judah, K.; Roy, S.; Fern, A.; and Dietterich, T. G. 2010. Reinforcement learning via practice and critique advice. In *Proc. 24th AAAI Conference on Artificial Intelligence (AAAI 2010)*, 481–486.

Maclin, R., and Shavlik, J. W. 1996. Creating advice-taking reinforcement learners. *Machine Learning* 22(1-3):251–281.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for inverse reinforcement learning. In *Proc. 17th Interantional Conference on Machine Learning (ICML 2000)*, 663–670.

Piot, B.; Geist, M.; and Pietquin, O. 2013. Learning from demonstrations: Is it worth estimating a reward function? In *Proc. 1st Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD 2013*, 17–32.

Puterman, M. 2005. *Markov decision processes: discrete stochastic dynamic programming*. Wiley series in probability and statistics. Wiley-Interscience.

Regan, K., and Boutilier, C. 2009. Regret-based reward elicitation for Markov decision processes. In *Proc. 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, 444–451.

Regan, K., and Boutilier, C. 2010. Robust policy computation in reward-uncertain MDPs using nondominated policies. In *Proc. 24th AAAI Conference on Artificial Intelligence (AAAI 2010)*, 1127–1133.

Regan, K., and Boutilier, C. 2011. Robust online optimization of reward-uncertain MDPs. In *Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, 2165.

Weng, P., and Zanuttini, B. 2013. Interactive value iteration for Markov decision processes with unknown rewards. In *Proc. 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, 2415–2421.

Weng, P. 2011. Markov decision processes with ordinal rewards: reference point-based preferences. In *Proc. 21st International Conference on International Conference on Automated Planning and Scheduling (ICAPS 2011)*, 282–289.

Weng, P. 2012. Ordinal decision models for Markov decision processes. In *Proc. 20th European Conference on Artificial Intelligence (ECAI 2012)*, 828–833.

Xu, H., and Mannor, S. 2009. Parametric regret in uncertain Markov decision processes. In *Proc. 48th International Joint Conference on Decision and Control (IEEE CDC 2009)*, 3606–3613.