

Preference elicitation and inverse reinforcement learning

Anonymous

No Institute Given

Abstract. We state the problem of inverse reinforcement learning in terms of preference elicitation, resulting in a principled (Bayesian) statistical formulation. This generalises previous work on Bayesian inverse reinforcement learning and allows us to obtain a posterior distribution on the agent's preferences, policy and optionally, the obtained reward sequence, from observations. We examine the relation of the resulting approach to other statistical methods for inverse reinforcement learning via analysis and experimental results. We show that preferences can be determined accurately, even if the observed agent's policy is sub-optimal with respect to its own preferences. In that case, significantly improved policies with respect to the agent's preferences are obtained, compared to both other methods and to the performance of the demonstrated policy.

1 Introduction

Preference elicitation is a well-known problem in statistical decision theory [5]. The goal is to determine, whether a given decision maker prefers some events to other events, and if so, by how much. The first main assumption is that there exists a partial ordering among events, indicating relative preferences. Then the corresponding problem is to determine which events are preferred to which others. The second main assumption is the expected utility hypothesis. This posits that if we can assign a numerical *utility* to each event, such that events with larger utilities are preferred, then the decision maker's preferred choice from a set of possible *gambles* will be the gamble with the highest *expected* utility. The corresponding problem is to determine the numerical utilities for a given decision maker.

Preference elicitation is also of relevance to cognitive science and behavioural psychology, where a proper elicitation procedure may allow one to reach more robust experimental conclusions. There are also direct practical applications, such as determining customer preferences. Finally, by analysing the apparent preferences of an expert while performing a particular task, we may be able to discover behaviours that match or even surpass the performance of the expert in the very same task.

This paper uses the formal setting of preference elicitation to determine the preferences of an agent acting within a discrete-time stochastic environment. We assume that the agent obtains a sequence of (hidden to us) rewards from the environment and that its preferences have a functional form related to the rewards.

We also suppose that the agent is acting nearly optimally (in a manner to be made more rigorous later) with respect to its preferences. Armed with this information, and observations from the agent’s interaction with the environment, we can determine the agent’s preferences and policy in a Bayesian framework. This allows us to generalise previous Bayesian approaches to inverse reinforcement learning.

In order to do so, we define a structured prior on reward functions and policies. We then derive two different Markov chain procedures for preference elicitation. The result of the inference is used to obtain policies that are significantly improved with respect to the *true preferences* of the observed agent.

Numerous other inverse reinforcement learning approaches exist [1, 12, 13]. Our main contribution we provide a clear Bayesian formulation of inverse reinforcement learning as preference elicitation, with a structured prior on the agent’s utilities and policies. This generalises the approach of Ramachandran and Amir [12] and paves the way to principled procedures for determining distributions on reward functions, policies and reward sequences. Performance-wise, we show that the policies obtained through our methodology easily surpass the agent’s actual policy with respect to its own utility. Furthermore, we obtain policies that are significantly better than those obtained other inverse reinforcement learning methods that we compare against.

Finally, the relation to *experimental design* for preference elicitation (see [2] for example) must be pointed out. Although this is a very interesting planning problem, in this paper we do not deal with *active* preference elicitation. We focus on the sub-problem of estimating preferences given a particular observed behaviour in a given environment and use decision theoretic formalisms to derive efficient procedures for inverse reinforcement learning.

This paper is organised as follows. The next section formalises the preference elicitation setting and relates it to inverse reinforcement learning. Section 3 presents the abstract statistical model used for estimating the agent’s preferences. Section 4 describes a model and inference procedure for joint estimation of the agent’s preferences and its policy. Section 5 discusses related work in more detail. Section 6 presents comparative experiments, which quantitatively examine the quality of the solutions in terms of both preference elicitation and the estimation of improved policies, concluding with a view to further extensions.

2 Formalisation of the problem

We separate the agent’s preferences (which are unknown to us) from the environment’s dynamics (which we consider known). More specifically, the environment is a controlled Markov process $\nu = (\mathcal{S}, \mathcal{A}, \mathcal{T})$, with state space \mathcal{S} , action space \mathcal{A} , and transition kernel $\mathcal{T} = \{ \tau(\cdot | s, a) : s \in \mathcal{S}, a \in \mathcal{A} \}$, indexed in $\mathcal{S} \times \mathcal{A}$ such that $\tau(\cdot | s, a)$ is a probability measure¹ on \mathcal{S} . The dynamics of the environment are Markovian: If at time t the environment is in state $s_t \in \mathcal{S}$ and the agent

¹ We assume the measurability of all sets with respect to some appropriate σ -algebra.

performs action $a_t \in A$, then the next state s_{t+1} is drawn with a probability independent of previous states and actions:

$$\mathbb{P}_\nu(s_{t+1} \in S \mid s^t, a^t) = \tau(S \mid s_t, a_t), \quad S \subset \mathcal{S}, \quad (2.1)$$

where we use the convention $s^t \equiv s_1, \dots, s_t$ and $a^t \equiv a_1, \dots, a_t$ to represent sequences of variables.

In our setting, we have observed the agent acting in the environment and obtain a sequence of actions and a sequence of states:

$$D \triangleq (a^T, s^T), \quad a^T \equiv a_1, \dots, a_T, \quad s^T \equiv s_1, \dots, s_T.$$

The agent has an *unknown utility function*, U_t , according to which it selects actions, which we wish to discover. Here, we assume that U_t has a structure corresponding to that of reinforcement learning discounted reward infinite-horizon problems and that the agent tries to maximise the expected utility.

Assumption 1 *The agent's utility at time t is defined in terms of future rewards from time t :*

$$U_t \triangleq \sum_{k=t}^{\infty} \gamma^k r_k, \quad (2.2)$$

where $\gamma \in [0, 1]$ is a discount factor, and the reward r_t is given by the reward function $\rho : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ so that the $r_t \triangleq \rho(s, a)$.

In our framework, this is only one of the many possible assumptions regarding the form of the utility function. This choice establishes correspondence with the standard reinforcement learning setting. However, unlike other inverse reinforcement learning approaches, ours is applicable to arbitrary functional forms of the utility.

The controlled Markov process and the utility define a Markov decision process [10] (MDP), denoted by $\mu = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho, \gamma)$. The agent uses some policy π to select actions with distribution $\pi(a_t \mid s_t)$, which together with the Markov decision process μ defines a Markov chain on the sequence of states, such that:

$$\mathbb{P}_{\mu, \pi}(s_{t+1} \in S \mid s^t) = \int_{\mathcal{A}} \tau(S \mid a, s_t) d\pi(a \mid s_t), \quad (2.3)$$

where we use a subscript to denote that the probability is taken with respect to the process defined jointly by μ, π . We shall use this notational convention throughout this paper. Similarly, the *expected utility* of a policy π is denoted by $\mathbb{E}_{\mu, \pi} U_t$. We also introduce the family of Q -value functions $\{Q_\mu^\pi : \mu \in \mathcal{M}, \pi \in \mathcal{P}\}$, where \mathcal{M} is a set of MDPs, with $Q_\mu^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ such that:

$$Q_\mu^\pi(s, a) \triangleq \mathbb{E}_{\mu, \pi}(U_t \mid s_t = s, a_t = a). \quad (2.4)$$

Finally, we use Q_μ^* to denote the optimal Q -value function for an MDP μ , such that:

$$Q_\mu^*(s, a) = \sup_{\pi \in \mathcal{P}} Q_\mu^\pi(s, a), \quad \forall s \in \mathcal{S}, a \in \mathcal{A}. \quad (2.5)$$

With a slight abuse of notation, we shall use Q_ρ when we only need to distinguish between different reward functions ρ , as long as the remaining components of μ remain fixed.

Loosely speaking, our problem is to estimate the reward function ρ and discount factor γ that the agent uses, given the observations s^T, a^T and some prior beliefs. As shall be seen in the sequel, this task is easier with additional assumptions on the structural form of the policy π . We derive two sampling algorithms. The first estimates a joint posterior distribution on the policy and reward function, while the second also estimates a distribution on the sequence of rewards that the agent obtains. We then show how to use those estimates in order to obtain a policy that can perform significantly better than that of the agent's original policy with respect to the agent's true preferences.

3 The statistical model

In the simplest version of the problem, we assume that γ is known and we only estimate the reward function, given some prior over reward functions and policies. This assumption can be easily relaxed, however via an additional prior on the discount factor.

Let \mathcal{R} be a space of reward functions ρ and \mathcal{P} to be a space of policies π . We define a (prior) probability measure $\xi(\cdot | \nu)$ on \mathcal{R} such that for any $B \subset \mathcal{R}$, $\xi(B | \nu)$ corresponds to our prior belief that the reward function is in B . Finally, for any reward function $\rho \in \mathcal{R}$, we define a conditional probability measure $\psi(\cdot | \rho, \nu)$ on the space of policies \mathcal{P} . Let ρ_a, π_a denote the agent's true reward function and policy respectively. Then our model is:

$$\rho \sim \xi(\cdot | \nu), \quad \pi | \rho_a = \rho \sim \psi(\cdot | \rho, \nu), \quad (3.1)$$

while the joint prior on reward functions and policies is denoted by:

$$\phi(P, R | \nu) \triangleq \int_{\mathcal{R}} \psi(P | \rho, \nu) d\xi(\rho | \nu), \quad P \subset \mathcal{P}, R \subset \mathcal{R}, \quad (3.2)$$

such that $\phi(\cdot | \nu)$ is a probability measure on $\mathcal{R} \times \mathcal{P}$.

For the moment we shall leave the exact functional form of the prior on the reward functions and the conditional prior on the policy unspecified. Nevertheless, the structure allows us to state the following:

Lemma 1. *For a prior of the form specified in (3.1), (3.2), and given a controlled Markov process ν and observed state and action sequences s^T, a^T , where the actions are drawn from a reactionary policy π , the posterior measure on reward functions is:*

$$\xi(B | s^T, a^T, \nu) = \frac{\int_B \int_{\mathcal{P}} \pi(a^T | s^T) d\psi(\pi | \rho, \nu) d\xi(\rho | \nu)}{\int_{\mathcal{R}} \int_{\mathcal{P}} \pi(a^T | s^T) d\psi(\pi | \rho, \nu) d\xi(\rho | \nu)}, \quad (3.3)$$

where $\pi(a^T | s^T) = \prod_{t=1}^T \pi(a_t | s_t)$.

Proof. Conditioning on the observations s^T, a^T via Bayes' theorem, we obtain the conditional measure:

$$\xi(B | s^T, a^T, \nu) = \frac{\int_B \psi(s^T, a^T | \rho, \nu) d\xi(\rho | \nu)}{\int_{\mathcal{R}} \psi(s^T, a^T | \rho, \nu) d\xi(\rho | \nu)}, \quad (3.4)$$

where $\psi(s^T, a^T | \rho, \nu)$ is a marginal likelihood term:

$$\psi(s^T, a^T | \rho, \nu) \triangleq \int_{\mathcal{P}} \mathbb{P}_{\nu, \pi}(s^T, a^T) d\psi(\pi | \rho, \nu).$$

It is easy to see via induction that:

$$\mathbb{P}_{\nu, \pi}(s^T, a^T) = \prod_{t=1}^T \pi(a_t | s_t) \tau(s_t | a_{t-1}, s_{t-1}), \quad (3.5)$$

where $\tau(s_1 | a_0, s_0) = \tau(s_1)$ is the initial state distribution. Thus, the reward function posterior is proportional to:

$$\int_B \int_{\mathcal{P}} \prod_{t=1}^T \pi(a_t | s_t) \tau(s_t | a_{t-1}, s_{t-1}) d\psi(\pi | \rho, \nu) d\xi(\rho | \nu).$$

Note that the $\tau(s_t | a_{t-1}, s_{t-1})$ terms can be taken out of the integral. Since they also appear in the denominator, the state transition terms cancel out.

4 Estimation

While it is entirely possible to assume that the agent's policy is optimal with respect to its utility (as is done for example in [1]), our analysis can be made more interesting by assuming otherwise. One simple idea is to restrict the policy space to stationary soft-max policies:

$$\pi_a(a_t | s_t) = \frac{\exp(\eta Q_{\mu}^*(s_t, a_t))}{\sum_a \exp(\eta Q_{\mu}^*(s_t, a))}, \quad (4.1)$$

where we assumed a finite action set for simplicity. Then we can define a prior on policies, given a reward function, by specifying a prior on the inverse temperature η , such that given the reward function and η , the policy is uniquely determined.

However, our framework's generality allows any functional form relating the agent's preferences and policies. As an example, we could define a prior distribution over the ϵ -optimality of the chosen policy, without limiting ourselves to soft-max forms. This would of course change the details of the estimation procedure.

For the chosen prior (4.1), inference can be performed using standard Markov chain Monte Carlo (MCMC) methods [3]. If we can estimate the reward function well enough, we may be able to obtain policies that surpass the performance of the original policy π_a with respect to the agent's reward function ρ_a .

Algorithm 1 Direct Metropolis-Hastings sampling from the joint distribution $\phi(\pi, \rho \mid a^T, s^T)$.

```

1: for  $k = 1, \dots$  do
2:    $\tilde{\rho} \sim \xi(\rho \mid \nu)$ .
3:    $\tilde{\eta} \sim \mathit{Gamma}(\zeta, \theta)$ 
4:    $\tilde{\pi} = \mathit{Softmax}(\tilde{\rho}, \tilde{\eta}, \tau)$ 
5:    $\tilde{p} = \mathbb{P}_{\nu, \tilde{\pi}}(s^T, a^T)$ .
6:   w.p.  $\max\{1, \tilde{p}/p_{(k-1)}\}$  do
7:      $\pi_{(k)} = \tilde{\pi}$ ,  $\epsilon_{(k)} = \epsilon_{(k-1)}$ ,  $\rho_{(k)} = \tilde{\rho}$ ,  $p_{(k)} = \tilde{p}$ .
8:   else
9:      $\pi_{(k)} = \pi_{(k-1)}$ ,  $\epsilon_{(k)} = \epsilon_{(k-1)}$ ,  $\rho_{(k)} = \rho_{(k-1)}$ ,  $p_{(k)} = p_{(k-1)}$ .
10:  done
11: end for

```

A Metropolis-Hastings procedure

Recall that a Metropolis-Hastings (MH) algorithm for sampling from some distribution with density $f(x)$ using a proposal distribution with conditional density $g(\tilde{x} \mid x)$, has the form:

$$x_{(k+1)} = \begin{cases} \tilde{x}, & \text{w.p. } \min\left\{1, \frac{f(\tilde{x})/g(\tilde{x} \mid x_{(k)})}{f(x_{(k)})/g(x_{(k)} \mid \tilde{x})}\right\} \\ x_{(k)}, & \text{otherwise.} \end{cases}$$

In our case, $x = (\rho, \pi)$ and $f(x) = \phi(\rho, \pi \mid s^T, a^T, \nu)^2$. We use *independent* proposals $g(x) = \phi(\rho, \pi \mid \nu)$. Since $\phi(\rho, \pi \mid s^T, a^T, \nu) = \phi(s^T, a^T \mid \rho, \pi, \nu)\phi(\rho, \pi)/\phi(s^T, a^T)$, it follows that:

$$\frac{\phi(\tilde{\rho}, \tilde{\pi} \mid s^T, a^T, \nu)}{\phi(\rho, \pi \mid s^T, a^T, \nu)} = \frac{\mathbb{P}_{\nu, \tilde{\pi}}(s^T, a^T)\phi(\tilde{\rho}, \tilde{\pi} \mid \nu)}{\mathbb{P}_{\nu, \pi_{(k)}}(s^T, a^T)\phi(\rho_{(k)}, \pi_{(k)} \mid \nu)}.$$

This gives rise to the sampling procedure described in Alg. 1, which uses a gamma prior for the temperature.

A hybrid Gibbs procedure

The second alternative is a two-stage hybrid Gibbs sampler, described in Alg. 2. The main interest of this procedure is that it conditions alternatively on a reward sequence sample $r_{(k)}^T$ and on a reward function sample $\rho_{(k)}$ at the k -th iteration of the chain. Thus, we also obtain a posterior distribution on *reward sequences*.

This sampler is of particular utility when the reward function prior is conjugate to the reward distribution, in which case: (i) The reward sequence sample can be easily obtained and (ii) the reward function prior can be conditioned on

² Here we abuse notation, using $\phi(\rho, \pi \mid \cdot)$ to denote the density or probability function with respect to a Lebesgue or counting measure associated with the probability measure $\phi(B \mid \cdot)$ on subsets of $\mathcal{R} \times \mathcal{P}$

the reward sequence with a simple sufficient statistic. While, sampling from the reward function posterior continues to require MH, the resulting hybrid Gibbs sampler remains a valid procedure [3], which may give better results than specifying arbitrary proposals for pure MH sampling.

As previously mentioned, the Gibbs procedure also results in a distribution over the reward sequences observed by the agent. On the one hand, this could be valuable in applications where the reward sequence is the main quantity of interest. On the other hand, this has the disadvantage of making a strong assumption about the distribution from which rewards are drawn.

Algorithm 2 Two stage Gibbs sampler with an M-H step

```

1: for  $k = 1, \dots$  do
2:    $\tilde{\rho} \sim \xi(\rho \mid r_{(k-1)}^T, \nu)$ .
3:    $\tilde{\eta} \sim \text{Gamma}(\zeta, \theta)$ 
4:    $\tilde{\pi} = \text{Softmax}(\tilde{\rho}, \tilde{\epsilon}, \tau)$ 
5:    $\tilde{p} = \mathbb{P}_{\nu, \tilde{\pi}}(s^T, a^T)$ .
6:   w.p.  $\max\{1, \tilde{p}/p_{(k-1)}\}$  do
7:      $\pi_{(k)} = \tilde{\pi}, \epsilon_{(k)} = \epsilon_{(k-1)}, \rho_{(k)} = \tilde{\rho}, p_{(k)} = \tilde{p}$ .
8:   else
9:      $\pi_{(k)} = \pi_{(k-1)}, \epsilon_{(k)} = \epsilon_{(k-1)}, \rho_{(k)} = \rho_{(k-1)}, p_{(k)} = p_{(k-1)}$ .
10:  done
11:   $r_{(k)}^T \mid s^T, a^T \sim \rho_{(k)}^T(s^T, a^T)$ 
12: end for

```

5 Related work

Linear programming

One interesting solution proposed by [8] is to use a linear program in order to find a reward function that maximises the gap between the best and second best action. Although elegant, this approach suffers from some drawbacks. (a) A good estimate of the optimal policy must be given. This may be hard in cases where the demonstrating agent does not visit all of the states frequently. (b) In some pathological MDPs, there is no such gap. For example it could be that for any action a , there exists some other action a' with equal value in every state.

Policy walk

Our framework can be seen as a generalisation of the Bayesian approach considered in [12], which does not employ a structured prior on the rewards and policies. In fact, they implicitly define the joint posterior over rewards and policies as:

$$\phi(\pi, \rho \mid s^T, a^T, \nu) = \frac{\exp[\alpha \sum_t Q_\mu^*(s_t, a_t)] \xi(\rho \mid \nu)}{\phi(s^T, a^T \mid \nu)},$$

which implies that the exponential term corresponds to $\xi(s^T, a^T, \pi | \rho)$. This *ad hoc* choice is probably the weakest point in this approach. Although, as mentioned in [12], such a choice could be justifiable through a maximum entropy argument, we note that the maximum-entropy based approach reported in [14] does not employ the value function in that way.

Rearranging, we write the denominator as:

$$\xi(s^T, a^T | \nu) = \int_{\mathcal{R} \times \mathcal{P}} \xi(s^T, a^T | \pi, \rho, \nu) d\xi(\rho, \pi | \nu), \quad (5.1)$$

which is still not computable, but we can employ a Metropolis-Hastings step using $\xi(\rho | \nu)$ as a proposal distribution, and an acceptance probability of:

$$\frac{\xi(\pi, \rho | s^T, a^T) / \xi(\rho)}{\xi(\pi', \rho' | s^T, a^T) / \xi(\rho')} = \frac{\exp[\alpha \sum_t Q_\rho^\pi(s_t, a_t)]}{\exp[\alpha \sum_t Q_{\rho'}^{\pi'}(s_t, a_t)]}.$$

We note that in [12], the authors employ a different sampling procedure than a straightforward MH, called a policy grid walk. In exploratory experiments, where we examined the performance of the authors' original method [11], we have determined that MH is sufficient and that the most crucial factor for this particular method was its initialisation.

The maximum entropy approach.

A maximum entropy approach is reported in [14]. Given a feature function $\Phi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^n$, and a set of trajectories $\{s_{(k)}^{T_k}, a_{(k)}^{T_k} : k = 1, \dots, n\}$, they obtain features $\Phi_{(k)}^{T_k} = (\Phi(s_{i,(k)}, a_{i,(k)}) : i = 1, \dots, T_k)$. They show that given empirical constraints $\mathbb{E}_{\theta, \nu} \Phi^{T_k} = \hat{\mathbb{E}} \Phi^{T_k}$, where $\hat{\mathbb{E}} \Phi^T = \frac{1}{n} \sum_{k=1}^n \Phi_{(k)}^{T_k}$ is the empirical feature expectation, one can obtain a maximum entropy distribution for actions of the form $\mathbb{P}_\theta(a_t | s_t) \propto e^{\theta^T \Phi(s_t, a_t)}$. If Φ is the identity, then θ can be seen as a scaled state-action value function.

In general, maximum entropy approaches have good minimax guarantees [7]. Consequently, the estimated policy is guaranteed to be close to the agent's. However, at best, by bounding the error in the policy, one obtains a two-sided high probability bound on the relative loss. Thus, one is almost certain to perform neither much better, nor much worse than the demonstrator.

Game theoretic approach

An interesting game theoretic approach was suggested by [13] for apprenticeship learning. This also only requires statistics of observed features, similarly to the maximum entropy approach. The main idea is to find the solution to a game matrix with a number of rows equal to the number of possible policies, which, although large, can be solved efficiently by an exponential weighting algorithm. The method is particularly notable for being (as far as we are aware of) the only

one with a high-probability upper bound on the loss relative to the demonstrating agent and no corresponding lower bound. Thus, this method may in principle lead to a significant improvement over the demonstrator. Unfortunately, as far as we are aware of, sufficient conditions for this to occur are not known at the moment.

6 Experiments

Domains

We compare the proposed algorithms on two different domains, namely on random MDPs and random maze tasks. The *Random MDP* task is a discrete-state MDP, with four actions, such that each leads to a different, but possibly overlapping, quarter of the state set.³ The reward function is drawn from a Beta-product hyperprior with parameters α_i and β_i , where the index i is over all state-action pairs. This defines a distribution over the parameters p_i of the Bernoulli distribution determining the probability of the agent of obtaining a reward when carrying out an action a in a particular state s .

For the *Random Maze* tasks we constructed *planar grid* mazes of different sizes, with four actions at each state, in which the agent has a probability of 0.7 to succeed with the current action and is otherwise moved to one of the adjacent states randomly. These mazes are also randomly generated, with the rewards function being drawn from the same prior. The maze structure is sampled by randomly filling a grid with walls through a product-Bernoulli distribution with parameter $1/4$, and then rejecting any mazes with a number of obstacles higher than $|\mathcal{S}|/4$.

Algorithms

We compared our methodology, using the pure MH and the hybrid Gibbs sampler, to three previous approaches. The linear programming based approach [8], the game-theoretic approach [13] and finally, the Bayesian inverse reinforcement learning method suggested in [12]. In all cases, each demonstration was a T -long trajectory s^T, a^T , provided by a demonstrator employing a softmax policy with respect to the optimal value function.

All algorithms have some parameters that must be selected. Since our methodology employs MCMC the sampling parameters must be chosen so that convergence is ensured. We found that 10^4 samples from the chain were sufficient, for both the MH and hybrid Gibbs sampler, with 2000 steps used as burn-in, for both tasks.

³ The transition matrix of the MDPs was chosen so that the MDP was communicating (c.f. [10]) and so that each individual action from any state results in a transition to approximately a quarter of all available states (with the destination states arrival probabilities being uniformly selected and the non-destination states arrival probabilities being set to zero).

We also compared our procedure with the Bayesian inverse reinforcement learning algorithm of Ramachandran and Amir [12]. For the latter, we used a MH sampler seeded with the solution found by [8], as suggested by [11] and by our own preliminary experiments. We also verified that the same number of samples used in our case was sufficient for this method.

The linear-programming based inverse reinforcement learning algorithm by Ng and Russell [8] requires the actual agent policy as input. For the random-MDP domain, we used the maximum likelihood estimate from our observations. For the maze domain, we used a Laplace-smoothed estimate (i.e. a product-Dirichlet prior with parameters equal to 1) instead, since this was more stable.

Finally, we examined the MWAL algorithm of Syed and Schapire [13]. This requires the cumulative discounted feature expectation as input, for appropriately defined features. Since we dealt with discrete environments, we used the state occupancy as a feature. Although the feature expectations can be calculated empirically, we obtained better performance in practice with the following procedure: We first computed the transition probabilities of the Markov chain induced by the maximum likelihood (or Laplace-smoothed) policy given the observed state-action sequences of the agent and the MDP transitions. Then we calculate the expectation of these features given this chain. We set all accuracy parameters of this algorithm to 10^{-3} , which was sufficient for a robust behaviour.

Performance measure

In order to measure performance, we plot the L_1 loss of the value function of each policy relative to the optimal policy with respect to the agent’s utility:

$$\ell(\pi) \triangleq \sum_{s \in \mathcal{S}} V_{\mu}^*(s) - V_{\mu}^{\pi}(s), \quad (6.1)$$

where $V_{\mu}^*(s) \triangleq \max_a Q_{\mu}^*(s, a)$ and $V_{\mu}^{\pi}(s) \triangleq \mathbb{E}_{\pi} Q_{\mu}^{\pi}(s, a)$.

In all cases, we average over 100 experiments on an equal number of randomly generated environments μ_1, μ_2, \dots . For the i -th experiment, we generate a T -step-long demonstration $D_i = (s^T, a^T)$ via an agent employing a softmax policy. The same demonstration is used across all methods to reduce variance.

Results

We consider the loss of five different policies, averaged over 100 runs. The first, **soft**, is the policy of the demonstrating agent itself. The second, **MH**, is the Metropolis-Hastings procedure defined in Alg. 1, while **G-MH** is the hybrid Gibbs procedure from Alg. 2. Finally, **Ng & Russel**, **Ramachandran & Amir**, and **Syed & Schapire** are our implementations of the methods described in the papers by the respective authors, summarised in Sec. 5.

We first examined the the loss of greedy policies,⁴ derived from the estimated reward function, as the demonstrating agent becomes greedier. Figure 1

⁴ Experiments with non-greedy policies (not shown) produced generally worse results.

shows results for the two different domains. It is easy to see that the **MH** sampler significantly outperforms the demonstrator, even when the latter is nearly greedy. While the hybrid **Gibbs** sampler’s performance lies between that of the demonstrator and the **MH** sampler, it also estimates a distribution over reward sequences as a side-effect. Thus, it could be of further value where estimation of reward sequences is important. We observed that the performance of the baseline methods is generally inferior, though nevertheless the **Syed & Schapire** algorithm tracks the demonstrator’s performance closely.

This suboptimal performance of the baseline methods in the *Random MDP* setting cannot be attributed to poor estimation of the demonstrated policy, as can clearly be seen in Figure 2(a), which shows the loss of the greedy policy derived from each method as the amount of data increases. While the proposed samplers improve significantly as observations accumulate, this effect is smaller in the baseline methods we compared against. As a final test, we plot the relative loss in the *Random MDP* as the number of states increases in Figure 2(b). We can see that the relative performance of methods is invariant to the size of the state space for this problem.

Overall, we observed that our suggested model consistently outperforms the agent in all settings, when the MH sampler is used, while the Gibbs sampler only manages to match the behaviour approximately. Presumably, this is due to the joint estimation of the reward sequence. Finally, the other methods under consideration on average do not improve upon the initial policy and can be, in a large number of cases, significantly worse. For the linear programming inverse RL method, perhaps this can be attributed to implicit assumptions about the MDP and the optimality of the given policy. For the policy walk inverse RL method, our belief is that its suboptimal performance is due to the very restrictive and somewhat *ad hoc* prior it uses. Finally, the performance of the game theoretic approach is slightly disappointing. Although it is much more robust than the other two baseline approaches, it never outperforms the demonstrator, even though technically this is possible. One possible explanation is that since this approach is worst-case by construction, it results in overly conservative policies.

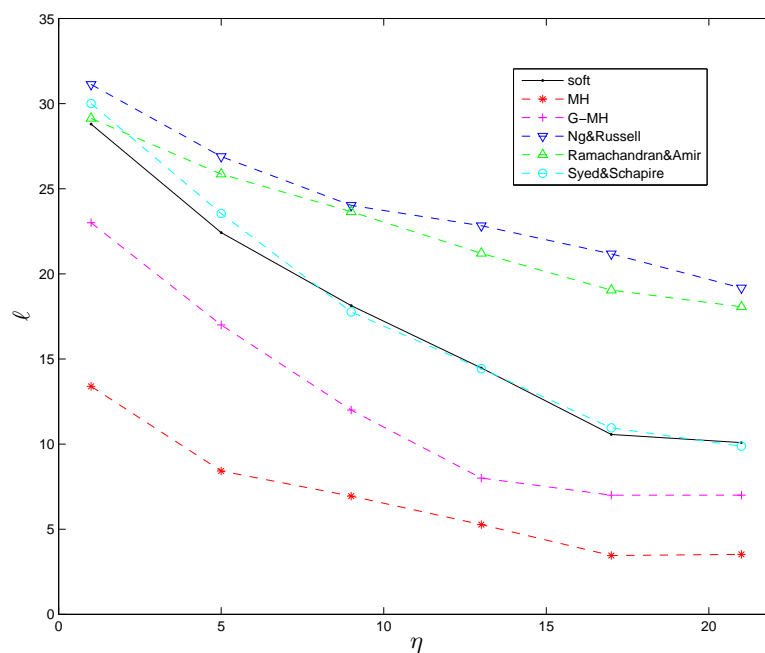
7 Discussion

We introduced a unified framework of preference elicitation and inverse reinforcement learning, presented a statistical model for inference and derived two different sampling procedures for estimation. Our framework is flexible enough to allow plugging in alternative priors on the form of the policy and of the agent’s preferences, although that would require adjusting the sampling procedures. In experiments, we showed that for a particular choice of policy prior, closely corresponding to previous approaches, our samplers can outperform not only other well-known inverse reinforcement learning algorithms, but the demonstrating agent as well.

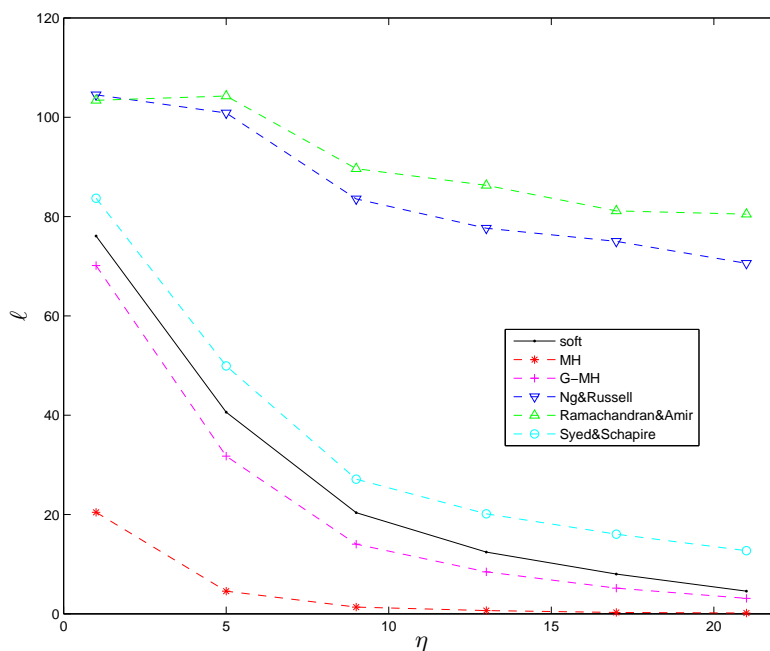
The simplest extension, which we have already alluded to, is the estimation of the discount factor, for which we have obtained promising results in preliminary experiments. A slightly harder generalisation occurs when the environment is not known to us. This is not due to difficulties in inference, since in many cases a posterior distribution over \mathcal{M} is not hard to maintain (see for example [4, 9]). However, computing the optimal policy given a belief over MDPs is harder [4], even if we limit ourselves to stationary policies [6]. We would also like to consider more types of preference and policy priors. Firstly, the use of spatial priors for the reward function, which would be necessary for large or continuous environments. Secondly, the use of alternative priors on the demonstrator’s policy.

The generality of the framework allows us to formulate different preference elicitation problems than those directly tied to reinforcement learning. For example, it is possible to estimate utilities that are not additive functions of some latent rewards. This does not appear to be easily achievable through the extension of other inverse reinforcement learning algorithms. It would be interesting to investigate this in future work.

Finally, although in this paper we have not considered the problem of *experimental design* for preference elicitation (i.e. *active* preference elicitation), we believe is a very interesting direction. First of all, it has many applications, such as the automated optimal design of behavioural experiments to give but one example. Nevertheless, a more effective preference elicitation procedure such as the one presented in this paper is absolutely essential for the complex planning task that experimental design is. Consequently, we hope that researchers in that area will find our methods useful.

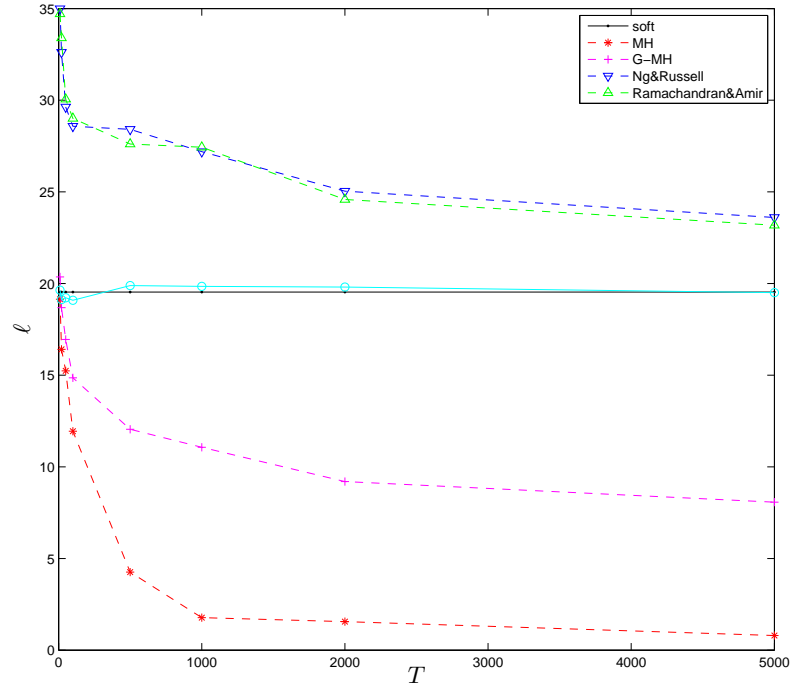


(a) Random MDP

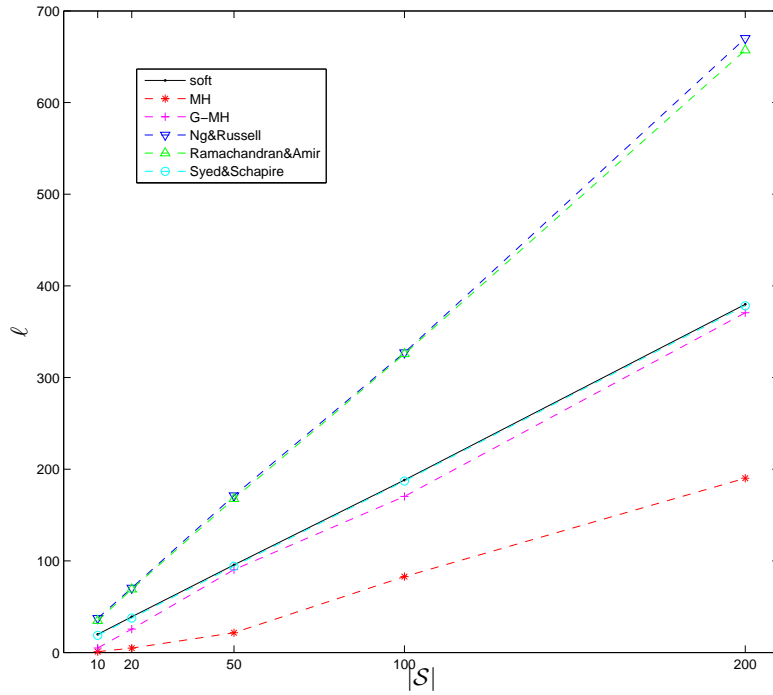


(b) Random Maze

Fig. 1. Total loss ℓ with respect to the optimal policy, for the given sub-optimal demonstrator policy, as well as for the estimated *greedy* policies, as a function of the inverse temperature η of the softmax policy of the demonstrator for (a) the Random MDP and (b) the Random Maze tasks, averaged over 100 runs.



(a) Effect of amount of data



(b) Effect of environment size

Fig. 2. Total loss ℓ with respect to the optimal policy, for the given sub-optimal demonstrator policy, as well as for the estimated *greedy* policies, in the Random MDP task. Figure 2(a) shows how performance improves as a function of the length T . of the demonstrated sequence. Figure 2(b) shows the effect of the number of states $|\mathcal{S}|$ of the underlying MDP. All quantities are averaged over 100 runs.

Bibliography

- [1] P. Abbeel and A.Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st international conference on Machine learning (ICML 2004)*, 2004.
- [2] C. Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of the National Conference on Artificial Intelligence*, pages 239–246. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.
- [3] George Casella, Stephen Fienberg, and Ingram Olkin, editors. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, 1999.
- [4] Michael O’Gordon Duff. *Optimal Learning Computational Procedures for Bayes-adaptive Markov Decision Processes*. PhD thesis, University of Massachusetts at Amherst, 2002.
- [5] Milton Friedman and Leonard J. Savage. The expected-utility hypothesis and the measurability of utility. *The Journal of Political Economy*, 60(6): 463, 1952.
- [6] Thomas Furnston and David Barber. Variational methods for reinforcement learning. In *AISTATS*, pages 241–248, 2010.
- [7] Peter D. Grünwald and A. Philip Dawid. Game theory, maximum entropy, minimum discrepancy, and robust bayesian decision theory. *Annals of Statistics*, 32(4):1367–1433, 2004.
- [8] Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *in Proc. 17th International Conf. on Machine Learning*, pages 663–670. Morgan Kaufmann, 2000.
- [9] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete Bayesian reinforcement learning. In *ICML 2006*, pages 697–704. ACM Press New York, NY, USA, 2006.
- [10] Marting L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New Jersey, US, 2005.
- [11] D Ramachandran, 2010. Personal communication.
- [12] D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *in 20th Int. Joint Conf. Artificial Intelligence*, volume 51, page 61801, 2007.
- [13] Umar Syed and Robert E. Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems*, volume 10, 2008.
- [14] Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. Modelling interaction via the principle of maximum causal entropy. In *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, Haifa, Israel, 2010.