Bayesian multitask inverse reinforcement learning

Christos Dimitrakakis[1] and Constantin A. Rothkopf[2]

[1] EPFL, Lausanne, Switzerland
christos.dimitrakakis@epfl.ch
[2] Frankfurt Institute for Advanced Studies, Frankfurt, Germany
rothkfopf@fias.uni-frankfurt.de

Abstract. We generalise the problem of inverse reinforcement learning to multiple tasks, from multiple demonstrations. Each one may represent one expert trying to solve a different task, or as different experts trying to solve the same task. Our main contribution is to formalise the problem as statistical preference elicitation, via a number of structured priors, whose form captures our biases about the relatedness of different tasks or expert policies. In doing so, we introduce a prior on policy optimality, which is more natural to specify. We show that our framework allows us not only to learn to efficiently from multiple experts but to also effectively differentiate between the goals of each. Possible applications include analysing the intrinsic motivations of subjects in behavioural experiments and learning from multiple teachers.

Keywords: Bayesian inference, intrinsic motivations, inverse reinforcement learning, multitask learning, preference elicitation

1 Introduction

This paper deals with the problem of multitask inverse reinforcement learning. Loosely speaking, this involves inferring the motivations and goals of an unknown agent performing a series of tasks in a dynamic environment. It is also equivalent to inferring the motivations of different experts, each attempting to solve the same task, but whose different preferences and biases affect the solution they choose. Solutions to this problem can also provide principled statistical tools for the interpretation of behavioural experiments with humans and animals.

While both inverse reinforcement learning, and multitask learning are well known problems, to our knowledge this is the only principled statistical formulation of this problem. Our first major contribution generalises our previous work [20], a statistical approach for single-task inverse reinforcement learning, to a hierarchical (population) model discussed in Section 3. Our second major contribution is an alternative model, which uses a much more natural prior on the optimality of the demonstrations, in Section 4, for which we also provide computational complexity bounds. An experimental analysis of the procedures is given in Section 5, while the connections to related work are discussed in Section 6. Auxiliary results and proofs are given in the appendix.

2 The general model

We assume that all tasks are performed in an environment with dynamics drawn from the same distribution (which may be singular). We define the environment as a controlled Markov process (CMP) $\nu = (\mathcal{S}, \mathcal{A}, \mathcal{T})$, with state space $\mathcal{S}$, action space $\mathcal{A}$, and transition kernel $\mathcal{T} = \{ \tau(\cdot \mid s, a) : s \in \mathcal{S}, a \in \mathcal{A} \}$, indexed in $\mathcal{S} \times \mathcal{A}$ such that $\tau(\cdot \mid s, a)$ is a probability measure[3] on $\mathcal{S}$. The dynamics of the environment are Markovian: If at time t the environment is in state $s_t \in S$ and the agent performs action $a_t \in A$, then the next state s_{t+1} is drawn with a probability independent of previous states and actions: $\mathbb{P}_\nu(s_{t+1} \in S \mid s^t, a^t) = \tau(S \mid s_t, a_t)$, $S \subset \mathcal{S}$, where we use the convention $s^t \equiv s_1, \ldots, s_t$ and $a^t \equiv a_1, \ldots, a_t$ to represent sequences of variables, with $\mathcal{S}^t, \mathcal{A}^t$ being the corresponding product spaces. If the dynamics of the environment are unknown, we can maintain a belief about what the true CMP is, expressed as a probability measure ω on the space of controlled Markov processes $\mathcal{N}$.

During the m-th demonstration, we observe an agent acting in the environment and obtain a T_m-long sequence of actions and a sequence of states: $\boldsymbol{d}_m \triangleq (a_m^{T_m}, s_m^{T_m})$, $a_m^{T_m} \triangleq a_{m,1}, \ldots, a_{m,T}$, $s_m^{T_m} \triangleq s_{m,1}, \ldots, s_{m,T_m}$. The m-th task is defined via an *unknown utility function*, $U_{m,t}$, according to which the demonstrator selects actions, which we wish to discover. Setting $U_{m,t}$ equal to the total discounted return,[4] we establish a link with inverse reinforcement learning:

Assumption 1 *The agent's utility at time t is defined in terms of future rewards: $U_{m,t} \triangleq \sum_{k=t}^{\infty} \gamma^k r_k$, where $\gamma \in [0, 1]$ is a discount factor, and the reward r_t is given by the reward function $\rho_m : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ so that $r_t \triangleq \rho_m(s_t, a_t)$.*

In the following, for simplicity we drop the subscript m whenever it is clear by context. For any reward function ρ, the controlled Markov process and the resulting utility U define a Markov decision process [17] (MDP), denoted by $\mu = (\nu, \rho, \gamma)$. The agent uses some policy π to select actions $a_t \sim \pi(\cdot \mid s^t, a^{t-1})$, which together with the Markov decision process μ defines a distribution[5] on the sequences of states, such that $\mathbb{P}_{\mu,\pi}(s_{t+1} \in S \mid s^t, a^{t-1}) = \int_{\mathcal{A}} \tau(S \mid a, s_t) \,\mathrm{d}\pi(a \mid s^t, a^{t-1})$, 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 $Q_\mu^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ such that: $Q_\mu^\pi(s, a) \triangleq \mathbb{E}_{\mu,\pi} (U_t \mid s_t = s, a_t = a)$. Finally, we use Q_μ^* to denote the optimal Q-value function for an MDP μ, such that: $Q_\mu^*(s, a) = \sup_{\pi \in \mathcal{P}} Q_\mu^\pi(s, a)$, $\forall s \in \mathcal{S}, a \in \mathcal{A}$. With a slight abuse of notation, we shall use Q_ρ when we only

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

[4] Other forms of the utility are possible. For example, consider an agent who collects gold coins in a maze with traps, and where the agent's utility is the logarithm of the number of coins it has after it has exited the maze.

[5] When the policy is reactive, then $\pi(a_t \mid s^t, a^{t-1}) = \pi(a_t \mid s_t)$, and the process reduces to first order Markov.

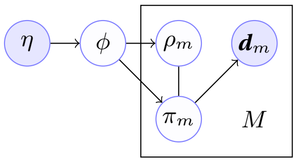


Fig. 1. Graphical model of general multitask reward-policy priors. Lighter colour indicates latent variables. Here η is the hyperprior on the joint reward-policy prior ϕ while ρ_m and π_m are the reward and policy of the m-th task, for which we observe the demonstration $\boldsymbol{d}_m$. The undirected link between π and ρ represents the fact that the rewards and policy are jointly drawn from the reward-policy prior. The implicit dependencies on ν are omitted for clarity.

need to distinguish between different reward functions ρ, as long as the remaining components of μ are clear from the context.

Loosely speaking, our problem is to estimate the sequence of reward functions $\boldsymbol{\rho} \triangleq \rho_1, \ldots, \rho_m, \ldots, \rho_M$, and policies $\boldsymbol{\pi} \triangleq \pi_1, \ldots, \pi_m, \ldots, \pi_M$, which were used in the demonstrations, given the data $\boldsymbol{D} = \boldsymbol{d}_1, \ldots, \boldsymbol{d}_m, \ldots, \boldsymbol{d}_M$ from *all* demonstrations and some prior beliefs. In order to do this, we define a multitask reward-policy prior distribution as a Bayesian hierarchical model.

2.1 Multitask priors on reward functions and policies

We consider two types of priors on rewards and policies. Their main difference is how the dependency between the reward and the policy is modelled. Due to the multitask setting, we posit that the reward function is drawn from some unknown distribution for each task, for which we assert a hyperprior, which is later conditioned on the demonstrations. The hyperprior η is a probability measure on the set of joint reward-policy priors $\mathscr{J}$. It is easy to see that, given some specific $\phi \in \mathscr{J}$, we can use Bayes' theorem directly to obtain, for any $A \subset \mathcal{P}^M, B \subset \mathcal{R}^M$, where $\mathcal{P}^M, \mathcal{R}^M$ are the policy and reward product spaces:

$$\phi(A, B \mid \boldsymbol{D}) = \frac{\int_{A \times B} \phi(\boldsymbol{D} \mid \boldsymbol{\rho}, \boldsymbol{\pi}) \, \mathrm{d}\phi(\boldsymbol{\rho}, \boldsymbol{\pi})}{\int_{\mathcal{R}^M \times \mathcal{P}^M} \phi(\boldsymbol{D} \mid \boldsymbol{\rho}, \boldsymbol{\pi}) \, \mathrm{d}\phi(\boldsymbol{\rho}, \boldsymbol{\pi})} = \prod_m \phi(\rho_m, \pi_m \mid \boldsymbol{d}_m).$$

When ϕ is not specified, we must somehow estimate some distribution on it. In the *empirical Bayes* case [19] the idea is to simply find a distribution η in a restricted class H, according to some criterion, such as maximum likelihood. In the *hierarchical Bayes* approach, followed herein, we select some prior η and then estimate the *posterior distribution* $\eta(\cdot \mid \boldsymbol{D})$.

We consider two models. In the first, discussed in Section 3 on the following page, we initially specify a product prior on reward functions and on *policy parameters*. Jointly, these determine a unique policy, for which the probability of the observed demonstration is well-defined. The policy-reward dependency is exchanged in the alternative model, which is discussed in Section 4 on page 5. There we specify a product prior on policies and on *policy optimality*. This leads to a distribution on reward functions, conditional on policies.

3 Multitask Reward-Policy prior (MRP)

Let $\mathcal{R}$ be the space of reward functions ρ and $\mathcal{P}$ the space of policies π. Let $\psi(\cdot \mid \nu) \in \mathscr{R}$ denote a conditional probability measure on the reward functions $\mathcal{R}$ such that for any $B \subset \mathcal{R}$, $\psi(B \mid \nu)$ corresponds to our prior belief that the reward function is in B, when the CMP is known to be ν. For any reward function $\rho \in \mathcal{R}$, we define a conditional probability measure $\xi(\cdot \mid \rho, \nu) \in \mathscr{P}$ on the space of policies $\mathcal{P}$. Let ρ_m, π_m denote the m-th demonstration's reward function and policy respectively. We use a product[6] hyperprior[7] η on the set of reward function distributions and policy distribution $\mathscr{R} \times \mathscr{P}$, such that $\eta(\Psi, \Xi) = \eta(\Psi)\eta(\Xi)$ for all $\Psi \subset \mathscr{R}$, $\Xi \subset \mathscr{P}$. Our model is specified as follows:

$$(\psi, \xi) \sim \eta(\cdot \mid \nu), \qquad \rho_m \mid \psi, \nu \sim \psi(\cdot \mid \nu), \qquad \pi_m \mid \xi, \nu, \rho_m \sim \xi(\cdot \mid \rho_m, \nu), \tag{3.1}$$

In this case, the joint prior on reward functions and policies can be written as $\phi(P, R \mid \nu) \triangleq \int_R \xi(P \mid \rho, \nu) \, \mathrm{d}\psi(\rho \mid \nu)$ with $P \subset \mathcal{P}$, $R \subset \mathcal{R}$, such that $\phi(\cdot \mid \nu)$ is a probability measure on $\mathcal{P} \times \mathcal{R}$ for any CMP ν.[8] In our model, the only observable variables are η, which we select ourselves and the demonstrations $\boldsymbol{D}$.

3.1 The policy prior

The model presented in this section involves restricting the policy space to a parametric form. As a simple example, we consider stationary soft-max policies with an inverse temperature parameter c:

$$\pi(a_t \mid s_t, \mu, c) = \mathcal{S}\mathit{oftmax}(a_t \mid s_t, \mu, c) \triangleq \frac{\exp(c Q^*_\mu(s_t, a_t))}{\sum_a \exp(c Q^*_\mu(s_t, a))}, \tag{3.2}$$

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 β on c. Inference can be performed using standard Monte Carlo methods. If we can estimate the reward functions well enough, we may be able to obtain policies that surpass the performance of the demonstrators.

3.2 Reward priors

In our previous work [20], we considered a product-Beta distribution on states (or state-action pairs) for the reward function prior. Herein, however, we develop

[6] Even if a prior distribution is a product, the posterior may not necessarily remain a product. Consequently, this choice does not imply the assumption that rewards are independent from policies.

[7] In order to simplify the exposition somewhat, while maintaining generality, we usually specify distributions on functions or other distributions directly, rather than on their parameters.

[8] If the CMP itself is unknown, so that we only have a probabilistic belief ω on $\mathcal{N}$, we can instead consider the marginal $\phi(P, R \mid \omega) \triangleq \int_{\mathcal{N}} \phi(P, R \mid \nu) \, \mathrm{d}\omega(\nu)$.

a more structured prior, by considering reward functions as a measure on the state space $\mathcal{S}$ with $\rho(\mathcal{S}) = 1$. Then for any state subsets $S_1, S_2 \subset \mathcal{S}$ such that $S_1 \cap S_2 = \emptyset$, $\rho(S_1 \cup S_2) = \rho(S_1) + \rho(S_2)$. A well-known distribution on probability measures is a Dirichlet process [11]. Consequently, when $\mathcal{S}$ is finite, we can use a Dirichlet prior for rewards, such that each sampled reward function is equivalent to multinomial parameters. This is more constrained than the Beta-product prior and has the advantage of clearly separating the reward function from the c parameter in the policy model. It also brings the Bayesian approach closer to approaches which bound the L_1 norm of the reward function such as [21].

3.3 Estimation

The simplest possible algorithm consists of sampling directly from the prior. In our model, the prior on the reward function ρ and inverse temperature c is a product, and so we can simply take independent samples from each, obtaining an approximate posterior on rewards an policies, as shown in Alg. 1. While such methods are known to converge asymptotically to the true expectation under mild conditions [12], stronger technical assumptions are required for finite sample bounds, due to importance sampling in step 8.

Algorithm 1 MRP-MC: Multitask Reward-Policy Monte Carlo. Given the data $\boldsymbol{D}$, we obtain $\hat{\eta}$, the approximate posterior on the reward-policy distirbution, and $\hat{\rho}_m$, the $\hat{\eta}$-expected reward function for the m-th task.

```
 1: for $k = 1, \ldots, K$ do
 2:     $\phi^{(k)} = (\xi^{(k)}, \psi^{(k)}) \sim \eta$, $\xi^{(k)} = \mathscr{G}amma(g_1^{(k)}, g_2^{(k)})$.
 3:     for $m = 1, \ldots, M$ do
 4:         $\rho_m^{(k)} \sim \xi(\rho \mid \nu)$, $c_m^{(k)} \sim \mathscr{G}amma(g_1^{(k)}, g_2^{(k)})$
 5:         $\mu_m^{(k)} = (\nu, \gamma, \rho_m^{(k)})$, $\pi_m^{(k)} = \mathscr{S}oftmax(\cdot \mid \cdot, \mu_m^{(k)}, c_m^{(k)})$, $p_m^{(k)} = \pi_m^{(k)}(a_m^T \mid s_m^T)$
 6:     end for
 7: end for
 8: $q^{(k)} = \prod_m p_m^{(k)} / \sum_{j=1}^K \prod_m p_m^{(j)}$
 9: $\hat{\eta}(B \mid \boldsymbol{D}) = \sum_{k=1}^K \mathbb{I}\left\{\phi^{(k)} \in B\right\} q^{(k)}$, for $B \subset \mathcal{R} \times \mathcal{P}$.
10: $\hat{\rho}_m = \sum_{k=1}^K \rho_m^{(k)} q^{(k)}$, $m = 1, \ldots, M$.
```

An alternative, which may be more efficient in practice if a good proposal distribution can be found, is to employ a Metropolis-Hastings sampler instead, which we shall refer to as MRP-MH. Other samplers, including a hybrid Gibbs sampler, hereafter refered to as MRP-Gibbs, such as the one introduced in [20] are possible.

4 Multitask Policy Optimality prior (MPO)

Specifying a parametric form for the policy, such as the softmax, is rather awkward and hard to justify. It is more natural to specify a prior on the *optimality* of the policy demonstrated. Given the data, and a prior over a policy class (e.g. stationary policies), we obtain a posterior distribution on policies. Then, via a simple algorithm, we can combine this with the optimality prior and obtain a posterior distribution on reward functions.

As before, let $\boldsymbol{D}$ be the observed data and let ξ be a prior probability measure on the set of policies $\mathcal{P}$, encoding our biases towards specific policy types. In addition, let $\{\, \psi(\cdot \mid \pi) : \pi \in \mathcal{P} \,\}$ be a set of probability measures on $\mathcal{R}$, indexed in $\mathcal{P}$, to be made precise later. In principle, we can now calculate the marginal posterior over reward functions ρ given the observations $\boldsymbol{D}$, as follows:

$$\psi(B \mid \boldsymbol{D}) = \int_{\mathcal{P}} \psi(B \mid \pi) \,\mathrm{d}\xi(\pi \mid \boldsymbol{D}), \qquad B \subset \mathcal{R}. \tag{4.1}$$

The main idea is to define a distribution over reward functions, via a prior on the optimality of the policy followed. The first step is to explicitly define the measures on $\mathcal{R}$ in terms of ε-optimality, by defining a prior measure β on $\mathbb{R}_{+}$, such that $\beta([0, \varepsilon])$ is our prior that the policy is ε-optimal. Assuming that $\beta(\varepsilon) = \beta(\varepsilon \mid \pi)$ for all π, we obtain:

$$\psi(B \mid \pi) = \int_{0}^{\infty} \psi(B \mid \varepsilon, \pi) \,\mathrm{d}\beta(\varepsilon), \tag{4.2}$$

where $\psi(B \mid \varepsilon, \pi)$ can be understood as the prior probability that $\rho \in B$ given that the policy π is ε-optimal. The marginal (4.1) can now be written as:

$$\psi(B \mid \boldsymbol{D}) = \int_{\mathcal{P}} \left(\int_{0}^{\infty} \psi(B \mid \varepsilon, \pi) \,\mathrm{d}\beta(\varepsilon) \right) \mathrm{d}\xi(\pi \mid \boldsymbol{D}) \tag{4.3}$$

We now construct $\psi(\cdot \mid \varepsilon, \pi)$. Let the set of ε-optimal reward functions with respect to π be: $\mathcal{R}^{\pi}_{\varepsilon} \triangleq \left\{ \rho \in \mathcal{R} : \|V^{*}_{\rho} - V^{\pi}_{\rho}\|_{\infty} < \varepsilon \right\}$. Let $\lambda(\cdot)$ be an arbitrary measure on $\mathcal{R}$ (e.g. the counting measure if $\mathcal{R}$ is discrete). We can now set:

$$\psi(B \mid \varepsilon, \pi) \triangleq \frac{\lambda(B \cap \mathcal{R}^{\pi}_{\varepsilon})}{\lambda(\mathcal{R}^{\pi}_{\varepsilon})}, \qquad B \subset \mathcal{R}. \tag{4.4}$$

Then $\lambda(\cdot)$ can be interpreted as an (unnormalised) prior measure on reward functions. If the set of reward functions $\mathcal{R}$ is finite, then a simple algorithm can be used to estimate preferences, described below.

We are given a set of demonstration trajectories $\boldsymbol{D}$ and a prior on policies ξ, from which we calculate a posterior on policies $\xi(\cdot \mid \boldsymbol{D})$. We sample a set of K policies $\mathit{\Pi} = \{\pi^{(i)} : i = 1, \ldots, K\}$ from this posterior. We are also given a set of reward functions $\mathcal{R}$ with associated measure $\lambda(\cdot)$. For each policy-reward pair $(\pi^{(i)}, \rho_j) \in \mathit{\Pi} \times \mathcal{R}$, we calculate the loss of the policy for the given reward function to obtain a loss matrix:

$$L \triangleq [\ell_{i,j}]_{K \times |\mathcal{R}|}, \qquad\qquad \ell_{i,j} \triangleq \sup_{s} V^{*}_{\rho_j}(s) - V^{\pi^{(i)}}_{\rho_j}(s), \tag{4.5}$$

where $V^*_{\rho_j}$ and $V^{\pi^{(i)}}_{\rho_j}$ are the value functions, for the reward function ρ_j, of the optimal policy and $\pi^{(i)}$ respectively.[9]

Given samples $\pi^{(i)}$ from $\xi(\pi \mid \boldsymbol{D})$, we can estimate the integral (4.3) accurately via $\hat{\psi}(B \mid \boldsymbol{D}) \triangleq \frac{1}{K} \sum_{i=1}^{K} \int_0^\infty \psi(B \mid \varepsilon, \pi^{(i)}) \, \mathrm{d}\beta(\varepsilon)$. In addition, note that the loss matrix L is finite, with a number of distinct elements at most $K \times |\mathcal{R}|$. Consequently, $\psi(B \mid \varepsilon, \pi^{(i)})$ is a piece-wise constant function with respect to ε. Let $(\varepsilon_k)_{k=1}^{K \times |\mathcal{R}|}$ be a monotonically increasing sequence of the elements of L. Then $\psi(B \mid \varepsilon, \pi^{(i)}) = \psi(B \mid \varepsilon', \pi^{(i)})$ for any $\varepsilon, \varepsilon' \in [\varepsilon_k, \varepsilon_{j+1}]$, and:

$$\hat{\psi}(B \mid \boldsymbol{D}) \triangleq \sum_{i=1}^{K} \sum_{k=1}^{K \times |\mathcal{R}|} \psi(B \mid \varepsilon_k, \pi^{(i)}) \beta([\varepsilon_k, \varepsilon_{k+1})). \tag{4.6}$$

Note that for an exponential prior with parameter c, we have $\beta([\varepsilon_k, \varepsilon_{k+1}]) = e^{-c\varepsilon_k} - e^{-c\varepsilon_{k+1}}$. We can now find the optimal policy with respect to the expected utility.

Theorem 1. *Let $\hat{\eta}_k(\cdot \mid \boldsymbol{D})$ be the empirical posterior measure calculated via the above procedure and assume ρ takes values in $[0,1]$ for all $\rho \in \mathcal{R}$. Then, for any value function V_ρ,*

$$\mathbb{E}_\eta(\|V_\rho - \hat{V}_\rho\|_\infty \mid \boldsymbol{D}) \leq \frac{1}{(1-\gamma)\sqrt{K}} \left(2 + \frac{1}{2}\sqrt{\ln K}\right), \tag{4.7}$$

where the expectation is taken w.r.t the marginal distribution on $\mathcal{R}$.

This theorem, whose proof is in the appendix, bounds the number of samples required to obtain a small loss in the value function estimation, and holds with only minor modifications for both the single and multi-task cases for finite $\mathcal{R}$. For the multi-task case and general $\mathcal{R}$, we can use MPO-MC (Alg. 2 on the next page), to sample N reward functions from a prior. Unfortunately the theorem does not apply directly for infinite $\mathcal{R}$. While one could define an ϵ-net on $\mathcal{R}$, and assume smoothness conditions, in order to obtain in optimality guarantees for that case, this is beyond the scope of this paper.

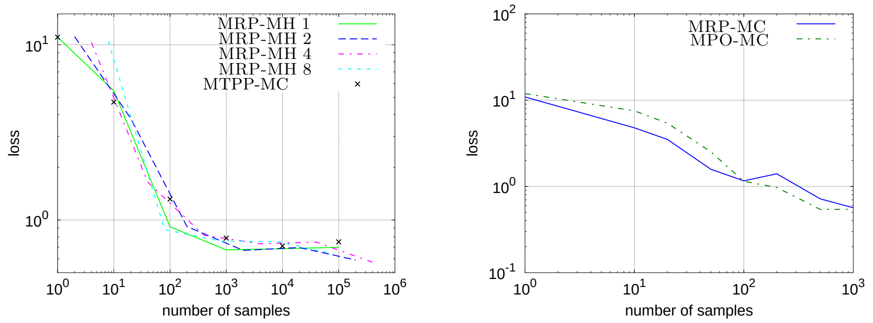
5 Experiments

Given a distribution on the reward functions ψ, and known transition distributions, one can obtain a stationary policy that is optimal with respect to this distribution via value iteration. This is what single-task algorithms essentially do, but it ignores differences among tasks. In the multi-task setting, we infer the optimal policy $\hat{\pi}^*_m$ for the m-th task. Its L_1-loss with respect to the optimal value function is $\ell_m(\hat{\pi}^*_m) \triangleq \sum_{s \in \mathcal{S}} V^*_{\rho_m}(s) - V^{\pi}_{\rho_m}(s)$. We are interested in minimising the total loss $\sum_m \ell_m$ across demonstrations. We first examined the efficiency

[9] Again, we abuse notation slightly and employ V_{ρ_j} to denote the value function of the MDP (ν, ρ_j), for the case when the underlying CMP ν is known. For the case when we only have a belief ω on the set of CMPs $\mathcal{N}$, V_{ρ_j} refers to the expected utility with respect to ω, or more precisely $V^{\pi}_{\rho_j}(s) = \mathbb{E}_\omega(U_t \mid s_t = s, \rho_j, \pi) = \int_{\mathcal{N}} V^{\pi}_{\nu,\rho_j}(s) \, \mathrm{d}\omega(\nu)$.

Algorithm 2 MPO-MC Multitask Policy Optimality Monte Carlo posterior estimate

```
Sample $N$ reward functions $\rho_1, \ldots, \rho_N \sim \psi$.
for $k = 1, \ldots, K$ do
    $(\xi^{(k)}, \psi^{(k)}) \sim \eta$, where $\psi^{(k)}$ is multinomial over $N$ outcomes.
    for $m = 1, \ldots, M$ do
        $\pi_m^{(k)} \sim \xi^{(k)}(\cdot \mid \boldsymbol{d}_m)$.
    end for
end for
Calculate $\hat{\phi}_m(\cdot \mid \boldsymbol{d}_m)$ from (4.6) and $\{\pi_m^{(k)} : k = 1, \ldots, K\}$.
```



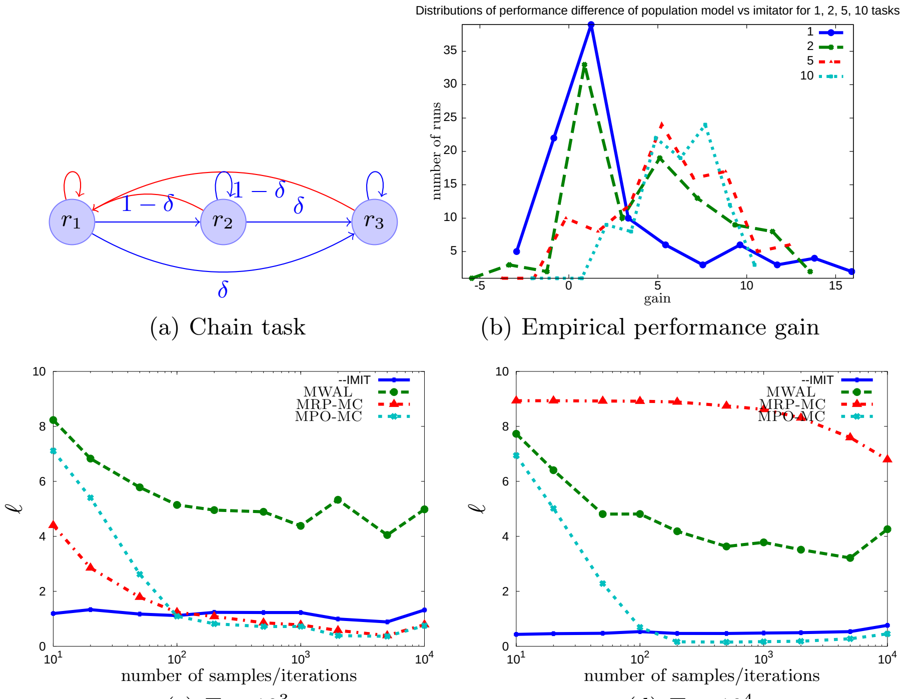
(a) Sampling comparison

(b) Model comparison

Fig. 2. Expected loss for two samplers, averaged over 10^3 runs, as the number of total samples increases. Fig. 2(b) compares the MRP and MPO models using a Monte Carlo estimate. Fig. 2(a) shows the performance of different sampling strategies for the MTPP model: *Metropolis-Hastings* sampling, with different numbers of parallel chains and simple *Monte Carlo* estimation.

of sampling. Initially, we used the *Chain* task [8] with 5 states (c.f. Fig. 3(a)), $\gamma = 0.95$ and a demonstrator using standard model-based reinforcement learning with ϵ-greedy exploration policy using $\epsilon = 10^{-2}$, using the Dirichlet prior on reward functions. As Fig. 2(a) shows, for the MRP model, results slightly favour the single chain MH sampler. Figure 2(b) compares the performance of the MRP and MPO models using an MC sampler. The actual computing time of MPO is larger by a constant factor due to the need to calculate (4.6).

In further experiments, we compared the multi-task perfomance of MRP with that of an imitator, for the generalised chain task where rewards are sampled from a Dirichlet prior. We fixed the number of demonstrations to 10 and varied the nnumber of tasks. The gain of using a multi-task model is shown in Fig. 3(b). Finally, we examined the effect of the demonstration's length, independently of the number of task. Fig. 3(c),3(d) show that when there is more data, then MPO is much more efficient, since we sample directly from $\xi(\pi \mid \boldsymbol{D})$. In that case, the MRP-MC sampler is very inefficient. For reference, we include the performance of MWAL and the imitator.



(a) Chain task

(b) Empirical performance gain

(c) $T = 10^3$

(d) $T = 10^4$

Fig. 3. Experiments on the chain task. (a) The 3-state version of the task. (b) Empirical performance difference of MRP-MC and Imit is shown for $\{1, 2, 5, 10\}$ tasks respectively, with 10 total demonstrations. As the number of tasks increases, so does the performance gain of the multitask prior relative to an imitator. (c,d) Single-task sample efficiency in the 5-state Chain task with $r_1 = 0.2$, $r_2 = 0$, $r_3 = 1$. The data is sufficient for the imitator to perform rather well. However, while the MPO-MC is consistently better than the imitator, MRP-MC converges slowly.

The second experiment samples *variants* of Random MDP tasks [20], from a hierarchical model, where Dirichlet parameters are drawn from a product of $\mathcal{Gamma}(1, 10)$ and task rewards are sampled from the resulting Dirichlets. Each demonstration is drawn from a softmax policy with respect to the current task, with $c \in [2, 8]$ for a total of 50 steps. We compared the loss of policies derived from MRP-MC, with that of algorithms described in [16, 18, 21], as well as a flat model [20]. Fig. 4(a) on the following page shows the loss for varying c, when the (unknown) number of tasks equals 20. While flat MH can recover reward functions that lead to policies that outperform the demonstrator, the multi-task model MRP-MH shows a clear additional improvement. Figure 4(b) shows that this increases with the number of available demonstrations, indicating that the task distribution is estimated well. In contrast, RP-MH degrades slowly, due to its assumption that all demonstrations share a common reward function.

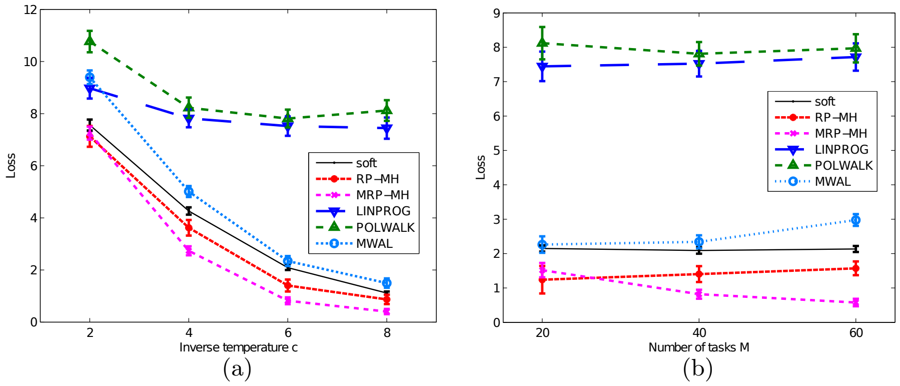


Fig. 4. Experiments on random MDP tasks, comparing MTPP-MH with the original (RP-MH) sampler[20], a demonstrator employing a softmax policy (soft), Policy Walk (PolWalk) [18] and Linear Programming (LinProg) [16] MWAL [21], averaged over 10^2 runs. Fig. 4(a) shows the loss as the inverse softmax temperature c increases, for a fixed number of $M = 20$ tasks Fig. 4(b) shows the loss relative to the optimal policy as the number of tasks increases, for fixed $c = 8$. There is one 50-step demonstration per task. The error bars indicate standard error.

6 Related work and discussion

A number of inverse reinforcement learning [1, 5, 7, 16, 18, 20, 23] and preference elicitation [4, 6] approaches have been proposed, while multitask learning itself is a well-known problem, for which hierarchical Bayesian approaches are quite natural [13]. In fact, two Bayesian approaches have been considered for multitask reinforcement learning. Wilson et al. [22] consider a prior on MDPs, while Lazaric and Ghavamzadeh [14] employ a prior on value functions.

The first work that we are aware of that performs multi-task estimation of utilities is [3], which used a hierarchical Bayesian model to represent relationships between preferences. Independently to us, [2] recently considered the problem of learning for multiple intentions (or reward functions). Given the number of intentions, they employ an expectation maximisation approach for clustering. Finally, a generalisation of IRL to the *multi-agent* setting, was examined by Natarajan et al. [15]. This is the problem of finding a good *joint* policy, for a number of agents acting simultaneously in the environment.

Our approach can be seen as a generalisation of [3] to the dynamic setting of inverse reinforcement learning; of [2] to full Bayesian estimation; and of [20] to multiple tasks. This enables significant potential applications. For example, we have a first theoretically sound formalisation of the problem of learning from multiple teachers who all try to solve the same problem, but which have different preferences for doing so. In addition, the principled Bayesian approach allows us to infer a complete distribution over task reward functions. Technically, the work presented in this paper is a direct generalisation of our previous paper [20], which proposed single task equivalents of the policy parameter priors discussed in Sec. 3, to the multitask setting. In addition to the introduction of multiple tasks, we provide an alternative policy optimality prior, which is a not only a much more natural prior to specify, but for which we can obtain computational complexity bounds.

In future work, we may consider non-parametric priors, such as those considered in [10], for the policy optimality model of Sec. 4. Finally, when the MDP is unknown, calculation of the optimal policy is in general much harder. However, in a recent paper [9] we show how to obtain near-optimal memoryless policies for the unknown MDP case, which would be applicable in this setting.

Acknowledgements

This work was partially supported by the BMBF Project Bernstein Fokus: Neurotechnologie Frankfurt, FKZ 01GQ0840, the EU-Project IM-CLeVeR, FP7-ICT-IP-231722, and the Marie Curie Project ESDEMUU, Grant Number 237816.

A Auxillary results and proofs

Lemma 1 (Hoeffding inequality). *For independent random variables $X_1, \ldots, X_n$ such that $X_i \in [a_i, b_i]$, with $\mu_i \triangleq \mathbb{E}\, X_i$ and $t > 0$:*

$$\mathbb{P}\left(\sum_{i=1}^n X_i \geq \sum_{i=1}^n \mu_i + nt\right) = \mathbb{P}\left(\sum_{i=1}^n X_i \leq \sum_{i=1}^n \mu_i - nt\right) \leq \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Corollary 1. *Let $g : X \times Y \to \mathbb{R}$ be a function with total variation $\|g\|_{TV} \leq \sqrt{2/c}$, and let P be a probability measure on Y. Define $f : X \to \mathbb{R}$ to be $f(x) \triangleq \int_Y g(x,y)\,\mathrm{d}P(y)$. Given a sample $y^n \sim P^n$, let $f^n(x) \triangleq \frac{1}{n}\sum_{i=1}^n g(x, y_i)$. Then, for any $\delta > 0$,with probability at least $1 - \delta$, $\|f - f^n\|_\infty < \sqrt{\frac{\ln 2/\delta}{cn}}$.*

Proof. Choose some $x \in X$ and define the function $h_x : Y \to [0,1]$, $h_x(y) = g(x,y)$. Let h_x^n be the empirical mean of h_x with $y_1, \ldots, y_n \sim P$. Then note that the expectation of h_x with respect to P is $\mathbb{E}\, h_x = \int h_x(y) dP(y) = \int g(x,y) dP(y) = f(x)$. Then $P^n\left(\{y^n : |f(x) - f^n(x))| > t\}\right) < 2e^{-cnt^2}$, for any x, due to Hoeffding's inequality. Substituting gives us the required result.

Proof (Proof of Theorem 1 on page 7). Firstly, note that the value function has total variation bounded $1/(1-\gamma)$. Then corollary 1 applies with $c = 2(1-\gamma)^2$. Consequently, the expected loss can be bounded as follows:

$$\mathbb{E}\, \|V - \hat{V}\|_\infty \leq \frac{1}{1-\gamma}\left(\sqrt{\frac{\ln 2/\delta}{2K}} + \delta\right).$$

Setting $\delta = 2/\sqrt{K}$ gives us the required result.

Bibliography

[1] P. Abbeel and A.Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML 2004*, 2004.

[2] Monica Babes, Vukosi Marivate, Michael Littman, and Kaushik Subramanian. Apprenticeship learning about multiple intentions. In *ICML 2011*.

[3] A. Birlutiu, P. Groot, and T. Heskes. Multi-task preference learning with gaussian processes. In *ESANN 2009*, pages 123–128, 2009.

[4] C. Boutilier. A POMDP formulation of preference elicitation problems. In *AAAI 2002*, pages 239–246, 2002.

[5] Jaedeug Choi and Kee-Eung Kim. Inverse reinforcement learning in partially observable environments. *Journal of Machine Learning Research*, 12:691–730, 2011.

[6] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *ICML 2005*, 2005.

[7] A. Coates, P. Abbeel, and A.Y. Ng. Learning for control from multiple demonstrations. In *ICML 2008*, pages 144–151. ACM, 2008.

[8] Richard Dearden, Nir Friedman, and Stuart J. Russell. Bayesian Q-learning. In *AAAI/IAAI*, pages 761–768, 1998.

[9] Christos Dimitrakakis. Robust Bayesian reinforcement learning through tight lower bounds. In *EWRL 2011*, 2011.

[10] Finale Doshi-Velez, David Wingate, Nicholas Roy, and Joshua Tenenbaum. Nonparametric Bayesian policy priors for reinforcement learning. In *NIPS 2010*, pages 532–540. 2010.

[11] Thomas S. Ferguson. Prior distributions on spaces of probability measures. *The Annals of Statistics*, 2(4):615–629, 1974. ISSN 00905364.

[12] J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339, 1989.

[13] T. Heskes. Solving a huge number of similar tasks: a combination of multi-task learning and a hierarchical Bayesian approach. In *ICML98*, pages 233–241. Citeseer, 1998.

[14] A. Lazaric and M. Ghavamzadeh. Bayesian multi-task reinforcement learning. In *ICML 2010*, 2010.

[15] S. Natarajan, G. Kunapuli, K. Judah, P. Tadepalli, K. Kersting, and J. Shavlik. Multi-agent inverse reinforcement learning. In *ICMLA 2010*, pages 395–400. IEEE.

[16] Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *ICML 2000*, pages 663–670. Morgan Kaufmann, 2000.

[17] Marting L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New Jersey, US, 2005.

[18] D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *IJCAI 2007*, volume 51, page 61801, 2007.

[19] Herbert Robbins. An empirical Bayes approach to statistics. 1955.

[20] Constantin A. Rothkopf and Christos Dimitrakakis. Preference elicitation and inverse reinforcement learning. In *ECML 2011*, 2011.

[21] Umar Syed and Robert E. Schapire. A game-theoretic approach to apprenticeship learning. In *NIPS 2008*, volume 10, 2008.

[22] A. Wilson, A. Fern, S. Ray, and P. Tadepalli. Multi-task reinforcement learning: a hierarchical Bayesian approach. In *ICML 2007*, pages 1015–1022. ACM, 2007.

[23] Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. Modelling interaction via the principle of maximum causal entropy. In *ICML 2010*, Haifa, Israel, 2010.