Biol Cybern (2013) 107:477–490
DOI 10.1007/s00422-013-0562-6

ORIGINAL PAPER

Biological Cybernetics

Modular inverse reinforcement learning for visuomotor behavior

Constantin A. Rothkopf · **Dana H. Ballard**

Received: 24 August 2012 / Accepted: 17 June 2013 / Published online: 6 July 2013

Abstract In a large variety of situations one would like to have an expressive and accurate model of observed animal or human behavior. While general purpose mathematical models may capture successfully properties of observed behavior, it is desirable to root models in biological facts. Because of ample empirical evidence for reward-based learning in visuomotor tasks, we use a computational model based on the assumption that the observed agent is balancing the costs and benefits of its behavior to meet its goals. This leads to using the framework of reinforcement learning, which additionally provides well-established algorithms for learning of visuomotor task solutions. To quantify the agent's goals as rewards implicit in the observed behavior, we propose to use inverse reinforcement learning, which quantifies the agent's goals as rewards implicit in the observed behavior. Based on the assumption of a modular cognitive architecture, we introduce a modular inverse reinforcement learning algorithm that estimates the relative reward contributions of the component tasks in navigation, consisting of following a path while avoiding obstacles and approaching targets. It is shown how to recover the component reward weights for individual tasks and that variability in observed trajectories can be explained succinctly through behavioral goals. It is demonstrated through simulations that good estimates can be obtained already with modest amounts of observation data, which in turn allows the prediction of behavior in novel configurations.

Keywords Inverse reinforcement learning · Visuomotor behavior · Spatial navigation · Task priorities

C. A. Rothkopf (✉)
Frankfurt Institute for Advanced Studies, Goethe University,
60438 Frankfurt, Germany
e-mail: rothkopf@fias.uni-frankfurt.de

C. A. Rothkopf
Institute of Cognitive Science, University Osnabrück,
49076 Osnabrück, Germany

C. A. Rothkopf
Technical University Darmstadt, 64283 Darmstadt, Germany

D. H. Ballard
Department for Computer Science, University of Texas at Austin,
Austin, TX 78712, USA

1 Introduction

Finding expressive and accurate models of animal and human behavior is an important goal within cognitive science, neuroscience, and artificial intelligence. Models of visuomotor behavior, specifically navigation, have been developed, which are able to describe well the observed average trajectories that human subjects tend to choose when walking toward a target while avoiding obstacles (Fajen and Warren 2003; Schöner and Dose 1992). This is achieved by assuming attractive and repulsive forces exerted by targets and obstacles on the navigating agent leading to a dynamical system governing the resulting trajectories. While these models are good mathematical models in that they are able to reproduce average trajectories, they are not necessarily computational models as it is difficult to relate the underlying model assumptions and parameters to visuomotor parameters and cognitive quantities. Thus, it is not clear what the assumed attractive and repulsive force fields in these models are.

Substantial empirical and theoretical work has shown that visuomotor behavior and visuomotor learning in animals and humans can be explained through mechanisms of reward-mediated learning (e.g., Glimcher 2004; Graybiel et al. 1994; Gold and Shadlen 2007; Daw and Doya 2006; Seymour et al. 2004; Haber 2003). Further evidence has demonstrated that

such behavior can be described well quantitatively using the formal framework of reinforcement learning (RL) (Sutton and Barto 1998). RL constitutes a large family of algorithms with the goal of solving the optimal control problem through some form of learning, i.e., the agent learns how to solve a task based on the experience it accumulates while interacting with an environment quantifying the costs and benefits of its actions. The generality of RL has lead to the successful interpretation of a large variety of problems in sequential decision making and visuomotor behavior (e.g Barto 1995; Daw and Doya 2006). Furthermore, RL has not only been shown to describe observed psychophysical measures of behavior but has also been successful in quantitatively explaining neuronal signals associated with visuomotor learning and behavior (e.g., Montague et al. 1996; Schultz et al. 1997; Daw et al. 2006; Bromberg-Martin et al. 2010). Specifically, at the algorithmic level temporal difference learning, a particular RL method for solving the optimal control problem has been shown to predict the neuronal activity of certain midbrain dopaminergic neurons. This success in explaining visuomotor behavior and learning through RL leads to the aim, which we pursue here, of finding methodologies for describing human visuomotor navigation behavior within the framework of reinforcement learning.

Methods for inferring the costs and benefits underlying observed behavior under the assumption of the agent acting according to RL are termed inverse reinforcement learning (IRL) (Ng and Russell 2000). These methods offer the possibility to infer the costs and benefits underlying observed behavior under the assumption that the animal or human is carrying out actions with the intention of maximizing or approximately maximizing its measure of success, the so called reward. In the case of navigation behavior, this means that utilizing IRL assumes that this behavior is governed by explicit costs and benefits that the organism is balancing to achieve its navigational goals. This is an appealing model as it is plausible to assume that navigation is a fundamental task both at the phylogenetic as well as ontogenetic levels and that therefore considerable optimization pressure exists on finding optimal solutions. As such, IRL is ideally suited to describe observed behavior in terms of a reward function expressing the preferences of the organism for different states of the world, which reflect the inherent costs and benefits of navigation.

An additional area of interest in building an accurate model of observed agent behavior through IRL is that of imitation learning. For an agent to learn a new task by trial and error can be very time-consuming and even impractical depending on the task difficulty. To that end researchers have studied ways of learning by demonstration (Whitehead and Ballard 1991; Whitehead 1991; Pastor et al. 2009; Billard and Mataric 2001). Learning by watching other agents' performance has gained wide acceptance with the discovery of specific motor neurons in monkeys that respond to a task component regardless of whether they themselves are performing a task of whether they are watching another perform it. Such results in part have motivated imitation learning whereby an agent observes the demonstrations of another agent and uses that data to infer sets of motor actions that would duplicate the behavior. One can think of this venue as having a demonstrator who produces the behavior and an observer who attempts to interpret the behavior in terms of an RL model using IRL.

Work by several authors has expanded on the IRL idea introduced by Ng and Russell (2000) within the framework of RL. A probabilistic approach was suggested in Ramachandran and Amir (2007), see also Lopes et al. (2009) and Neu and Szepesvári (2007), in which the authors provide an intuitively appealing formulation of the likelihood of observing state–action pairs from a demonstrator given a reward function. By sampling from a prior distribution over reward functions and evaluating the samples' likelihood within a Markov chain Monte Carlo framework, Ramachandran and Amir (2007) compute the posterior mean over reward functions as the best estimate explaining the observed actions of the agent. A technique related to IRL has also been applied to human navigation data in Ziebart et al. (2010). Subsequently, a general Bayesian formulation for general relationships between rewards and values and general action selection functions including possibly suboptimal ones was developed in Rothkopf and Dimitrakakis (2001) and a hierarchical Bayesian extension allowing for principled modeling in the case of multiple demonstrators or multiple tasks was developed in Dimitrakakis and Rothkopf (2011).

Here,[1] we start from the formulation given in Ramachandran and Amir (2007) (see also Lopes et al. 2009) in which it is assumed that state–action pairs corresponding to high future expected rewards are more likely to be observed compared to less valuable ones. But in the present paper, we propose methods that are specifically targeted at the challenges encountered when trying to infer the costs and benefits underlying human navigation behavior. First, usually only limited data are available from a single subject, limited by the experimental conditions, but it is essential to estimate the implicit rewards on a trial by trial and subject by subject basis to avoid only estimating average behavior. Secondly, substantial additional domain knowledge about the underlying reward functions in such tasks can be used. It is, e.g., reasonable to assume that the reward associated with reaching a target is determined by the position of that target itself and not by some arbitrary position relative to an obstacle. Thirdly, full knowledge of the underlying tran-

[1] The basis of the model developed in this paper was published previously as part of a PhD thesis (Rothkopf 2008).

sition functions governing the agent may be difficult to obtain so that it may be easier to come up with a family of value functions as the result of a simulation or a suitable prior.

A further important assumption, for which evidence is abundant, is that of the modular organization of behavior. The general idea of a modular organization of cognitive processes has appeared in the literature in many different variations (see e.g., Minsky 1988; Barrett and Kurzban 2006; Fodor 1983; Pinker 1999; Brooks 1986). Concrete and direct evidence for a modular organization of task solutions in the human brain conform with the modular RL framework used in the present paper comes from experiments by Gershman et al. (2009). Human subjects made simultaneous decisions with their left and right hands and received separate rewards for each hand movement. Not only was the choice behavior better described by a modular learning model that decomposed the values of bimanual movements into separate values for each component but also blood-oxygen-level-dependent activity in reward-learning-related areas of the subjects' brains reflected specific values of modular RL models. This suggests that the human brain can use modular decompositions to solve RL problems allowing for the factorization used in this paper.

Under the above assumptions, we propose a methodology for estimating the relative reward weights associated with the component tasks underlying human navigation such as obstacle avoidance, path following, and target approach. We improve on Ramachandran and Amir (2007) in several ways. First, we show that if one can assume a parametric form of the reward function by incorporating additional knowledge about which states are rewarded, it is not required to solve the RL problem for each sample in the Markov chain as the reward function's parameters can be computed directly. The problem reduces to finding the linear scaling of the expected total discounted reward associated with each individual task component, the so called Q-functions. Secondly, as only parameterizations of Q-functions are used, it is not obligatory to have transition functions for the considered tasks but one can start from appropriately parametrized Q-functions or from a prior over Q-functions. On the other hand, if one has access to the transition functions and knows which states are rewarded, it is possible to precompute the corresponding Q-functions. Furthermore, in the particular navigation task presented, we show that reasonably accurate estimates for the component reward weights can be obtained from a single experimental trajectory of only 40 m. Finally, the modular IRL formulation not only allows estimating rewards in cases where the composite state space would be intractable for other IRL methods because of its high dimensionality, but leads to an appealing interpretation of the estimated reward weights as priorities within a modular cognitive architecture.

2 Background

The problem setting is that of a Markov decision processes (MDP) (Puterman 1994). An individual MDP consists of a 4-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$ with $\mathcal{S}$ being the set of possible states, $\mathcal{A}$ the set of possible actions, $\mathcal{T}$ the transition model describing the probabilities $P(s_{t+1}|s_t, a_t)$ of reaching a state s_{t+1} when being in state s_t at time t and executing action a_t, and $\mathcal{R}$ is a reward model that describes the expected value of the reward r_t, which is distributed according to $P(r_t|s_t, a_t)$ and is associated with the transition from state s_t to some state s_{t+1} when executing action a_t.

In RL, the dynamics of the environment $\mathcal{T}$ and the reward function $\mathcal{R}$ may not be known in advance. One central goal in classical RL is to assign a value $V^\pi(s)$ to each state, which represents the expected total discounted reward obtainable when starting from the particular state s and following the policy π thereafter:

$$V^\pi(s) = E^\pi\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) \tag{1}$$

Alternatively, the values can be parametrized by state and action pairs, leading to the so called "Q" values $Q^\pi(s, a)$. RL attempts finding a policy π that maps from the set of states $\mathcal{S}$ to actions $\mathcal{A}$ so as to maximize the expected total discounted future rewards through some form of learning (Sutton and Barto 1998), where Q^* denotes the Q-value associated with the optimal policy π^*, and the optimal value of a state s can be written as $V^*(s) = \max_a Q^*(s, a)$. The optimal Q-values can be expressed through the recursive Bellman optimality equation:

$$Q^*(s, a) = \sum_r r P(r|s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^*(s') \tag{2}$$

Temporal difference learning (Sutton 1988) is one specific algorithm for computing the optimal values through continuous experience with the environment and uses the error between the current estimated values of states and the observed reward to drive learning. Evidence for temporal difference learning in animals comes from a multitude of studies (e.g., Schultz et al. 1997). In a related Q-learning form, the value of state–action pairs is adjusted by this temporal difference error δ_Q between current Q-value estimates and observed rewards using a learning rate α:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \delta_Q \tag{3}$$

Evidence for the representation of action values in the brain has also been found (e.g., Samejima et al. 2005). One specific classic RL algorithm for updating the Q-values according to the temporal difference error is the SARSA algorithm (Rummery and Niranjan 1994), an on-policy temporal difference learning rule, i.e., one in which the updates of the state and

action values reflect the current policy derived from these value estimates. The update in this case is:

$$\delta_Q = r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t). \tag{4}$$

2.1 Modular RL

Early work in RL noted that the problems associated with learning task solutions in high dimensional state spaces, i.e., spaces for which the number of states increases exponentially in the number of dimensions, could be simplified by taking the statistical structure present in the respective domain into account so as to somehow factor the problem. This idea has been proposed by several authors early on and has reappeared in many different settings (see e.g., Dayan and Hinton 1992; Kaelbling 1993; Humphrys 1996; Singh and Cohn 1998). More recent approaches (Chang et al. 2004; Sprague and Ballard 2003; Russell and Zimdars 2003) use decompositions of the transition function $\mathcal{T}$ and reward function $\mathcal{R}$ to simplify learning tasks. The idea is that separate representations for the states of individual tasks are available and that actions by the agent influence state transitions and rewards individually and independently for the separate tasks. This allows evaluating the value of individual state–action pairs for the overall task by assessing the value of the module components separately.

For the most general composite state spaces, the relationship between the optimal value functions for each of the individual component tasks and the global task in which multiple objectives are pursued depend on the overall structure of the problem and can be very complex. But specifically in the considered visuomotor tasks such as navigation, the assumptions of independence hold. Moving with respect to any obstacle and moving with respect to a target are in general independent of each other, i.e., the transition functions governing these tasks can be factored. Similarly, rewards associated with obstacles are independent of rewards associated with targets. We use a formulation of modular RL that is consistent with previous work (e.g., Sprague and Ballard 2003; Russell and Zimdars 2003; Rothkopf and Ballard 2010). A module can be defined as an MDP, i.e., the n-th module is given by:

$$M^{(n)} = \{\mathcal{S}^{(n)}, \mathcal{A}, \mathcal{T}^{(n)}, \mathcal{R}^{(n)}\} \tag{5}$$

where the superscripts reflect that the information is referred to the particular MDP. As these modules are all embedded within a single agent, the action space is unitary and shared among all modules.

The expression in Eq. 5 incorporates the assumption that the state transitions and reward functions are independent in the respective modules, which can be directly expressed as:

$$P(s_{t+1}|s_t, a_t) = \prod_{n=1}^{N} P(s_{t+1}^{(n)}|s_t^{(n)}, a_t) \tag{6}$$

$$P(r_t|s_t, a_t) = \prod_{n=1}^{N} P(r_t^{(n)}|s_t^{(n)}, a_t) \tag{7}$$

While this may seem a rather restrictive set of assumptions for general tasks, note again that this holds for the navigation tasks considered in this paper and evidence that the human brain indeed uses such factored representations has been provided (Gershman et al. 2009).

Given the computations of optimal Q-values for the individual task components, a single action needs to be selected by the agent. As in previous literature, we consider the action selection to be based on the aggregate value estimate:

$$Q(s_t, a_t) = \sum_{n=1}^{N} Q^{(n)}(s_t^{(n)}, a_t^{(n)}) \tag{8}$$

In general, one may consider some form of action selection in order to mediate the competition between actions proposed by individual modules. In accordance to both theoretical work on RL as well as empirical results in animal and human sequential behavior, we consider here the probabilistic softmax action selection. This formulation includes an inverse temperature parameter, which expresses the degree of randomness in the action selection. Once the action a has been selected, it is used for all modules. Note that the problem of learning in this modular setting has been considered previously (Sprague and Ballard 2003; Russell and Zimdars 2003). The basic result is that if an on-policy learning method such as SARSA is used, as described in Eq. 4, the locally learned Q-functions can be combined to give the globally optimal policy.

3 Modular IRL

Ng and Russell (2000) formulate the problem of inverse reinforcement learning as inferring the reward function $\mathcal{R}$ an agent implicitly maximizing by only observing its behavior and knowing the transition function governing the state dynamics. This corresponds to assuming that the transition function $\mathcal{T}$ of the MDP is given and that there are T observations of state–action pairs (s_t, a_t) at each moment in time t, which together constitute the observed data D. Implicitly, this therefore also assumes that the specific representation of the state space and action spaces is know, which we also assume in the present study. Probabilistic formulations of the IRL problem were subsequently provided (Ramachandran and Amir 2007; Lopes et al. 2009). Following these authors, we model the likelihood of observing a single state–action pair to be given by:

$$P(s_t, a_t|Q^*, \eta) = \frac{e^{\eta Q^*(s_t,a_t)}}{\sum_b e^{\eta Q^*(s_t,b)}} \tag{9}$$

This captures the intuition that the higher the Q-value of a state–action pair, the more likely you are to observe it. In this context, η expresses the degree of confidence with which the optimal Q-values are actually selected by the observed agent. This expression can also be interpreted as a softmax action selection together with a uniform distribution over state visitations. Furthermore, we assume the observed state–action pairs to be conditionally independent so that the likelihood L of the parameters given the entire observation D consisting of T individual state–action pairs can be written as:

$$\begin{aligned} L &= P(D|Q^*, \eta) \\ &= P((s_1, a_1), \ldots, (s_T, a_T))|Q^*, \eta) \\ &= \prod_{t=1}^{T} \frac{e^{\eta Q^*(s_t,a_t)}}{\sum_b e^{\eta Q^*(s_t,b)}} \end{aligned} \tag{10}$$

In the case of the modular RL formulation, the Q-functions of the composite tasks need to be summed to find the global Q-values. Substituting Eq. 8 into Eq. 10 gives:

$$\begin{aligned} P(D|Q^{*(n)}, \eta) &= \prod_{t=1}^{T} \frac{e^{\eta \sum_{n=1}^{N} Q^{*(n)}(s_t^{(n)},a_t)}}{\sum_b e^{\eta \sum_{n=1}^{N} Q^{*(n)}(s_t^{(n)},b)}} \\ &= \prod_{t=1}^{T} \prod_{n=1}^{N} \frac{e^{\eta Q^{*(n)}(s_t^{(n)},a_t)}}{\sum_b e^{\eta Q^{*(n)}(s_t^{(n)},b)}} \end{aligned} \tag{11}$$

where the second line is obtained by expanding all terms in the exponentials and collecting terms associated with individual actions b for all individual modules n in the denominator.

As mentioned before, the problem we want to address is that of estimating how much individual component tasks contribute to the observed behavior, i.e., the relative rewards in situations in which the agent follows multiple objectives. We furthermore may only have very limited amounts of observation data. The solution we propose here is to use additional prior knowledge about the Q-functions, underlying the observed behavior. This utilizes the result by Neumann et al. (1947) that a positive linear transform of reward functions leads to a rescaling of the corresponding Q-function by the same scalar, so that the $Q^{*(n)}$ can be written as scaled versions of the Q-function $Q_{\mathbf{1}}^{*(n)}$, which represents the Q-function for a total reward of 1:

$$Q^{*(n)}(s^{(n)}, a) = w^{(n)} Q_{\mathbf{1}}^{*(n)}(s^{(n)}, a) \tag{12}$$

This has the advantage that now we are neither searching for reward functions nor searching for Q-functions $Q^{*(n)}(s^{(n)}, a)$ over state–action pairs but only for coefficients $w^{(n)}$, as the normalized Q-functions $Q_{\mathbf{1}}^{*(n)}(s^{(n)}, a)$ can be precomputed. Importantly, this formulation does not restrict the origin of the Q-functions, i.e., they may be obtained by solving a known MDP, they may be sampled from a suitable prior over Q-functions, or they may simply be obtained by simulation. By substituting Eq. 12 into Eq. 8, we obtain an expression for the joint Q-values with the corresponding weights:

$$Q(s_t, a_t) = \sum_{n=1}^{N} w^{(n)} Q_{\mathbf{1}}^{*(n)}(s_t^{(n)}, a_t) \tag{13}$$

Note, however, that we now require the sum of the weights $w^{(n)}$ to sum to one $\sum_{n=1}^{N} w^{(n)} = 1$ because increasing the weights for individual modules would lead to an increase in the likelihood of observing corresponding state–action pairs without bound. This alteration of the original problem (Ramachandran and Amir 2007) is significant as it allows estimating the modular reward contributions without suffering from the indeterminacy in the general formulation of the IRL problem (Ng and Russell 2000; Ramachandran and Amir 2007). The factors $w^{(n)}$ now quantify the relative contributions of the component task.

In the limit, in which it is known where in the respective state spaces reward is available, there will be one basis-Q-function for each component RL module:

$$\begin{aligned} L &= P(D|Q_{\mathbf{1}}^{*(n)}, w^{(n)}, \eta) \\ &= P((s_1, a_1), \ldots, (s_T, a_T))|Q_{\mathbf{1}}^{*(n)}, w^{(n)}, \eta) \\ &= \prod_{t=1}^{T} \prod_{n=1}^{N} \frac{e^{\eta w^{(n)} Q_{\mathbf{1}}^{*(n)}(s_t^{(n)},a_t)}}{\sum_b e^{\eta w^{(n)} Q_{\mathbf{1}}^{*(n)}(s_t^{(n)},b)}} \end{aligned} \tag{14}$$

subject to:

$$\sum_{n=1}^{N} w^{(n)} = 1 \tag{15}$$

where we used the notation $Q_{\mathbf{1}}^{(n)}$ to denote the basis-Q-function for the component module n obtained with a reward of 1 and $w^{(n)}$ is the factor scaling Q-function n.

Inference about the latent Q-function weights $w^{(n)}$ that best describes the data D obtained from the demonstrator can then proceed by adopting the original policy-walk algorithm (Ramachandran and Amir 2007). The main difference is that now one requires a suitably chosen prior distribution over task weights $w^{(n)}$ and not over reward functions $\mathcal{T}$. But crucially, the modular framework not only renders the estimation more tractable computationally when using the policy-walk algorithm but also allows for the estimation of the unknown scaling factors conveniently by using maximum likelihood estimation as will be shown subsequently.

First, we adapt policy-walk (Ramachandran and Amir 2007) by sampling along a grid in weight space andevaluating

the corresponding posterior. The costly computation in the original grid-walk algorithm is to do policy iteration at each step in the Markov chain to obtain a policy from a new reward vector. Although computation is saved in the original algorithm by reutilizing the previously calculated policy in the next iteration of the Markov chain, this is still costly. We avoid this by rescaling the precomputed policies.

Utilizing additional knowledge about the composition of the tasks, it is desirable to express this in form of a prior over reward weights $w^{(n)}$. This allows computing a posterior over reward functions R, or more precisely given the above formulation, a posterior over the reward weights $w^{(n)}$. Using Bayes theorem:

$$P(w^{(n)}|D, Q_{\mathbf{1}}^{*(n)}, \eta) \propto P(D|w^{(n)}, Q_{\mathbf{1}}^{*(n)}, \eta) P(w^{(n)}, Q_{\mathbf{1}}^{*(n)}, \eta) \tag{16}$$

For the above formulation using individual weights for basis-Q-functions as given in Eq. 14 such additional knowledge could be that we know a priori that a particular component task has a much higher reward associated with it, say because it is known what the main task of the demonstrator was. Accordingly, a prior over reward functions is now a prior over individual weights $w^{(n)}$. A suitable prior in this case is the Dirichlet distribution that can assign different prior probabilities to individual sets of weights which sum to one and represent the individual task weights:

$$P(w^{(1)}, \ldots, w^{(N-1)}; \alpha^{(1)}, \ldots, \alpha^{(N)}) = \frac{\prod_{i=1}^{N} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{N} \alpha_i)} \prod_{i=1}^{N} w^{(i)\alpha_i - 1} \tag{17}$$

A noninformative prior can be implemented by setting all the α_i to 0.5 while, e.g., the prior belief that one particular task has a higher associated reward than the other tasks can be expressed by $\alpha_1 > \alpha_i$ for all $i \neq 1$.

Algorithm 1 Gridwalk: MCMC on the grid of Q-value weights $w^{(n)}$.

1: Pick initial random weights $w^{(n)}$ with $\sum_{i=1}^{n} w^{(n)} = 1$ from prior
2: Compute $Q^* = w^{(n)} Q_{\mathbf{1}}^{*(n)}(s, a, \eta)$
3: **repeat**
4: Pick random weights $\tilde{w}^{(n)}$ on grid of size δ with $\sum_{i=1}^{n} \tilde{w}^{(n)} = 1$
5: Compute $\tilde{Q}^* = \tilde{w}^{(n)} Q_{\mathbf{1}}^{*(n)}(s, a, \eta)$
6: Compute $\alpha = \frac{P(\tilde{Q}^*)}{P(Q^*)}$
7: **if** $\alpha \geq 1$ **then**
8: $w^{(n)} \leftarrow \tilde{w}^{(n)}$
9: **else**
10: $w^{(n)} \leftarrow \tilde{w}^{(n)}$ with probability α
11: **end if**
12: **until** criterion

4 ML estimates for regularized modular IRL

While the above adaptation of the policy-walk algorithm to searching in the space of task weights has clear computational advantages, it is possible to directly obtain a maximum likelihood solution, which will be developed in the following.

Maximum likelihood obtains point estimates for the factors $w^{(n)}$ of each basis-Q-function by maximizing the likelihood in Eq. 14. Note that because the Q-values for a reward of 1 can be precomputed and the data D consists of state–action pairs, the only unknowns in Eq. 14 are the $w^{(n)}$. The maximum likelihood estimate of the reward function R can be obtained as follows: We maximize the logarithm of the likelihood function Eq. 14 by setting its derivative to 0 and incorporate the constraint Eq. 15 that the individual factors $w^{(n)}$ sum to one using a Lagrange multiplier. Thus, the maximization problem to solve is:

$$\max_{w^{(m)}} \sum_{t=1}^{T} \sum_{n=1}^{N} \eta w^{(n)} Q_{\mathbf{1}}^{*(n)}(s_t^{(n)}, a_t) - \log\left(\sum_b e^{\eta w^{(n)} Q_{\mathbf{1}}^{*(n)}(s_t^{(n)}, b)} \right) + \lambda \left(\sum_{n=1}^{N} w^{(n)} - 1 \right) \tag{18}$$

We proceed by calculating the gradient of the log-likelihood with respect to some weight $w^{(m)}$:

$$\frac{\partial \log L}{\partial w^{(m)}} = \sum_{t=1}^{T} \eta Q_{\mathbf{1}}^{*(m)}(s_t^{(m)}, a_t) - \eta \frac{\sum_b Q_{\mathbf{1}}^{*(m)}(s_t^{(m)}, b) e^{\eta w^{(m)} Q_{\mathbf{1}}^{*(m)}(s_t^{(m)}, b)}}{\sum_b e^{\eta w^{(m)} Q_{\mathbf{1}}^{*(m)}(s_t^{(m)}, b)}} + \lambda$$

and with respect to λ:

$$\frac{\partial \log L}{\partial \lambda} = \sum_{n=1}^{N} w^{(n)} - 1 \tag{19}$$

We can now proceed using numerical optimization to find the maximum of the log-likelihood. We note that one can change the objective to take into account that the critical points may occur at saddle points instead of at local maxima of the log-likelihood. This is accomplished by minimizing the squared magnitude of the gradient, so that the function to minimize is:

$$h(w^{(1)}, \ldots, w^{(N)}, \lambda) = \left(\sum_{t=1}^{T} \eta Q_{\mathbf{1}}^{*(m)}(s_t^{(m)}, a_t) - \eta \frac{\sum_b Q_{\mathbf{1}}^{*(m)}(s_t^{(m)}, b) e^{\eta w^{(m)} Q_{\mathbf{1}}^{*(m)}(s_t^{(m)}, b)}}{\sum_b e^{\eta w^{(m)} Q_{\mathbf{1}}^{*(m)}(s_t^{(m)}, b)}} + \lambda \right)^2 + \left(\sum_{n=1}^{N} w^{(n)} - 1 \right)^2 \tag{20}$$

Finally, it may be advantageous to incorporate a sparsifying prior on the weights $w^{(n)}$ especially in cases in which the number of component tasks may be large. This can be achieved by including an ℓ_1 regularization term. This leads to the following maximization problem for the regularized modular IRL:

$$\max_{w^{(m)}} \sum_{t=1}^{T}\sum_{n=1}^{N} \eta w^{(n)} Q_{\mathbf{1}}^{*(n)}(s_t^{(n)}, a_t) - \log\left(\sum_b e^{\eta w^{(n)} Q_{\mathbf{1}}^{*(n)}(s_t^{(n)},b)}\right) + \lambda_1\left(\sum_{n=1}^{N} w^{(n)} - 1\right) + \lambda_2\left(\sum_{n=1}^{N} ||w^{(n)}||_1\right) \quad (21)$$

While many different approaches to ℓ_1 regularization have been proposed (see, e.g., Schmidt et al. 2007), we use a smooth approximation to the non-differentiable ℓ_1 norm proposed in (Schmidt et al. 2007). Following Lee and Mangasarian (2001), we define the $(x)_+$ function as:

$$(x)_+ = x + \frac{1}{\alpha}\log(1 + e^{-\alpha x}) \quad (22)$$

which derives from the integral of the sigmoid function. The parameter α is strictly positive and in practice sufficiently large to render the approximation appropriate. This allows approximating the absolute value function using the sum of the integral of two sigmoid functions by writing $|x| \approx (x)_+ + (-x)_+$. The first derivative necessary in the ℓ_1 regularization can be approximated by differentiating the $(x)_+$ function and defining the derivative as:

$$(x)_+^{'} = (1 + e^{-\alpha x})^{-1} \quad (23)$$

leading to the smooth approximation $\frac{d}{dx}|x| \approx (x)_+^{'} - (-x)_+^{'}$, which is the sigmoid function $\tanh(\alpha x/2)$. Similarly as before, this objective can be maximized by various numerical methods including gradient descent on the squared magnitude of the gradient:

$$h(w^{(1)}, \ldots, w^{(N)}, \lambda_1, \lambda_2) = \left(\sum_{t=1}^{T} \eta Q_{\mathbf{1}}^{*(m)}(s_t^{(m)}, a_t) - \eta \frac{\sum_b Q_{\mathbf{1}}^{*(n)}(s_t^{(n)}, b) e^{\eta w^{(n)} Q_{\mathbf{1}}^{*(n)}(s_t^{(n)},b)}}{\sum_b e^{\eta w^{(n)} Q_{\mathbf{1}}^{*(n)}(s_t^{(n)},b)}} + \lambda_1 + \lambda_2\left((w^{(m)})_+^{'}\right)\right)^2 + \left(\sum_{n=1}^{N} w^{(n)} - 1\right)^2 + \left(\sum_{n=1}^{N} (w^{(n)})_+\right)^2 \quad (24)$$

5 Human avatar simulation methodology

We tested the algorithm on navigation tasks inspired by the work by Fajen and Warren (2003) and Schöner and Dose (1992) and a specific setting described in Sprague and Ballard (2007). An agent has to walk down a sidewalk and attend to three tasks: following the path of the walkway, approaching targets, and avoiding obstacles as shown in Fig. 1. In all of the following simulations, we used regularized maximum likelihood estimation of the reward weights with the objective function specified in Eq. 24. The parameter governing the influence of the regularization term was determined empirically by selecting the value giving the smallest total deviation in the reward estimates across all tested reward settings.

The representation of the state space is punctate and similar as in previous work (Sprague and Ballard 2007): There is only one state that corresponds to the delivery of a reward, i.e., the distance d of the human to the center of a cylinder is $d < 0.2$m. The other distances are discretized according to:

$$\lceil |S_d|(1 - e^{-0.5 * d * \log(2)}) \rceil$$

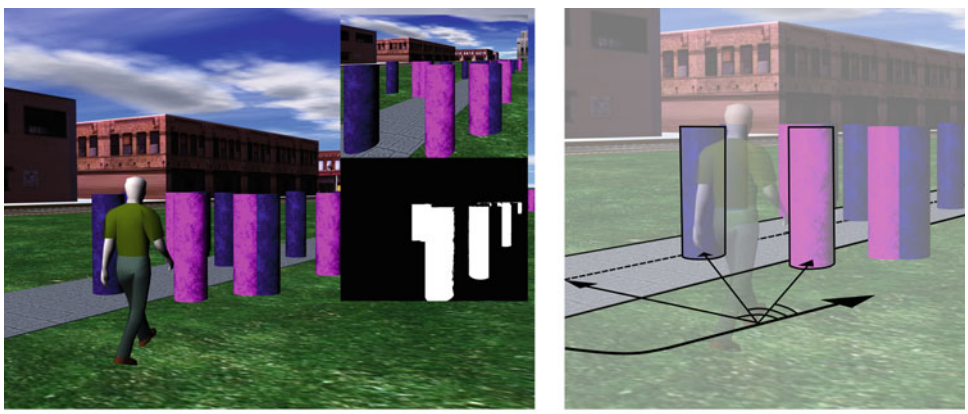

Fig. 1 Avatar simulation environment. **a** A frame from the human embedded vision system simulation. The insets show the use of vision to guide the humanoid through a complex environment. The *lower inset* shows the particular visual routine that is running at any instant. This instant shows the detection of target objects. The *upper inset* shows the visual field in a head-centered viewing frame. **b** Corresponding state space parameterization of the modules for target approach, obstacle avoidance, and walkway following

where $|S_d|$ is the dimensionality of the distance state space. See Fig. 3 for a depiction of this discretization of the state space.

Heading angles are $-50 \leq \theta \leq 50$ degrees, with linear spacing. The parametrization used for the distance to the walkway is:

$$\lceil |S_d|(0.5 + 0.5(1 - (2^{-0.5*|\rho)|}))\text{sign}(\rho)) \rceil$$

where ρ is the signed distance from the center line of the walkway. Given these representations, data can be collected for a given walk on the sidewalk for each of the three modules. For the current study, this results in a total of 121 states for each of the three component tasks giving a total of 363 states. Note that the full joint state space would give a total of more than 1.7 million states. Accordingly, the Q-functions have 605 state–action pairs, resulting in a total of 1815 state–action pairs for the four component tasks, compared with over 8 million state–action pairs for the joint Q-function. Accordingly, we are not aware of a current IRL method capable of inferring the full joint reward function.

First, it is necessary to determine how expressive this model is, i.e., what range of different behaviors can be obtained. Different values of reward for the modules lead to radically different behaviors. For example, Fig. 2 shows the different avatar trajectories for three different reward sets. In the topmost traces, the avatar is rewarded for all three component tasks, such that obstacles are avoided, targets are approached, and the path is maintained on the walkway. In the second case, the avatar is only rewarded for approaching targets, resulting in trajectories that leave the walkway and also walk through obstacles. Finally, in the third case, the avatar is only rewarded for avoiding obstacles, and he accordingly wanders off to the corners of the defined area. Note that although the trajectories in this latter case show a high degree of variability, they are all obtained by a single set of task weights. This demonstrates the importance of a computational model of visuomotor behavior that does not average over individual trajectories but accommodates a probabilistic relationship between behavioral goals and observed behavior.

6 Experimental results

The first and essential question to ask is: how well can the algorithm recover known rewards? To answer this question, we choose reward sets and have the avatar learn MDP modules. The important assumption here is that the learning avatar is allowed to have the state space of the avatar generating the data. This assumption may seem quite restrictive, but it is the common assumption in IRL (including Ng and Russell 2000; Ramachandran and Amir 2007; Lopes et al. 2009). Furthermore, for basic ambulatory behaviors, the agents in question operate under very similar constraints owing to common physical environments and movement systems. The experimental protocol for each individual episode has multiple steps:

1. Chosen rewards are used by the avatar to learn module MDP policies using Eqs. 2 and 3.
2. Runs using the learned policy are made to generate trajectories for the data sequences.

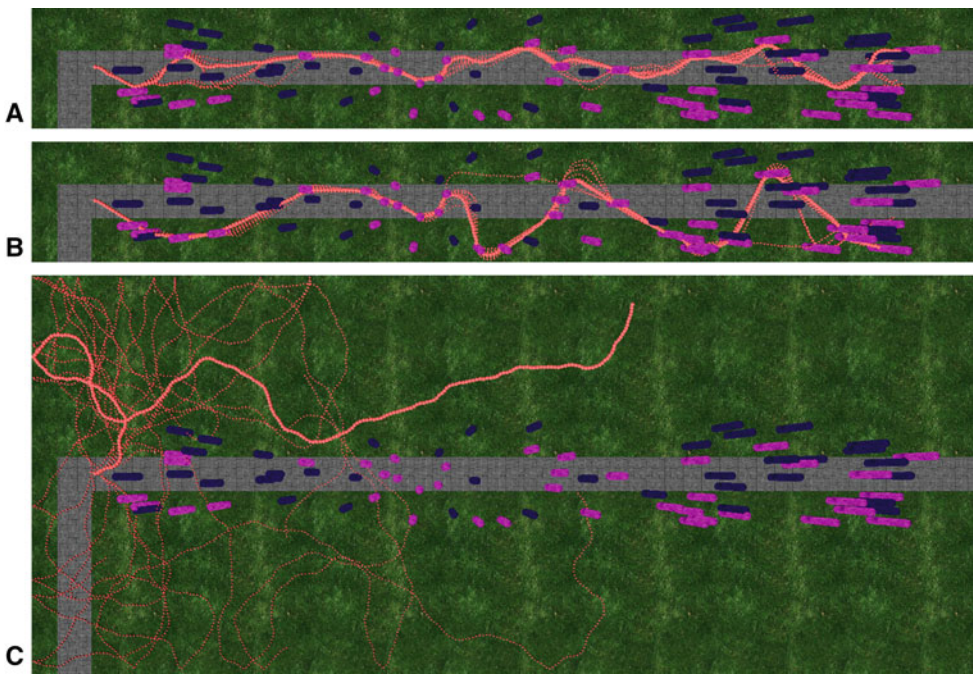

Fig. 2 Comparison of trajectories of the agent with different sets of task weights. **a** Target approach, obstacle avoidance, and walkway navigation are weighted as (0.5, 0.3, 0.2). **b** With only incentive to pick up targets, the avatar wonders off the walkway and hits obstacles. When targets are valuable, a large number of targets (*pink*) are collected. **c** With reward only being given for avoiding obstacles, there is neither an incentive to stay on the walkway nor to pick up targets (color figure online)

3. The IRL algorithm uses the data sequences to estimated the Q-functions' parameters solving the minimization problems of either Eq. 20 or 24.
4. The avatar uses the estimated rewards, which may not be exactly the same, to learn a new policy.
5. The avatar is given the same initial points and uses the computed policy to generate new trajectories that can be compared to the originals.

First, Q-functions are learned by solving the individual component task for a reward of 1, i.e., the Q-functions are precomputed solving the corresponding MDP and assigning a reward value of 1 to all rewarded states. Figure 3a shows the value functions corresponding to these Q-functions for the three modules' state spaces. The next step is to generate trajectories through different courses using this policy to provide data for testing the IRL algorithm. A new layout is sampled and the avatar is run for 300 time steps on this layout before a new layout is sampled. The state–action pairs constitute the data D that is used in the IRL algorithm. Figure 3b shows the recovered values for each of the component task, i.e., the value functions multiplied by the respective estimated weight $w^{(n)}$. Finally, Fig. 3c shows that the log-likelihood of the parameters given the data is well behaved and convex for the given example so that the global maximum is easily computable.

To get a more complete picture of the accuracy of the recovered reward weights we ran one hundred simulations for each possible reward weight vector on a reward grid of size 0.1 and quantified the estimation error by computing the root mean square (RMS) error of the estimated reward weights. The results are shown in Fig. 4a. Note that the two axes show the weights for the obstacle avoidance and target tasks, and the corresponding weight for walkway navigation can be obtained by computing the weight value that brings the sum of all weights to one. Therefore, the plot is on a simplex over the task weights. As can be seen, the accuracy of the estimation is quite good with an RMS error not exceeding

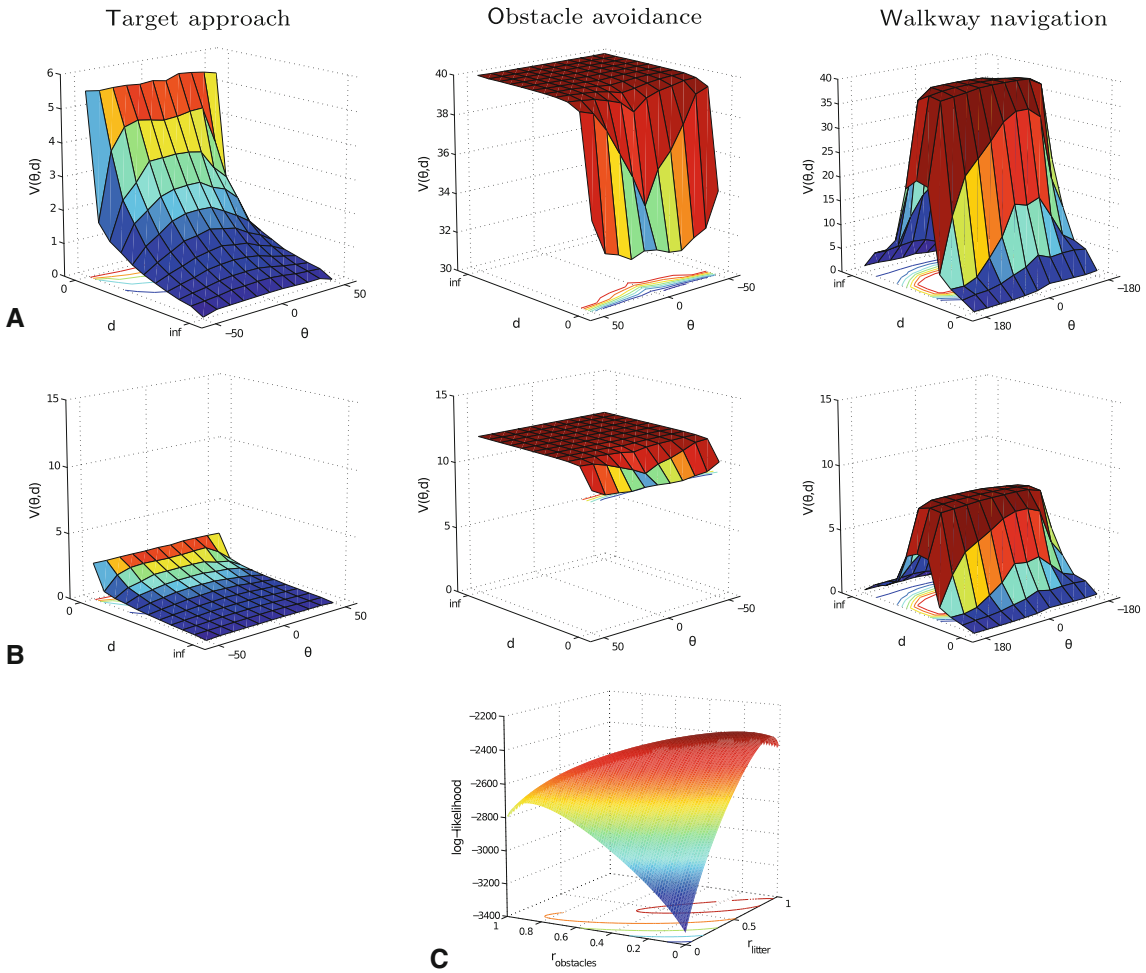

Fig. 3 **a** The value functions obtained for a reward of one unit. **b** The scaled value functions as recovered by the IRL algorithm, where the weights for the tasks are represented as (target approach, obstacle avoidance, walkway following) $w = (0.5, 0.3, 0.2)$. **c** Log-likelihood for the weights given the demonstrator's data

0.3 and being around 0.1 reward units for the majority of reward weights. The improvement of using the regularized problem formulation is significant as can be seen in Fig. 4b. The RMS error of the estimated reward weights is below 0.15 for almost all tested reward weights, despite the relatively modest amount of observed data.

Similarly, it is important to explore the estimates' sensitivity to the number of active behaviors. Ideally one would like to have a match between the number of active behaviors used by the trainer and the number assumed by the learner. But what are the consequences of a mismatch? Figure 4 also shows results relevant for answering this question, as it shows the cases in which the reward values for the obstacle avoidance or target approach are set to 0, i.e., cases in which the demonstrator only follows two of the three tasks. The plot demonstrates that again the RMS error is smaller than 0.08 and that the estimated reward weights are close to the true ones, also in the case of modules' rewards being 0.

Given that the reward functions are well estimated, one may now explore the extent to which the generated trajectories match. Figure 5a shows the original trajectory used to generate the data overlain with ten trajectory generated by starting from the same initial condition and using Q-functions scaled with the estimated weights. For comparison, we also show trajectories obtained by perturbing the true reward weights with uniform noise corresponding to 5, 10, and 20 % changes in the reward values.

To gain further insight in the variability of the trajectories, we again ran 100 simulations per reward weight vector obtained by considering all possible reward weights on a grid of step size 0.1. We first computed the RMS errors between the demonstrator trajectory and 100 trajectories sampled with the exact same weights. The results are shown in Fig. 6a. This was compared to the RMS error in both spatial dimensions between the demonstrator's trajectory and the trajectory obtained using the estimated weights. The corresponding plot is shown in Fig. 6b. The results show that large variability is present in the trajectories obtained from the simulated avatar especially under conditions in which the overall navigation behavior is dominated by obstacle avoidance. This agrees well with the observation made in Fig. 2c. Note that the same data on which Fig. 6b is based were used to estimate the reward weights in Fig. 4b. This shows that even if the variability in the trajectories may be large, this does not necessarily mean that the estimated reward weights are highly variable, capturing the intuition that variability in the trajectories can nevertheless be reflecting a clear behavioral goal. Accordingly, the differences between the observed trajectories and the trajectories carried out when utilizing the estimated reward weights show the largest deviation for behavior emphasizing obstacle avoidance. The total deviation of the trajectories for the vast majority of reward weights by contrast is small, as confirmed by the observations in Fig. 5.

When learning the rewards from data, one obvious point is that there must be sufficient data to produce accurate reward estimates. However the number of data points required can vary due to the different sensitivities to distal rewards in different parts of the data space. To elucidate this point, we

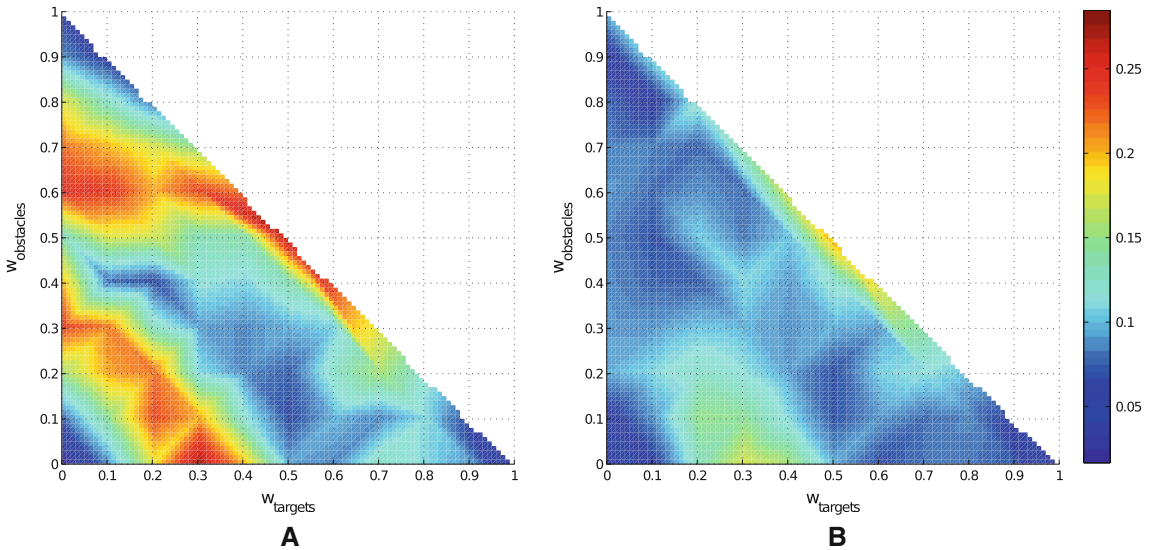

Fig. 4 RMS error of the estimated weight vector across possible reward weights. The weights were varied in steps of 0.1 reward units. The RMS error was computed on the basis of 100 trials with different object arrangements on the walkway for each weight value. **a** Results for maximum likelihood estimation using Eq. 20. **b** Results for regularized maximum likelihood estimation using Eq. 24 with $\lambda_2 = 10^{-4}$

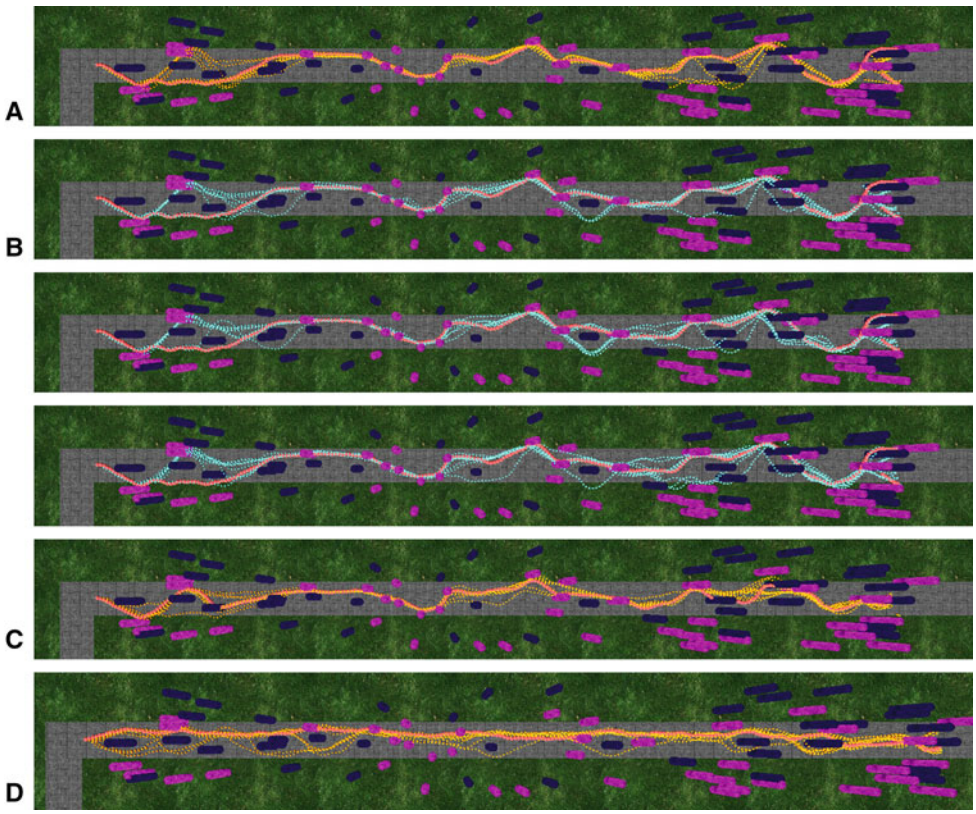

Fig. 5 Comparing trajectories generated from policies using recovered rewards with original trajectories. **a** Original trajectory for $w = (0.5, 0.3, 0.2)$ and ten simulated trajectories obtained using the estimated reward weights. **b** trajectories obtained by perturbing the original reward weights $w = (0.5, 0.3, 0.2)$ with different amounts of uniform noise. From top to bottom: 5, 10, and 20 % noise added. **c** Original trajectory and ten sample trajectories using estimated reward weights for $w = (0.55, 0, 0.45)$. **d** Original trajectory and ten sample trajectories using estimated reward weights for $w = (0, 0.55, 0.45)$

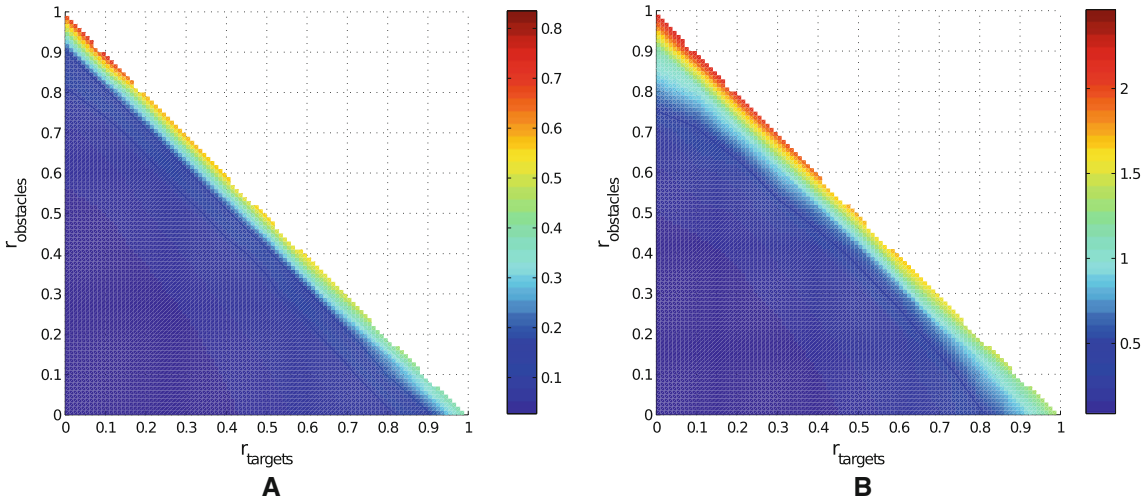

Fig. 6 Error in the trajectories. **a** Mean squared deviation between trajectories obtained with the same reward weights. **b** Mean squared deviation between the demonstrator's trajectory and trajectories obtained from the estimated weights

explore the sensitivity of the reward estimates to the size of the data by varying the total number of state–action pairs in the simulation together with different starting points and object arrangements. Thus, we simulated the avatar 100 times with different object arrangements, initial conditions, and reward weights. We then unitized different amounts of the observed data to estimate the reward weights. Using this data, we then calculate the mean and standard deviation of the learned reward values. Figure 7a shows that the estimation error for the reward weights on average are within an RMS error of 0.02 given only a single observed trajectory of 300 state–actions pairs.

The core property of the parameterization that emerges from our Q-function set is that the different reward values

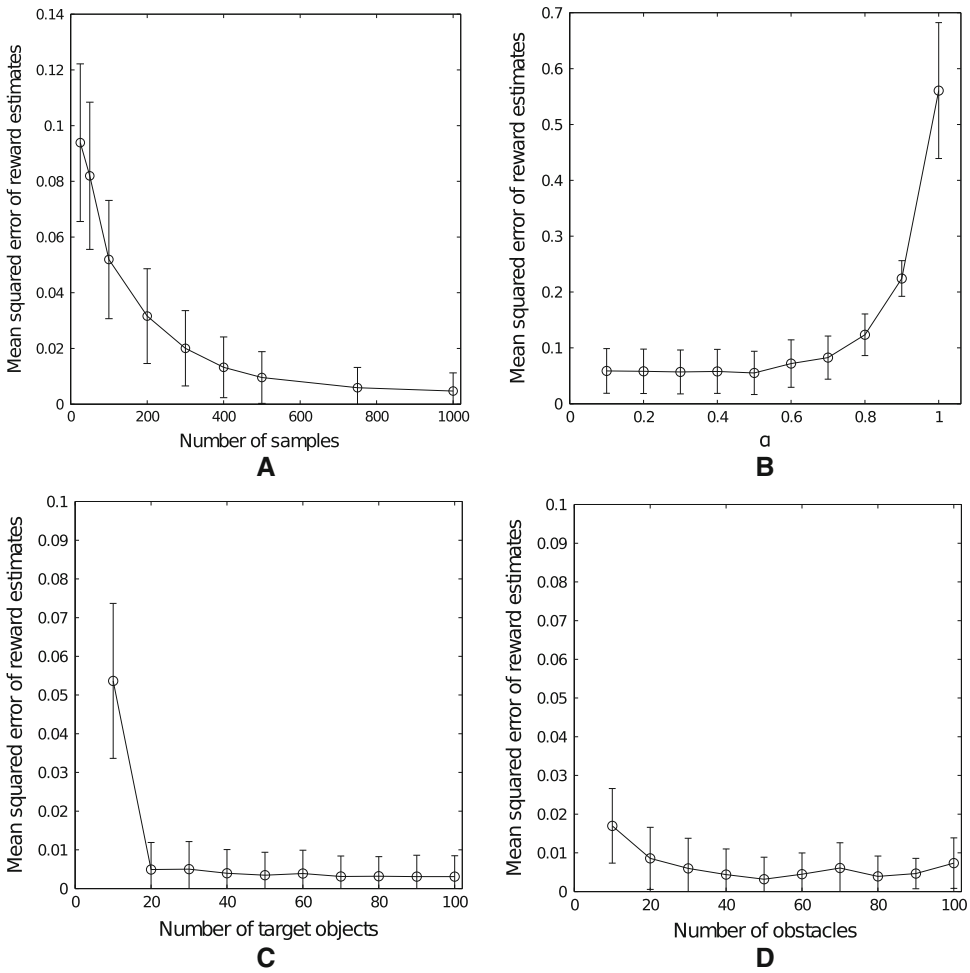

Fig. 7 Accuracy and precision of estimated reward weights. **a** Effect of the amount of data on the values of recovered rewards. Mean squared error of reward weights for different numbers of state–action samples, where a single trajectory of length 40 m along the walkway consists of 300 samples. **b** Testing the robustness to state estimation errors. Root mean squared error in the estimated reward vectors for different probabilities α of changing a state estimate weighted by the respective compression of the state spaces. **c** RMS error in weights estimates as a function of the total number of targets along the walkway. **d** RMS error in weights estimates as a function of the total number of obstacles along the walkway

for each behavior are just scaled version of each other. In effect, the assumption is that the joint reward functions and transition functions can be factorized. In the current navigation tasks, these assumptions are fulfilled. But the question arises, how many different states in each module's state space need to be visited in the data available for estimation to obtain accurate weight estimates? We ran the simulations with 10 to 100 target objects or obstacles and estimated the Q-functions weights for 100 examples per number of objects. Fig. 7c, d compares the RMS error in the reward estimates given different numbers of obstacles or targets. These figures demonstrate that the algorithm is very robust even when the number of interactions with the respective objects are small. Only when the number of objects is smaller than about 20, the avatar visits only a small subset of states in the respective state space and the estimates are significantly off.

The final issue we tackle is that of noise. Up to this point, the state values have all been assumed to be accurate to the precision used by the Q-tables, but in any practical system, there will be measurement errors so there remains a question as to their significance. While it is possible to incorporate a specific measurement noise model, this is beyond the scope of this paper. Instead, here, we establish empirically the robustness of the introduced algorithm. One reason that noise might be unimportant is that the basic sensory system that we assume logarithmically compresses the space around the avatar. This has the consequence of generating accurate measurements for near distances and headings to objects and less accurate measurements for those measurements to distant objects. The overall effect of these uncertainties will in the end be that the estimated state of the observer is not the correct state of the demonstrator. To that end, we model the noise in the observer's system by perturbing the current state of the observed agent by randomly changing individual state–action pairs to adjacent values with a uniform probability α normalized by the inverse area of the corresponding state. Thus, the probability of obtaining a perturbed state instead of the correct state variable is larger for states with a large surface area, i.e., states that are far away from the center of the obstacles or targets. This means that if $\alpha = 1$, the probability of changing a state variable for the states most distant to the targets or obstacles is 1, while the probability of changing the states closest to the target are 0.09, corresponding to the respective ratio of state space element areas. Fig. 7 shows the

results of these simulations by plotting the RMS error across all weights for each trajectory and confirms the robustness of the presented algorithm to state estimation errors. Because of the logarithmic state compression and the shape of the Q-functions as shown in Fig. 3, even severely perturbed state estimates still lead to small errors in the estimated weights.

7 Discussion and conclusions

Expressive and accurate models of agent behavior are required in many scientific fields, ranging from the need to quantify agent behavior to learning from other agents' behavior. While mathematical models of human navigation behavior are available, which describe the average trajectories across subjects and conditions well (Schöner and Dose 1992; Fajen and Warren 2003), these models are not framed in cognitive or behavioral quantities. As substantial research has demonstrated that visuomotor learning and behavior can be understood as an optimization process in the framework of reinforcement learning, it is desirable to develop methods connecting these approaches. This would also establish a connection to data in biological agents that has revealed that neuronal activity is correlated with several quantities in the above RL models.

In the present paper, we developed a methodology to estimate the relative reward contributions of multiple basic visuomotor tasks to observed navigation behavior. We introduced a modular IRL algorithm based on a parameterizations of Q-functions that reduces the IRL problem to the estimation of linear weights. This allows estimating the respective contributions of several goals such as obstacle avoidance, target approach, and path following in visuomotor behavior. The modular formulation uses additional assumptions on the shape of the reward functions to constrain the problem and makes inference computationally tractable even with small amounts of data. Specifically, instead of estimating the reward function for the full joint state space, we infer the relative weighting of the component Q-functions, which can be obtained by assuming that the shape of the reward function and the environment dynamics are known. For example, in the considered navigation task, we assume that the reward associated with obstacle avoidance is obtained when not intersecting with the obstacle, the reward associated with target approach is obtained when intersecting with the target, and the reward associated with walkway following is obtained when the agent is on the walkway. Together with known dynamics, this determines the shape of the Q-functions. Implicitly, as in other IRL methods, we assume that we know the state and action spaces required for the representations of the Q-functions. Based on these assumptions, we derived methods for doing maximum likelihood estimation of the respective task contributions. Finally, we derive a regularized formulation that uses an ℓ_1 norm to obtain sparse reward component weights, which favors only a small number of weights to be different from zero.

The simulations demonstrate that reward functions used by the agent that mimic human's performance on the task of traversing a walkway with multiple independent goals can be well recovered with modest amounts of observation data. An important result of the empirical evaluations is that the inherently probabilistic formulation of the walking task leads to variability in the trajectories, depending on the weighting of the component tasks. Despite this large variability of the trajectories, the corresponding weights can be recovered well. This means that models describing agent navigation behavior on the basis of average trajectories (Schöner and Dose 1992; Fajen and Warren 2003) may not capture the common behavioral goals underlying the agent's trajectories. Thus, variability in trajectories may reflect a very succinct task, which cannot be captured by modeling the trajectory themselves without reference to latent cause of the observed behavior, the composite behavioral goal.

The presented methodology has further appeal because it is based on the assumption of a modular cognitive architecture. Direct recent empirical support for such a modularity in reward-mediated visuomotor behavior gives further credence to this approach. Importantly, the computational framework naturally allows interpretation of the observed navigational behavior as the consequence of balancing costs and benefits in visuomotor tasks and can readily accommodate algorithms for the learning of these task solutions. We therefore hope that this methodology has broad appeal in cognitive science and neuroscience, where it can quantify behavior in terms of underlying costs and benefits with respect to behavioral goals.

Acknowledgments The research reported herein was supported by NIH Grants RR009283 and NSF grant 0932277. CR was additionally supported by the BMBF Project Bernstein Fokus: Neurotechnologie Frankfurt, FKZ 01GQ0840 and EU-Project IM-CLeVeR, FP7-ICT-IP-231722.

References

Barrett HC, Kurzban R (2006) Modularity in cognition: framing the debate. Psychol Rev 113(3):628

Barto AC (1995) Adaptive critics and the basal ganglia. In: Houk JC, Davis JL, Beiser DG (eds) Models of information processing in the basal ganglia. MIT Press, Cambridge, MA, pp 215–232

Billard A, Mataric MJ (2001) Learning human arm movements by imitation: evaluation of a biologically inspired connectionist architecture. Robotics Auton Syst 37:145–160

Bromberg-Martin ES, Matsumoto M, Hikosaka O (2010) Dopamine in motivational control: Rewarding, aversive, and alerting. Neuron 68:815–834

Brooks R (1986) A robust layered control system for a mobile robot. IEEE J Robotics Autom 2(1):14–23
Chang Y-H, Ho T, Kaelbling LP (2004) All learning is local: multi-agent learning in global reward games. In: Thrun S, Saul L, Schölkopf B (eds) Advances in neural information processing systems 16. MIT Press, Cambridge, MA
Daw ND, O'Doherty JP, Dayan P, Seymour B, Dolan RJ (2006) Cortical substrates for exploratory decisions in humans. Nature 441(7095): 876–879. ISSN 1476–4687. doi:10.1038/nature04766. URL http://www.ncbi.nlm.nih.gov/pubmed/16778890
Daw ND, Doya K (2006) The computational neurobiology of learning and reward. Curr Opin Neurobiol 16(2):199–204
Dayan P, Hinton GE (1992) Feudal reinforcement learning. In: Advances in neural information processing systems 5. Morgan Kaufmann Publishers, Burlington, pp 271–271
Dimitrakakis C, Rothkopf CA (2011) Bayesian multitask inverse reinforcement learning. In: European workshop on reinforcemnt learning (EWRL)
Fajen BR, Warren WH (2003) Behavioral dynamics of steering, obstable avoidance, and route selection. J Exp Psychol Hum Percept Perform 29(2):343
Fodor JA (1983) Modularity of mind. MIT Press, Cambridge, MA
Gershman SJ, Pesaran B, Daw ND (2009) Human reinforcement learning subdivides structured action spaces by learning effector-specific values. J Neurosci 29(43):13524–13531
Glimcher PW (2004) Decisions, uncertainty, and the brain: the science of neuroeconomics. MIT Press, Bradford Books, Cambridge, MA
Gold JI, Shadlen MN (2007) The neural basis of decision making. Annu Rev Neurosci 30(1):535–574. ISSN 0147–006X. doi:10.1146/annurev.neuro.29.051605.113038
Graybiel AM, Aosaki T, Flaherty AW, Kimura M (1994) The basal ganglia and adaptive motor control. Science 265(5180):1826–1831
Haber SN (2003) The primate basal ganglia: parallel and integrative networks. J Chem Neuroanat 26(4):317–330
Humphrys M (1996) Action selection methods using reinforcement learning. In: Maes P, Mataric M, Meyer J-A, Pollack J, Wilson SW (eds) From animals to animats 4: proceedings of the fourth international conference on simulation of adaptive behavior. MIT Press, Bradford Books, Cambridge, MA, pp 135–144
Kaelbling LP (1993) Hierarchical learning in stochastic domains: preliminary results. In: Proceedings of the tenth international conference on machine learning, vol 951, pp 167–173
Lee YJ, Mangasarian OL (2001) Ssvm: a smooth support vector machine for classification. Comput Optim Appl 20(1):5–22
Lopes M, Melo F, Montesano L (2009) Active learning for reward estimation in inverse reinforcement learning. In: Buntine W, Grobelnik M, Mladenić D, Shawe-Taylor J (eds) Machine learning and knowledge discovery in databases. Lecture notes in computer science, vol 5782. Springer, Berlin, Heidelberg, pp 31–46. http://dx.doi.org/10.1007/978-3-642-04174-7_3
Minsky M (1988) The society of mind. Simon and Schuster
Montague PR, Dayan P, Sejnowski TJ (1996) framework for mesencephalic dopamine systems based on predictive hebbian learning. J Neurosci 16:1936–1947
Neu G, Szepesvári C (2007) Apprenticeship learning using inverse reinforcement learning and gradient methods. In: Proceedings of the 23 conference on uncertainty in, artificial intelligence, pp 295–302
Ng AY, Russell S (2000) Algorithms for inverse reinforcement learning. In: Proceedings 17th international conference on machine learning, Morgan Kaufmann, pp 663–670
Pastor P, Hoffmann H, Asfour T, Schaal S (2009) Learning and generalization of motor skills by learning from demonstration. In: International conference on robotics and automation
Pinker SA (1999) How the mind works. Ann N Y Acad Sci 882(1):119–127
Puterman ML (1994) Markov decision processes. Wiley, New York, NY
Ramachandran D, Amir E (2007) Bayesian inverse reinforcement learning. In: 20th internatinal joint conference artificial intelligence
Rothkopf CA (2008) Modular models of task based visually guided behavior. PhD thesis, Department of Brain and Cognitive Sciences, Department of Computer Science, University of Rochester
Rothkopf CA, Ballard DH (2010) Credit assignment in multiple goal embodied visuomotor behavior. Frontiers in Psychology, 1, Special Issue on Embodied, Cognition (00173)
Rothkopf CA, Dimitrakakis C (2001) Preference elicitation and inverse reinforcement learning. In: 22nd European conference on machine learning (ECML)
Rummery GA, Niranjan M (1994) On-line Q-learning using connectionist systems. Technical report CUED/F-INFENG/TR 166, Cambridge University Engineering Department
Russell S, Zimdars AL (2003) Q-decomposition for reinforcement learning agents. In: Proceedings of the international conference on machine learning, vol 20, p 656
Samejima K, Ueda Y, Doya K, Kimura M (2005) Representation of action-specific reward values in the striatum. Science 310(5752):1337
Schmidt M, Fung G, Rosales R (2007) Fast optimization methods for l1 regularization: a comparative study and two new approaches. In: Kok J, Koronacki J, Mantaras R, Matwin S, Mladenic D, Skowron A (eds) Machine learning: ECML 2007, volume 4701 of Lecture notes in computer science, Springer, Berlin, 2007, pp 286–297. ISBN 978-3-540-74957-8
Schöner G, Dose M (1992) A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. Robotics Auton Syst 10(4):253–267
Schultz W, Dayan P, Montague PR (1997) A neural substrate of prediction and reward. Science 275:1593–1599
Seymour B, O'Doherty JP, Dayan P, Koltzenburg M, Jones AK, Dolan RJ, Friston KJ, Frackowiak RS (2004) Temporal difference models describe higher-order learning in humans. Nature 429(6992):664–667
Singh S, Cohn D (1998) How to dynamically merge Markov decision processes. In: Neural information processing systems 10, pp 1057–1063
Sprague N, Ballard D (2003) Multiple-goal reinforcement learning with modular sarsa(0). In: International joint conference on artificial intelligence, Acapulco, August 2003
Sprague N, Ballard DH (2007) Modeling embodied visual behaviors. ACM Trans Appl Percept 4(2):11
Sutton RS (1988) Learning to predict by the methods of temporal differences. Mach Learn 3:9–44
Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press, Cambridge, MA
Von Neumann J, Morgenstern O, Rubinstein A, Kuhn HW (1947) Theory of games and economic behavior. Princeton University Press, Princeton, NJ
Whitehead SD (1991) A complexity analysis of cooperative mechanisms in reinforcement learning. In: Proceedings of the association for artificial intelligence
Whitehead SD, Ballard DH (1991) Learning to perceive and act by trial and error. Mach Learn 7:45–83
Ziebart BD, Bagnell JA, Dey AK (2010) Modeling interaction via the principle of maximum causal entropy. In: Johannes F, Thorsten J (eds) Proceedings of the 27th international conference on machine learning (ICML-10), June 21–24, 2010. Haifa, Israel, pp 1255–1262