Biol Cybern
DOI 10.1007/s00422-014-0599-1

ORIGINAL PAPER

Biological Cybernetics

Learning strategies in table tennis using inverse reinforcement learning

Katharina Muelling · Abdeslam Boularias · Betty Mohler · Bernhard Schölkopf · Jan Peters

Received: 2 April 2013 / Accepted: 20 March 2014

Abstract Learning a complex task such as table tennis is a challenging problem for both robots and humans. Even after acquiring the necessary motor skills, a strategy is needed to choose where and how to return the ball to the opponent's court in order to win the game. The data-driven identification of basic strategies in interactive tasks, such as table tennis, is a largely unexplored problem. In this paper, we suggest a computational model for representing and inferring strategies, based on a Markov decision problem, where the reward function models the goal of the task as well as the strategic information. We show how this reward function can be discovered from demonstrations of table tennis matches using model-free inverse reinforcement learning. The resulting framework allows to identify basic elements on which the selection of striking movements is based. We tested our approach on data collected from players with different playing styles and under different playing conditions. The estimated reward function was able to capture expert-specific strategic information that sufficed to distinguish the expert among players with different skill levels as well as different playing styles.

Keywords Computational models of decision processes · Table tennis · Inverse reinforcement learning

This article forms part of a special issue of *Biological Cybernetics* entitled "Structural Aspects of *Biological Cybernetics*: Valentino Braitenberg, Neuroanatomy, and Brain Function.

K. Muelling (✉)· A. Boularias · B. Schölkopf · J. Peters
Max Planck Institute for Intelligent Systems, Spemannstr. 38, 72076 Tuebingen, Germany
e-mail: muelling@tuebingen.mpg.de
e-mail: kmuelling@nrec.ri.cmu.edu
e-mail: muelling@ias.tu-darmstadt.de

A. Boularias
e-mail: boularias@tuebingen.mpg.de

B. Schölkopf
e-mail: bs@tuebingen.mpg.de

J. Peters
e-mail: jrpeters@tuebingen.mpg.de

K. Muelling · A. Boularias
Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

A. Boularias
e-mail: aboularias@nrec.ri.cmu.edu

B. Mohler
Max Planck Institute for Biological Cybernetics, Spemannstr. 44, 72076 Tuebingen, Germany
e-mail: mohler@tuebingen.mpg.de

K. Muelling · J. Peters
FG Intelligente Autonome Systeme, Technische Universität Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany
J. Peters
e-mail: peters@ias.tu-darmstadt.de

1 Introduction

Understanding the complex interplay between learning, decision making and motion generation is crucial both for creating versatile, intelligent robot systems and for understanding human motor control. To make headway toward this goal, parsimonious models that "sculpt motor commands" based on a notion of optimal performances are needed (Braitenberg et al. 1997). Braitenberg (1984) showed more than 25 years ago that the key to understand this complex interplay is to create simple, elementary structures, such as his *Braitenberg Vehicles*, that nevertheless allow synthesizing complex behavior. Braitenberg vehicles correspond to control policies in reinforcement learning, which we can by today learn from demonstrations and by self-improvement. In the last decade, such approaches have matured in the robot learning context and led to robot systems that can learn the complex motor

Published online: 23 April 2014

skills including even basic robot table tennis (Muelling et al. 2013; Kober et al. 2012).

In complex competitive and cooperative motor tasks, mastering the task is not merely a matter of perfect execution of a specific movement pattern. For example, in table tennis, a player usually cannot win the game by always returning the ball safely to the same position. Instead, players need a good strategy that defines where and how to return the ball to the opponent's court. An action should always be chosen to have a high probability to successfully return the ball as well as to make the task of the opponent harder, i.e., it should improve the chance to win the game. In this paper, we want to make a first step toward understanding the decision processes underlying such a behavior. We follow Braitenberg's example of finding straightforward synthetic constituents of strategies rather than using complex physical models of the world. To accomplish this goal, we create a simplified model of human-human table tennis and study how basic strategic elements can be extracted from a game play.

In racket science, researcher identified so-called *winning patterns* in tennis video sequences in order to help trainers analyze their game (Wang et al. 2004; Wang and Parameswaran 2005; Vis et al. 2010). Here, specific repetitive movement patterns of both the players and the ball were turned into tactical templates. In table tennis, Hohmann et al. (2004) determined the transition probabilities of different stroke positions, directions and types individually. Such transition probabilities allow identifying the components that were used most efficiently. Diaz et al. (2013) showed that memory-based information is used for predictive eye movements in racquetball, and Seve et al. (2004) showed that such memory-based information is also used for strategies in table tennis. Seve et al. (2004) concluded from interviews with professional table tennis players that those selected their actions in a match not only based on the current situation, but also on the knowledge of sequences that have proven to be effective in the past in similar situations. Rather than identifying the frequencies and effectiveness of specific movement patterns in large data sets, we want to model this situation-based knowledge from a computational point of view and extract it from collected table tennis data. Such an approach would enable us to yield a better insight into the reasons for choosing a given action in a specific state and to use the learned model for artificial systems, such as table tennis robots (Muelling et al. 2013). Creating a model that accounts for the complexity of this task can easily lead to an intractable problem formulation. For this reason, we use a straightforward approximation to this problem and only consider basic features available to the player as well as perfect knowledge about the environment. In particular, we account for positional features of the players and the ball, but not for opponent-specific strategies, changes in such an opponent-specific strategy and spin. As a result, we are able to model this decision process as a Markov decision problem (MDP, Puterman (1994)).[1]

In an MDP framework, an agent interacts with a dynamic environment. It chooses and executes an action that will change the state of the agent and its environment (see Fig. 2). The agent can observe this state change and may receive a reward for its action. A strategy defines the general plan of choosing actions in specific states in order to achieve a goal. A strategy in the MDP framework is usually called a *policy* and is denoted by π. Given a MDP model, one can find an optimal policy using optimal control techniques (Sutton and Barto 1998; Powell 2011). The goal is to find a policy that maximizes the expected reward. The reward thus encodes the goal of the task. While it is possible to learn a policy directly from demonstrations using supervised learning (Schaal 1999; Argall et al. 2009), such behavioral cloning approaches usually have limited generalization abilities since they are restricted to the demonstrated scenarios. As they do not consider the underlying dynamics, they cannot be applied in a task with altered or constantly changing dynamics. In table tennis, the dynamics of the environment changes as the opponent changes. The player may also encounter new states and hence need to learn new strategic elements while his experience increases with training. Therefore, blindly following the strategy of an observed expert will not lead to a successful strategy. In this paper, we do not intend to mimic an observed strategy, instead we want to learn an underlying reward function that connects the information available to the player with his chosen actions.

Given an exact model, simple reward functions that only specify an immediate positive reward for winning, a negative one for losing a rally and zero reward of nonterminal actions may be sufficient. However, such simplified rewards will cause slow convergence rates for behavior generation as the system will need to pass through several state–action pairs before receiving a reward. Although winning the game remains a driving factor in their behavior, it remains unclear whether a simple winning strategy explains human playing behavior or whether humans learn subgoals leading to win. In artificial systems, however, such simplified reward functions are unsuited for learning table tennis due to the curse of dimensionality. Instead of predefining the reward function, we seek to identify it from human game play. Such an approach will also allow us to reveal memory-based knowledge and individual preferences of table tennis players. The process of determining the reward function from an expert demonstration is referred to as *inverse reinforcement learning* (IRL) or inverse optimal control (Boyd et al. 1994; Ng and Russel 2000). IRL has been applied to many problems such

[1] Note that in order to include such uncertain state information as assumptions about the strategy of the opponent or spin, a problem formulation in form of partial observable MDPs would be necessary.

as helicopter control (Abbeel et al. 2010), parking lot navigation (Abbeel et al. 2008), navigating a quadruped robot across different terrains (Kolter and Ng 2011), human navigation behavior (Rothkopf and Ballard 2013), routing preferences of drivers (Ziebart et al. 2008), modeling goal-directed trajectories of pedestrians (Ziebart et al. 2009) and user simulation in spoken dialog management systems (Chandramohan et al. 2011). In most of these approaches, the underlying dynamics of the system is assumed to be known. However, the dynamics of human behavior is usually difficult to model. We avoid modeling these complex dynamics by learning the strategies directly from human demonstration. Thus, the dynamics model underlying the task is implicitly encoded in the observed data. To collect demonstrations, we asked skilled and naive table tennis players to compete in several matches. We recorded the ball trajectories as well as the Cartesian position and orientation of the upper body joints for all players with a VICON motion capture system (see Fig. 1).

This paper does not focus on the introduction of new IRL methods for solving this kind of problem. We rather intend to apply existing methods on this new challenging problem. During the course of this paper, we will answer the following questions: (1) Can we infer a reward function that captures expert-specific information using model-free inverse reinforcement learning? (2) Using this reward function, can we distinguish players with different playing styles and skill levels? (3) Which parts of the sensory information are the key elements for selecting the movement parameters?

In the remainder of this paper, we will proceed as follows. In Sect. 2, we present the theoretical background for modeling decision processes, including MDPs and the used IRL algorithms. We present the experimental setup and evaluations in Sect. 3. In Sect. 4, we summarize our approach and the results.

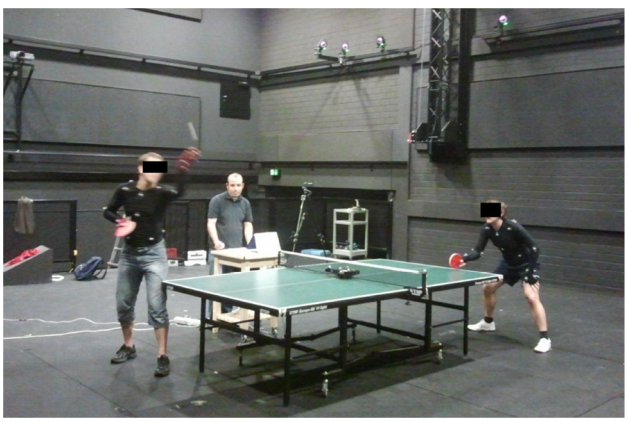


Fig. 1 Considered Scenario. Two people playing a competitive match of table tennis. The movements of the player and the ball were recorded with a VICON motion capture system and analyzed afterward

2 Modeling human strategies

As discussed in the introduction, we use model-free inverse reinforcement learning (IRL) to learn human strategies. Here, we will first introduce the notation and basic elements necessary for the table tennis model. Subsequently, we will discuss different model-free IRL approaches and show how the states, actions and reward features in the table tennis task can be represented.

2.1 Preliminaries

To employ IRL, the problem at hand needs to be modeled as a Markov decision problem (MDP). Formally, a MDP is a tuple $(S, A, \mathcal{T}, R, d_0, \gamma)$, where S is the state space, A is the action space, and $\mathcal{T}$ is a transition function

$$\mathcal{T}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) = \Pr(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t),$$

with states $\mathbf{s}_t, \mathbf{s}_{t+1} \in S$ and actions $\mathbf{a}_t \in A$. The function $R(\mathbf{s}, \mathbf{a})$ defines the reward for executing action $\mathbf{a}$ in state $\mathbf{s}$, the initial state distribution $d_0(\mathbf{s})$ models the start conditions, and the discount factor $\gamma \in [0, 1)$ determines the effective planning horizon.

A deterministic policy π is a mapping: $S \mapsto A$ and defines which action is chosen in a state $\mathbf{s} \in S$. A stochastic policy is a probability distribution over actions in a given state $\mathbf{s}$ and is defined as $\pi(\mathbf{s}|\mathbf{a}) = Pr(\mathbf{a}|\mathbf{s})$. The performance of a policy is measured with the so-called *value function* $V^\pi(\mathbf{s})$. The value function of a policy π evaluated at state $\mathbf{s}$ is given by

$$V^\pi(\mathbf{s}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(\mathbf{s}_t, \mathbf{a}_t) \middle| \pi, \mathcal{T}, \mathbf{s}_0 = \mathbf{s}\right],$$

and corresponds to the expected reward following policy π starting from state $\mathbf{s}$. The optimal value function is defined by $V^*(\mathbf{s}) = \max_\pi V^\pi(\mathbf{s}) \ \forall \mathbf{s} \in S$. The goal of an agent in a MDP is to find the optimal policy π^*, i.e., a policy that maximizes the value for every $\mathbf{s} \in S$.

We assume that the reward function R is given by a linear combination of m feature functions f_i with weights w_i. The reward function is therefore defined by

$$R(\mathbf{s}, \mathbf{a}) = \sum_{i=1}^{m} w_i f_i(\mathbf{s}, \mathbf{a}) = \mathbf{w}^{\mathrm{T}} \mathbf{f}(\mathbf{s}, \mathbf{a}),$$

where $\mathbf{w} \in \mathbb{R}^m$ and $\mathbf{f}(\mathbf{s}, \mathbf{a}) \in \mathbb{R}^m$. The features f_i are fixed, known, bounded basis functions mapping from $S \times A$ into $\mathbb{R}$. For a given trajectory $\tau = s_1 a_1, \ldots, s_T a_T$, the feature counts are given by $f_i^\tau = \sum_{t=1}^{H} \gamma^t f_i(\mathbf{s}_t, \mathbf{a}_t)$. Similarly to the value function, we can define the feature count f_i^π under policy π by

$$f_i^{\pi}(\mathbf{s}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t f_i(s_t, a_t) \middle| \pi, \mathcal{T}, \mathbf{s}_0 = \mathbf{s}\right]$$

as the expected features observed when following policy π. Since the reward function can be represented as a linear combination of features f_i, the expected return of policy π can be written as

$$V_{\mathbf{w}}^{\pi}(\mathbf{s}) = \sum_{i=1}^{m} w_i f_i^{\pi}(\mathbf{s}) = \mathbf{w}^{\mathrm{T}} \mathbf{f}^{\pi}(\mathbf{s}),$$

where $\mathbf{f}^{\pi} \in \mathbb{R}^m$ is a vector containing the single feature counts $f_i^{\pi}(\mathbf{s})$ as entries.

2.2 Learning the reward function

The reward function is a crucial part of the MDP as it defines the goal of the task and shapes the policy optimization process. Usually, it is assumed that the reward function is given. However, it is hard to specify the reward function for solving a complex task beforehand, and the learned behavior is sensitive to the provided reward function. This problem is especially evident when the task requires modeling the dynamics of human actions. The problem of designing the right reward function led to the development of IRL methods. Given the actions of an agent that is assumed to behave in an optimal manner, the available sensory information about the environment and, if possible, a model of the environment, the goal of IRL is to determine a reward function that can (mostly) justify the demonstrated behavior.

The IRL problem was originally formulated within the MDP framework by Ng and Russel (2000). Many researches provided further refinements in order to improve the original algorithms suggested by Ng and Russel (2000) and Abbeel and Ng (2004). For example, Ratliff et al. (2006) suggested a max-margin planning approach. Ziebart et al. (2008) suggested an algorithm where the principle of maximum entropy was exploited. Ramachandran and Amir (2007) modeled the uncertainties involved as probabilities where the demonstrations are treated as evidence of the unknown reward function. Rothkopf and Dimitrakakis (2011) extended this approach by suggesting a general Bayesian formulation. Levine et al. (2011) used GPs to model the reward as a nonlinear function of the features. A recent review of IRL algorithms can be found in (Zhifei and Joo 2012).

However, most IRL approaches rely on a given model of the environment $\mathcal{T}$ or assume that it can be accurately learned from the demonstrations. The reward function is found by first computing a policy that optimizes a reward function for an initial weight vector $\mathbf{w}$. Subsequently, the expected feature count of the new policy $\mathbf{f}^{\pi}$ can be computed. Based on this feature count, a new weight vector that separates the values of the expert feature $\mathbf{f}^{\pi_E}$ and the features of the current policy $\mathbf{f}^{\pi}$ can be computed. These steps are repeated until the weight vector converges. This general algorithm is displayed in Algorithm 1. Generally, a model of the dynamics is used to iteratively generate optimal trajectories (optimization step in Algorithm 1) under different reward functions until the generated trajectories match the ones provided by the expert.

Since modeling the dynamics of the table tennis task is highly challenging, we adopt in this paper a slightly different methodology. The policy optimization step in Algorithm 1 is performed by searching in a finite set of policies and retaining the policy with the highest average value. Each one of these policies is obtained by recording the state–action trajectories of a particular player. The skills of the players vary from novice to expert.

Only few model-free IRL methods have been suggested: Boularias et al. (2011) derived a relative entropy (RE) approach which, was evaluated on a ball-in-a-cup scenario. Mori et al. (2011) used least squares policy iteration and least squares temporal difference learning and applied their algorithm on human impedance control. We apply both RE-IRL and the method suggested by Abbeel and Ng (2004) to solve this problem and compare their performances. Boularias et al. (2011) already used the same sample-based technique described in the previous paragraph. We use the same methodology to obtain a model-free variant of Abbeel and Ng (2004).

We use both expert and nonoptimal data to compute the weight vector $\mathbf{w}^*$ that maximizes the differences between the nonexpert and the expert reward values. Here, we assume that the actions chosen by the expert are to be favored over those chosen by the less skilled players as they enable the player to win the game. The demonstrations given by the less skilled players under different playing conditions and goals provide arbitrary and suboptimal policies that stand in contrast to the policy demonstrated by the expert. To compute the reward weights, we tested three different methods, where the results can be found in Sect. 3.2. The first two evaluated methods that are based on the max-margin method of Abbeel and Ng (2004), while the third algorithm is the model-free IRL algorithm of Boularias et al. (2011). In the following sections, we assume that we are given a set of expert demonstrations

Algorithm 1 General IRL Algorithm

Input: $D^E = \{\tau\}_{p=1}^{P}$ expert demonstrations
Initialize: reward feature weights $\mathbf{w}^0$, $j = 1$
expert feature counts $\mathbf{f}^{\pi_E} = \frac{1}{P}\sum_{\tau \in D^E} \mathbf{f}^{\tau}$
repeat
  Optimize π_j based on $\mathbf{w}^{j-1}$
  Estimate f
  Update $\mathbf{w}^j$ such that $(\mathbf{w}^j)^T f^{\pi_j} < \mathbf{w}^j \mathbf{f}^{\pi_E}$
  $j \leftarrow j + 1$
until $\|\mathbf{w}^j - \mathbf{w}^{j-1}\|_2 < \varepsilon$

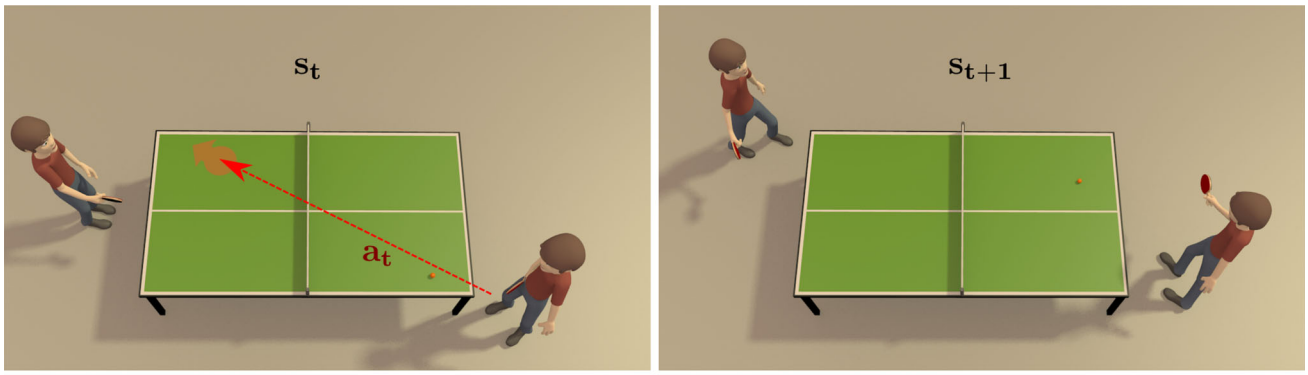


Fig. 2 Considered scenario: A table tennis player (agent) plays a game of table tennis. At time point t, he has to decide how to return the approaching ball to the opponents court such that the chance of winning the point will increase. Returning the ball to a specific goal on the opponent's court (with a specific orientation and velocity) corresponds to an action $\mathbf{a}_t$ executed by the agent. The player chooses this action based on his current state $\mathbf{s}_t$ (**a**). Due to this action, the system will transfer to the state $\mathbf{s}_{t+1}$ defining a new situation for the player (**b**)

$D^E = \{\tau_p\}_{p=1}^P$, where $\tau_p = \mathbf{s}_1^p \mathbf{a}_1^p, \ldots, \mathbf{s}_{T_p}^p \mathbf{a}_{T_p}^p$ corresponds to one rally (i.e., state–action trajectory), as well as a set of nonoptimal demonstrations $D^N = \{\tau_l\}_{l=1}^L$. Here, T_p is the number of volleys (i.e., state–action pairs) in the observed rally τ_p.

Please note that the following IRL methods are only discussed briefly to illustrate how the chosen IRL methods were applied in this (model-free) context. The reader is referred to the original literature as referenced in the following for a detailed description and analysis of the presented approaches.

2.2.1 Model-free max-margin for game values

The max-margin method of Abbeel and Ng (2004) aims at finding a policy π that has an expected return close to that of the expert, i.e., $\max_{\mathbf{w}} |V_{\mathbf{w}}^{\pi}(\mathbf{s}) - V_{\mathbf{w}}^{\pi_E}(\mathbf{s})| \leq \epsilon$, where $\|\mathbf{w}\|_2 \leq 1$. As the value is a linear function of the reward, it suffices to find an optimal policy π that has feature counts close to the ones of the expert's trajectories, i.e., $\|\mathbf{f}^{\pi} - \mathbf{f}^{\pi_E}\|_2 \leq \epsilon$. The policy π needs to be chosen from the set of previously recorded nonoptimal policies due to the lack of a model for generating policies. We use the projection algorithm of Abbeel and Ng (2004) to solve the following optimization problem

$$\max_{\xi,\mathbf{w}} \xi \qquad s.t. \;\; \mathbf{w}^T \mathbf{f}^{\pi_E} \geq \mathbf{w}^T \mathbf{f}^{\pi_j} + \xi, \qquad \|\mathbf{w}\| \leq 2,$$

where ξ is the difference of the value of the expert and the value of the nonexpert, and π_j are the policies of nonexpert players. $\mathbf{f}^{\pi_j}$ therefore corresponds to the average feature count for all rallies demonstrated by a player in one game. The corresponding algorithm is displayed in Algorithm 2. In the following, we will refer to this algorithm as max-margin for game values (MMG).

Algorithm 2 Max-Margin for Game Values

Input: $D^E = \{\tau\}_{p=1}^P$ expert demonstrations
$D^N = \{\tau\}_{l=1}^L$ nonoptimal demonstrations
Initialize: $\mathbf{f}^{\pi_E} = \frac{1}{P} \sum_{\tau \in D^E} \mathbf{f}^{\tau}$
$\mathbf{f}^{\pi_i} = \frac{1}{L} \sum_{\tau \in D^{N_i}} \mathbf{f}^{\tau}$ with $D^{N_i} \subset D^N$
$\mathbf{w}^0 = \mathbf{0}$, $j = 1$
repeat
$i = \arg\min_i (\mathbf{w}^{j-1})^T (\mathbf{f}^{\pi_E} - \mathbf{f}^{\pi_i})$
$\mathbf{f}^{j-1} = \mathbf{f}^{\pi_i}$
Compute $\bar{\mathbf{f}}^{j-1}$, the projection of $\mathbf{f}^{\pi_E}$ on $(\bar{\mathbf{f}}^{j-2}, \mathbf{f}^{j-1})$
$\mathbf{w}^j = \mathbf{f}^{\pi_E} - \bar{\mathbf{f}}^{j-1}$
$\Delta f = \|\mathbf{f}^{\pi_E} - \bar{\mathbf{f}}^{j-1}\|_2$
$j \leftarrow j + 1$
until $\Delta f < \varepsilon$

2.2.2 Model-free max-margin of states values

Using the max-margin method of Abbeel and Ng (2004) in a model-free setup as described above has one drawback. We assume that the initial state of the rally largely defines all following state–actions pairs. However, in table tennis, it is unlikely that any player plans the strokes for more than only a few steps ahead. Computing the value function based on only a few state–action pairs after the initial serve would cause the agent to lose important information that led to winning or losing the rally. To avoid this information loss, we need to compare the values of the expert in every state in the recorded trajectories to the ones of the nonexperts in the same state. As the states are continuous, it is unlikely that exactly the same state is encountered in both the expert and nonexpert trajectories. Nevertheless, we can find the weight vector $\mathbf{w}$ by solving the quadratic optimization problem

$$\max_{\mathbf{w}} \sum_{p=1}^{P} \sum_{t=0}^{T_p} \left(V_{\mathbf{w}}^{\pi_E}(\mathbf{s}_t^p) - \hat{V}_{\mathbf{w}}^{\pi_N}(\mathbf{s}_t^p) \right) - \lambda ||\mathbf{w}||_2, \tag{1}$$

Algorithm 3 Max-Margin of States

Input: $D^E = \{\tau_p\}_{p=1}^P$ expert demonstrations
$D^N = \{\tau_l\}_{l=1}^L$ nonoptimal demonstrations
Initialize: $n = 1$
for all $p \in D^E$ **do**
  for all $\mathbf{s}_t^p \in \tau_p$ **do**
    $[\mathbf{F}^{\pi_E}]_{n:} = \sum_{i=t}^{H_t^p} \mathbf{f}(\mathbf{s}_t^p, \mathbf{a}_t^p)$
    Compute k-nearest neighbors $\mathcal{N}_k(\mathbf{s}_t^p)$
    $[\mathbf{F}^{\pi_N}]_{n:} = \frac{1}{k} \sum_{\mathbf{s}_t^l \in \mathcal{N}_k(\mathbf{s}_t^p)} \sum_{i=t}^{H_t^l} \mathbf{f}(\mathbf{s}_i^l, \mathbf{a}_i^l)$
    $n \leftarrow n + 1$
  end for
end for
$\mathbf{w} = \arg\max_{\mathbf{w}} \mathbf{w}(\mathbf{F}^{\pi_E} - \mathbf{F}^{\pi_N}) - \lambda||\mathbf{w}||_2$

where $\hat{V}_{\mathbf{w}}^{\pi_N}(\mathbf{s}_t^p)$ is an estimated value of the nonexpert players in the current state $\mathbf{s}_t^p$ of the expert. Estimating the value $\hat{V}^{\pi_N}$ in a given state $\mathbf{s}$ is a regression problem that we propose to solve by using the k-nearest neighbors method,

$$\hat{V}_{\mathbf{w}}^{\pi_N}(\mathbf{s}) = \frac{1}{k} \sum_{\mathbf{s}' \in \mathcal{N}_k(\mathbf{s})} V_{\mathbf{w}}^{\pi_N}(\mathbf{s}'),$$

where $\mathcal{N}_k(\mathbf{s})$ is the set of k-nearest neighbors of $\mathbf{s}$ among all the states that have been observed in trajectories of the nonexpert players.[2] The metric used to find the k-nearest neighbors is a Gaussian kernel $K(\mathbf{s}, \mathbf{s}') = \exp(-(\mathbf{s}-\mathbf{s}')^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{s}-\mathbf{s}')^{\mathrm{T}})$ that defines the similarity measure between states. The diagonal matrix $\boldsymbol{\Sigma}$ contains the measured standard deviation of the data. Note that one can also use other nonparametric methods, such as kernel regression.

The value functions V^{π_E} and V^{π_N} of the expert's policy π_E and nonexperts policies π_N are computed as

$$V_{\mathbf{w}}^{\pi}(\mathbf{s}_t^p) = \frac{1}{H_t^p - t + 1} \sum_{i=t}^{H_t^p} \mathbf{w}^{\mathrm{T}} \mathbf{f}^{\pi}(\mathbf{s}_i^p, \mathbf{a}_i^p),$$

where $H_t^p = \min\{t + H - 1, T_p\}$ and H is the planning horizon, i.e., the number of steps we look into the future. The corresponding algorithm is displayed in Algorithm 3. In the following, we will refer to this algorithm as max-margin of state values (MMS).

2.2.3 Relative entropy method

The relative entropy IRL method (Boularias et al. 2011) finds a distribution $\mathcal{P}$ over trajectories that minimizes the KL-divergence to a reference distribution Q, while ensuring that the feature counts under $\mathcal{P}$ are similar to the feature counts in the expert trajectories. The reference distribution Q encodes prior preferences and constraints of the learned behavior, which makes this method well suited for transferring the expert's policy to a robot. The solution to this problem takes the following form

$$\mathcal{P}(\tau|\mathbf{w}) = \frac{1}{Z(\mathbf{w})} Q(\tau) \exp\left(\mathbf{w}^T f_i^{\tau}\right),$$

where $Z(\mathbf{w}) = \sum_{\tau} Q(\tau) \exp\left(\mathbf{w}^T f_i^{\tau}\right)$. The reward weight vector $\mathbf{w}$ is found by solving the optimization problem

$$\max_{\mathbf{w}} \mathbf{w}^T \mathbf{f}^{\pi_E} - \ln Z(\mathbf{w}) - \lambda \|\mathbf{w}\|_1. \quad (2)$$

The gradient of this objective function is calculated by reusing the expert and nonexpert trajectories with importance sampling. For our experiments, we choose the reference distribution Q to be uniform, as we are mainly interested in extracting the most informative reward function and not in transferring the expert's policy. The corresponding algorithm is displayed in Algorithm 4. In the following, we will refer to this algorithm as RE.

Algorithm 4 Relative Entropy IRL Algorithm

Input: $D^E = \{\tau_p\}_{p=1}^P$ expert demonstration
$D^N = \{\tau_l\}_{l=1}^L$ nonoptimal demonstration
Initialize: $\mathbf{f}^{\pi_E} = \frac{1}{P} \sum_{\tau \in D^E} \mathbf{f}^{\tau}$
$\mathbf{w}^0 = \mathbf{0}$, $j = 1$
repeat
  Compute $\mathcal{P}(\tau|\mathbf{w}^{j-1}) = \frac{Q(\tau) \exp(\sum_{i=1}^m w_i^{j-1} f_i)}{\sum_{\tau \in D^N} Q(\tau) \exp(\sum_{i=1}^m w_i^{j-1} f_i)}$
  for all $\tau \in D^N$
  for all features f_i **do**
    $\frac{\partial}{\partial w_i} g(\mathbf{w}) = f_i^{\pi_E} - \sum_{\tau \in D^N} \mathcal{P}(\tau|\mathbf{w}^{j-1}) f_i(\tau) - \alpha_i \lambda_i$
    $w_i^j = w_i^{j-1} + \frac{\partial}{\partial w_i} g(\mathbf{w})$
  end for
  $\Delta w = \|\mathbf{w}^{j-1} - \mathbf{w}^j\|_2$
  $j \leftarrow j + 1$
until $\Delta w < \varepsilon$

2.3 Computational model for representing strategies in table tennis

In the previous sections, we have given a general description of how the decision processes in table tennis can be modeled as a MDP. We also showed several approaches for obtaining the reward function from the table tennis player's demonstrations. As a next step, we now need to specify the states, actions and reward features of the table tennis task.

2.3.1 States

Ideally, the state of the system would contain all information experienced by the agent. However, such an approach is not feasible for two reasons: First, we do not have access to all information. For example, we do not know what kind of assumptions the player makes about the opponent's strategy

[2] Please note that the performance of k-NN regression depends on the density of the data. In the table tennis context, most of the data were adequately concentrated in a small region.

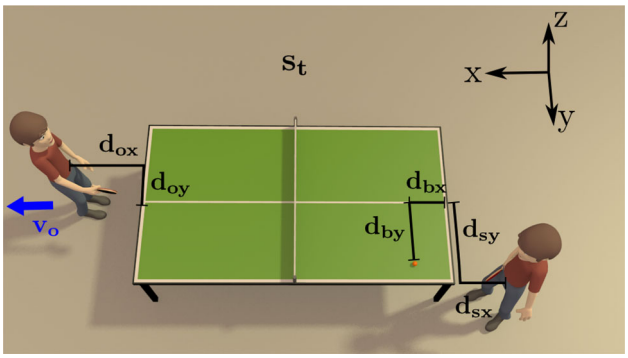


Fig. 3 The state of the system is defined by the relative position of the agent (d_{sx}, d_{sy}) and the relative position (d_{ox}, d_{oy}) and velocity ($\mathbf{v}_o$) of the opponent toward the table, as well as the position (d_{bx}, d_{by}) and velocity ($\mathbf{v}_b$) of the ball when bouncing on the table

or the spin of the ball. Modeling such hidden and uncertain information in the state space leads to the formulation of partial observable MDPs (PoMDPs, (Monahan 1982)). Second, modeling such high-dimensional continuous state domains in the context of PoMDPs requires a large data set and is likely to be intractable. Hence, we approximate the problem by assuming perfect knowledge about the environment and remove redundant and irrelevant information. We assume that the player has to decide where and how to hit the ball when the hitting movement is initiated and that the decision depends on the following information: the planar Cartesian position of the agent $\mathbf{d}_s = [d_{sx}, d_{sy}]$, the opponent's position $\mathbf{d}_o = [d_{ox}, d_{oy}]$ and velocity $\mathbf{v}_o$, the state of the rally $\mathbf{g} \in \{$player serve, opponent serve, not served$\}$, the elbow position of the opponent $\mathbf{e}_o = [e_{o_x}, e_{o_y}]$ as well as the ball position $\mathbf{d}_b = [d_{bx}, d_{by}]$, velocity $|\mathbf{v}_b|$ and direction given by the angles θ_{py} and θ_{pz} (see Fig. 3).

Thus, the state can be represented by the parameters $\mathbf{s}_i = [\mathbf{d}_b, |\mathbf{v}_b|, \theta_{\text{py}}, \theta_{\text{pz}}, \mathbf{d}_s, \mathbf{d}_o, \mathbf{e}_o, \mathbf{v}_o, \mathbf{g}]$. The variables θ_{py} and θ_{pz} are defined as the horizontal and vertical bouncing angles of the ball at the moment of impact on the player's side of the table, respectively. θ_{pz} defines the bouncing angle in the xz-plane and therefore corresponds to how flat the ball was played. θ_{py} defines the bouncing angle in the xy-plane (see Fig. 5). Playing the ball diagonal to the backhand area of the opponent results in a smaller negative angle for θ_{py}, while playing the ball diagonal to the forehand area results in an increased angle. Playing the ball straight corresponds to an angle of zero. Additionally, we define a set of terminal states $s_T \in \{W, L\}$. A rally will end when either the subject won the rally ($s_T = W$), or the subject lost the rally ($s_T = L$).

2.3.2 Actions

To perform a hitting movement, the system needs the following information: (i) where and when to hit the ball, (ii) the velocity of the racket and (iii) the orientation of the racket at impact. While the first may directly result from the current state of the system, the second and third points are determined by where and how the player decides to return the ball to the opponent's court. This decision includes the desired bouncing point $\mathbf{p}_b$ of the ball on the opponent's court, the corresponding bouncing angles θ_{oy} and θ_{oz}, the overall velocity of the ball $||\mathbf{v}_b||$ and the spin of the ball. Here, the desired bouncing point refers to the bouncing point on the opponent's court desired by the player. Since the different kinds of spin are hard to capture without an expert classifying the sampled data, we discard the spin and use only basic strategic elements. Therefore, an action can be defined as $\mathbf{a} = [\mathbf{p}_b, ||\mathbf{v}_b||, \theta_{\text{oy}}, \theta_{\text{oz}}]$. We do not distinguish between serves and nonserves for the actions, as the first bounce of the serve will be fully described by the second bounce.

2.3.3 Reward features

In order to estimate the desired unknown reward function, we assume that the reward function is given by a linear combination of observable reward features. Usually, those reward features are chosen manually by the experimenter. An automatic approach for choosing these reward features was suggested by Levine et al. (2010). Here, it was suggested to construct the features from a logical combinations of components that are the most relevant to the task. Nevertheless, this approach also requires the definition of the most relevant components of the state space beforehand. Even if it would be possible to consider the whole state space as components, some features might be the result of a nontrivial combination of these elements. Other feature combinations might be redundant and could dominate the behavior due to their multiple occurrences. Therefore, we choose the features manually taking into account the logical combination of state components that seemed most relevant for the task.

We choose the features as a combination of the state information of the ball and the position of the opponent. In order to be able to distinguish whatever the relevant features depend on the opponent or not, we choose features that depend only on the state information of the ball but are independent of the opponent and features that depend on the state information of the ball and the opponent. In the following, we list the chosen reward features $f_i(\mathbf{s}, \mathbf{a})$.

Position on the table This feature corresponds to the bouncing point of the ball in the opponent's court. Players do not usually target a particular point on the table but rather a small region. Therefore, we discretize the court into nine regions (see Fig. 4). Each region i is identified by its center $\mathbf{c_i}$. We use as features the relative distances between the observed bouncing point $\mathbf{p}_b$ of the ball on the opponent's court and each center $\mathbf{c_i}$, given by

$$p_{c_i} = \frac{\exp(-\|\mathbf{p}_b - \mathbf{c}_i\|_2)}{\sum_j \exp(-\|\mathbf{p}_b - \mathbf{c}_j\|_2)}.$$

This computation is based on the euclidean distance between $\mathbf{p}_b$ and the cell center $\mathbf{c}_i$. $\mathbf{p}_b$ corresponds here to chosen action of the player.

Bouncing angles We computed two bouncing angles θ_{oz} and θ_{oy} which define the direction of the ball when bouncing on the opponent's side of the court (see Fig. 5). This feature allows us to tell whether the ball was played rather cross or straight, or if there where any preferences in how flat the ball was played.

Distance to the edges of the table We provided two features defining the proximity of the bouncing point p_b to the edge of the table $\mathbf{e}_t$. One for the x-direction $\delta_{tx} = \exp(-1.5|e_{t_x} - p_{b_x}|)$ and one for the y-direction $\delta_{ty} = \exp(-1.5|e_{t_y} - p_{b_y}|)$. These features were chosen in order to see whether the expert plays in general closer to the edges than the naive player.

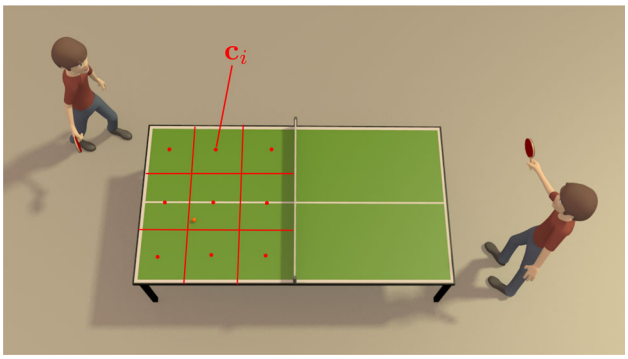


Fig. 4 In order to compute the table preferences on the opponent's court, the table was divided into nine cells. Each cell was assigned a center (*red points*) $\mathbf{c}_i$ (color figure online)

Velocity of the ball The velocity of the ball $\|\mathbf{v}_b\|$ in meters per second was used as another feature.

Smash One of the features defined whether the ball was a smash. When the ball velocity was higher than 10 m/s, this feature was set to one, otherwise this feature was set to zero. The velocity of 10 m/s was defined empirically.

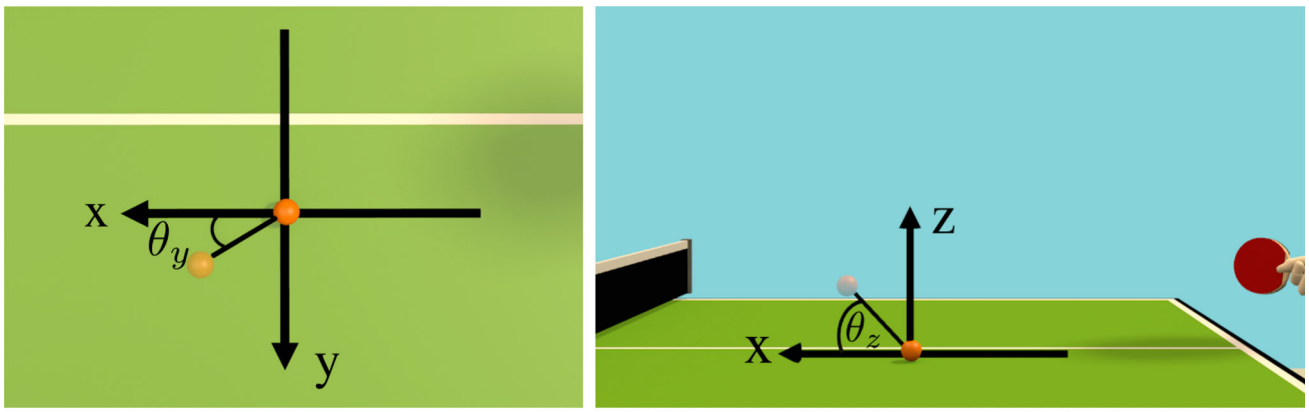
Distance to the opponent Two features define the distance of the bouncing point of the ball on the opponent's court and the right hand of the opponent. One of the features is defined by the distance in x-direction $\delta_{ox} = |p_{o_x} - p_{b_x}|$, while the second is defined by the distance in y-direction $\delta_{oy} = |p_{o_y} - p_{b_y}|$. This feature allows to evaluate whether the skilled player chose the bouncing point such that the distance between the player and the ball is maximized or not.

Elbow One feature is the closeness of the ball to the elbow, and therefore, it measures if the ball was played to the elbow of the opponent $\mathbf{e}_o$. It is defined by $\delta_{elbow} = \exp(-|e_{o_y} - p_{b_y} + \tan(\theta_y)(e_{o_x} - p_{b_x}|))$, where $\tan(\theta_y)(e_{o_x} - p_{b_x})$ is an extrapolation of the ball position. This feature also provides a measurement of how close the ball bounces relative to the opponent. Playing the ball close to the opponent makes it harder for the opponent to return the ball.

Movement direction of the opponent One feature was derived in order to define the velocity of the opponent and the ball in y-direction. It is defined by $v_o = (p_{o_y} - p_{b_y})v_{o_y},$. This feature indicates whether the ball was played in the opposite moving direction of the opponent.

Winning and loosing One binary feature was used to assign a reward to the terminal states (i.e., winning and loosing). For all nonterminal states, this feature was set to zero. For the terminal states, a value of one was assigned to the feature for $s_T = W$ and a value of -1 for $s_T = L$.

All features are scaled to lie in an interval of [0 1], except for the direction sensitive features θ_{oy} and v_o, which lie in



(a) θ_y **(b)** θ_z

Fig. 5 The bouncing angles θ_y and θ_z in the xy- and xz-surface define the orientation of the ball. While θ_z corresponds to the horizontal bouncing angle, θ_y corresponds to the direction of the ball and thereby defines if the ball is played cross to the *left*, cross to the *right* or straight

an interval of [−1 1]. Some of the features reflect aspects of other features. For example, the position of the bouncing point on the table can reflect a preference of a bouncing angle. The position on the table might depend on the position of the opponent or opponent specific weakness. Nevertheless, we choose these feature since each of them seemed to be likely to be a strategic component and as they allow us to analyze the influences of the state components individually.

3 Experiments and evaluations

To validate the suitability of using IRL algorithms in order to extract basic strategic elements, we recorded table tennis players with various skill levels. The subjects played under three different conditions. These data were used to compute the reward feature weights and to validate the potential reward functions.

In the following, we will first describe the experiment and the data processing procedure. Subsequently, we will present the results.

3.1 Experimental setup and data collection

The purpose of the experiment was to investigate basic strategic elements in table tennis (excluding all types of spin which are difficult to capture), using IRL techniques. Therefore, a data set with expert demonstrations and a data set with different suboptimal policies were collected. In this study, there were both participants serving as subjects who rarely played table tennis, as well as subjects who played on a regular basis in a table tennis club.

3.1.1 Participants

Eight healthy right-handed subjects of all genders (seven males and one female) participated in this study. The mean age of the participants was 26.25 years (standard deviation (SD) 3.38 years). All subjects had normal or corrected-to-normal eye sight. All participants gave their consent prior to the experiment and completed a form about their playing skills according to which they were grouped in one of two classes: (1) naive players and (2) skilled players.

The group of naive players consisted of five subjects (four males and one female) with a mean age of 28.4 years (SD 1.14 years). The subjects were recruited from the Max Planck Campus in Tübingen and the University of Tübingen. All naive players fulfilled the following requirements: (i) never played in a table tennis club, (ii) did not train on a regular basis (weekly or daily) in the last five years, (iii) did not participate in table tennis tournaments and (iv) did not play any other racket sports on a regular basis. The group of skilled players consisted of three subjects (all male) with a mean age of 22.67 years (SD 2.08 years). The subjects were recruited from a local table tennis club and fulfilled the following requirements: (i) played for at least eight years in a table tennis club, (ii) trained on a weekly basis (at least twice a week) and (iii) participated regularly in table tennis competitions.

One of the skilled players was used as a permanent fixed *opponent* and, therefore, was not considered part of the subject set. Furthermore, only one of the skilled subjects was used for the expert demonstrations since the other skilled player was not able to win against the opponent. All other subjects were used as nonoptimal demonstrations. Due to the fact that the nonoptimal data set also contains a skilled player, we have the possibility to test the approach not only to detect the differences between naive and skilled players, but also between skilled players which have the same level of training.

3.1.2 Apparatus

In order to collect information about the position of the participants, the table and the ball during the game, we used a VICON motion capture system (VICON MX-13 with the VICON IQ 2.5 Software, 16 cameras, 120 frames per second). Therefore, 25 VICON infrared reflecting markers were attached to the hands, wrists, elbows, shoulders, hips and the back and front of the participants. With this setup and a 3D kinematic model of the upper body of each individual, we could capture their whole body movement during the game. To identify the table and the net, we placed four markers at each corner of the table and one marker on one of the edges of the net. A standard table tennis table (length 2.74 m, width 1.53 m and height 0.76 m) and rackets conform with the rules of the International Table Tennis Federation (2011) were used. The surfaces of the rackets were chosen such that they did not allow for spin on both sides. The table tennis ball was covered with a gray green infrared reflecting powder in order to detect it with the VICON system. As a result, the ball had an additional weight of 2 g. This coating slightly changed its physical properties (e.g., it additionally reduced the spin during the game). Additionally, the subjects were recorded with two video cameras. The experimental setup is also shown in Fig. 6.

3.1.3 Procedure

The participants were asked to play a game of table tennis under three different conditions.

Condition 1. The subject played a cooperative game of table tennis. The goal for the subjects is to maximize the number of returns in a rally for a ten minute period.

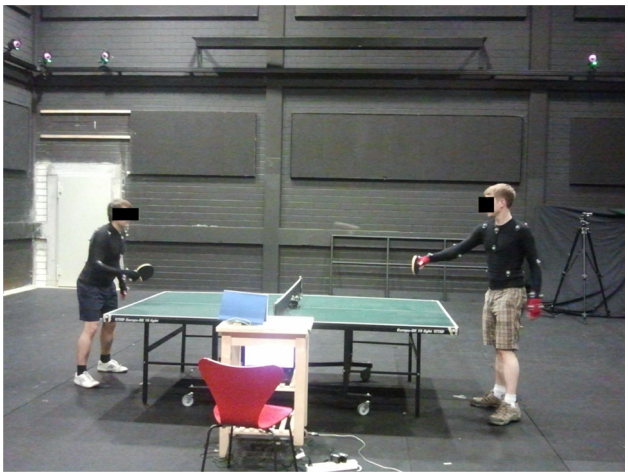


Fig. 6 Experimental setup. A naive player (*right side*) plays against an expert opponent (*left side*). The *upper* body of both players and the ball are tracked by a motion capture system

Condition 2. The subject was told to perform a competitive game of table tennis, while the opponent was instructed to return the ball "nicely" (i.e., the opponent was instructed to play toward the subject when possible in a cooperative way).

Condition 3. Both the subject and the opponent were instructed to play a competitive game of table tennis.

Each of the seven subjects played against the opponent one game under each of the three conditions. The participants were required to play table tennis according to the standard table tennis rules defined by the International Table Tennis Federation (2011) with the following exceptions: (i) The players did not switch sides after a game, (ii) the expedite system[3] did not apply during the game and (iii) the first serve of the match was always executed by the subject (never by the opponent). A game consisted of the best of five matches, i.e., the game was won by the player who first won three matches. Before the experiment started, the subjects played a friendly game with the opponent for 10 minutes in order to get used to the slightly altered bouncing properties of the table tennis ball (due to the coating with reflective powder). Each subject was required to read the rules before the experiment. The current score of the game in Conditions 2 and 3 were displayed on a scoreboard visible for both of the two players. In each game, a referee ensured that the game was conducted in accordance with the rules. The score was protocolled by two of the experimenters independently and reconciled afterward.

[3] Expedite system: additional rules to discourage slow play in a table tennis match. It is used after 10 minutes of play or if requested by both players.

3.1.4 Data processing

The captured motion was post-processed using the VICON IQ 2.5 software. The marker labels were automatically assigned to each marker using the VICON IQ 2.5 trajectory labeler. Errors that occurred during this automatic labeling process were manually corrected afterward. The ball had to be labeled manually as it was tracked similar to a single VICON marker. The VICON IQ 2.5 kinematic fitting function computed the 3D kinematic information of the subjects automatically. Bouncing and hitting events for all data were then automatically labeled during another MATLAB post-processing step and manually reassigned if necessary. For each point, the score was automatically computed based on this information and reconciled with the score information recorded by the experimenters. Finally, for each time where the ball was hit by the subject, the corresponding state and reward features were extracted and saved in a MATLAB file.

3.2 Results and discussion

Only one of the subjects was able to win against the opponent in the competitive game under Condition 3. All other games were won by the skilled opponent. The scoring results of the subjects that lost the game can be found in Table 1. The skilled player who won the game in Condition 3 was able to win 41 out of 75 rallies. Based on these results, the data were divided into two subsets: (1) a nonexpert data set and (2) an expert data set. The nonexpert data set included all games of the subjects who lost against the fixed opponent, i.e., all naive subjects and one of the skilled players, as well as all cooperative games. We will refer to the players that lost as Naive 1–5 and Skilled 1. The expert data set consisted of all rallies in the competitive game (Condition 3) played by the skilled player that won against the opponent. We will refer to this player as Expert. When asked which player performed worst, the opponent stated that Naive 3 was the worst.

We tested all three IRL methods as described in Sect. 2.2. To evaluate the potential reward functions, we performed a leave-one-subject-out testing scheme. We computed the reward feature weights for each of the three methods seven times. Every time leaving out all rallies (i.e., state–action trajectories) of one of the subjects that lost or the rallies of the cooperative game of the expert respectively. We also excluded 20 rallies of the expert for the validations. To this spared data of the expert and the naive players, we refer to as spared test data. The obtained reward functions were tested for the different skill levels of the subjects using the excluded rallies demonstrated in the game under Condition 3 only and the different styles using the cooperative game of the expert.

All resulting reward functions yielded the highest rewards for the feature of the terminal state for losing or winning the rally. Winning the rally was therefore highly desirable for the

agent while losing should be avoided. For the evaluations, we did not consider this feature in order to see how well we can distinguish the subjects based on the other strategic elements.

Analyzing the scores yielded by the subjects in Condition 2 and Condition 3, one can see that the scores yielded by the naive players are higher in Condition 3 than in Condition 2. This might seem contradicting on a first glance. While the opponent was playing always nicely back toward the subject in Condition 2, there was a lower chance of making a fault. In Condition 3, however, the opponent played the ball such that there is a higher chance that the subject is not able to return the ball. By doing so, he also takes a higher risk of making a fault. It seems reasonable to assume that a player takes a higher risk when he has a reasonable advance or is quite certain that he can beat his opponent. This assumption seems to be reflected in the data, where it can be observed that the opponent loses more points in Condition 3 when his opponent was not as good (as reflected in Condition 2).

Statistical significance values can be computed by repeating the game of each player several times. However, it is anticipated that the behavior of the individual players will change overtime due to his increased experience and knowledge of the opponent. Consequently, also their expected feature counts will change overtime. Significance tests might not be able to capture such time varying behaviors of contestants during an extended match.

Due to the complex and multidimensional nature of the task, the feature scores within a game usually have a large variance. For this reason, we reported only the average reward for each player. From the results reported in Table 1, it can be concluded that the predicted performance (average reward) of each player is correlated with the observed performance (actual score).

In the following, we will first present the overall results of the three methods showing that we were able to distinguish between different playing skills and styles. Subsequently, we will discuss the influence of the horizon for the MMS algorithm. Finally, we discuss the results for all features separately.

3.2.1 Classifying the skill levels of the players

We computed the differences in the average reward for a state–action pair of the spared expert and nonexpert data for the reward functions obtained from the three methods described in Sect. 2.2 abbreviated as before as MMG, MMS and RE. The results in terms of the differences in the average reward between expert and nonexpert are displayed in Table 1. All three reward functions were able to distinguish between the nonexpert games and the expert game, as well as between the different playing styles of the expert (competitive vs cooperative). In general, the average reward for each player reflected the skill level of the players with the exception of Naive 2. For all naive players except Naive 2, the differences were high, while the difference between Skilled 1 and the expert was moderate. These differences were more distinctive for the MMS algorithm.

The player Naive 2 yielded similar scores as the expert and the player Skilled 1 with respect to the analyzed features (see Table 1; Fig. 8). Although the subject did not yield as many points as player Skilled 1, he did achieve a better feature score. There are two possible explanations for this result. First, it can be argued that the subject did use a similar strategy as the expert, but suffered from an inaccurate movement execution due to his lack of practice. As a consequence, he made many mistakes as playing the ball into the net or missing the court. Second, it is possible that we are missing features that would distinguish the naive and the expert. However, Naive 2 was the best of the naive players and came close to the score observed for the skilled player. Given the high scores in Condition 2 and 3 (compared to Skilled 1), it seems reasonable to assume that player Naive 2 chooses his actions based on the same principles as the expert in a game without spin. In comparison, Skilled 1 has a very good movement execution due to his long training and experience. However, he was not able to win against the opponent, although this player had the most experience in terms of years. This suggests that Skilled 1 was a very good player in terms of playing the ball successfully back to the opponent, but was not efficient in choosing his actions without the strategic element of spin.

Table 1 Summary of the results of the evaluations for the different methods

	Method	Naive 1	Naive 2	Naive 3	Naive 4	Naive 5	Skilled 1	Cooperative
Average reward difference	MMG	1.01	0.28	0.90	1.16	0.69	0.49	0.55
with respect to the expert	MMS	1.16	0.07	1.24	0.86	0.71	0.33	0.50
	RE	0.70	0.11	0.60	0.80	0.42	0.31	0.55
Scores in Condition 2		5:33	12:33	2:33	5:33	2:33	21:34	
Scores in Condition 3		13:33	17:33	10:33	5:33	17:33	20:33	

The differences in the average rewards with respect to the expert define the differences between the reward of the expert and the spared test subject of the nonexpert data set. The feature of winning and loosing the rally was not included. MMG corresponds to the model-free max-margin of game values, MMS corresponds to the model-free max-margin of states values with an horizon of three, and RE corresponds to the relative entropy method (see Sect. 2.2)

The close feature scores of subject Naive 2 the expert also show that all tested algorithms are able to deal with nonoptimal data containing strategies similar to the one of the expert.

3.2.2 Comparison of the tested IRL methods

All three reward functions obtained in the evaluation show a very small difference in the average reward of the expert and Naive 2, followed by Skilled 1 and Naive 5. Furthermore, all three methods showed relatively large differences between the expert and the players Naive 1, Naive 3 and Naive 4. However, they disagree in the ranking of these three players. While the reward function obtained by the MMG and RE algorithm shows the highest difference for the expert and Naive 4, the reward function obtained by the MMS algorithm yields the highest difference between the expert and Naive 3. Naive 4 being the worst player is in compliance with the scoring results of Experiment 3, while Naive 3 being the worst player is in compliance with the statement of the permanent opponent.

3.2.3 Influence of the planning horizon

For the max-margin of the state values algorithm given by the MMS algorithm, we evaluated the setup with three different horizons. We chose the horizons of $H = 1, H = 2$ and $H = 3$. The horizon of one only considers one state–action pair. The horizon of two also considers the state–action pair presented directly after the current one. A horizon of three means that we consider up to two state–action pairs following the current one.

The results of the average reward differences of the nonoptimal policies and the expert for the whole game and the states directly before the terminal are displayed in Table 2. In general, the average reward difference was reduced slightly with increasing horizon, while the average reward difference for the last $H - 1$ states before the terminal state increases with growing planning horizon, reaching its maximum with a horizon of three. Horizons larger than three did not improve the differences in the reward.

3.2.4 Individual reward features

Analyzing the reward weights individually, the different methods showed similar weights for the most important features (i.e., the features with the highest weights and highest resulting reward differences). The largest influence resulted from the bouncing angles θ_y and θ_z, the table preferences and the distance between the desired bouncing point and the racket of the opponent. For simplicity, we will only discuss the parameter values for the individual features of the reward functions obtained by the MMS and RE algorithm (MMG had the worst performance in terms of individual feature classification).

The reward weights for the individual features are displayed in Fig. 7a, b. We also showed the average reward differences for the spared test data sets for each feature individually in Fig. 7b and for the different time steps in Fig. 7c. The individual differences of each player are displayed in Fig. 7d. Figure 8 shows the various characteristics of the features for each subjects individually. We will discuss all features in the next sections.

A paired t-test was performed on the average rewards of the expert and the nonexert subject for each feature (Fig. 8). The results are reported below.

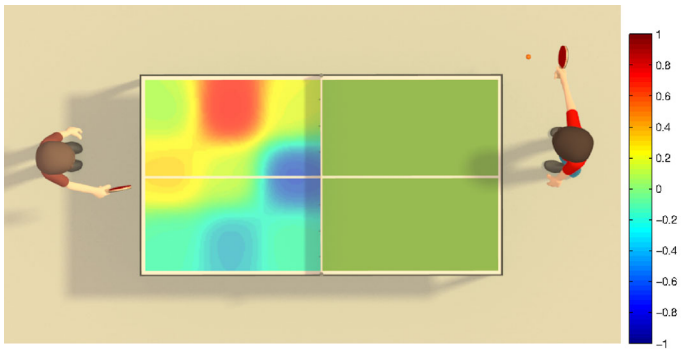
3.2.5 Goal preferences on the table

The preferences of the locations on the table are independent from the state information of the opponent, but they do reflect parts of the strategy that will also be covered by other features. The resulting reward functions of the different algorithms showed a preference for the areas where the opponent would have to return the ball using the backhand, while the areas that are suited for returning the ball with the forehand and the areas directly after the net are often rather avoided (see Fig. 7a). The differences in the average reward for the goal preferences on the table were signifi-

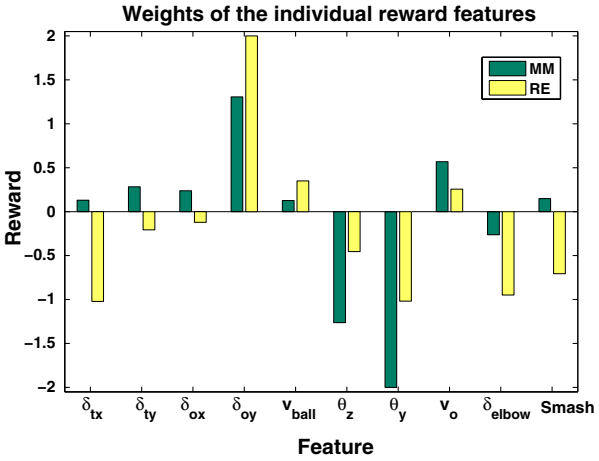
Table 2 Summary of the results for the different horizons with Algorithm 3

	horizon	Naive 1	Naive 2	Naive 3	Naive 4	Naive 5	Skilled 1	Cooperative
Average reward difference	1	1.30	0.04	1.17	0.91	0.74	0.30	0.43
with respect to the expert	2	1.20	0.07	1.22	0.87	0.72	0.33	0.47
	3	1.16	0.07	1.24	0.86	0.71	0.33	0.50
Average reward differences	2	0.91	−0.21	0.92	0.57	0.38	−0.12	0.23
directly before terminal state	3	1.12	0.04	1.23	0.89	0.76	0.24	0.53

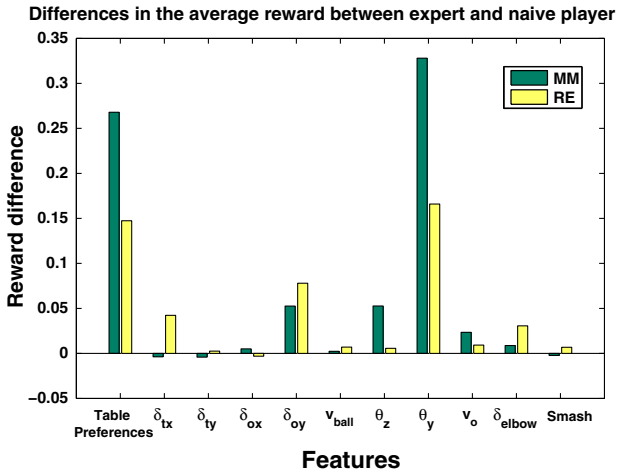
The differences in the average reward with respect to the expert trained with the different horizons. The differences in the average reward directly before the terminal define the differences of the reward of the expert and the spared test subject for the state before the terminal or the average reward of the two states before the terminal for the horizons 2 and 3, respectively



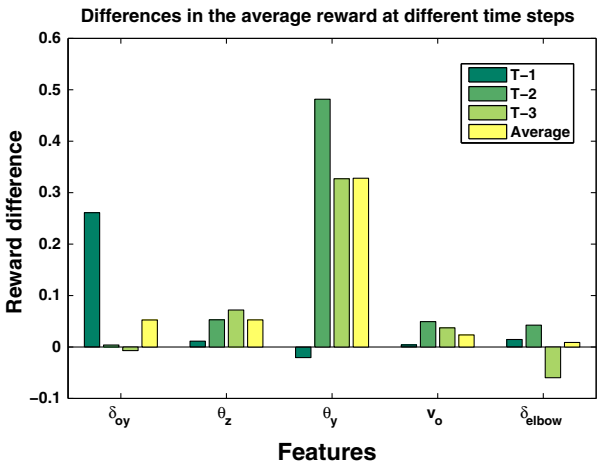
(a) Reward function for table preferences



(b) Reward feature weights



(c) Average reward differences



(d) Reward differences features at different time steps

Fig. 7 Resulting parameter values for the individual features. **a** The resulting reward function of the table preferences for Algorithm 3 (MM). **b** The weights of all other features for Algorithm 3 (MM) and Algorithm 4 (RE), respectively. **c** The differences of the average reward of the expert and the naive player for each feature separately using the reward function of the max-margin algorithm (*green*) and the relative entropy algorithm (*yellow*). **d** The differences of the average rewards for the most important features at different time steps before the terminal state (win or loss) for the reward function yield with the max-margin algorithm. **a** Reward function for table preferences. **b** Reward feature weights. **c** Average reward differences. **d** Reward differences features at different time steps (color figure online)

cant for both MMS ($t(4) = -4.22$, $p = 0.008$) and RE ($t(4) = -3.06$, $p = 0.03$).

3.2.6 Distance to the edges of the table

The distance of the bouncing point of the ball to the edges of the table had only a small positive influence in the reward function yielded by the max-margin algorithm. The reward function yielded by the RE algorithm assigned a little negative reward for playing the ball close to the edge in the y-direction (i.e., along the width of the table) and a relatively high negative reward for playing the ball close to the edge in the x-direction (direction toward the player). The average reward differences in the evaluations indicate that the reward assigned by the reward function of the RE method is to be favored (see Fig. 7b). However, the average reward differences in x- and y-directions are not significant for both MMS ($t(4) = 2.07$, $p = 0.09$; $t4) = 1.18$, $p = 0.29$) and RE ($t(4) = -1.85$, $p = 0.12$; $t(4) = -0.91$, $p = 0.40$).

3.2.7 Distance to the opponent

Maximizing the difference between the position of the bouncing point and the position of the opponent in the x-direction (i.e., direction toward the opponent) received only a small reward (Fig. 7a) and also had only a small effect in the evaluations (Fig. 7b). While the reward function of the maxi-

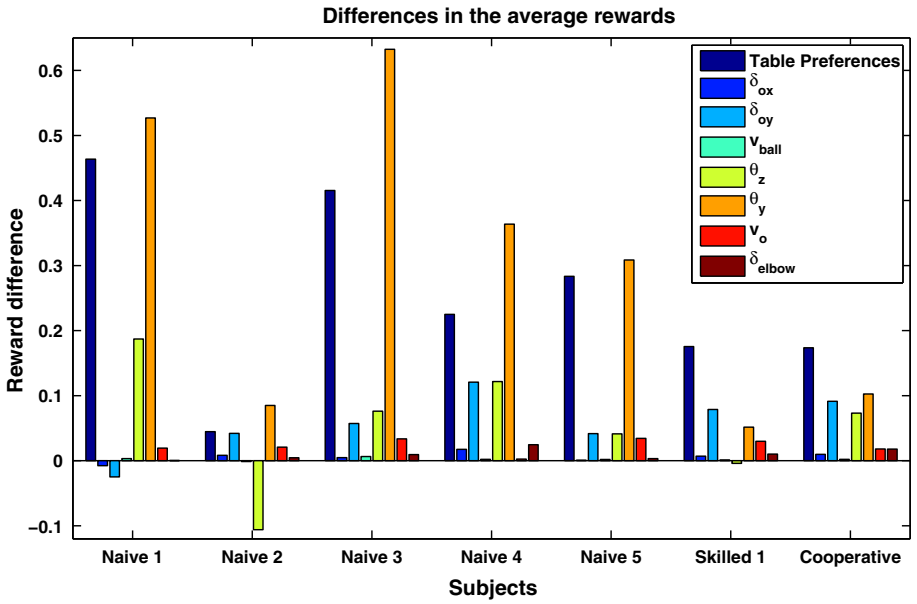


Fig. 8 Individual player preferences. Histogram of the average reward differences between the expert and nonoptimal players for each player and each feature individually. The reward function was received by the MMS algorithm with a horizon of three (color figure online)

mum margin algorithm assigned a slightly positive reward for maximizing this distance, the reward function yielded by the relative entropy algorithm assigned a slightly negative reward. The evaluations on the spared test data were in favor for the positive reward weights. The differences in the average reward were not significant for both MMS ($t(4) = -1.5$, $p = 0.19$) and RE ($t(4) = 1.25$, $p = 0.26$).

The distance in y-direction (i.e., along the width of the table) between the bouncing point and the racket of the opponent resulted in a high reward in both reward functions. This feature also influences the differences in the reward yield by the naive and expert table tennis player. The difference in the average reward of the expert and the subjects was significant for both MMS ($t(4) = -2.67$, $p = 0.044$) and RE($t(4) = -2.69$, $p = 0.046$).

The overall performance on average only increased by $\sim [0.05|0.08]$.[4] The differences in the average reward for the features before a terminal state increased dramatically by $\sim [0.26|0.40]$ and became a dominant factor in the reward function (see Fig. 7d). The differences between the average reward two states before the terminal were below average. This observation suggests that the chance of winning a point increases with an increasing distance between the bouncing point and the racket between the player.

3.2.8 Proximity to the elbow

Playing toward the elbow of the opponent had a negative effect. The weights for the elbow features were negative and increased the differences in the average reward between nonexpert players and the expert player (see Fig. 7b). The differences in the average rewards between expert and subjects were significant for RE ($t(4) = -3.01$, $p = 0.03$), but not for MMS ($t(4) = -2.47$, $p = 0.06$).

3.2.9 Velocity of the ball and opponent

The feature for the velocity of the ball had only a small positive weight and almost no influence on the difference between the players (see Fig. 7a, b) in the evaluations. This feature was also not significant for both MMS ($t(4) = -2.24$, $p = 0.07$) and RE ($t(4) = -2.25$, $p = 0.07$).

The movement direction of the opponent relative to the ball had a moderate positive weight (see Fig. 7a), but only a small influence in the evaluations on the differences between the nonexpert and expert data set. These differences were significant in both MMS ($t(4) = -4.7$, $p = 0.005$) and RE ($t(4) = -3.8$, $p = 0.01$). This observation indicates that this feature was used by the expert but did not dominate his behavior.

3.2.10 Direction of the ball

We evaluated the direction of the ball by means of two angles: θ_z and θ_y. The horizontal angle θ_z had a high negative reward value, i.e., smaller angles were preferred. The overall difference in the performance between the expert and the naive players did increase the overall reward difference only slightly. Hence, the ball was in general played in a slightly flatter manner by the expert. However, this feature was not significant for both MMS [$t(4) = -1.26$, $p = 0.26$] and RE [$t(4) = -0.35$, $p = 0.73$].

[4] In the following, the first value will correspond to the reward differences obtained by MMS algorithm and the second value will correspond to the reward differences obtained by the RE algorithm.

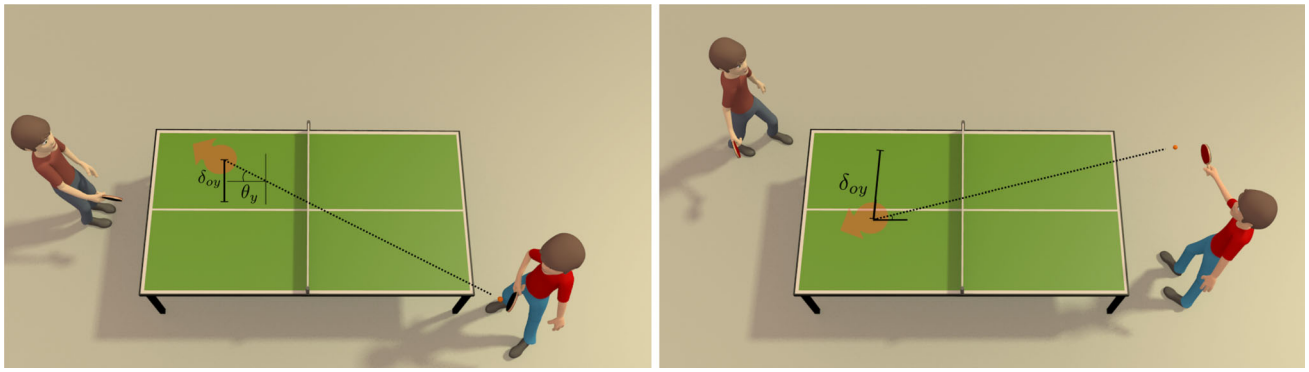


Fig. 9 Possible strategy that distinguished the expert player that won the game, from the nonexpert players that lost the game against the opponent. If the expert had the chance, he would play the ball very cross to the *backhand area* (left side). As a result, the opponent was forced to move more into the *left* corner. The expert could then play the ball to the *forehand area* in order to increase the distance between the ball and the opponent (right side)

The angle θ_y also had a high negative weight, i.e., playing the ball cross to the backhand area was preferred to playing the ball cross toward the forehand area. These results are conform with the table preferences as displayed in Fig. 7a. This feature was one of the dominating factors in the reward function and in the evaluations of the excluded subjects. The average difference between expert and naive players for the state right before the terminal state was only decreased by $\sim [0.02|0.01]$. The average reward two states before the terminal state on the other side were much higher than the overall average reward ($\sim [0.48|0.25]$). The differences in the average reward of the expert and the subjects were significant for this feature for both MMS ($t(4) = -3.46$, $p = 0.018$) and RE ($t(4) = -3.56$, $p = 0.016$).

This observation together with the results of the distance of the bouncing point and the racket suggests the following strategy successfully applied by the Expert. When playing the ball very cross to the outer backhand area of the opponent, the opponent was forced to move to his left. The expert used this opportunity to play the ball to the other side of the table in order to increase the distance between the ball and the opponent, although he usually did not play to the forehand area (see Fig. 9).

The observation that the overall difference in the reward between the expert and Naive 2 and the expert and Skilled 1 is not high indicates that these two players use similar techniques in terms of playing the ball cross to the backhand area. However, when comparing the results in the last hits before the terminal state, we notice that (i) the expert usually plays the ball more cross in the backhand area, forcing the opponent to move further in this direction and (ii) the other two players did not play the ball into the other direction afterward in order to increase the distance.

4 Conclusion

In this paper, we modeled table tennis games as a MDP. We have shown that it is possible to automatically extract expert knowledge on effective elements of basic strategy in the form of a reward function using model-free IRL. To accomplish this step, we collected data from humans playing table tennis using a motion capture system. Participants with different skill levels played in both a competitive and a cooperative game during this study. Based on their performance, we divided the data into an expert and a nonoptimal data set. These data sets have been used to infer and evaluate the reward functions.

We have tested three different model-free inverse reinforcement learning methods. Two were derived from the model-based IRL method of Abbeel and Ng (2004). The third algorithm was the model-free relative entropy method of Boularias et al. (2011). The resulting reward functions were evaluated successfully in a leave-one-subject-out testing scheme. All learned reward functions were able to distinguish strategic information of players with different playing skills and styles. The findings of all tested IRL methods support each other and demonstrate that they are all suitable for the challenging task context presented in this paper.

The presented approach used information about the position of the player and the opponent as well as the ball position, velocity and orientation. However, assumptions made by the player about the spin or the strategy of the opponent were not included in this setup. The reward function was able to capture the goal of the task, in terms of winning the rally while avoiding to lose it. The key elements revealed by the model were (i) playing cross to the backhand area of the opponent, (ii) maximizing the distance of the bouncing point of the ball and the opponent and (iii) playing the ball in a flat manner.

Other elements as playing against the moving direction and the velocity of the ball were also positively correlated.

The presented approach is not limited to analyzing individual preferences of players and successful strategic components against a specific opponent. Rather, the learned reward function can also be used within the MDP framework for artificial systems such as table tennis robots or virtual reality-based table tennis games. Thus, the robot can learn a strategy against a human opponent. The described method allows an artificial system to analyze the strategy of the opponent, and as a result, the system will be able to anticipate the actions of its opponent. Such anticipation can allow artificial systems to adapt their own strategies to improve their chances.[5]

In this paper, we modeled table tennis as an MDP, assuming the task consists of one agent that has perfect knowledge about its environment. This approach is a good starting point, but might be an overly strong assumption. In the current model, we did not account for the opponent's personal weaknesses, his strategy, spin of the ball and the possibility of imperfect sensory information. Here, PoMDPs could be useful. In contrast to modeling the task using a MDP, PoMDPs assume that the agent cannot completely observe its environment. PoMDPs model uncertainty of the state the agent is currently in such that we are able to include beliefs about the intentions of the opponent. Here, it should be investigated whether it is possible to extend the model-free methods presented in this paper to PoMDPs.

In future work, we will also investigate whether it is possible to use the Kinect cameras instead of the VICON system in order to track the players. Furthermore, we plan to integrate the results of this study into a robot table tennis setup.

Acknowledgments We would like to thank Ekaterina Volkova for her support with the calibration and advise for the motion suits and VICON system, as well as Volker Grabe for his technical support for the integration of Kinect and VICON with ROS. We also like to thank Dr. Tobias Meilinger for helpful comments on the psychological part of this experiment and Oliver Kroemer for proof reading this paper.

References

Abbeel P, Coates A, Ng A (2010) Autonomous helicopter aerobatics through apprenticeship learning. Int J Robotics Res 29:1608–1679

Abbeel P, Dolgov D, Ng A, Thrun S (2008) Apprenticeship learning for motion planning with application to parking lot navigation. In: Proceedings of the international conference on intelligent robots and systems (IROS)

Abbeel P, Ng A (2004) Apprenticeship learning via inverse reinforcement learning. In Proceedings of the 21st international conference of machine learning (ICML)

Argall B, Chernova S, Veloso MM, Browning B (2009) A survey of robot learning from demonstration. Robot Auton Syst 57(5):469–483

Boularias A, Kober J, Peters J (2011) Relative entropy inverse reinforcement learning. In: Proceedings of the artificial intelligences and statistics (AISTATS), pp 20–27

Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory, volume 15 of studies in applied mathematics. SIAM, Philadelphia

Braitenberg V (1984) Vehicles: experiments in synthetic psychology. MIT Press, Cambridge

Braitenberg V, Heck D, Sultan F (1997) The detection and generation of sequences as a key to cerebellar function: experiments and theory. Behav Brian Sci 20:229–277

Chandramohan S, Geist M, Lefevre F, Pietquin O (2011) User simulation in dialogue systems using inverse reinforcement learning. In: Proceedings of the 12th annual conference of the international speech communication association

Diaz G, Cooper J, Rothkopf C, Hayhoe M (2013) Saccades to future ball location reveal memory-based prediction in a natural interception task. J Vis 13(1):1–14

Hohmann A, Zhang H, Koth A (2004) Performance diagnosis through mathematical simulation in table tennis. In: Lees A, Kahn J-F, Maynard I (eds) Science and racket sports III. Routledge, London, pp 220–226

International Table Tennis Federation (2011) Table tennis rules

Kober J, Wilhelm A, Oztop E, Peters J (2012) Reinforcement learning to adjust parameterized motor primitives to new situations. Auton Robot 33(4):361–379

Kolter Z, Ng A (2011) The Stanford LittleDog: A learning and rapid replanning approach to quadruped locomotion. Int J Robot Res 30(2):150–174

Levine S, Popovic Z, Koltun V (2010) Feature construction for inverse reinforcement learning. In: Advances in neural information processing systems (NIPS), pp 1342–1350

Levine S, Popovic Z, Koltun V (2011) Nonlinear inverse reinforcement learning with gaussian processes. Adv Neural Inf Process Syst 19–27

Monahan G (1982) A survey of partially observable markov decision processes: theory, models and algorithms. Manag Sci 28:1–16

Mori T, Howard M, Vijayakumar S (2011) Model-free apprenticeship learning for transfer of human impedance behaviour. In: Proceedings of the 11th IEEE-RAS international conference on humanoid robots (HUMANOIDS), pp 239–246

Muelling K, Kober J, Kroemer O, Peters J (2013) Learning to select and generalize striking movements in robot table tennis. Int J Robot Res 32(3):263–279

Ng A, Russel X (2000) Algorithms for inverse reinforcement learning. In: Proceedings of the 17th international conference of, machine learning, pp 663–670

Powell W (2011) Approximate dynamic programming: solving the curses of dimensionality, 1st edn. Wiley, New York

Puterman M (1994) Markov decision processes: discrete stochastic dynamic programming, 1st edn. Wiley, New York

Ramachandran D, Amir E (2007) Bayesian inverse reinforcement learning. In: Proceedings of the 20th international joint conference of artificial intelligence (IJCAI), pp 2586–2591

Ratliff N, Bagnell J, Zinkevich M (2006) Maximum margin planning. In: Proceedings of the 23rd international conference on machine learning (ICML), pp 729–736

Rothkopf C, Ballard D (2013) Modular inverse reinforcement learning for visuomotor behavior. Biol Cybern 107:477–490

Rothkopf C, Dimitrakakis C (2011) Preference elicitation and inverse reinforcement learning. In: 22nd European conference on machine learning (ECML)

[5] Please note, such a reward function could also contain agent-specific intrinsic cost, which might not be straightforward to transfer to an artificial system.

Schaal S (1999) Is imitation learning the route to humanoid robots? Trends Cogn Sci 6:233–242

Seve C, Saury J, Leblanc S, Durand M (2004) Course-of-action theory in table tennis: a qualitative analysis of the knowledge used by three elite players during matches. Revue europeen de psychologie appliquee

Sutton R, Barto A (1998) Reinforcement learning: an introduction. The MIT Press, Cambridge

Vis J, Kosters W, Terroba A (2010) Tennis patterns: player, match and beyond. In: 22nd Benelux conference on artificial intelligence

Wang J, Parameswaran N (2005) Analyzing tennis tactics from broadcasting tennis video clips. In: Proceedings of the 11th international multimedia modelling conference, pp 102–106

Wang P, Cai R, Yang S (2004) A tennis video indexing approach through pattern discovery in interactive process. Adv Multimed Inf Process 3331:56–59

Zhifei S, Joo E (2012) A survey of inverse reinforcement learning techniques. Int J Intell Comput Cybern 5(3):293–311

Ziebart B, Maas A, Bagnell A, Dey A (2008) Maximum entropy inverse reinforcement learning. In: Proceedings of the 23th national conference of artificial intelligence (AAAI), pp 1433–1438

Ziebart B, Ratliff N, Gallagher G, Mertz C, Peterson K, Bagnell A, Herbert M, Srinivasa S (2009) Planning based prediction for pedestrians. In: Proceedings of the international conference on intelligent robotics and systems (IROS)