# Learning to Play Soccer using Imitative Reinforcement

Sven Behnke and Maren Bennewitz
University of Freiburg, Computer Science Institute
Georges-Khler-Allee 52, 79110 Freiburg, Germany

**Abstract**

The reinforcement framework is a principled approach for agents learning to act in an environment. In the long run, reinforcement learning finds optimal policies. However, a physical agent, such as a humanoid robot, acting in the real world can perform only a limited number of trails, and consequently has only access to limited experience. With such limitations, the exhaustive exploration of high-dimensional state and action spaces is not feasible. One approach to this dilemma is to utilize experiences of other agents by imitating their behavior. If the agents are sufficiently similar, this can speed-up learning dramatically.

We propose to give the learning agent access to the Q-values of an experienced agent. The learner combines them with its own Q-values in order to determine its policy. This should head-start learning. We plan to evaluate the effects of this knowledge transfer in a task derived from the RoboCup soccer domain using a humanoid robot.

## 1 Introduction

Reinforcement learning (RL) offers principled methods for agents to improve upon their actions in a reward-generating environment [17]. The framework underlying RL is that of Markov decision processes (MDPs), which describe the effects of actions in a stochastic environment and the possible rewards at various environmental states. The goal of the agent is to maximize the expected (discounted) future reward, without knowing the MDP or the reward function in advance.

Both, model-based and model-free approaches exist to find optimal policies when agents are allowed to act for unlimited time. The impressive power of RL has been demonstrated in several simulated environments (see e.g. [18]), where it is easy to run many trials. For physical agents, such as humanoid robots acting in the real world, it is much more difficult to gain experience. Hence, the exhaustive exploration of high-dimensional state and action spaces is not feasible. For a physical robot, it is essential to learn from few trials in order to have some time left for exploitation.

Several methods have been proposed to speed-up RL. Among them are hierarchical RL [4, 10], subtask decomposition [6], and imitation. Imitating experienced agents offers the possibility to leverage their experiences in order to head-start RL. Such a social learning allows overcoming the limitations of standard RL, where each agent has to reinvent good policies.

Imitation learning is a well established concept in robotics and social sciences [1, 3, 5, 7, 8, 9, 11]. Several approaches already exist that use imitation to accelerate RL. Behavior cloning [14, 20] transfers the policy of the experienced agent to the learner. In implicit imitation [13], the learner observes state changes of the experienced agent and infers the transition model of the underlying MDP. This approach assumes that the actions of the experienced agent are not observable. The rational constraint technique [21] and LQ controller induction [16] attempt to transfer the Q-values, representing expected discounted future rewards for state-action pairs, from one agent to another.

While many of these techniques have been developed using highly simplified settings, such as grid-worlds, imitative RL has also been successfully applied to real robots. For instance, Schaal [15] showed that learning by demonstration can significantly speed up model-based RL in the case of teaching a robot arm to balance a pole and pendulum-swing up. The technique presented by Ng et al. [12] learned a stochastic model of the dynamics of a helicopter by applying supervised learning on a data set consisting of logged

commands of a human pilot and the resulting states of the helicopter. Afterwards, they used Monte Carlo sampling to approximate the optimal policy.

## 2 Value Transfer

Transferring Q-values offers the possibility for combining experience of one agent with the experience of another agent. It assumes that both agents are sufficiently similar, act in the same world, and pursue the same goals. Hence, they share a MDP $\langle \mathcal{S}, \mathcal{A}, T, R \rangle$, where $\mathcal{S}$ denotes the set of states, $\mathcal{A}$ is the set of possible actions, $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ describes the state changes caused by actions, and $R : \mathcal{S} \rightarrow \Re$ assigns rewards to the states.

RL attempts to find a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the discounted rewards $\sum_{t=0}^{\infty} \gamma^t r_t$, where $r_t$ denotes the reward received at time $t$ and $0 \leq \gamma \leq 1$ is the discount rate of future rewards.

One popular RL-algorithm is Q-learning [22]. The algorithm maintains a function $Q(s, a)$ that represents the expected discounted future reward when the learning agent takes action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. This value function is updated according to the rule

$$Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha \big( r + \gamma \operatorname*{argmax}_{a'} Q(s', a') \big),$$

whenever a new experience $(a, s, r, s')$ is made, where $s'$ is the state reached after taking action $a$ in state $s$. The learning rate $0 \leq \alpha \leq 1$ is chosen according to the stochasticity of the MDP.

While the greedy policy $\pi_g(s) = \text{argmax}_a \, Q(s, a)$ is optimal when the Q-values are accurate, at the beginning of the learning nothing is known about $Q(s, a)$. In $\epsilon$-greedy policies, random actions are chosen with decreasing probability $\epsilon$ in order to explore the state-action space. In high-dimensional state-action spaces, such a random exploration takes exponentially long. Q-value initialization and reward-shaping are equivalent ways [23] of providing extra hints to accelerate Q-learning. These hints are based on potential functions, which show the agent the direction to goal states. Such functions are hard to provide if a good policy is not already known.

We propose to give the learning agent access to the Q-values $\hat{Q}(s, a)$ of an experienced agent in order to head-start Q-learning. To weight the experiences of both agents, we maintain counters $c(s)$ for the learner and $\hat{c}(s)$ for the experienced agent that keep track of the number of visits of a state $s$. An $\epsilon$-greedy imitative policy can now be defined as

$$\pi_i(s) = \text{argmax}_a \big( \lambda Q(s, a) + (1 - \lambda) \hat{Q}(s, a) \big), \text{ with } \lambda = \begin{cases} 1 & \text{if } c(s) + \beta \hat{c}(s) = 0 \\ c(s) / \big( c(s) + \beta \hat{c}(s) \big) & \text{otherwise} \end{cases}.$$

The imitation rate $0 \leq \beta \leq 1$ depends on the similarity of the agents. With probability $(1 - \epsilon)$ the agent chooses $\pi_i(s)$. Otherwise it performs a random action.

If the learner encounters a new state ($c(s) = 0$), it will imitate the experienced agent. The influence of the learner's Q-values increases the more often it encounters a state, relative to the experienced agent. In the long run, the influence of the experienced agent vanishes completely. This is important to account for differences between the learner and the experienced agent. Ultimately, it allows the learner to exceed the performance of the experienced agent.

## 3 Soccer Task

We plan to evaluate the effects of such an imitative policy in a task from the RoboCup soccer domain, using a humanoid robot.

Fig. 1(a) shows one of the robots we use for our research. The robot base, RoboSapien, was developed by Tilden [19] for the toy market. It is driven by seven DC motors and powered by batteries located in its feet. The low center of mass makes RoboSapien very stable. Using three motors it can walk dynamically with two speeds in sagittal direction and also turn on the spot. The other four motors move its arms. The original RoboSapien is controlled by the user with a remote control. We made it autonomous using a camera-equipped Pocket PC [2].

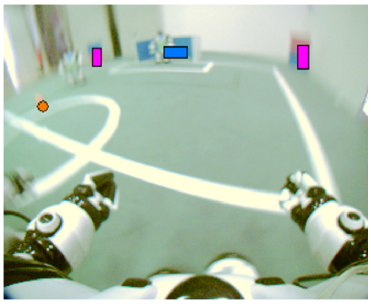
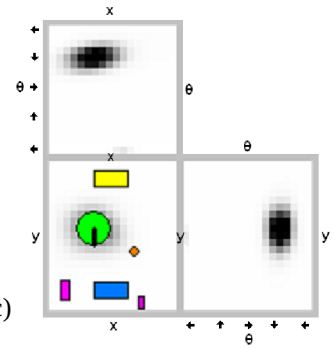
(a)  (b)  (c) 


Figure 1: (a) Augmented RoboSapien. (b) Image captured from RoboSapien's perspective while it was walking with detected objects: goal (blue horizontal rectangle), ball (orange circle), and markers (magenta vertical rectangles). (c) Three two-dimensional projections of the grid representing the probability distribution of robot poses $(x, y, \theta)$. The green circle is drawn at the estimated robot location $(x, y)$. The black line represents its estimated orientation $\theta$. The detected objects are drawn relative to the robot.

The only source of information about the environment of the robot is the wide-angle camera. Its large field of view allows seeing the ball at the robot's feet and the goal simultaneously (see Fig. 1(b)). The Pocket PC runs higher-level behavior control, computer vision, self-localization (Fig. 1(c)), and wireless communication and sends motion commands to the robot base via infrared. To simplify the behavior control interface to the robot base, we implemented parameterized motion functions, like walking straight for a certain distance or turning for a certain angle. To implement more complex behaviors, we use a finite state machine to decompose complex tasks into subtasks. For example, the task of scoring a goal consists of positioning the robot behind the ball such that it faces the goal and than moving towards the goal. The transitions between the states are triggered based on visual input.

Initially, we want to apply imitative RL to the task of dribbling the ball into an empty goal. As experienced agents we will use a human-controlled RoboSapien, an autonomous RoboSapien controlled by our hand-coded behaviors, and autonomous RoboSapiens with behavior trained by RL.

To vary the complexity of the task, we will use different encodings of the state and action spaces. One possibility to encode the state space is to discretize relative coordinates (angle and distance) of the ball and the goal, which are estimated by the computer vision module. One possibility to encode the action space is to discretize walk and turn commands, sent to the robot. Another option would be to use macro-actions, such as moving behind the ball and walking towards the goal.

Rewards will be given for scored goals. We will also test the effect of including the ball-goal distance into the reward function.

# 4 Conclusions

We proposed to give a soccer-playing humanoid robot which acts in an RL framework access to the Q-values of experienced robots. The learner will combine them with its own Q-values in order to determine its policy. This should head-start RL and hence allow for learning within the few trials which are possible with real robots.

The proposed imitative policy is different from Q-value initialization. Initial Q-values decay exponentially fast whereas in the proposed policy state counters are used to weight the Q-values of the learner and the experienced agent.

# References


[1] Minoru Asada, Masaki Ogino, Shigeo Matsuyama, and Jun'ichiro Ooga. Imitation learning based on visuo-somatic mapping. In *Proc. of International Symposium on Experimental Robotics (ISER)*, 2004.

[2] Sven Behnke, Tobias Langner, Jürgen Müller, Holger Neub, and Michael Schreiber. NimbRo RS: A low-cost autonomous humanoid robot for multi-agent research. In *Proc. of the Workshop on Methods and Technology for Empirical Evaluation of Multi-Agent Systems and Multi-robot Teams (MTEE) at the German Conference on Artificial Intelligence (KI)*, 2004.

[3] Darrin C. Bentivegna, Christopher G. Atkeson, and Gordon Cheng. Learning tasks from observation and practice. *Journal of Robotics & Autonomous Systems*, 47(2-3):163–169, 2004.

[4] Thomas G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

[5] Rüdiger Dillmann. Teaching and learning of robot tasks via observation of human performance. *Journal of Robotics & Autonomous Systems*, 47(2-3):109–116, 2004.

[6] Chris Drummond. Accelerating reinforcement learning by composing solutions of automatically identified subtasks. *Journal of Artificial Intelligence Research*, 16:59–104, 2002.

[7] Auke J. Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2002.

[8] Tetsunari Inamura, Hiroaki Tanie, and Yoshihiko Nakamura. From stochastic motion generation and recognition to geometric symbol development and manipulation. In *Proc. of the International Conference on Humanoid Robots (Humanoids)*, 2003.

[9] Masato Ito and Jun Tani. Joint attention between a humanoid robot and users in imitation game. In *Proc. of the Int. Conf. on Development and Learning (ICDL)*, 2004.

[10] Alexander Kleiner, Markus Dietl, and Bernhard Nebel. Towards a life-long learning soccer agent. In *Proc. of the International RoboCup Symposium*, 2002.

[11] Maja J. Mataric. Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics. In K. Dautenhahn and C.L. Nehaniv, editors, *Imitation in Animals and Artifacts*. MIT Press, 2002.

[12] Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. Inverted autonomous helicopter flight via reinforcement learning. In *Proc. of International Symposium on Experimental Robotics (ISER)*, 2004.

[13] Bob Price and Craig Boutilier. Accelerating reinforcement learning through implicit imitation. *Journal of Artificial Intelligence Research*, 19:569–629, 2003.

[14] Claude Sammut, Scott Hurst, Dana Kedzier, and Donald Michie. Learning to fly. In *Proc. of Ninth International Workshop on Machine Learning*, pages 385–393, 1992.

[15] Stefan Schaal. Learning from demonstration. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 1997.

[16] Dorian Suc and Ivan Bratko. Skill reconstruction as induction of LQ controllers with subgoals. In *Proc. of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, pages 914–919, 1997.

[17] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[18] Gerald Tesauro. Temporal difference learning and TD-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[19] Mark W. Tilden. Neuromorphic robot humanoid to step into the market. *The Neuromorphic Engineer*, 1(1), 2004.

[20] Tanja Urbancic and Ivan Bratko. Reconstructing human skill with machine learning. In *Proc. of Eleventh European Conference on Artificial Intelligence (ECAI), Amsterdam*, pages 498–502, 1994.

[21] Paul E. Utgoff and Jeffrey A. Clouse. Two kinds of training information for evaluation function learning. In *Proc. of the 9th National Conference on Artificial Intelligence (AAAI)*, volume 2, pages 596–600, 1991.

[22] Christopher J.C.H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[23] Eric Wiewiora. Potential-based shaping and Q-value initialization are equivalent. *Journal of Artificial Intelligence Research*, 19:205–208, 2003.