Algorithmic and Human Teaching of Sequential Decision Tasks

Maya Cakmak, Manuel Lopes

Algorithmic and Human Teaching of Sequential Decision Tasks

Maya Cakmak
Georgia Institute of Technology
Atlanta, GA, USA
maya@cc.gatech.edu

Manuel Lopes
INRIA
Bordeaux Sud-Ouest, France
manuel.lopes@inria.fr

Abstract

A helpful teacher can significantly improve the learning rate of a learning agent. Teaching algorithms have been formally studied within the field of Algorithmic Teaching. These give important insights into how a teacher can select the most informative examples while teaching a new concept. However the field has so far focused purely on classification tasks. In this paper we introduce a novel method for optimally teaching sequential decision tasks. We present an algorithm that automatically selects the set of most informative demonstrations and evaluate it on several navigation tasks. Next, we explore the idea of using this algorithm to produce instructions for humans on how to choose examples when teaching sequential decision tasks. We present a user study that demonstrates the utility of such instructions.

Introduction

We extend the field of *Algorithmic Teaching* (AT) to *Markov Decision Processes* (MDP)s. AT formally studies the optimal teaching problem, that is, finding the smallest sequence of examples that uniquely identifies a target concept to a learner. Work within AT, tries to identify the teachability of different concept classes and devise efficient algorithms that produce optimal teaching sequences (Balbach and Zeugmann 2009; Goldman and Kearns 1995). So far, AT has focused purely on classification problems and no algorithm exists for optimal teaching of sequential decision tasks.

Benefits of extending AT to sequential decision tasks are two-fold. First, this would enable automatic tutoring systems involving such sequential tasks, *e.g.* flight/surgical training, physical rehabilitation or surveillance training. In these applications, the training of humans requires extensive practice and time, and an optimal teaching algorithm could reduce this cost without compromising the final training quality. Secondly, an optimal teaching algorithm gives insights into what constitutes informative examples for a learning agent. These insights can in turn be used by human teachers while training a learning agent to perform sequential decision tasks, *e.g.* a personal robot or virtual assistant.

In this paper we present a teaching algorithm for sequential decision problems. We consider a learning agent that uses Inverse Reinforcement Learning to learn a reward model. We present an algorithm that selects the most informative demonstrations for the learner. Intuitively, an informative demonstration allows the learner to compute the relevant features of the task and, more importantly, informs the learner about the relative importance of each feature. An illustrative example is given in Fig. 1. We show that a learner trained with non-optimal selected expert demonstrations require significantly more demonstrations to achieve a similar performance as the optimally taught learner.

Next, we analyze the impact of describing the intuition of the optimal teaching algorithm to human teachers, and find that it improves the informativeness of examples that they provide while teaching sequential decision problems.

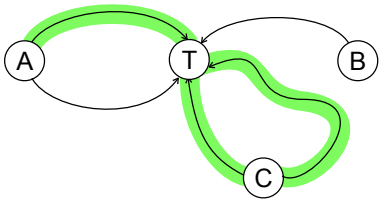

Figure 1: An example that motivates optimal teaching. Assume that a teacher wants to train an agent that prefers navigating through green shaded areas to reach a target state T. In order to give a demonstration of this policy, the teacher needs to select a start location (A, B or C) and the path to take. In this case, starting at A and following the shaded road will best communicate the teacher's preference. Starting at B would not reveal anything and starting at C would only reveal a preference between the lengths of the roads.

Related Work

The Algorithmic Teaching literature presents a range of metrics that characterize the teachability of a concept, such as the Teaching Dimension (Goldman and Kearns 1995), or its variants (Natarajan 1987; Anthony, Brightwell, and Shawe-Taylor 1995; Balbach 2008). An important challenge in the field is to devise polynomial time algorithms for the complex optimal teaching problem. Other work on teaching takes a more human-inspired approach. For instance, *curriculum design* suggests that the teacher should select ex-

amples by starting with *easy* examples and gradually increasing the difficulty (Elman 1993; Bengio et al. 2009; Zang et al. 2010; Taylor 2009). (Khan, Zhu, and Mutlu 2011) demonstrated that this is the most prominent strategy in human teaching and provided a theoretical account for it. *Pedagogical sampling* assumes a benevolent teacher and learner; consequently the teacher chooses most informative examples for the learner (Shafto and Goodman 2008; Warner, Stoess, and Shafto 2011). Our work contributes to both the theoretical accounts of algorithmic teaching and the empirical accounts of human teaching by extending them to sequential decision tasks.

A closely related area for the work presented in this paper is *Active Learning* (AL) (Angluin 1988; Settles 2010). The goal of AL, like in AT, is to reduce the number of demonstrations needed to train an agent. AL gives the learner control of what examples it is going to learn from, thereby steering the teacher's input towards useful examples. In many cases, a teacher that chooses examples optimally will teach a concept significantly faster than an active learner choosing its own examples (Goldman and Kearns 1995). Thus, the idea explored in this paper, of trying to improve the teacher, has a lot of potential for making the teaching process more efficient. AL has been applied to problems such as *preference elicitation* from users (Fürnkranz and Hüllermeier 2010; Viappiani and Boutilier 2010) or interactive learning agents (Chernova and Veloso 2007; Grollman and Jenkins 2007), with extensions to the IRL framework (Lopes, Melo, and Montesano 2009; Cohn, Durfee, and Singh 2011). These works serve as inspiration for the computational methods developed in this paper.

Background

We start with related preliminaries on MDPs and *Inverse Reinforcement Learning* (IRL)(Ng and Russell 2000).

Markov Decision Processes. We consider a standard MDP defined as a five element tuple (S, A, P, R, γ) (Sutton and Barto 1998). S and A are the state and action spaces, R is a reward function, P is a state transition model and γ is the discount factor. The goal of Reinforcement Learning (RL) is to find a policy $\pi : S \rightarrow p(A)$ that tells the learning agent what action to take in a given state such that its total reward is maximized. We consider both *deterministic* policies that map a state to an action and *stochastic* policies that associate a certain probability to each action. Stochastic policies are assumed to give equal probability to all optimal actions, and zero probability to sub-optimal actions.

A *value function* corresponds to the expected return for a state when following policy π: $V^\pi(s_0) = E_{\pi,s_0}[\sum_{t=0}^{\infty} \gamma^t R(s_t)]$, where s_t is the state reached at step t when the agent starts at state s and follows policy π. A *Q-function* is the value associated with taking an action in a state, and can be written in terms of the value function as $Q^\pi(s,a) = R(s) + \gamma E_y[V^\pi(y)]$, where y_a is the state reached by taking action a in state s.

Inverse Reinforcement Learning. Learning from Demonstration involves an agent learning a policy by observing demonstrations given by a teacher, rather than interacting with the environment (Argall et al. 2009). Two main approaches for achieving this are direct policy learning and IRL (Ng and Russell 2000). The former models the policy directly from the observed state-action pairs. The latter assumes that the teacher is trying to maximize a reward and models this reward based on the demonstrations. In IRL the reward function R is unknown and it cannot be sampled from the process.

In many situations the reward function is more accurately described as a combination of features. We assume, without loss of generality, that the reward can be written as a linear combinations of features $R(s) = \sum_i w_i f_i(s)$ as in (Abbeel and Ng 2004), where $f_i(s)$ is a function on the ith feature of state s. With this substitution the value function can be re-written as $V^\pi(s) = \sum_i w_i E_{s,\pi}\left[\sum_t \gamma^t f_i(s_t)\right]$.

The inner term in the second summation is known as the feature counts (Abbeel and Ng 2004). We denote these with $\mu_i^{\pi,s} = E_{s,\pi}\left(\sum_t \gamma^t f_i(s_t)\right)$. When the action on the initial state is a we represent it by μ^{π,s_a}. Note that this represents the value of a state if the system follows policy π and the reward is $R(s) = f_i(s)$. Thus the value function is:

$$V^\pi(s) = \sum_i w_i \mu_{\pi,s,i} = \mathbf{w}^{\mathrm{T}} \bar{\mu}_{\pi,s}$$

In order to learn the reward function, an IRL agent uses the following intuition. If the teacher chooses a particular action a in state s, then action a must be at least as good as all the other available actions in s: $\forall b,\ Q^*(s,a) \geq Q^*(s,b)$. We can re-write this in terms of the value functions and thus in terms of the feature counts as follows:

$$\forall b,\ \mathbf{w}^{\mathrm{T}} \bar{\mu}_{\pi,s_a} \geq \mathbf{w}^{\mathrm{T}} \bar{\mu}_{\pi,s_b}. \tag{1}$$

This presents a constraint directly on the weight vector. With additional prior information, these constraints can be used to estimate the reward function. Several methods proposed in the literature, allow estimation of the reward and the policy from such constraints obtained from demonstrations, see a review in (Neu and Szepesvári 2009).

Optimal Teaching

Based on the formulation of the IRL agent, an informative set of demonstrations is one that allows the agent to compute relevant feature counts and infer the weights as accurately as possible. Hence, the most informative demonstrations are the ones that reduce the uncertainty in the reward estimation.

A demonstration given by the teacher is a trajectory of state-action pairs. Assume that all state-action pairs from all trajectories provided by the teacher are pooled together in a demonstration set $D = \{(s_t, a_t)\}_{t=1}^{M}$. Based on Equation 1 the constraints placed by this demonstration set on available reward functions can be summarized as:

$$\forall (s,a) \in D,\ \forall b,\ \mathbf{w}^{\mathrm{T}}(\bar{\mu}_{\pi,s_a} - \bar{\mu}_{\pi,s_b}) \geq 0 \tag{2}$$

Note that these inequalities give a set of half-spaces defined by hyperplanes going through the origin in the space of weight vectors. The true weight vector lies in the intersection of these half-spaces. We assume that the weights are

bounded within a hypercube described by $(-M_w < w_i < M_w)$. These bounds will depend on the task and should provide a meaningful scale for the reward function.

We denote the subspace described by the combined constraints as $\subset(D)$. To compare alternative demonstrations given this bounded space of hypotheses, we need a measure of uncertainty. While a number of measures are possible (see (Krause and Guestrin 2005a) for an analysis), we use the volume of the space of possible weights. This volume is estimated using a sampling approach. We uniformly sample weight vectors from the hypercube $(-M_w < w < M_w)$ and count the ones that are within $\subset(D)$. This gives us an indicator of the uncertainty in the reward estimation, denoted as $G(D)$.

$$G(D) = -\frac{1}{N}\sum_{j}^{N} \delta(w_j \in \subset(D))$$

$\delta()$ is an indicator function which is 1 if the argument is true and 0 otherwise. $G(D)$ is the negative of a Monte Carlo estimate for the ratio of the valid space with respect to D, to the space of all possible reward functions.

Finding the set of demonstrations that maximize G is a hard-to-solve combinatorial problem. Thus, we rely on a greedy approximation. This involves sequentially choosing demonstrations that increase $G(D)$ as much as possible. Our algorithm evaluates a set of potential start states in terms of the reduction in uncertainty provided by all possible trajectories that starts at that state.

If the teacher's policy is *deterministic*, then this evaluation can be done directly with the complete trajectory that starts at s and follows the teacher's optimal policy. Thus, we can write $s_0 = \arg\max_s(G(\{D \cup \tau_\pi(s)\})$ where $\tau_\pi(s)$ is a trajectory starting at s and following π for a certain horizon. Then the demonstration provided to the learner is $\tau_\pi(s_0)$.

If the teacher's policy is *stochastic*, then the teacher can sample a set of trajectories that start at s_0 and demonstrate the one that results in the largest $G(\{D \cup \tau_\pi(s_0)\})$. This results in a two step algorithm that first selects the start state for the demonstration, and after committing to the best start state, performs a second step that selects a particular trajectory allowed by the stochastic policy. To maximize $G(.)$, the demonstration that reduces the uncertainty in the feature weights **w** as much as possible is greedily selected at each step. This algorithm which assumes a stochastic policy is outlined in Algorithm 1. Based on the particular structure of the uncertainty measure, we obtain the following result about the optimality bounds of this algorithm.

Theorem 1. *Given a set of candidate demonstrations, Algorithm 1 selects a subset of demonstrations D_g which satisfies the inequality $G(D_g) \geq (1 - 1/e)G(D_{OPT})$ where e is the base of the natural logarithm.*

Proof. (Sketch for the case of deterministic policies) The theorem follows from the fact that $G(.)$ is a monotonous sub-modular function. Every new sample τ only *removes* hypotheses from w, never adding new hypotheses. Thus, $G(w)$ is a non-decreasing monotonous function. $G(.)$ is sub-modular iff for $A \subseteq B,\ s \notin B,\ G(A \cup \{s\}) - G(A) \geq G(B \cup \{s\}) - G(B)$. We can verify this by observing that a sample added in later stages cannot remove more hypotheses than if it was added earlier. The same demonstration presented at a later stage can increase G at most as much as if it was presented in the beginning. From (Nemhauser, Wolsey, and Fisher 1978) we know that for monotonous sub-modular functions, the value of the function for the set obtained with the greedy algorithm $G(D_g)$ is lower-bounded by the value corresponding to the optimal set $G(D_{OPT})$ with a factor of $(1 - 1/e)$. □

Algorithm 1 Optimal Teaching for IRL

Require: Set of possible initial states S_0
Require: Feature weights **w** of the optimal reward function
  Initialize $D \leftarrow \emptyset$
  Compute optimal policy π^* based on **w**
  while $G(D) < \epsilon$ **do**
    $s_0 \leftarrow \arg\max_{s \in S_0} E_\pi(G(\{D \cup \tau_\pi(s)\}))$
    Generate K trajectories $\tau_\pi(s_0)$
    for all $\tau_j, j = 1 \ldots K$ **do**
      Compute $G(D \cup \tau_j)$
    end for
    $\tau \leftarrow \arg\max_j G(D \cup \tau_j)$
    if $G(D \cup \tau)) > G(D)$ **then**
      $D \leftarrow D \cup \{\tau\}$
    end if
  end while
  return Demonstration set D

This result shows that the greedy approximate algorithm for the combinational maximization problem, yields a good approximation to the optimal solution. For stochastic policies, explicit maximization might not be possible and therefore we cannot ensure that the best demonstration is chosen at each step. Nevertheless, if the approximation error in the maximization is small, then we know that the loss we incur will also be small (Krause and Guestrin 2005b).

Evaluation of Optimal Teaching

We present an evaluation of the proposed algorithm on a navigation task. We consider two maps shown in Fig. 2. Both domains have three features. Each square on the map (other than the obstacles) includes one of the three features. This means that the observed feature vector when the agent is on a particular square, is a 3-dimensional vector where only one of the components is 1 and the others are 0. This simple domain was chosen to allow teaching tasks that are understandable by humans and easy to visualize.

For each map, we consider two teaching tasks. A task corresponds to a particular reward function, *i.e.* a weight vector. For the first map (Fig. 2(a)), both tasks have one terminal state – the corner that has the star shaped feature. For Task 1, the weights given to the other two features are equal (both negative). Thus the resulting optimal policy always chooses the shortest path to the terminal state. For Task 2, the negative weight is larger for the dark gray feature. Depending on the ratio of the weights, this results in policies that prefer light gray paths that are longer.

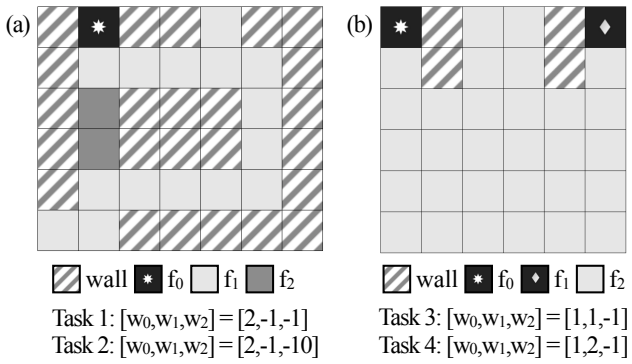

Figure 2: The two maps considered in the experiments and the associated reward weights.

For the second map (Fig. 2(b)), we consider tasks that have two terminal states – the two corners that have the star and diamond shaped features. Both of these features are given a positive weight in the reward function. The rest of the map has the light gray feature that has a certain negative weight associated with it. For Task 3, the positive weights for the two terminal states are equal. This results in a policy that goes to the nearest terminal state. For Task 4, the diamond shaped feature has a larger weight. Depending on the ratio of the weights this results in policies that prefer to go towards the diamond even though it is further away. The actual weight vectors for each task are shown in Fig. 2.

Examples of Optimal Demonstrations

The start states and trajectories produced by our algorithm for Tasks 1 and 2 are shown in Fig. 3. For both tasks the algorithm decides that a single demonstration is sufficient. The significance of the chosen start states is that they are at the mid point between the two possible ways that have similar reward returns. In Task 1, the two terrains have equal cost. Hence the start state of the chosen demonstration is at the furthest location on the map where both paths have equal length. In Task 2, the chosen start state is at the critical point that balances the length of the path and the different costs of the two terrains. Intuitively this communicates that the shortest path is so costly that it is better to take the longest road. The start state that was selected in for Task1 would not be as informative in this task because it could also be explained by having equal costs.

Fig. 4 shows the demonstrations chosen by our algorithm in Tasks 3 and 4. In Task 3, each goal attracts the trajectories that start closer to them. The best places to start a demonstration, are the states around the mid point between the two goals. Presenting a single trajectory is not sufficient, since this could be explained by weights that try to avoid the other goal. In Task 4, the mid point is shifted towards the goal that has a higher reward.

From observing example outcomes of the optimal teaching algorithm we get a better intuition about what constitutes an informative demonstration for the learner. A good teacher must show the range of important decision points

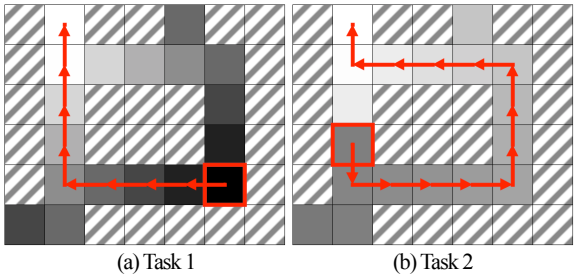

Figure 3: The start states and trajectories chosen by our algorithm for Tasks 1 and 2. The gray-scale color of the squares on the map indicate the value function according to the optimal policy (light colors have high value).

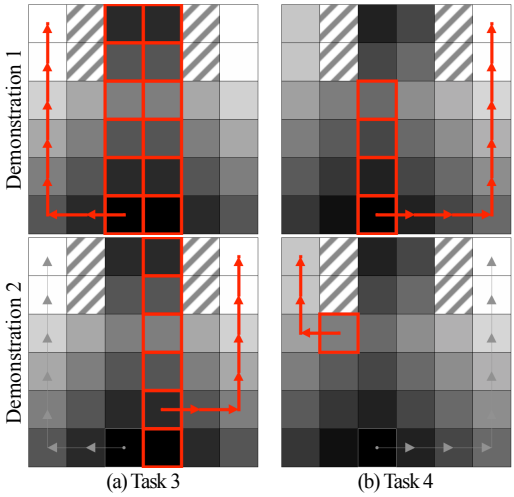

Figure 4: The start states and trajectories chosen by our algorithm for Tasks 3 and 4. The gray-scale color of the squares on the map indicate the value function according to the optimal policy for corresponding reward functions.

that are relevant for the task. The most informative trajectories are the ones where the demonstrator makes rational choices among different alternatives, as opposed to those where all possible choices would result in the same behavior. The teaching instructions provided to human teachers in our human subject experiment is based on this intuition.

Learning Gains

We evaluate the gains achieved with optimal teaching by comparing it with learning from demonstrations whose starting states are chosen randomly. Note that all presented demonstrations follow the optimal policy. We compare the two in terms of (i) the uncertainty in the reward function $G(D)$ achieved by the produced demonstration sets and (ii) the performance of an IRL agent trained with the produced demonstrations. The learning agent uses a gradient IRL approach (Neu and Szepesvári 2007). The performance of the IRL agent is measured by the overlap between the learned policy and the true policy. This is the percentage of actions

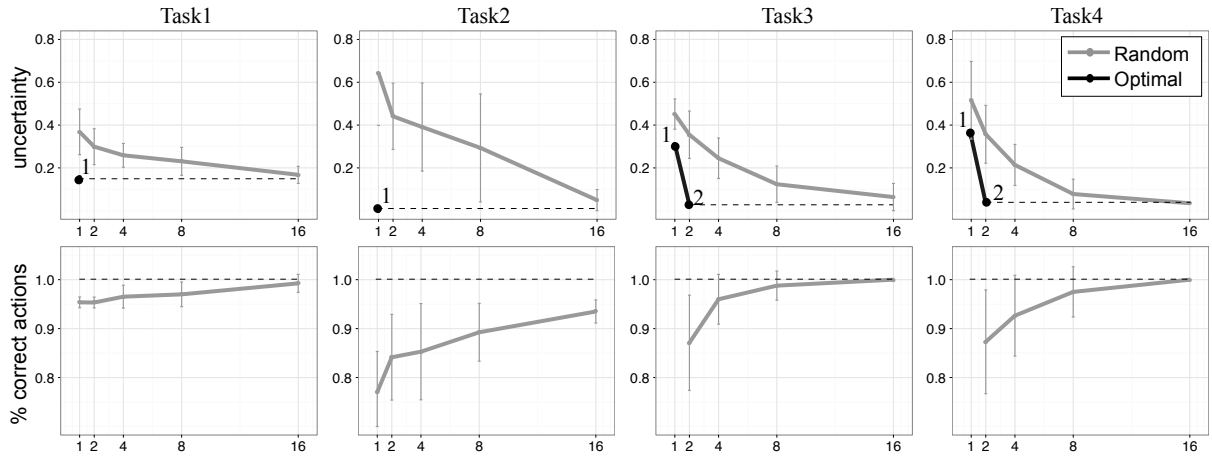

Figure 5: Comparison of learning from optimally selected demonstrations and randomly selected demonstrations. The top row shows the decrease in the uncertainty on the rewards, and the bottom row shows the change in the percentage of correctly chosen actions with the policy obtained from the estimated rewards. The x-axes are the increasing number of demonstrations.

chosen by the agent that are consistent with the true policy.

The learning curves for all four tasks are shown in Fig. 5. We observe that optimal teaching allows much faster convergence in all four tasks. On average, for the first two tasks 16 times more demonstrations are required to reach a similar performance. For the last two tasks this fraction is 1:8. From these graphs we observe that the difficulty of the problem is increased when the weights are less balanced (Task 2 and 4).

Human Teaching

Next we investigate the use of optimal teaching algorithms in improving human teaching. We present a user study demonstrates sub-optimality of natural human teaching and the utility of providing them with instructions that explain the intuition of the optimal teaching algorithm. We refer to such instructions as *teaching guidance*. The utility of teaching guidance was demonstrated in previous work for classification tasks (Cakmak and Thomaz 2010).

Experimental Design

We use the navigation tasks from the previous section. Our experiment has a between-groups design with two factors. The first factor is whether or not the participant receives teaching guidance (*Natural* group versus *Guided* group). The second factor is the teaching task, which can be one of the four tasks illustrated in Fig. 2. Thus each participant performs a single task, with or without guidance, and we have a total of eight groups.

Procedure. Our experiments are web-based. The webpage layout consists of three parts separated by horizontal lines. The top part has general instructions to the participant. The middle part is a Java applet that lets the teacher interact with the map, and present demonstrations to the learning agent. The bottom part has questions to the teacher answered after completing teaching. The participants were solicited and compensated through Amazon Mechanical Turk. Each individual was allowed to take part only once in any of our experiments and was compensated with $\$0.25$.

The applet allows interaction with the map to create a demonstration and submit it to the learner. When a square on the map is clicked, the trajectory that starts in this square and follows the optimal policy is plotted on the map. This avoids mistakes by participants in following the optimal policy. In addition, it clarifies the optimal policy that is verbally described to the teacher. A "demonstrate" button on the applet submits the currently displayed demonstration to the learner.

Instructions. The instructions motivate the teaching task with a scenario: Participants are told that a robot will be sent to a foreign planet for a mission and that their goal is to teach the robot how to navigate the terrain on this planet. For the tasks that involve the first map (Fig. 2(a)) they are told that the robot needs to get to the square that has the star. For the tasks that involve the second map (Fig. 2(b)) they are told that the robot needs to get to the square that has the star or the one with the diamond. The individual task descriptions have the following details:

- **Task 1:** Navigating through dark or light gray areas has no difference in terms of consumed battery energy.
- **Task 2:** Navigating through dark gray areas consumes ten times more battery energy.
- **Task 3:** Each of the targets have equal treasures.
- **Task 4:** Diamond has two times more treasures as Star.

For Task 1 and 2 participants are told to choose a single demonstration that will be most informative to the learner. For Task 3 and 4 they are told that they can give more than one demonstration, but that they should try to teach the task with as few examples as possible.

For the two teaching conditions we give the following instructions:

- **Natural:** *Try to choose the most informative paths for the robot.*

	Number of optimal teachers		*Uncertainty*			
	Natural	Guided	Natural	Guided	Optimal	t-test
Task1	3/10	4/10	0.34 (SD=0.25)	0.31 (SD=0.04)	0.16	t(10.51) = 0.41, p=0.69
Task2	0/10	4/10	0.54 (SD=0.16)	0.3 (SD=0.25)	0.0006	t(15.23) = 2.59, p<0.05
Task3	2/10	6/10	0.23 (SD=0.14)	0.11 (SD=0.11)	0.035	t(13.14) = 1.96, p=0.07
Task4	0/10	3/10	0.39 (SD=0.15)	0.24 (SD=0.17)	0.027	t(16.19) = 1.97, p=0.06

Table 1: Results of human subject experiments. First two columns show the number of participants that taught with an optimal sequence. The next two columns show the average uncertainty in the reward estimation achieved by the demonstrations given by the participants. The optimal value is given for comparison. The last column reports t-test results between the two groups.

- **Guided:** *The most informative paths for the robot are those that allow him to see all types of terrains and demonstrate what path to choose when there are multiple viable alternatives. A path is more informative if the alternative paths that start around that location are also good options. If the alternatives are obviously bad, then demonstrating that path is not as informative.*

Note that the instructions are exactly the same for all four tasks. To further motivate good teaching in both conditions, participants are told that an award of $\$4.0$ will be given to the teacher who trains the best learner with the fewest examples.

Evaluation. Our main measure for comparison is the performance of a learner that is trained by the demonstrations given by the participants. For this we use the uncertainty in the learned reward models ($G(D)$). In addition one open-ended question was asked after completing the teaching task, regarding the participants teaching strategy (Natural) or the intuitiveness of the teaching guidance (Guided).

Results

Our human subject experiment was completed by a total of 80 participants (ages between 22-56), 10 in each of the eight conditions. Table 1 presents the results of this experiment. We summarize our observations as follows.

Natural teaching is sub-optimal but spontaneous optimality is possible. Only five out of the 40 people in this condition spontaneously produced optimal demonstrations. The participants' descriptions of how the demonstration was chosen reveals that these were not chosen by chance, but were indeed insightful. For instance, two participants describe their teaching strategy as: "[I tried] to demonstrate a route that contains several different elements the robot may encounter", and "I tried to involve as many crossroads as possible and to use both green and blue tiles in the path." As a result of such intuitions being rare, the performance of the trained learners averaged across participants is far from the optimal values.

Teaching guidance improves performance. From Table 1 we observe that the number of optimal teachers is increased in all four tasks. The uncertainty in the estimation of the rewards is reduced for all four tasks. This shows that the teaching guidance was effective in eliciting mode informative demonstrations. In addition, this shows that our teaching guidance was generic enough, such that it resulted in positive effects across four different tasks. The size of the effect varies across tasks; we see a statistically significant effect only in Task 3, and a positive trend in Tasks 2 and 4, however the difference is insignificant in Task 1. One observed trend is that the task difficulty impacts the usefulness of teaching guidance. If the task is more difficult then the guidance is more helpful for human teachers.

The number of participants who provide an optimal demonstration set is increased in all tasks, however a large portion of the participants are still not optimal. Nonetheless, we see that the demonstrations provided are more informative. For instance in Task 2, only four participants provided the optimal trajectory that starts at the lower dark gray square. However three more participants provided a path that starts in one of the squares at the bottom left corner, which all go through the junction where the agent has the choice between the long light gray path and the short dark gray path. Such demonstrations are also relatively informative.

Although we see an improvement due to teaching guidance, the average uncertainties are still far from the optimal values. Some of the common mistakes causing the sub-optimality were, assuming that the "longest" path would be the most informative (Task 2, 4), giving demonstrations very close to the goal in (Task 1,2), trying to involve obstacles in the path (Task 4), or not providing sufficient demonstrations (Task 3,4). This points towards the challenges in creating intuitive and understandable teaching guidance for users that have no prior knowledge in RL.

Overall, the experiment shows the utility of guiding teachers with instructions based on the proposed optimal teaching algorithm for sequential decision tasks.

Conclusions

This work extends Algorithmic Teaching to sequential decisions tasks. We contribute a novel method for selecting demonstrations optimally while training an Inverse Reinforcement Learner. We present an algorithm that selects demonstration sets, so as to allow the learner to reduce its hypothesis space of possible reward functions, as fast as possible. We provide example outcomes of the algorithm to give an intuition of how it works and demonstrate the learning gains obtained from teaching optimally. Next we present a user study that investigates the potential of using optimal teaching algorithms to improve human teaching in sequential decision tasks. We find that naive humans are in general sub-optimal teachers, but when provided with instructions

describing the intuition of our teaching algorithm, they select more informative examples.

Acknowledgment

Authors would like to thank Dr. Andrea L. Thomaz for their valuable feedback on the drafts of this paper. This work was partially supported by the Conseil Régional d'Aquitaine and the ERC grant EXPLORERS 24007 and the National Science Foundation, under grant number IIS-1032254 and by the Flowers Team (an INRIA/ENSTA-Paristech joint-lab).

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, 1–8.

Angluin, D. 1988. Queries and concept learning. *Machine Learning* 2:319–342.

Anthony, M.; Brightwell, G.; and Shawe-Taylor, J. 1995. On specifying boolean functions by labelled examples. *Discrete Applied Mathematics* 61(1):1–25.

Argall, B.; Chernova, S.; Browning, B.; and Veloso, M. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5):469–483.

Balbach, F. J., and Zeugmann, T. 2009. Recent developments in algorithmic teaching. In *3rd International Conference on Language and Automata Theory and Applications*, LATA '09, 1–18.

Balbach, F. 2008. Measuring teachability using variants of the teaching dimension. *Theoretical Computer Science* 397(1–3):94–113.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Cakmak, M., and Thomaz, A. 2010. Optimality of human teachers for robot learners. In *Proceedings of the IEEE International Conference on Development and Learning (ICDL)*.

Chernova, S., and Veloso, M. 2007. Confidence-based policy learning from demonstration using gaussian mixture models. In *Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Cohn, R.; Durfee, E.; and Singh, S. 2011. Comparing action-query strategies in semi-autonomous agents. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, 1287–1288.

Elman, J. L. 1993. Learning and development in neural networks: The importance of starting small. *Cognition* 48(1):71–9.

Fürnkranz, J., and Hüllermeier, E. 2010. Preference learning: An introduction. *Preference Learning* 1.

Goldman, S., and Kearns, M. 1995. On the complexity of teaching. *Computer and System Sciences* 50(1):20–31.

Grollman, D., and Jenkins, O. 2007. Dogged learning for robots. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*.

Khan, F.; Zhu, X.; and Mutlu, B. 2011. How do humans teach: On curriculum learning and teaching dimension. In *Advances in Neural Information Processing Systems (NIPS)*.

Krause, A., and Guestrin, C. 2005a. Near-optimal nonmyopic value of information in graphical models. In *Uncertainty in AI*.

Krause, A., and Guestrin, C. 2005b. A note on the budgeted maximization of submodular functions.

Lopes, M.; Melo, F. S.; and Montesano, L. 2009. Active learning for reward estimation in inverse reinforcement learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, 31–46.

Natarajan, B. K. 1987. On learning boolean functions. In *19th Annual ACM symposium on Theory of computing*, STOC'87, 296–304.

Nemhauser, G.; Wolsey, L.; and Fisher, M. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14(1):265–294.

Neu, G., and Szepesvári, C. 2007. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Uncertainty in Artificial Intelligence (UAI)*, 295–302.

Neu, G., and Szepesvári, C. 2009. Training parsers by inverse reinforcement learning. *Machine learning* 77(2):303–337.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for inverse reinforcement learning. In *Proc. 17th Int. Conf. Machine Learning*.

Settles, B. 2010. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Shafto, P., and Goodman, N. 2008. Teaching games: Statistical sampling assumptions for learning in pedagogical situations. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, 1–6.

Sutton, R., and Barto, A. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press.

Taylor, M. E. 2009. Assisting transfer-enabled machine learning algorithms: Leveraging human knowledge for curriculum design. In *Proc. AAAI Spring Symposium on Agents that Learn from Human Teachers*.

Viappiani, P., and Boutilier, C. 2010. Optimal bayesian recommendation sets and myopically optimal choice query sets. *Advances in Neural Information Processing Systems* 23:2352–2360.

Warner, R.; Stoess, T.; and Shafto, P. 2011. Reasoning in teaching and misleading situations. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 1430–1435.

Zang, P.; Irani, A.; Zhou, P.; Isbell, C.; and Thomaz, A. 2010. Using training regimens to teach expanding function approximators. In *Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS)*.