MODELLING THE PROBLEM OF LEARNING BY DEMONSTRATION AS A BAYESIAN GAME

BRUNO N. DA SILVA

1. Introduction

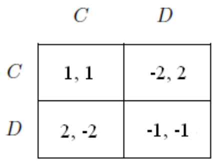
The motivation of our work comes from the field of human teaching by demonstration. In this problem, a robot observes a human demonstrator perform a certain task. Subsequently, this robot will try to perform the task on its own by generalizing over the human executions. Traditionally, this field has concentrated on the simplest scenario where a single demonstrator tries to teach a single robot. However, if we consider a situation where multiple demonstrators are available, then one might imagine a more efficient learning algorithm where the computational agent is able to collect execution samples from multiple data sources. This might alleviate the burden on the task experts, who would now be able to share the tedious task of executing the task multiple times, and might also contribute to an improved robot behavior, since by observing multiple demonstrators the robot is more likely to observe more diverse behavior that explores different areas of the problem space.

Unfortunately, this diversity in behavior might at the same time be detrimental to the learning algorithm. If we considered that multiple demonstrators would present different executions to the agent, we might as well accept that these demonstrators will have different perspectives on the task and about the robot execution. In particular, humans might disagree about preference relations over final outcomes, or even about different strategies that would achieve the same outcome but under different risk profiles. Therefore, the robot performance after learning the task might be strongly dependant on how well it collects and integrates the data from multiple disagreeing demonstrators.

As an inspiring example, imagine a married couple who tries to teach a robot how to drive their kids to school. More specifically, one of the human agents has a very aggressive driving style, while the other is too passive. Unfortunately, no individual driving profile can be singled out as better than the other. While in some cases a passive approach will diminish the risk of exposing the passengers to accidents, there may be situations where there is room for a more aggressive (i.e. less defensive) course of action that won't increase the likelihood of accidents by much, while incurring in a considerable decrease in the duration of the ride.

Coming back to our abstract scenario, if we observe the problem from the demonstrator's perspective, additional insights might be available. If the robot is to learn solely from the demonstrations of the humans, and if the humans have different preference relations over the robot's behavior, then the task demonstration becomes a strategic tool for the human. The problem is that different demonstration

Date: April 22, 2009.



	C	D
C	1, 1	-2, 2
D	2, -2	-1, -1

FIGURE 1. Learning from multiple demonstrators as a Prisoner's Dilemma game.

styles will be combined with the demonstration from other humans to lead to different robot behaviors. Consequently, if demonstrators care about having a robot whose behavior is similar to their own, they might exploit the mechanism and think about demonstrating in a malicious way in order to have robots that resemble their idiosyncratic behavior as much as possible. Therefore, instead of having demonstrations that show how to perform a task, we will have demonstrations that will try to neutralize other demonstrations and highlight some arbitrary idiosyncratic behavior.

Unfortunately, the different task conditions (states of the world) would dictate the right idiosyncratic demonstration to follow. And since the robot is learning from the demonstrations, it might be overwhelming (or even impossible) to demand that the robot evaluates the nature of the human demonstration. Consequently, if the robot collects the demonstrations naively, the underlying game played by two demonstrators would look exactly like the Prisoner's Dilemma (Figure 1). Here, Cooperate (C) means to demonstrate honestly and reasonably mindful, whereas Defect (D) means demonstrate in order to overshadow the other demonstrator. If one agent demonstrates honestly (trying to avoid their own vices, and reflecting on what the right action for the robot might be), while her parter replicates all his mindless behavior, then the robot will be biased towards the behavior of the second agent, making the first one worst off. Naturally, agents will prefer the behavior that looks like their own, and so defecting is a dominant strategy in this game.

Unfortunately, this outcome is very bad for the robot learner. Now, instead of efficiently collecting valid data from multiple sources, the robot is collecting demonstrations that are uninformative of how to perform the task. Even demonstrators who would routinely execute the task similarly (demonstrators of *similar types*) would produce useless data. Therefore, a mechanism that will induce honest demonstrations is in order for this problem.

In this paper, we will study one such mechanism. Since we cannot tell *a-priori* how players would execute the task (otherwise, we wouldn't need to collect the demonstrations), we will model this problem as a Bayesian game. But first, we will present a brief review of the related literature in Section 2. Next Section 3 will introduce a more concrete representation of the learning by demonstration scenario and Section4 presents our model. Finally, we conclude in Section5.

2. Related Literature

In this section, we will briefly review some of the literature that used Bayesian games to model solutions in Artificial Intelligence. In the ARMOR system [5], security is presented as an important motivation for using models including Bayesian games in Artificial Intelligence models. In particular, the authors present their analysis in the context of Stackelberg games. In this specific class of games, we have two player who take their action in sequence. First, a leader makes an initial move, and then a follower observes the move from the first player and takes an action. These games provide a very natural model for many security situations, where usually a defender secures an environment, and subsequently an attacker decides whether and how to attack the environment. In the ARMOR system, a software agent acts as an assistant to the Los Angeles International Airport to place security checkpoints and other resources in order to minimize threats to the facility. In order to produce responsive results to eminent threats, this system overcomes the inherent complexity of Bayesian games [1] by implementing a fast optimal algorithm for Bayesian Stackelberg games [4] under the assumption that the set of types for the leader player is a singleton.

One different use of Bayesian games is presented in [3]. This paper deals with the problem of planning in multiagent systems. One solution model for decentralized robot teams is the class of Partially Observable Stochastic games (POSG). This is a generalization of Stochastic games but where the robot cannot directly observe either the actions executed by other agents but also the state of the world it is in. Therefore, a belief over different world states allow for the characterization of the Stochastic games. The main contribution of this paper lies in overcoming the intractability of POSG by decomposing smaller sequences of states into a Bayesian game. For this Bayesian games, they are able to estimate utilities using domain-specific utility functions and the types of each players become the sequences of states of the original POSG. The agents can then coordinate after solving the reduced Bayesian games and limit the amount of communication necessary for planning and execution.

3. The Learning from Demonstration Mechanism

In this Section, we will very briefly introduce a concrete mechanism for collecting task demonstrations from humans. A detailed description of this mechanism is presented in [2]. The task demonstration will occur in two stages. In the first stage, each demonstrator will independently provide the robot with multiple task executions. The robot then collects these independent sets of executions and will store all of them in a (very heterogeneous) knowledge base. This stage is depicted in Figure 2.

After this data collection, the robot will generate an initial policy for the datapoints in the knowledge base. One way to understand this knowledge base is as a sequence of (p_m, a_n, c), where p_m is the percepted state of the world at the time an action a_n was executed by the demonstrator. And c is the confidence that a_n is the best action to be followed under the state p_m. This confidence is first set to an initial value and should be updated in the second stage of the mechanism.

The holistic objective of the second stage (Figure 3) is to update the confidence levels of the datapoints in the knowledge base. In this second step of this mechanism, the robot will employ the policy induced from the knowledge base and will execute the task repeated times under the supervision of the humans. After each of these task executions, the robot will collect criticisms to its execution from the humans and will adapt future executions based on these criticisms. More concretely, if the critique $feedback$ from demonstrator d_i praises/scorns the execution,

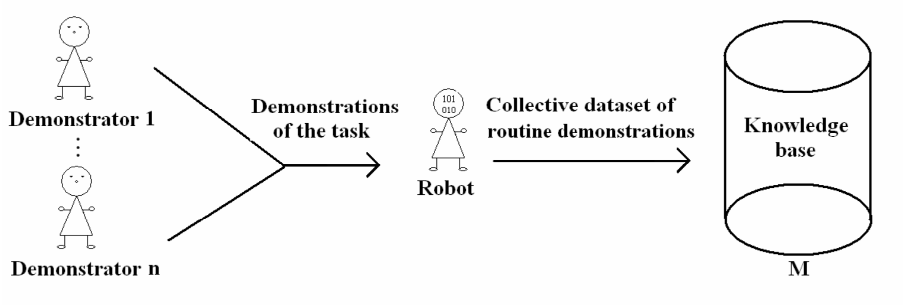

Demonstrator 1
Demonstrator n
Demonstrations of the task
101
010
Robot
Collective dataset of routine demonstrations
Knowledge base
M

FIGURE 2. The first stage of the Learning from Demonstration Mechanism.

the robot will update the confidence c of those datapoints associated with the execution as $c := c + r_i * f(feedback)$. First, $f(feedback)$ simply returns $+1/-1$ for praises/scorns. Second, r_i is the reputation level of demonstrator d_i. This is a variable that will measure how past critiques from d_i have improved the robot policy in the past.

Finally, we must explain how the mechanism updates the values of r_i after each task execution of the second stage. After collecting all the criticisms, the robot evaluates each of them in isolation in order to estimate the effectiveness of the criticism. In this evaluation, the robot will sample different task scenarios that are affected by the newly submitted criticism. We call this procedure $evaluation(feedback)$, and it returns a value proportional to the improvement from $feedback$. After computing this function, we can therefore update the reputation levels as $r_i := max(0, min(1, r_i + \alpha * evaluation(feedback)))$. This formula makes sure that r_i remains within the interval $[0, 1]$.

In order to have meaningful influence on future executions of the robot, each demonstrator will have to try to maintain as high a reputation level as possible. Therefore, if we have a good heuristic function $evaluation()$, we will be able to collect mindful (honest) critisms from the agents in this second stage of the mechanism.

4. A Bayesian Game Model

In this section, we will introduce a model of the Learning from Demonstration problem described above. We therefore must describe the behavior of agents in the second stage of our mechanism. For this scenario, we will adopt a Bayesian game setting, and will adopt a few simplifying assumptions. We will restrict ourselves to the case where we have two agents, and each agent has two types, θ_i = agressive or θ_i = passive. We characterize the set of actions for each agent as A_i = Critiquing mindfully, Critiquing mindlessly. Distributions over types are given by $P(\theta_i, \theta_j)$.Critiquing mindlessly represent the action where the agent does not reflect on their routine behavior and transfers her idiosyncrasies to the robot through the critique. Analogously, critiquing mindfully represents the case where the agent actually reflects on their critique and is able to detach and provide a thoughtful critique to the demonstration.

Figure 4 presents the payoff distributions of this game for each type of each player. These values were chosen in order to emulate the outcomes of each possible

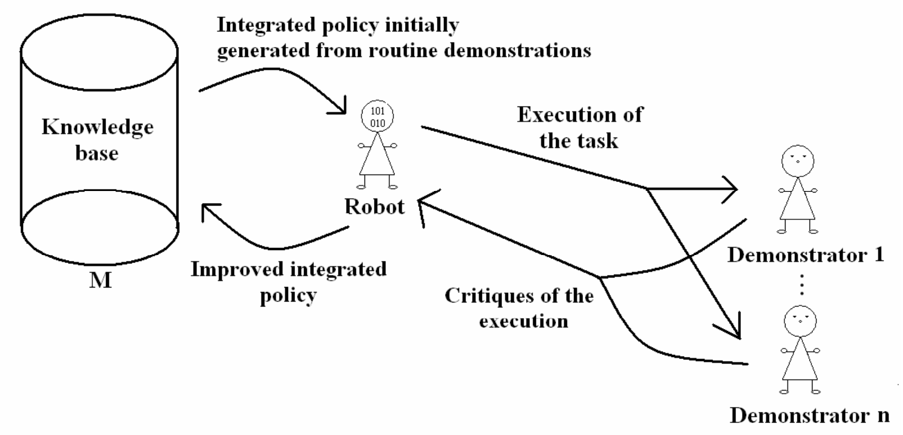
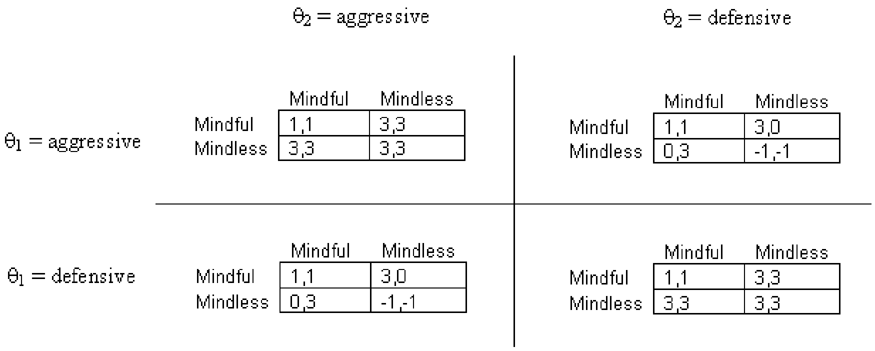


FIGURE 3. The second stage of the Learning from Demonstration Mechanism.



		θ_2 = aggressive		θ_2 = defensive	
		Mindful	Mindless	Mindful	Mindless
θ_1 = aggressive	Mindful	1,1	3,3	1,1	3,0
	Mindless	3,3	3,3	0,3	-1,-1
θ_1 = defensive	Mindful	1,1	3,0	1,1	3,3
	Mindless	0,3	-1,-1	3,3	3,3

FIGURE 4. The game.

approach to the critiquing stage. Consequently, if both agents choose to produce mindful critiques, then both derive utility for constructing a policy that will execute the task in a neutral but mindful manner. However, when players are of opposing type and they both produce mindless criticisms, then they will be providing bimodal biased corrections to the robot. This will only maintain conflicting datapoints in the knowledge base from the first stage of the mechanism, and therefore, they will enjoy neither a robot that performs the task rightfully, nor a robot that performs the task in the particular way they behave. Still in the case where an agent i faces an adversary with type $\theta_j \neq \theta_i$, if he chooses to provide mindful criticisms while j starts providing mindless feedback, then the initial noise introduced by j will be easily compensated by i who will possess sole influence over the robot's demonstrations in subsequent stages of the game. However, if both agents are of the same type, it doesn't really matter which agent loses influence over the agent because after all they would provide similar feedback to the robot's demonstrations.

5. Conclusions

In this study, we presented a Bayesian game model of a Leaning from Demonstration scenario. There are several directions for future research. One *ex-post* equilibrium of this game is both agents of the same type to provide meaningless criticisms. Therefore, it would be interesting to study a signaling game where agents would try to identify or learn which is the type of the other agents, in order to figure out how to converge to an optimal critiquing strategy. Another possibility would be to actually challenge the mechanism introduced in Section 4 and verify what are the different equilibria for this repeated game.

Computer Science Department, University of British Columbia
E-mail address: bnds@cs.ubc.ca

REFERENCES

[1] Conitzer, V., and Sandholm, T. 2006. Computing the Optimal Strategy to Commit to. In *Proceedings of the ACM Conference on Electronic Commerce* (ACM-EC), 82-90

[2] da Silva, B. 2009. Learning from Disagreeing Demonstrators. In Proceedings of the AAAI Spring Symposium Agents that Learn from Human Teachers, 36-39

[3] Emery-Montemerlo, R., Gordon, G., Schneider, J., Thrun, S. 2005. Game Theoretic Control for Robot Teams. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, 1175—1181

[4] Parachuri, P., Pearce, J., Marecko, J., Tambe, M., Ordóñez, F., Kraus, S. 2008. Playing Games for Security: An Efficient Exact Algorithm for Solving Bayesian Stackelberg Games. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 895-902

[5] Pita, J., Jain, M, Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., Kraus, S. 2008. ARMOR Security for Los Angeles International Airport. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 1884-1885