Exploring Teachable Humans and Teachable Agents: Human Strategies Versus Agent Policies and the Basis of Expertise

John Stamper(✉) and Steven Moore

Carnegie Mellon University, Pittsburgh, PA 15213, USA
jstamper@cs.cmu.edu, stevenjamesmoore@gmail.com

Abstract. In this research, we explore how expertise is shown in both humans and AI agents. Human experts follow sets of strategies to complete domain specific tasks while AI agents follow a policy. We compare machine generated policies to human strategies in two game domains, using these examples we show how human strategies can be seen in agents. We believe this work can help lead to a better understanding of human strategies and expertise, while also leading to improved human-centered machine learning approaches. Finally, we hypothesize how a continuous improvement system of humans teaching agents who then teach humans could be created in future intelligent tutoring systems.

Keywords: Policies · Agents · Strategies · Expertise

1 Introduction

In this research, we explore how AI agent policies might be used to teach humans. In complex tasks humans generate strategies which can be applied in many different situations. Combinations of strategies that lead to optimal outcomes can lead to expertise in a domain, although there is still no consensus among researchers as to what makes a person an expert and how expertise is defined. We explore the interactions of policies and strategies, looking at how both relate to expertise. Our long term goal is to see how humans can help teach agents and agents can help teach humans in a continuous loop. A start to this goal is a comparison of agent policies, generated with different techniques on several complex game domains, with strategies generated from human players.

2 Background and Domains

Expertise has been the subject at the crossroads of Psychology and Computer Science for some time. *The Nature of Expertise* [14] explored a wide variety of domains from human typing to sports to ill-defined domains. A key insight from this work is that in the early development of AI systems, expertise was tightly related to the concept of encoding human strategies into machines, such as early work involving chess players and intelligent tutors [4]. As work continued, the Psychology field moved into

© Springer Nature Switzerland AG 2019
S. Isotani et al. (Eds.): AIED 2019, LNAI 11626, pp. 269–274, 2019.
https://doi.org/10.1007/978-3-030-23207-8_50

architectures of cognition defined by ACT-R [1] and Soar [18] as examples. Computer Science moved towards agents and policy creation focusing early on reinforcement learning [29] and now advanced techniques built on deep learning [19].

The question of what exactly defines someone as an expert is still an open question and has a lot to do with the particular domain that is being studied. In chess, Chase and Simon posited that it takes 10,000 h of study to become an expert [4]. That number has also been suggested as the rough number of hours to become an expert musician [11] and is a general theory of expertise [10], although largely due to Simon's chess work. In the case of learning systems, we often define mastery using some form of knowledge tracing. These systems often set "mastery" as a probabilistic value that a learner knows a particular skill. The value of mastery varies on skills and domains, but a value of 90% or 95% are assumed to have achieved mastery [8]. Understanding skills that are used to solve problems has also been explored in many domains [16, 25]. Tasks to elicit knowledge from experts, such as cognitive task analysis (CTA) have been used by cognitive scientists to better understand the strategies that experts use, but may not explicitly recognize [6].

AI has been used now for decades to create agents that mimic human behavior. These agents are generally driven by a policy created by some form of machine learning, such as Q-learning [29]. The policy tells the AI agent what to do given a certain set of conditions. This is most often defined as a state-action graph that suggests the best possible next action for an agent assigned to a given state. In education, agents driven by policies have long been a foundational part of intelligent tutors and adaptive learning. Work has been done in modeling learning as a policy generated to predict what a student knows and what the next best instructional lesson is for a particular student [24]. Other research has been done using reinforcement learning (RL) with a focus on what pedagogical action would be best to use for a student when multiple actions are available [5]. Most closely associated with the research we are doing is work on the automatic generation of hints and feedback [23, 27]. This work uses state graphs and RL to identify the best path for solving problems. Then generates a just in time hint or provides feedback that can lead the student down a better path for learning.

We focus on two complex game domains: **connect four** (C4) and **Space Invaders** (SI). Both are well known games and chosen because of their simplicity of play and known human strategies for winning. They also have multiple agent implementations that we can exploit, which are explained in detail below.

The objective of C4 is to align four game pieces of the same color in a row, either diagonally, horizontally, or vertically. There are three possible states for each of the forty-two available game spaces. The board spaces can be occupied by the turn players piece, the opponent's piece, or it can be empty. This means there are 3^{42} ($\geq 10^{20}$) moves possible on the game board of seven columns by six rows, ranging from zero to forty-two pieces on it. Using binary decision diagrams, it has been shown there are exactly 4,531,985,219,092 legal board configurations [9]. Additionally, C4 is a solved zero-sum game, of moderate complexity, where the outcome of the game can correctly be predicted from any state [31]. There are many variants of C4 agents [12, 13]. Recent agents that solve the game using temporal difference learning, achieved a win percentage close to perfect, but require several millions of self-play games for training, thus being far off human performance [2]. Another study found that using 1,565,000 games for training data, their agent could reach an 80% success rate, but it required between 2–4 million games to produce what would be considered a strong-playing one [30]. The most successful agents of C4 make use of the MiniMax algorithm, which consists of heuristic evaluation function that is akin to these human strategies. It is often cited as the standard to compare different agent implementations against, as MiniMax can win virtually every time, depending on its search depth, with no training data required [30]. This is powerful since C4 is a zero sum game, and the heuristic function has the agent follow a set of optimal human-like strategies. The evaluation function can be summarized by five strategy points: (1) If there is a winning move, take it (2) If the opponent has a winning move, prevent it (3) Take the center square over edges and corners (4) Take corner squares over edges (5) Take edges if nothing else is available.

Just implementing a simple human strategy can have a profound effect on the size of the agents search space and number of game plays needed to generate an expert agent. For example, one basic strategy is when given the opportunity to go first, a player should always take the center position on the board, and if going second the player should take this position if available. From a simple computation we can see that this prunes 6 of the 7 high level branches in the initial graph leading to tremendously less possible game states in the expert player.

Space Invaders was a classic arcade game and one of the games available in the Atari Grand Challenge dataset (AGC) [17] based on the classic Atari 2600 home console game system. In dataset-1 of the AGC, there are 445 human game plays of SI. SI also represents a potentially easier game to follow in the Atari game space because the game dynamics remove some of the available moves. While Atari games allow for the use of four directional movements (left, right, up, down) plus a button, SI only allows the player to move left or right and use the button to shoot. This limits the complexity of this game compared to some others.

There are a number of human strategies that we have discovered from discussions with an expert of the game. This expert was able to achieve scores greater than 98% of all human players as reported by the Atari Grand Challenge site. The human strategies include (1) because only one shot can be on the screen at a time shooting lower invaders leads to faster shooting, (2) shooting entire columns from the left side first give additional time because of the right to left movement of the invaders, and (3) when the invaders reach the left side and begin moving right shoot the bottom row and move to shooting the rightmost column. These strategies keep the invaders largely in a square formation. It is disadvantageous to split the invaders into two squares, because that requires additional movement to get a shot off.

Using the Arcade Learning Environment (ALE) [3], agents have been created and trained to play Space Invaders. A summary of the scores of three agent based on different algorithms in a replication study of a number of previously built agents in the ALE framework [21] claimed agents did exceed human capabilities at times, although they did not average a score that was higher than the top 5% of human players presented in the AGC dataset. When we looked at data from the DQN agent, which was driven by a deep learning algorithm, visualizing the RAM states based on a t-SNE embedding [22] shows that many of the clusters do show evidence of human strategies, such as keeping the invaders in a single square formation. Watching replays of expert agent players also shows expert human strategies, but more work is needed to delve into the actual policy to find clear evidence of a particular strategy.

3 Discussion and Conclusions

Heuristic driven policies, by means of a given evaluation function, are widely used to solve games such as chess, C4, Othello and Go [7]. The evaluation functions in these agents use information about the game. Much of this is directly related to a strategy that a human player would follow, as addressed previously with C4. These strategies represent expertise in a human player and are clearly identifiable in agent play. In the development of agents, it is the human encoding the strategy into the AI using their knowledge of the game. The majority of game-playing agents, however, make use of deep neural nets to develop their policies, which makes them black-box and often difficult to interpret by a human. Recent work has looked at making policies developed this way programmatically interpretable, but much works remains for humans to be able to clearly articulate what many of these agents have learned from their training [32].

It is debatable if these deep reinforcement learning agents make use of explicit strategies as they execute their given policies. A recent approach uses saliency maps to highlight key decision regions for agents playing Atari 2600 games, and found that their SI agent learned a sophisticated aiming strategy [15]. Another way to make policies less black-box, is to break the policy down into smaller subtasks that are comprised of a few actions that feed back into the overall policy [20]. These techniques of breaking down policies into smaller interpretable strategies and visually representing the mechanisms of an agent's policy are steps toward having humans learn strategies from agents, without directly encoding any into the agent itself.

Some previous work looks to use human seeding of policies in educational domains [26]. Another such study found that training on human data; they could achieve comparable scores to state-of-the-art reinforcement learning techniques and even beat the scores using just the top 50% of their collected data for more complicated games [17]. Combining a method that not only trains agents on expert human data, but also encodes their strategies into a form of an evaluation function has the potential to yield successful agents that require less computational time, while performing at greater levels than comparable agents.

We can identify human strategies in the policies generated by agent through post hoc human inspection. In the future, we will explore how to automate the process of identifying strategies within the agent policies similar to previous work on less complex educational domains [28]. This will require progress on explainable AI to extract human readable information from increasingly black-box policies. We plan to explore a number of additional domains where data and agents are available for study.

References

1. Anderson, J.R., Matessa, M., Lebiere, C.: ACT-R: a theory of higher level cognition and its relation to visual attention. Hum.-Comput. Interact. **12**(4), 439–462 (1997)
2. Bagheri, S., Thill, M., Koch, P., Konen, W.: Online adaptable learning rates for the game Connect-4. IEEE Trans. Comput. Intell. AI Games **8**(1), 33–42 (2016)
3. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: an evaluation platform for general agents. J. Artif. Intell. Res. **47**, 253–279 (2013)
4. Chase, W.G., Simon, H.A.: Perception in chess. Cognit. Psychol. **4**(1), 55–81 (1973)
5. Chi, M., VanLehn, K., Litman, D., Jordan, P.: An evaluation of pedagogical tutorial tactics for a natural language tutoring system: a reinforcement learning approach. Int. J. Artif. Intell. Educ. **21**(1–2), 83–113 (2011)
6. Clark, R.E., Estes, F.: Cognitive task analysis for training. Int. J. Educ. Res. **25**(5), 403–417 (1996)
7. Clune, J.: Heuristic evaluation functions for general game playing. In: AAAI, vol. 7, pp. 1134–1139, July 2007
8. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. User Model. User-Adap. Interact. **4**(4), 253–278 (1994)
9. Edelkamp, S., Kissmann, P.: Symbolic classification of general two-player games. In: Dengel, A.R., Berns, K., Breuel, T.M., Bomarius, F., Roth-Berghofer, T.R. (eds.) KI 2008. LNCS (LNAI), vol. 5243, pp. 185–192. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85845-4_23
10. Ericsson, K.A., Smith, J. (eds.): Toward a General Theory of Expertise: Prospects and Limits. Cambridge University Press, Cambridge (1991)
11. Ericsson, K.A., Prietula, M.J., Cokely, E.T.: The making of an expert. Harvard Bus. Rev. **85** (7/8), 114 (2007)
12. Faußer, S., Schwenker, F.: Neural approximation of monte carlo policy evaluation deployed in connect four. In: Prevost, L., Marinai, S., Schwenker, F. (eds.) ANNPR 2008. LNCS (LNAI), vol. 5064, pp. 90–100. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69939-2_9
13. Ghory, I.: Reinforcement learning in board games. Department of Computer Science, University of Bristol, Technical report 105 (2004)
14. Glaser, R., Chi, M.T., Farr, M.J. (eds.): The Nature of Expertise. Lawrence Erlbaum Associates, Hillsdale (1988)
15. Greydanus, S., Koul, A., Dodge, J., Fern, A.: Visualizing and understanding atari agents. arXiv preprint arXiv:1711.00138 (2017)
16. Koedinger, K.R., Stamper, J.C., Leber, B., Skogsholm, A.: LearnLab's datashop: a data repository and analytics tool set for cognitive science. Top. Cognit. Sci. **3**(5), 668–669 (2013)
17. Kurin, V., Nowozin, S., Hofmann, K., Beyer, L., Leibe, B.: The atari grand challenge dataset. arXiv preprint arXiv:1705.10998 (2017)
18. Laird, J.E., Newell, A., Rosenbloom, P.S.: SOAR: an architecture for general intelligence. Artif. Intell. **33**(1), 1–64 (1987)
19. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
20. Lyu, D., Yang, F., Liu, B., Gustafson, S.: SDRL: interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. arXiv preprint arXiv:1811.00090 (2018)

21. Machado, M.C., Bellemare, M.G., Talvitie, E., Veness, J., Hausknecht, M., Bowling, M.: Revisiting the arcade learning environment: evaluation protocols and open problems for general agents. arXiv preprint arXiv:1709.06009 (2017)
22. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518** (7540), 529 (2015)
23. Moore, S., Stamper, J.: Decision support for an adversarial game environment using automatic hint generation. In: International Conference on Intelligent Tutoring Systems (ITS 2019). Springer, Heidelberg (2019, to Appear)
24. Rafferty, A.N., Brunskill, E., Griffiths, T.L., Shafto, P.: Faster teaching by POMDP planning. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS (LNAI), vol. 6738, pp. 280–287. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21869-9_37
25. Stamper, J., et al.: PSLC DataShop: a data analysis service for the learning science community. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6095, p. 455. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13437-1_112
26. Stamper, J., Barnes, T., Croy, M.: Enhancing the automatic generation of hints with expert seeding. Int. J. AI Educ. **21**(1–2), 153–167 (2011)
27. Stamper, J., Barnes, T.: Unsupervised MDP value selection for automating ITS capabilities. In: Educational Data Mining 2009, pp. 180–188 (2009)
28. Stamper, J.C., Barnes, T., Croy, M.: Extracting student models for intelligent tutoring systems. In: Proceedings of the National Conference on Artificial Intelligence, vol. 22, no. 2, p. 1900. AAAI Press, Menlo Park, CA. MIT Press, Cambridge (1999, 2007)
29. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2018)
30. Thill, M.: Temporal difference learning methods with automatic step-size adaptation for strategic board games: Connect-4 and Dots-and-Boxes. Doctoral dissertation, Master thesis, Cologne University of Applied Sciences, June 2015
31. Tromp, J.: Solving Connect-4 on medium board sizes. ICGA J. **31**(2), 110–112 (2008)
32. Verma, A., Murali, V., Singh, R., Kohli, P., Chaudhuri, S.: Programmatically Interpretable Reinforcement Learning. arXiv preprint arXiv:1804.02477 (2018)