2018

Simulating Human-AI Collaboration with ACT-R and Project Malmo

Zachary M. Brill
Bucknell University, zmb004@bucknell.edu

Christopher L. Dancy
Bucknell University, cld028@bucknell.edu

Recommended Citation

Brill, Z. M., & Dancy, C. L. (2018). Simulating Human-AI Collaboration with ACT-R and Project Malmo. In proceedings of the 16th International Conference on Cognitive Modeling, Madison, WI: University of Wisconsin, 9-10.

Simulating Human-AI Collaboration with ACT-R and Project Malmo

Zachary M. Brill (zmb004@bucknell.edu), Christopher L. Dancy (christopher.dancy@bucknell.edu)
Department of Computer Science, Bucknell University
701 Moore Ave, Lewisburg, PA 17837 USA

Keywords: ACT-R; Project Malmo; teamwork; human-AI interaction; cognitive architectures, UCT.

Introduction

We use the ACT-R cognitive architecture (Anderson, 2007) to explore human-AI collaboration. Computational models of human and AI behavior, and their interaction, allow for more effective development of collaborative artificial intelligent agents. With these computational models and simulations, one may be better equipped to predict the situations in which certain classes of intelligent agents may be more suited to collaborate with people. One can more tractably understand and predict how different AI agents affect task behavior in these situations. To simulate human-AI collaboration, we are developing ACT-R models that work with more traditional AI agents to solve a task in Project Malmo (Johnson et al., 2016). We use existing AI agents that were originally developed as the AI portion of the Human-AI collaboration. In addition, creating a model in ACT-R to simulate human behavior gives us the opportunity to play out these interactions much faster than would be possible in real time.

Malmo Collaborative Challenge

The Collaborative AI Challenge was designed by Microsoft to test the collaborative capabilities of artificial intelligence. Built on top of Minecraft to create various environments and agents, Project Malmo is an attractive platform for experimenting with AI agents. Microsoft challenged teams to design and implement an artificial agent capable of playing alongside a human teammate in a game of Pig Chase, which is an extension of the Stag Hunt task developed by Yoshida, Dolan, and Friston (2008). This game requires a two-member team to track down and catch a pig within an enclosed meadow. Having only a limited number of actions (moving one square takes one action), cooperation is key to success. Cornering or "pinching" the pig with no escape route provides the maximum points to each agent for that round. However, the game also allows for a different kind of reward to be earned. If an agent gives up and exits the pen, they will also receive points, albeit fewer than if they had caught the pig.

The AI Agent

The primary AI agent, created by the Bacon Gulch team (Garriga, 2017), uses Bayesian inference and a planning algorithm to 1) determine the other player's approach and 2) plan its next move. The Bacon Gulch agent's Bayesian inference identifies the other player's strategy as either 'focused' or 'random'. The actions taken by the partner determine this inferred state. A 'focused' state is inferred when the teammate appears to share the same pig-catching goal as our artificial agent. If no clear movement towards the pig is being made, perhaps the teammate is standing in one spot spinning in circles, we assign their state to be 'random', and promptly exit the pig pen, acquiring a small reward. Using Bayes' theorem, we see the probability of action *a* being performed given the agent is in state *t* $P(a \mid t)$. With a strategy established, the AI agent must plan its next move, taking it closer to the goal state of capturing the pig.

UCT and Planning

The planning of moves is carried out by a domain-adapted variant of the Upper Confidence bounds for Trees (UCT) algorithm (Kocsis & Szepesvári, 2006). This variation on the Monte Carlo Tree Search (MCTS) algorithm shines in a simple task, such as the Pig Chase, where there is not sufficient complexity to require a learning algorithm (i.e., the playing field is small enough that all moves can be considered in a reasonable amount of time). The UCT algorithm builds a search tree that has as its root the initial state. This initial state is always the agent's current square. From this initial state, stochastic trajectories are simulated. A stochastic simulation is desired because it diminishes the detrimental effects of the pig's random movement on the agent's move planning, and allows for a better exploration of possible moves, since any valid move can be considered. A trajectory would be represented by a path in the tree, from which we could sum all the reward values along that path. This determines our expected outcome of following said trajectory. The path with the highest reward is chosen.

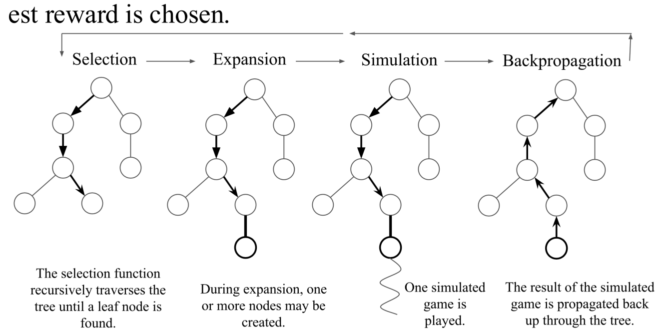

Figure 1: The Monte Carlo Tree Search algorithm procedure (based on Chaslot et al., 2008)

Figure 1 provides a general illustration of how the UCT algorithm works. Important to our research, is the simulation stage as previously mentioned. From this simulation, the final outcome, i.e. the score for the simulated round is backpropagated through the tree, allowing the agent to decide its best move given the current state.

Simulating the Human with ACT-R

We are developing a cognitive model of the human acting as part of a team in this task. We use these models to simulate and predict how differences in AI agents, and human cognitive states, may affect performance in human-AI collaboration. The declarative module houses all factual knowledge about the game's current state, primarily locations of all relevant objects e.g. 'me', my 'teammate', and the pig. ACT-R's procedural system (including utility) is used to complete actions. Lastly, we use ACT-R's motor module to make keystrokes, thus enabling the agent to play the game using perceptual-motor mechanisms similar to a human.

Model Strategy Overview

Previous work has shown that humans model their teammates as having similar decision-making processes to their own (Kennedy et al., 2008). In Kennedy et al.'s work, this projection of strategy improved the performance of a human-robot team. Similarly, our initial approach involves mirroring the AI agent's strategy within our model. This design choice will prove important in our analysis, enabling us to show shared strategies as a predictor of performance. This is the basis of our ACT-R model. Moreover, the model will grow to possess gameplay strategies that differ from the AI's, expanding our ability to analyze the effect of strategy similarity on the scores. Figure 2 details a high-level view of the model's approach to playing the game.

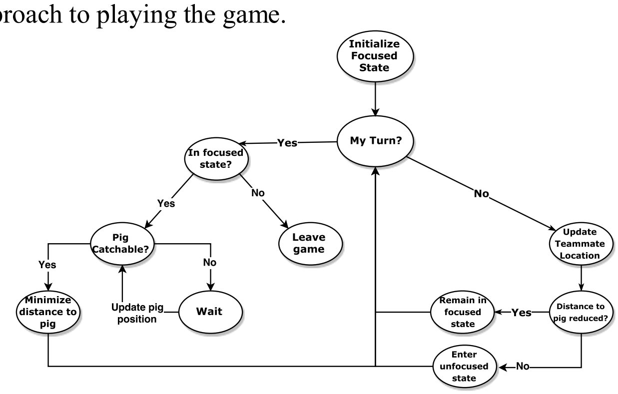

Figure 2: Initial model strategy. This strategy is designed to be as similar as possible to the AI's.

While many of the actual implementations within ACT-R are abstracted out of this flowchart, the visualization of this strategy shows the initial focused state and the ways in which the ACT-R agent not only makes its own moves, but how it responds to the moves of its teammate.

Foundations of the Model

The foundation of our model's strategy is an assumption of shared goals. To achieve the highest possible scores in the pig-chase game, both agents must attempt to catch or corner the pig. Thus, our model initializes itself with the assumption that its teammate will always attempt to reduce its distance to the pig. Our model identifies this as a focused state, the same state that is inferred by the AI agent. This state decision is implemented using a simple algorithm in which our agent pays attention to the teammate's movements. If, during the AI agent's turn, a negative move is made, i.e. their distance to the pig is increased, our agent determines them to be in an unfocused state, analogous to the AI agent's 'random' state. This unfocused state determination then shifts our agent's goal towards leaving the pen and earning itself a (small) reward. However, some adjustments to the model's goal change transition time would be valuable, as we could analyze the value of quitting and exiting immediately or waiting some amount of moves to allow the AI agent to recover a focused state.

Conclusion

The modelling of human behavior in a teamwork environment gives us the opportunity to not only add to previous research focused on human-machine interaction, but also contribute to our understanding of the aspects of human cognition that enable effective collaboration. Our primary goal is to observe the impact a shared strategy between teammates has on game performance. Using ACT-R and Project Malmo allows potential future expansion to more complex collaboration and environments, and the simulation of physiological and affective modulation of human-AI collaboration (e.g., using ACT-R/Φ, Dancy, 2013); both of these are important for the understanding of the consequences of integrating AI systems in different environments.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York, NY: OUP.

Chaslot, G. M. J. B., Winands, M. H. M., Uiterwijk, J. W. H. M., van der Herik, H. J., & Bouzy, B. (2008). Progressive strategies for monte-carlo tree search. *New Mathematics and Natural Computation, 4*(3), 343.

Dancy, C. L. (2013). ACT-RΦ: A cognitive architecture with physiology and affect. *Biologically Inspired Cognitive Architectures, 6*(1), 40–45.

Garriga, A. malmo-challenge, (2017), GitHub repository, https://github.com/rhaps0dy/malmo-challenge

Johnson, M., Hofmann, K., Hutton, T., & Bignell, D. (2016). The Malmo platform for artificial intelligence experimentation. *In proceedings of the Twenty-Fifth International joint conference on artificial intelligence (IJCAI)*, New York, NY, 4246-4247.

Kennedy, W. G., Bugajska, M. D., Adams, W., Schultz, A.C., & Trafton, J. G. (2008). Incorporating mental simulation for a more effective robotic teammate. *In proceedings of the Twenty-Third Association for the Advancement of Artificial Intelligence conference on artificial intelligence (AAAI),* Chicago, IL, 1300-1305.

Kocsis, L., & Szepesvári, C. (2006, 2006//). Bandit based monte-carlo planning. *In proceedings of the 17th European Conference on Machine Learning (ECML)*, Berlin, Heidelberg, 282-293.

Yoshida, W., Dolan, R. J., & Friston, K. J. (2008). Game theory of mind. *PLOS Computational Biology, 4*(12), e1000254.