

Memory and Analogy in Game-Playing Agents

Jonathan Rubin and Ian Watson

Department of Computer Science
University of Auckland, New Zealand
<http://www.cs.auckland.ac.nz/research/gameai>
jrub001@aucklanduni.ac.nz, ian@cs.auckland.ac.nz

Abstract. We present our views and ideas about a possible approach to general game playing by utilising memory and analogy. We initially discuss the importance of memory in game playing agents. The forms that memory can take are examined and examples of successful agents who utilise memory are presented. Following this we focus on *experience-based*, *lazy learners* and justify why we believe they may be beneficial in a general game playing domain. Analogical reasoning is then introduced and its benefits considered. We conclude by formulating some example analogies and speculating how an experience-based, lazy learner could apply these to a general game playing environment.

1 Introduction

In this position paper we wish to describe the possibility of a lazy learning agent used to play multiple, arbitrary games using memory and analogy. Memory refers to the concept of storing scenarios in a database or case-base to represent an agent's knowledge. Analogy refers to the ability to recognise similarities between separate problem domains and to generalise solutions from one domain to another. Through examples and previous research we will attempt to outline how this system could be constructed and the possible benefits of this approach.

The following sections begin with a summary of the types of memory an agent has available and various systems that have achieved success through the use of memory. Followed by a review of experience-based, lazy learners that have been developed to play specific games such as chess, checkers and poker. Finally, analogy is discussed where an attempt is made to generalise memories an agent holds about a specific game to aid it in playing a game it has not previously encountered.

2 Memory in Games

Memory in a game playing agent can refer to any kind of persistent knowledge an agent has at it's disposal that it does not need to deduce algorithmically. Some examples include:

- Databases of powerful strategies in games such as chess or checkers.

- Tables that record opponent based information in games such as poker.
- Or, a collection of cases, in a case-based reasoning system, illustrating various plays and their level of success.

The addition of some kind of memory component to a game-playing agent can be beneficial for numerous reasons. Some of which are outlined below:

- Memory can take the form of knowledge gained through expert play such as in Grandmaster databases. Experts with years of game playing experience can encode their knowledge and strategies into databases which form the basis of the agent's memory. This provides the agent with a persistent knowledge of the game that may not be available through other strategies alone such as game-tree search. These proven lines of sophisticated play can then be used by the agent to aid its game playing decisions.
- A memory component can be used to hold perfect information about a game at a certain position and the outcome from that position i.e. win/loss/draw. This improves the performance of the agent by allowing it to identify when a win is available. If only draws or losses are present it also allows the agent to avoid moves that will lead to a loss.
- Memory allows an agent to learn from experience. By maintaining a memory the agent can record which decisions were beneficial and which were harmful.
- For games that rely heavily on how an opponent plays e.g. poker, memory is imperative. An adaptive agent will be required to remember how an opponent has played in the past and what type of playing style they may employ. Using this information a strong game-playing agent can then exploit weak opponents and avoid being exploited itself. To achieve this the agent will need to encode some sort of long-term memory about specific opponents as well as general playing styles.

While it is true that for extremely simple games such as *Tic-Tac-Toe* optimal agents can be constructed algorithmically without relying on any memory component [12], this is not necessarily true for games that involve more sophisticated strategies such as *Chess*, *Checkers* or *Go*. As the complexity of the game increases so does the resulting search space required for the game tree. Reasonably complex, deterministic games such as checkers typically rely on the use of the *Alpha-Beta* pruning algorithm to determine the next best move for the agent to make, however as the number of ply required to search further into the future increases so too does the computational complexity. World class game playing agents such as *Chinook* in Checkers [15] have resorted to the use of end-game databases to address this issue. End-game databases provide a persistent memory of the exact outcome of a game from a certain position. The use of this memory component has substantially improved the performance of the agent [16]. The world-class chess machine *Deep Blue* also included an end-game database although its success depended less upon it and more on the inclusion of a database of Grandmaster games which were used to influence *Deep Blue's* decisions [4].

Chess and checkers are regarded as *deterministic* games with *perfect information*. There is no chance involved and both players can look upon the board and get all the information they need to make the best move possible. A separate category of games, classed as *stochastic* games with *imperfect information* are different, in that elements of chance play a role and information is hidden from the player. Poker is a game that involves chance and hidden information. Approaches to computer poker have mainly focussed around the use of game theory and adaptive imperfect information search [2, 21]. To be successful at games like poker a player has to be able to read his/her opponent, i.e. to compensate for missing information they have to make decisions based upon how their opponent has played in the past. We argue that at this point memory in a game playing agent not only becomes beneficial, but imperative. An early attempt to solve this problem was the poker playing program nicknamed *Poki*. *Poki* was developed by the University of Alberta Computer Poker Research Group¹ and used opponent modelling tables to keep track of how opponents played [3]. This memory for an opponent aided the agent in adapting its playing style accordingly.

The above examples have demonstrated the benefits that various forms of memory can have. This idea can be extended by considering programs that make decisions primarily based on memory, such as experience-based, lazy learners.

3 Lazy Learners

One approach to game AI focuses around the construction and traversal of game trees [4]. Another approach is to use machine learning algorithms to develop game playing agents [7, 18]. Many machine learning algorithms are classified as *eager learners*. An eager learner learns an approximation to a target function through training examples before any novel queries are encountered [11]. *Lazy learners* have also been developed and applied to game playing [13, 14]. Lazy learners usually bypass any computationally expensive training period and simply construct a local approximation to a target function when a new query is encountered [1]. When considering the problem of playing multiple, arbitrary games we believe that the use of lazy learners could prove beneficial due to the fact that lazy learners are highly adaptive to novel situations [1]. Experienced based or case-based systems can be considered lazy learners [10]. Detailed below are a number of experience-based agents that have been developed for specific games, with varying degrees of success.

Experience-based learning techniques were applied to the game of Othello in [6] with some success. The result was a system called GINA. De Jong and Schultz augmented a search-based Othello playing program with an experience base that was added to as GINA played more games. Each experience in the experience base was assigned a success rating which approximated the value that would have been found in a minimax search tree, coupled with a frequency counter that represented the confidence of the estimate. The results showed that the

¹ <http://poker.cs.ualberta.ca/>

use of experience-based learning was highly effective in improving both speed of decision making and skill in the game of Othello when challenging non-adaptive, minimax based opponents.

[13] produced CHEBR, a system to play the game of checkers via *automatic case elicitation*, whereby an agent with no prior domain knowledge acquired experience by simply playing games of checkers in real-time. CHEBR taught itself to play checkers better than an agent with initial knowledge of the game.

[17] reasoned that approaches to computer chess that used alpha-beta pruning algorithms employed a brute-force search strategy that considered many unnecessary lines of play. Sinclair focussed on forward pruning using example-based reasoning based on games played by human experts. An example base was built by analysing a collection of 16,728 expert games using *Principle Component Analysis* to reduce dimensionality and recording the move made given a board position. A separate test set of unseen positions was then used to assess the chosen move by the system. The results indicate that stronger moves were generated during earlier stages of the game when the example base held many instances and therefore similarity was high. However, this deteriorated as the move number increased as the example base became more sparse. Sinclair concludes that search based solutions do not transfer well to other problem domains they weren't designed for and proposes that example-based pruning may be well suited to handle imperfect information and problems where domain knowledge is incomplete [17].

Finally, case-based reasoning was applied to a stochastic, imperfect information environment in [14, 20]. A case-based poker player was developed (*Casper*) that made decisions in the game of Texas Hold'em by retrieving similar scenarios from it's memory and re-using these decisions. *Casper* was able to play even against the University of Alberta's *Pokibot*, upon which it was based, whilst avoiding the need for an intensive knowledge engineering effort.

As mentioned before the approaches discussed above have all focussed on specific domains. The next problem we wish to address is how an experience-based approach could be extrapolated to handle playing any arbitrary game it was presented with.

4 Analogy

Programs such as *Casper*, GINA and CHEBR described above have been created to focus on one particular domain - poker, othello and checkers respectively. Furthermore, one of the advantages of experience-based learning is an ability to generalise well [10]. Therefore, it is our opinion that a system with an initial memory of one or more game-related domains coupled with an ability to analogise beyond that domain could be used to address the problem of general game playing, where the same system uses knowledge it has obtained from previous games to inform it's decisions for a totally novel game.

Analogical reasoning has been successfully applied in the *Prodigy/Analogy* system [19]. Veloso combined derivational analogy with a base-level planning sys-

tem. Derivational analogy considers how a problem was solved rather than simply reusing old solutions for new problems [5]. This is achieved by taking the problem solving context into account. [19] describes the use of the *Prodigy/Analogy* system within a logistics transportation domain. When a problem is successfully solved an episodic solution trace is retained in a case base. This trace highlights justifications that support the decisions made. As new problems are encountered similar episodes are retrieved and their justifications are interpreted within the context of the new problem. One or more of the retrieved cases are then used to guide the problem solving process. This analogical reasoning process has allowed the successful transfer of skills within a complex domain, without a dependence on axiomatic domain knowledge. Resulting in a large increase in the amount and complexity of problems that can be solved compared to the base-level planning system [19].

Hinrichs and Forbus combine experimentation, analogy and qualitative modelling to the domain of a turn based strategy game [9]. The sub goal of optimizing food production within the *Freeciv* [8] strategy game is evaluated. Hinrichs and Forbus report that with the addition of learning from past failed cases, their experimental results indicate an improvement in the task of optimizing food production [9]. They propose that the use of analogy and qualitative reasoning is a viable approach to transfer learning, whereby a system is trained on one set of tasks and its learning subsequently measured on a set of related, but distinct tasks.

Analogical reasoning could perhaps produce similar results in a general game playing environment. Consider an example involving card games. The *Casper* system [14] plays Texas Hold'em poker entirely from memory. Through *Casper's* collection of experiences it has learned that cards such as *Jacks, Queens, Kings* and *Aces* are high valued cards. This knowledge could be generalised to other poker variants or even other card games as an initial assumption. If in fact this assumption proved incorrect (e.g. *Aces* are low) the system could compensate for its initial incorrect assumption by its ability to quickly adapt to new situations. Of course for general game playing, games entirely outside the realm of card-games would need to also be considered. This requires further extrapolation, however we believe this is not unreasonable. Take, for example, a situation where we wish to generalise knowledge a system may have about card values to aid it in playing a game that involves the throw of a dice. Given that the system assigns high value to high card values an initial assumption the system could infer would be that the same applies for dice values. Hence, the system would value rolling double sixes opposed to double ones. By recording the outcome of a game, the system could successively evaluate whether this analogy is relevant or not.

As the system played more and more games and accumulated more knowledge about the games it has played its experience-base would grow, allowing it to make further inferences and generalisations about different games it encountered. By consistently maintaining its knowledge-base the system could drop analogies that proved incorrect and strengthen others that contributed to

successful outcomes. It is hoped this process would improve the general game playing abilities of the system.

We believe that an experience-based, lazy learner would provide the flexibility required to handle the type of generalisation described above.

5 Conclusion

In conclusion, the idea of a lazy learning agent has been proposed that relies on memory and analogy to generalise knowledge gained in one domain with the intention of applying it to another. We believe this approach could be beneficial to general game playing due to the fact that experience-based, lazy learners are able to adapt well to new situations and have been shown to be successful in a range of game environments e.g. deterministic vs. stochastic. Furthermore, analogical reasoning has demonstrated an ability to generalise skills within complex domains.

A discussion of the importance of memory in game-playing agents was presented. It was shown that memory can take many forms, but mostly relies on the encoding of specific game knowledge into databases or case-bases. Successful agents in board and card games have been used as examples to highlight the types of memory available and how it has been used effectively.

The Prodigy/Analogy system was discussed as an example that has achieved success via analogical reasoning. Finally, we speculated about the possibilities of analogy coupled with experience-based learners to generalise game knowledge which could be used as a basis for a general game playing agent.

References

1. Aha, D, W.: Editorial. *Artificial Intelligence Review*. 11(1-5):7-10, (1997)
2. Billings, D., Burch, N., Davidson, A., Holte, R, C., Schaeffer, J., Schauenberg T., Szafron D.: Approximating Game-Theoretic Optimal Strategies for Full-scale Poker. *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann. 661-668 (2003)
3. Billings, D., Davidson, A., Schaeffer, J., Szafron, D.: The challenge of poker. *Artificial Intelligence*. (134): 201-240 (2002)
4. Campbell, M., Hoane, J., Hsu, F.: Deep Blue. *Artificial Intelligence*. (134). 57-83 (2002)
5. Carbonell, J, G.: Derivational Analogy and Its Role in Problem Solving. *Proceedings of the National Conference on Artificial Intelligence*. AAAI Press. 64-69 (1983)
6. De Jong, K., Schultz, A, C.: Using Experience-Based Learning in Game Playing. *ML*. 284-290 (1988)
7. Fogel, D, B.: Evolving a checkers player without relying on human experience. *Intelligence*. (11). 20-27 (2000)
8. Freeciv. Freeciv official website. <http://www.Freeciv.org/> (2006)
9. Hinrichs, T, R., Forbus, K, D.: Analogical Learning in a Turn-Based Strategy Game. *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*. 853-858 (2007)

10. Leake, D, B.: Case-based reasoning : experiences, lessons & future directions. Menlo Park, Calif. : AAAI Press ; Cambridge, Mass. : MIT Press, (1996)
11. Mitchell, T, M.: Machine learning. New York : McGraw-Hill, (1997)
12. Ostermiller, S.: Tic-Tac-Toe Strategy. <http://ostermiller.org/tictactoeexpert.html> January (2009)
13. Powell, J, H., Hauff, O, M., Hastings, J, D.: Utilizing case-based reasoning and automatic case elicitation to develop a self-taught knowledgeable agent. Challenges in Game Artificial Intelligence: Papers from the AAAI Workshop (Technical Report WS-0404). AAAI Press. 77-81 (2004)
14. Rubin, J., Watson, I.: Investigating the Effectiveness of Applying Case-Based Reasoning to the Game of Texas Hold'em. Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference. AAAI Press. 417-422 (2007)
15. Schaeffer, J.: One jump ahead: challenging human supremacy in checkers. Springer-Verlag New York, Inc. New York, NY, USA. (1997)
16. Schaeffer, J., Björnsson, Y., Burch, N., Lake, R., Lu, P., Sutphen, S.: Building the Checkers 10-piece Endgame Databases. Advances in Computer Games, Many Games, Many Challenges, 10th International Conference. Kluwer, 193-210 (2003)
17. Sinclair, D.: Using Example-Based Reasoning for Selective Move Generation in Two Player Adversarial Games. Advances in Case-Based Reasoning, 4th European Workshop, EWCBR-98. Springer, Lecture Notes in Computer Science (1488). 126-135 (1998)
18. Tesauro, G.: Temporal Difference Learning and TD-Gammon. Commun. ACM. (38) 58-68 (1995)
19. Veloso, M, M.: Flexible Strategy Learning: Analogical Replay of Problem Solving Episodes. Proceedings of the 12th National Conference on Artificial Intelligence. AAAI Press. 595-600 (1994)
20. Watson, I., Rubin, J.: CASPER: A Case-Based Poker-Bot. AI 2008: Advances in Artificial Intelligence, 21st Australasian Joint Conference on Artificial Intelligence. Springer, Lecture Notes in Computer Science (5360). 594-600 (2008)
21. Zinkevich, M., Johanson, M., Bowling, M., Piccione, C.: Regret Minimization in Games with Incomplete Information. Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems. MIT Press. (2007)