Mach Learn (2006) 63:211–215
DOI 10.1007/s10994-006-8919-x

GUEST EDITORIAL

Machine learning and games

**Michael Bowling · Johannes Fürnkranz ·
Thore Graepel · Ron Musick**

Published online: 10 May 2006

The history of the interaction of machine learning and computer game-playing goes back to the earliest days of Artificial Intelligence, when Arthur Samuel worked on his famous checker-playing program, pioneering many machine-learning and game-playing techniques (Samuel, 1959, 1967). Since then, both fields have advanced considerably, and research in the intersection of the two can be found regularly in conferences in their respective fields and in general AI conferences. For surveys of the field we refer to Ginsberg (1998), Schaeffer (2000), Fürnkranz (2001); edited volumes have been compiled by Schaeffer and van den Herik (2002) and by Fürnkranz and Kubat (2001).

In recent years, the computer games industry has discovered AI as a necessary ingredient to make games more entertaining and challenging and, vice versa, AI has discovered computer games as an interesting and rewarding application area. The industry's perspective is witnessed by a plethora of recent books on gentle introductions to AI techniques for game programmers (Collins, 2002; Champanard, 2003; Bourg & Seemann, 2004; Schwab, 2004) or a series of edited collections of articles (Rabin, 2002, 2003, 2006). AI research on computer games began to follow developments in the games industry early on, but since John Laird's keynote address at the AAAI 2000 conference, in which he advocated Interactive Computer Games as a challenging and rewarding application area for AI (Laird & van Lent, 2001), numerous workshops (Fu & Orkin, 2004; Aha et al., 2005), conferences, and special issues of journals (Forbus & Laird, 2002) demonstrate the growing importance of game-playing applications for Artificial Intelligence.

M. Bowling (✉)
e-mail: bowling@cs.ualberta.ca

J. Fürnkranz
e-mail: fuernkranz@informatik.tu-darmstadt.de

T. Graepel
e-mail: thoreg@microsoft.com

R. Musick
e-mail: musick@ikuni.com

Games, whether created for entertainment, simulation, or education, provide great opportunities for machine learning. The variety of possible virtual worlds and the subsequent ML-relevant problems posed for the agents in those worlds is limited only by the imagination. Furthermore, not only is the games industry large and growing (having surpassed the movie industry in revenue a few years back), but it is faced with a tremendous demand for novelty that it struggles to provide. Against this backdrop, machine learning driven successes would draw high-profile attention to the field. Surprisingly however, the more commercial the game to date, the less impact learning has made. This is quite unlike other great matches between application and data-driven analytics such as data mining and OLAP.

Topics of particular importance for successful game applications include learning how to play the game well, player modeling, adaptivity, model interpretation and of course performance. These needs can be recast as a call for new practical and theoretical tools to help with:

learning to play the game: Game worlds provide excellent test beds for investigating the potential to improve agents' capabilities via learning. The environment can be constructed with varying characteristics, from deterministic and discrete as in classical board and card games to non-deterministic and continuous as in action computer games. Learning algorithms for such tasks have been studied quite thoroughly. Probably the best-known instance of a learning game-playing agent is the Backgammon-playing program TD-Gammon (Tesauro, 1995).

learning about players: Opponent modeling, partner modeling, team modeling, and multiple team modeling are fascinating, interdependent and largely unsolved challenges that aim at improving play by trying to discover and exploit the plans, strengths, and weaknesses of a player's opponents and/or partners. One of the grand challenges in this line of work are games like Poker, where opponent modeling is crucial to improve over game-theoretically optimal play (Billings et al., 2002).

behavior capture of players: Creating a convincing avatar based on a player's in-game behavior is an interesting and challenging supervised learning task. For example, in Massive Multiplayer Online Role-playing Games (MMORGs) an avatar that is trained to simulate a user's game-playing behavior could take his creator's place at times when the human player cannot attend to his game character. First steps in this area have been made in commercial video games such as *Forza Motorsport* (Xbox) where the player can train a "Drivatar" that learns to go around the track in the style of the player by observing and learning from the driving style of that player and generalizing to new tracks and cars.

model selection and stability: Online settings lead to what is effectively the unsupervised construction of models by supervised algorithms. Methods for biasing the proposed model space without significant loss of predictive power are critical not just for learning efficiency, but interpretive ability and end-user confidence.

optimizing for adaptivity: Building opponents that can just barely lose in interesting ways is just as important for the game world as creating world-class opponents. This requires building highly adaptive models that can substantively personalize to adversaries or partners with a wide range of competence and rapid shifts in play style. By introducing a very different set of update and optimization criteria for learners, a wealth of new research targets are created.

model interpretation: "What's my next move" is not the only query desired of models in a game, but it is certainly the one which gets the most attention. Creating the illusion of intelligence requires "painting a picture" of an agent's thinking process. The ability to describe the current state of a model and the process of inference in that model from decision to decision enables queries that provide the foundation for a host of social actions in a game such as predictions, contracts, counter-factual assertions, advice, justification, negotiation, and demagoguery. These can have as much or more influence on outcomes as actual in-game actions.

performance: Resource requirements for update and inference will always be of great importance. The AI does not get the bulk of the CPU or memory, and the machines driving the market will always be underpowered compared to typical desktops at any point in time.

This special issue contains three articles and one research note that span the wide range of research in the intersection of game playing and machine learning.

In the first contribution, *Adaptive Game AI with Dynamic Scripting*, Spronck et al. tackle the problem of adaptivity by dynamically modifying the rules which govern character behavior in-game. This paper is targeted at the commercial games industry, and provides some good insight into problems faced by the creators of today's role playing games. The authors propose four functional and four computational requirements for on-line learning in games. They then proceed to show how dynamic scripting fits into those requirements, and provide experimental evidence of the potential promise of this approach. Dynamic scripting can be characterized as stochastic optimization. The authors evaluate dynamic scripting on both the task of providing the toughest opponent possible, and on the task of difficulty scaling. Good difficulty scaling underpins what makes most games fun, and solving this problem is often very challenging and the solutions are almost always ad-hoc. The authors present experimental data that compares dynamic scripting to static opponents and those controlled by Q-Learning and Monte Carlo. The test environments include both simulated games and an actual commercial game (Neverwinter Nights), and help to present a very interesting study which is sure to blaze a path for further interesting research.

The second paper, *Universal Parameter Optimization in Games Based on SPSA* by Szepesvari and Kocsis, considers the problem of optimizing parameters to improve the performance of parameterized policies for game play. They consider the Simultaneous Perturbation Stochastic Approximation (SPSA) method introduced by Spall (1992) which is a general gradient free optimization method that is applicable to a wide range of optimization problems. The authors demonstrate that SPSA is applicable to a wide range of typical optimization problems in games and propose several methods to enhance the performance of SPSA. These enhancements include the use of common random numbers and antithetic variables, a combination with RPROP and the reuse of samples. The application to games considers the domain of learning to play Omaha Hi-Lo Poker with their poker program *McRaise*. SPSA combined with their proposed enhancements leads to poker performance competitive with TD-learning, the method so successfully used by Tesauro (1995), for learning a world-class evaluation function for Backgammon and still used in today's world class backgammon programs such as *JellyFish* and Snowie.

The third contribution, *Learning to Bid in Bridge* by Markovitch and Amit, addresses the problem of bidding in the game of Bridge. While research in Bridge playing has pioneered Monte Carlo search algorithms for the playing phase of card games and resulted in programs of considerable strength (Ginsberg, 1999), the bidding phase, in which the goal (the so-called *contract*) of the subsequent playing phase is determined, is still a major weakness of existing Bridge programs. This paper is about an approach that supports the difficult bidding phase in the game Bridge with techniques from machine learning, in particular opponent modeling via the learning of decision nets and via model-based Monte Carlo sampling to address the problem of hidden information. The evaluation clearly establishes that the system improves with learning, and it seems that the level of play achieved by this program surpasses the level of the bidding module of current state-of-the-art programs and approaches that of an expert player.

Finally, Sadikov and Bratko present a research note on *Learning Long-term Chess Strategies from Databases*. They address the problem of knowledge discovery in game databases. For many games or subgames (such as chess endgames), there are game databases available, which contain perfect information about the game in the sense that for every possible position, the game-theoretic outcome is stored in a database. However, although these databases contain all information to allow perfect play, they are not amenable to human analysis, and are typically not very well understood. For example, chess Grandmaster John Nunn analyzed several simple chess endgame databases resulting in a series of widely acknowledged endgame books (Nunn, 1992, 1994b, 1995), but readily admitted that he does not yet understand all aspects of the databases he analyzed (Nunn, 1994a). This paper reports on an attempt to make headway by automatically constructing playing strategies from chess endgame databases. It describes a method for breaking up the problem into different game phases. For each phase, it is then proposed to learn a separate evaluation function via linear regression. Experiments in the the king and rook vs. king, or king and queen vs. king and rook endgames show encouraging results, but also illustrate the difficulty of the problem.

Machine learning has been instrumental to date in building some of the world's best players in Backgammon and has lead to interesting results in games like Chess and Go. To move into mainstream commercial games, machine learning research has to face what in many ways are the harder problems of losing in interesting ways, creating more useful illusions of intelligence, hyper-fast adaptation, and taking on persona. The articles in this special issue provide a glimpse into different facets of all of these problems.

References

Aha, D. W., Muñoz-Avila H. M., & van Lent, M. (Eds.), (2005). *Reasoning, representation, and learning in computer games: Proceedings of the IJCAI workshop.* Edinburgh, Scotland: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence. Technical Report AIC-05-127.

Billings, D., Peña, L., Schaeffer, J., & Szafron, D. (2002). The challenge of poker. *Artificial Intelligence*, *134*(1–2), 201–240, Special Issue on Games, Computers and Artificial Intelligence.

Bourg, D. M., & Seemann G. (2004). *AI for game developers—Creating intelligent behavior in games*. O'Reilly.

Champanard, A. (2003). *AI game development*. New Riders Publishing.

Collins, M. (2002). *Advanced AI game development*. Wordware Publishing Inc.

Forbus, K. D., & Laird J. E. (2002). Guest editors' introduction: AI and the entertainment industry. *IEEE Intelligent Systems*, *17*(4), 15–16.

Fu, D. & Orkin J. (Eds.), (2004). *Challenges of game AI: Proceedings of the AAAI-04 workshop.* AAAI Press. Technical Report WS-04-04.

Fürnkranz, J. (2001). Machine learning in games: A survey. In (Fürnkranz and Kubat, 2001), (pp. 11–59).

Fürnkranz, J., & Kubat M. (Eds.), (2001). *Machines that learn to play games*. Huntington, NY: Nova Science Publishers.

Ginsberg, M. L. (1998). Computers, games and the real world. *Scientific American Presents* 9:4. Special Issue on *Exploring Intelligence*.

Ginsberg, M. L. (1999). GIB: Steps toward an expert-level bridge-playing program. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-99)*, (pp. 584–589), Stockholm, Sweden.

Laird, J. E., & van Lent, M. (2001). Human-level AI's killer application: Interactive computer games. *AI Magazine*, *22*(2), 15–26.

Nunn, J. (1992). *Secrets of rook endings*. Batsford.

Nunn, J. (1994a). Extracting information from endgame databases. In H. J. van den Herik, I. S. Herschberg, and J. W. H. M. Uiterwijk (Eds.), *Advances in computer chess 7*, (pp. 19–34). Maastricht, The Netherlands: University of Limburg.

Nunn, J. (1994b). *Secrets of pawnless endings*. Batsford.

Nunn, J. (1995). *Secrets of minor-piece endings*. Batsford.

Rabin, S. (Ed.), (2002). *AI game programming wisdom*. Charles River Media.

Rabin, S. (Ed.), (2003). *AI game programming wisdom 2*. Charles River Media.

Rabin, S. (Ed.), (2006). *AI game programming wisdom 3*. Charles River Media.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*(3), 211–229.

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. II—Recent progress. *IBM Journal of Research and Development*, *11*(6), 601–617.

Schaeffer, J. (2000). The games computers (and people) play. In M. V. Zelkowitz (Ed.), *Advances in computers*, vol. 50, (pp. 189–266). Academic Press.

Schaeffer, J., & van den Herik, H. J. (Eds.), (2002). *Chips challenging champions: Games, computers and artificial intelligence*. North-Holland Publishing. Reprint of a Special Issue of *Artificial Intelligence*, 134(1–2).

Schwab, B. (2004). *AI game engine programming*. Charles River Media.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, *37*, 332–341.

Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM*, *38*(3), 58–68.