Can Meta-Interpretive Learning outperform Deep Reinforcement Learning of Evaluable Game strategies?

Céline Hocquette[1] , **Stephen H. Muggleton**[1]
[1]Department of Computing, Imperial College London, London, UK
{celine.hocquette16, s.muggleton}@imperial.ac.uk

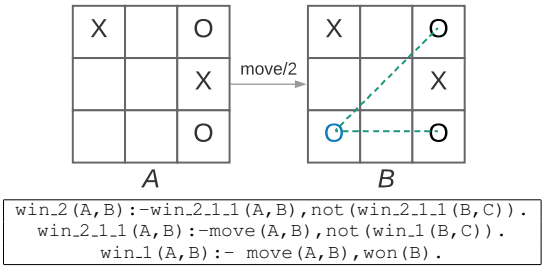

Figure 1: Noughts and Crosses: example of optimal move for O from board A to board B

1 Introduction

World-class human players have been outperformed in a number of complex two person games such as Go by Deep Reinforcement Learning systems [Silver D. *et al.*, 2016]. However, several drawbacks can be identified for these systems: 1) the data efficiency is unclear given they appear to require far more training games to achieve such performance than any human player might experience in a lifetime, 2) These systems are not easily interpretable as they provide limited explanation about how decisions are made, 3) these systems do not provide transferability of the learned strategies to other games.

We study in this work how machine learning strategies as logic programs and from an explicit logical representation can overcome these limitations. For example, an applicable strategy for playing Noughts-and-Crosses is to lead double attacks when possible, an example of which is shown in Figure 1. Player O executes a move from board A to board B which creates two threats represented in green, and results in a forced win for O. The rules presented in Figure 1 describe such a strategy. A and B are variables representing states that encode both the board description and the active player. A move from A to B is a winning move if the opponent can not immediately win and cannot make a move to prevent an immediate win. These rules provide an understandable strategy for winning in two moves. Moreover, they are transferable to more complex games as they are generally true for describing double attacks.

We introduce a new logical system called *MIGO* [1] based upon Meta-Interpretive Learning and designed for learning two player game optimal strategies of the form presented in Figure 1. It benefits from a strong inductive bias which provides the capability to learn efficiently from a few examples of games played. Additionally, *MIGO*'s learned rules are relatively easy to comprehend, and are demonstrated to achieve significant transfer learning. *MIGO* uses Meta-Interpretive Learning (MIL), a form of inductive logic programming which supports predicate invention and learning recursive theories [Muggleton and Lin, 2013].

2 Related Work

Various early approaches to game strategies [Shapiro and Niblett, 1982; J.R. Quinlan, 1983] used the decision tree learner ID3 to classify minimax depth-of-win for positions in chess end games. These approaches used a set of carefully selected board attributes as features. Conversely, MIGO is provided with a set of three relational primitives (move/2, won/1, drawn/1) representing the minimal information a human would expect to know before playing a two person game.

Classical reinforcement learning approaches, and more recently Deep Q-learning [Mnih *et al.*, 2015], are based upon the identifciation of a Q-function [Watkins, 1989]. The learned strategy is implicitly encoded into the Q-value parameters. However, these frameworks generally require the execution of many games to converge. Moreover, the learned strategy is implicitly encoded into the Q-value parameters, which do not provide interpretability of the learned strategy.

In the relational reinforcement learning (RRL) framework [Džeroski *et al.*, 2001], states, actions and policies are represented relationally. The learning is also based upon the identification of Q-values whereas *MIGO* learns hypotheses from examples of moves. Both RRL and *MIGO* provide the ability to carry over the policies learned in simple domains to more complex situations. However, most RRL systems aim at learning single agent policy and, in contrast to *MIGO*, are not designed to learn to play two person games.

3 Theoretical Framework

3.1 Credit Assignment Protocol

MIGO solves the Credit Assignment Problem from the Theorems below for identifying moves that are necessarily positive

[1]From the children's game-playing phrase *My go!* and the literal translation into English of the French word *Ordinateur* which means computer.

examples for the task of winning or drawing.
We assume the learner P_1 plays against opponent P_2 which follows an optimal strategy, and that games start from a randomly chosen initial board B. We consider the following ordering over the different outcomes for P_1:

$$won \succ drawn \succ lost$$

Then, the following Lemma and Theorems hold:

Lemma 1: The outcome of P_1 monotonically decreases during a game.

Theorem 2: If the outcome is won for P_1, then every move of P_1 is a positive example for the task of winning.

Theorem 3: If an accurate winning strategy S_W is known and its execution from B fails, then if the outcome of the game is drawn, then any move played by P_1 or P_2 is a positive example for the task of drawing.

3.2 Meta-Interpretive Learning

MIGO is a MIL system [Muggleton *et al.*, 2014; 2015]. MIL is a form of ILP based on a Prolog meta-interpreter, and which supports predicate invention, the learning of recursive programs and Abstraction. MIGO is an extension of the MIL system Metagol [Cropper and Muggleton, 2016].

4 MIGO Algorithm

4.1 Learning from positive examples

Theorems above provide a way of assigning only positive labels to moves. Therefore, the learning is based upon positive examples only. This is possible because of Metagol's strong language bias and ability to generalise from a few examples only. However, one pitfall is the risk of over-generalisation due to the absence of negative examples.

4.2 Dependent Learning

The learning operates in a staged fashion: simple definition are first learned and added to the background knowledge, allowing them to be reused during further learning tasks, and thus to build up more and more complex definitions. For successive values of k a series of inter-related definitions are learned for predicates $win\_k(A, B)$ and $draw\_k(A, B)$. These predicates define maintenance of minimax win and draw in k-ply when moving from position A to B. The learning algorithm is presented as Algorithm 1, each action 'learn' represents a call to Metagol. This approach is related to Dependent Learning [Lin *et al.*, 2014].

4.3 Primitives and Metarules

Learned programs are formed of dyadic predicates, representing actions, and monadic predicates, representing fluents. The background knowledge contains a general move generator move/2, which is an action executing a valid move on a board, and a won and a drawn classifiers won/1 and drawn/1, which are two fluents. In other words, the background knowledge encodes the rules of the game. Additional primitives based upon geometrical considerations have been considered in a follow-up experiment aiming at improving the running time of the learned strategy.
The metarules used are *postcondition* and a variant of postcondition which includes negation of primitive predicates.

Algorithm 1 MIGO Algorithm

Input: Positive examples for win_k and draw_k
Output: Strategy for win_k and draw_k

```
for k in [1,Depth] do
    for each example of win_k/2 do
        one shot learn a rule and add it to the BK
    end for
    Learn win_k/2 and add it to the BK
end for
for k in [1,Depth] do
    for each example of draw_k/2 do
        one shot learn a rule and add it to the BK
    end for
    Learn draw_k/2 and add it to the BK
end for
```

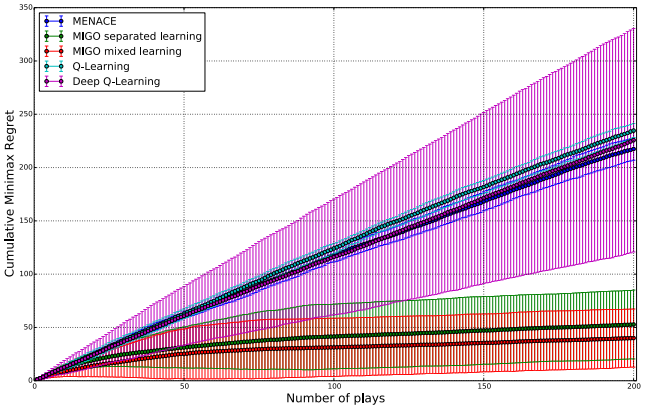

Figure 2: Cumulative regret versus the number of games played for Noughts-and-Crosses

5 Results

5.1 Cumulative Minimax Regret

Owing to tractability considerations, minimax regret of a learning system cannot be evaluated in complex games. We first consider simple games (Noughts-and-Crosses and Hexapawn) for which minimax regret can be efficiently evaluated. The reinforcement learning systems considered for comparison are MENACE [Michie, 1963] which is the world's first reinforcezment learning system, Tabular Q-learning [Watkins, 1989] and Deep Q-learning [Mnih *et al.*, 2015]. In our experiment all tested variants of both normal and deep reinforcement learning have worse performance (higher cumulative minimax regret) than both variants of *MIGO* on Noughts-and-Crosses as shown in Figure 2.

Depth	Rule
1	`win_1(A,B):-win_1_1_1(A,B),won(B).` `win_1_1_1(A,B):-move(A,B),won(B).`
2	`win_2(A,B):-win_2_1_1(A,B),not(win_2_1_1(B,C)).` `win_2_1_1(A,B):-move(A,B),not(win_1(B,C)).`
3	`win_3(A,B):-win_3_1_1(A,B),not(win_3_1_1(B,C)).` `win_3_1_1(A,B):-win_2_1_1(A,B),not(win_2(B,C)).`

Table 1: Winning strategy learned for Noughts and Crosses

5.2 Comprehensibility

The winning strategy learned for Noughts and Crosses is presented on Table 1. Strategies learned in this form provide a certain form of comprehensibility. However, they seem less comprehensible as the depth augments. We aim at studying whether MIGO additionally fulfils Michie's ultra-strong Machine Learning criteria, which requires the learner to be able to teach such learned strategies to humans, whose performance is consequently increased to a level beyond that of the human studying the training data alone [Muggleton *et al.*, 2018]. Initial experiments have been conducted, in which school children were provided with feedback on positional play based on MIGO's learned rules. Additional primitives have been considered to reduce the execution time of the learned strategy and therefore improve the comprehensibility for large depth.

6 Conclusion and Future Work

This work introduces a novel logical system named *MIGO* for learning two-player-game strategies and based upon the MIL framework. Our experiment have demonstrated that *MIGO* achieves lower Cumulative Minimax Regret compared to Deep and classical Q-Learning. Moreover, strategies learned with *MIGO* are general enough to be transferable to more complex games. Learned strategies are also relatively easy to comprehend.

One current limitation of *MIGO* is the limited scalability. The execution of learned strategies is computationally expensive as it browses the minimax tree to evaluate whether a move is a winning move. Therefore the running time increases rapidly with the state dimensions. The scalability is also limited by initial assumptions: the current version of *MIGO* requires a minimax player as opponent which is intractable in large dimensions. We further plan to extend this framework by relaxing our credit assignment protocol and weakening the optimal opponent assumption. A solution would be to learn from self-play.

Despite these limitations, we believe the novel approach introduced in this work opens exciting new avenues for machine learning game strategy.

References

[Cropper and Muggleton, 2016] A. Cropper and S.H. Muggleton. Metagol system. https://github.com/metagol/metagol, 2016.

[Džeroski *et al.*, 2001] S. Džeroski, L. De Raedt, and K. Driessens. Relational reinforcement learning. *Machine Learning*, 43(1):7–52, Apr 2001.

[J.R. Quinlan, 1983] J.R. J.R. Quinlan. *Learning Efficient Classification Procedures and Their Application to Chess End Games*, pages 463–482. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983.

[Lin *et al.*, 2014] D. Lin, E. Dechter, K. Ellis, J.B. Tenenbaum, and S.H. Muggleton. Bias reformulation for one-shot function induction. In *In Proceedings of the 23rd European Conference on Artificial Intelligence (ECAI 2014)*, pages 525–530,. IOS Press, 2014.

[Michie, 1963] D. Michie. Experiments on the mechanization of game-learning part i. characterization of the model and its parameters. *The Computer Journal, Volume 6, Issue 3*, pages 232–236, 1963.

[Mnih *et al.*, 2015] V. Mnih, K. Kavukcuoglu, and D. Silver et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 02 2015.

[Muggleton and Lin, 2013] S.H. Muggleton and D. Lin. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. In *In Proceedings of the 23rd International Joint Conference Artificial Intelligence*, pages 1551–1557, 2013.

[Muggleton *et al.*, 2014] S.H Muggleton, D. Lin, N. Pahlavi, and A. Tamaddoni-Nezhad. Meta-interpretive learning: application to grammatical inference. *Machine Learning 94*, pages 25–49, 2014.

[Muggleton *et al.*, 2015] S.H. Muggleton, D. Lin, and A. Tamaddoni-Nezhad. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. *Machine Learning*, 100(1):49–73, 2015.

[Muggleton *et al.*, 2018] S.H. Muggleton, U. Schmid, C. Zeller, A. Tamaddoni-Nezhad, and T. Besold. Ultra-strong machine learning: comprehensibility of programs learned with ilp. *Machine Learning*, 107(7):1119–1140, July 2018.

[Shapiro and Niblett, 1982] A. Shapiro and T. Niblett. Automatic induction of classification rules for a chess endgame. In M.R.B. Clarke, editor, *Advances in Computer Chess*, volume 3, pages 73–91. Pergammon, Oxford, 1982.

[Silver D. *et al.*, 2016] Huang A. Silver D., C. Maddison, and A. et al. Guez. Mastering the game of go with deep neural networks and tree search. 529:484–489, 01 2016.

[Watkins, 1989] C. Watkins. Learning from delayed rewards, phd thesis. 1989.