Short Term Opponent Exploitation in Poker

Wesley Tansey
University of Texas at Austin
1 Inner Campus Drive
Austin, TX 78712 USA
tansey@cs.utexas.edu

ABSTRACT

Effectively exploiting opponents in incomplete information, extensive-form games is a difficult task for an online learning agent. Previous work has focused on either maintaining an explicit model that is updated directly based on observed opponent actions or implicit modeling via a portfolio of methods.

This paper introduces four new approaches to playing exploitive poker. First, a one-step temporal differencing version of counterfactual regret minimization called TD(0)-CFR is presented. Second, an alternative implicit modeling approach based on the notion of subpolicies is explored. Third, a hybrid implicit and explicit algorithm is described that leverages a portfolio of models to bootstrap an explicit model. Finally, a combination of the second and third models is discussed along with an approach automtically deriving subpolicies from a portfolio of complete policies.

The game of Leduc poker is used as a benchmark domain to compare the two standard and four novel exploitive approaches. Results from these experiments indicate that everything failed and I should just give up on life.

General Terms

Algorithms, Experimentation

Keywords

poker, opponent modeling, online learning

1. INTRODUCTION

2. BACKGROUND

2.1 Counterfactual Regret Minimization

The Counterfactual Regret minimization (CFR) algorithm [12] is an iterative algorithm that extends regret matching [5] to incomplete-information, extensive-form games to provably walk a strategy profile towards a correlated ϵ-equilibrium.[1]

[1]Since Leduc poker is a two-player, zero-sum game, all correlated equilibria are also Nash equilibria.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

Later work has introduced speedups to CFR in the form of game tree compression [6] and Monte Carlo variants [9]. Among the MC-CFR implementations, Outcome Sampling CFR (OS-CFR) can be used for online regret minimization. The TD-CFR approach presented in this paper is based on OS-CFR.

2.2 Explicit opponent modeling

Explicit modeling of an opponent is perhaps the most natural way to conceive of an exploitive strategy. A probability distribution is estimated at each information set in the opponent's incomplete information gametree, with the results being updated after every observed outcome. Given the explicit model, a best response can be calculated and played. Data-Biased Responses (DBRs) [7] take a frequentist approach to explicit modeling, leveraging a large database of perfect information hands of opponent actions to build a model of opponent play. Once the opponent model is derived, DBRs compute a Restricted Nash Response (RNR) [8] that enables the agent to trade off exploitation and exploitability. While the resulting model is highly effective, the required large number of perfect-information hands and the sensitivity to the source of the samples makes DBRs impractical for short-term opponent exploitation.

An alternative, Bayesian approach to explicit modeling is to maintain a dirichlet prior at every information set [10, 11]. After every revealed hand, priors can then be updated; after a folded hand, the priors can be updated by marginalizing out the unobserved holecards. An exact opponent strategy can then be sampled via importance sampling or taken as the *maximum a posteriori* (MAP) strategy. Importance sampling is the more robust strategy since MAP inference may be a poor approximation if the distribution is multimodal. This paper uses importance sampling over a set of dirichlet priors, initialized to have a mode at the Nash equilibrium, for its explicit opponent modeling agent.

2.3 Implicit modeling

For large games, simple explicit modeling of an opponent may be impractical since the number of samples observed will generally cover only a small fraction of total game tree. Implicit opponent modeling [11] instead assumes the opponent is using a strategy drawn from some portfolio of potential strategies that is made available to the agent. After each outcome, the agent updates the likelihood of the opponent using each specific strategy, $s \in S$, given some observations, O, made online. Given that the parameters of s are known, and assuming a fair deck that deals each hand equally likely, the likelihood of a player using a given strategy can be com-

puted via Bayes rule: $p(s|O) = \frac{p(O|s)p(s)}{p(O)}$. As in explicit Bayesian opponent modeling, the opponent model can be chosen either as the MAP strategy or with importance sampling. Alternatively, an adversarial bandit algorithm such as Exp4 [1] can be used to find the best strategy from the best responses to the portfolio strategies. Implicit modeling with a portfolio of 2010 ACPC competitors is currently the state-of-the-art in exploitive computer poker agents [2]. This paper uses importance sampling over a set of five opponent models for its implicit opponent modeling agent.

3. ALGORITHMS

3.1 TD(0)-CFR

3.2 Implicit Subpolicy Modeling

3.3 Policy Bootstrapping

3.4 Subpolicy Bootstrapping

4. EXPERIMENTS

Experiments were performed on all four of our proposed algorithms and the standard approaches from the literature. We next detail our experimental setup and the results for each model in each experiment.

4.1 Setup

To analyze the performance of each model, we conducted a series of experiments in the game of Leduc poker. Leduc is a two-player poker variant where the deck contains six cards (two jacks, two queens, and two kings), each player is dealt a single hole card, there two rounds of betting, and a single community card is dealt after the first round. In total, Leduc contains 144 information sets for the second player, making this game non-trivial for agent modeling while still being tractable to analye in-depth. For more information on Leduc poker, see [10].

Experiments were conducted against three types of stationary opponents. First, a population of 100 "simple" agents was generated by generating two random percentage triplets for each agent; the first triplet was then used to skew the preflop actions of a Nash equilibrium strategy and the second was used to skew the flop actions. A population of 100 "complex" agents was then generated in a similar manner, where with equal probability the agent's preflop or flop actions were skewed, with a similar approach used to skew actions based on the holecard of the opponent (J, Q, or K). These two populations of agents aim to capture the systematic bias presumably found in human players that often over- or under- play a given hand or play looser or tighter at different points in the game. Finally, a Nash equilibrium strategy was used as a baseline to examine how the agents perform when trying to model an optimal player.

In each experiment, the algorithms were tested for 200 matches of 200 hands each, with each algorithm playing against each simple and complex opponent twice. The TD-CFR agent used $\epsilon = 0.1$ with an exponential ϵ-decay factor of 0.99 and a learning rate of $\alpha = 0.05$. For portfolio-based methods, the portfolio consisted of the Nash equilibrium strategy and four skewed strategies, chosen at random from the population at the start of each match. For explicit models, two times the Nash strategy was used as an initial prior. The subpolicy discovery algorithm used the Nash equilibrium strategy as the baseline method, $\delta^b_{min} = 0.05$, $\delta^{sp_i}_{max} = 0.1$, and a minimum subpolicy size of 3.

4.2 Results

5. DISCUSSION

6. FUTURE WORK

The experiments conducted in this paper reveal several opportunities for future research. In this section, we highlight some of the areas that we believe are ripe for future work:

- **Learning opponent models from incomplete information.** DBRs [7] require a large number of well-chosen, perfect-information samples to form strong opponent models that can then be used online. In contrast, an approach like Deviation-Based Best Resonses (DBBRs) [4] can potentially model opponents through incomplete information observations, but requires a startup phase of T hands to exploit an opponent. Ideally an exploitive agent would develop models that are both immediately exploitable and feasible to gather in practice.
- **Increasing sample efficiency.** It seems intuitive that most real-world opponents' exploitable actions are correlated in a systematic way; this intuition has been noted by other researchers [11] as well. Increasing the information derived from opponent observations, whether from subpolicy analysis or some other approach, is critical to the success of future exploitive agent research.
- **Subpolicy discovery.** Uncovering which subregions of opponent policies may contain the same systematic bias holds a large amount of potential. The straightforward approach to subpolicy discovery through better feature engineering is a pragmatic approach, but a more general approach would be more powerful and be applicable to a wider class of games.
- **Incremental best response calculation.** When maintaining an explicit opponent model where updates are made to only a portion of the model at each point, calculating a full best response may be unnecessary. In games such as Texas Hold'em, the full best response may be too computationally expensive to compute after every hand. An approach that can update only the parts of the model that are effected at each turn could vastly speed up the algorithm, with important practical implications.
- **Modeling non-stationary opponents.** The vast majority of exploitive play in poker has assumed a stationary opponent, likely due to the presumed drastic increase in difficulty of opponent modeling. Fortunately, while this may be the case for game theoretic non-stationary agents, humans appear to use relatively simple models that are weighted heavily towards recent observations [3]. Broadening the class of opponents to include such simple agents may hit the sweet spot of analytic tractability and real-world applicability.

- **Deceptively teaching opponents.** All exploitive poker research to date has ignored the potential for teaching opponents an exploitable model. It seems reasonable that situations would present themselves, particularly against humans or agents using "foresight-free" models [3], where an agent could cheaply teach an opponent a policy that is exploitable for a much larger amount.

The above list has highlighted only some of the opportunities in the area of online opponent modeling. Given the small amount of research in the area, there may be numerous additional fruitful avenues of research.

7. CONCLUSIONS

8. REFERENCES

[1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331. IEEE, 1995.

[2] N. Bard, M. Johanson, N. Burch, and M. Bowling. Online implicit agent modelling. *Autonomous Agents and Multiagent Systems (AAMAS'13)*, 2013.

[3] I. Erev and A. E. Roth. Multi-agent learning and the descriptive value of simple models. *Artificial intelligence*, 171(7):423–428, 2007.

[4] S. Ganzfried and T. Sandholm. Game theory-based opponent modeling in large imperfect-information games. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 533–540. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[5] S. Hart and A. Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 98(1):26–54, 2001.

[6] M. Johanson, N. Bard, M. Lanctot, R. Gibson, and M. Bowling. Efficient nash equilibrium approximation through monte carlo counterfactual regret minimization. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'12)*, pages 837–846. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[7] M. Johanson and M. Bowling. Data biased robust counter strategies. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS'09)*, pages 264–271, 2009.

[8] M. Johanson, M. Zinkevich, and M. Bowling. Computing robust counter-strategies. *Advances in neural information processing systems (NIPS'07)*, 20:721–728, 2007.

[9] M. Lanctot, K. Waugh, M. Zinkevich, and M. Bowling. Monte carlo sampling for regret minimization in extensive games. *Advances in Neural Information Processing Systems (NIPS'09)*, 22:1078–1086, 2009.

[10] F. Southey, M. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, and D. C. Rayner. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI'05)*, pages 550–558, 2005.

[11] F. Southey, B. Hoehn, and R. C. Holte. Effective short-term opponent exploitation in simplified poker. *Machine learning*, 74(2):159–189, 2009.

[12] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione. Regret minimization in games with incomplete information. *Advances in neural information processing systems (NIPS'08)*, 20:1729–1736, 2008.