The application of imperfect information game theory in social games

Ni Guo¹, and Zehui Qu¹*

¹ School of Computer and Information Science, Southwest University, Chongqing, 400700, P.R. China.
*Corresponding author’s e-mail: quzehui@swu.edu.cn

Abstract. The research on game AI has always been one of the focuses of artificial intelligence research. More complex real-world problems can be solved by the study of game AI. Since our cognition and information acquisition of the world are imperfect, we proposed single deep counterfactual regret minimization method with advantage baselines, a deep reinforcement learning algorithm based on imperfect information game. The new approach combines the single deep counterfactual regret minimization approach with the baseline network to achieve a better performance than the benchmark in poker. At last, this paper applies the new algorithm to two Chinese poker games with imperfect information game, and obtains better performance. Meanwhile, it also provides the possibility for the extension to real life.

1. Introduction

We consider the problem of calculating an ε-equilibrium in a two-player zero sum game with imperfect-information. This means that two agents play optimally against the other, and no agent defers to the other, looking for a policy that can’t be improved.

There are a number of deep reinforcement learning algorithms that calculate the best strategy for a single-agent setup and develop perfect-information games [11,15,18,20]. Nevertheless, these algorithms often fail to calculate the best strategy for an imperfect-information games.

In perfect information games, players usually try to play an optimal deterministic strategies. On the other hand, strong policy optimization algorithms for imperfect information games converge towards an ε-equilibrium, a distribution strategy characterized by reducing losses to worse competitors. The most common family of algorithms for finding these balance is Counterfactual Regret Minimization (CFR). Traditional CFR methods may cross the game tree to improve strategies taken in each state. For example, CFR + is the faster version of CFR used to solve the limited game of Texas hold'em poker, another version of poker that people usually play.

Recently, deep reinforcement learning algorithms have been formed in theory or empirically with an ε-equilibrium in two-player zero-sum imperfect-information games [16,26,28,32,36,37,39]. Of these, the neural form of counterfactual regret minimization (CFR) [4,6,9,29] have made the best performance [26,28,36]. This reflects the success of tabular CFR, whose version is in the state-of-the-art in tabular equilibrium-finding algorithms [29]. It has been used in every major AI milestone in poker [13,21,23,30].

While these abstract technologies have succeeded in poker, they require extensive knowledge in the field to build and are not suitable for all games. Existing forms of neural CFR are available only for accurate game simulators, allowing them to sample and explore multiple actions at one point of decision, thereby reducing variability.
In this article, we introduce SDRA, a neural form of CFR that displays only one action at each decision point. SDRA uses a sample-based learning-based neural form, which gives SDRA the characteristics of low variability when sampling individual actions.

SDRA minimizes regret and converges to an \( \epsilon \)-equilibrium in a two-player zero-sum game, proportional to neural modeling errors. We experimented to demonstrate that DRAM achieves the most advanced performance in model-free self-play deep reinforcement learning algorithms in two popular Chinese poker games: Doudizhu (DDZ) and Paodekuai (PDK).

2. Character and Background

Based on the partially observable stochastic strategy, we consider using \( N = \{1, 2, ..., N\} \) agents participating in the game. The agent index is represented by \( i \), and the rest of the proxy except \( i \) is represented by \(-i\). Trajectory (also known as history) is an ordered sequence of actions and states, represented by \( tr \). Final reward of agent \( i \) in the trajectory is denoted by \( R_i(tr) \), and legal behavior of agent \( i \) is denoted by \( A_i(tr) \). The information-state of agent \( i \) (also known as action-observation history, AOH) is a sequence of agent observations and action \( s_i \). The set of all information states of agent \( i \) is \( I_i \). The unique information-state of agent \( i \) containing trace \( tr \) is represented by \( s_i(tr) \). The trajectory set corresponding to the information state \( s_i \) is expressed as \( tr(s_i) \).

The agent policy \( \pi_i \) is a mapping function from the information-state to the probability distribution of actions. Policy profile \( \pi \) is a tuple \( (\pi_1, \pi_2, ..., \pi_N) \). In contrast, \( \pi_{-i} \) represents the policy of all agents except \( i \). The policy of trajectory \( tr \) is represented by \( \pi_i(tr) = \pi_i(s_i(tr)) \) and \( \pi(tr) = (\pi_1(s_1(tr)), \pi_2(s_2(tr)), ..., \pi_N(s_N(tr))) \). Define the transition function as \( T(tr, a_i, \pi_{-i}) \), plot the actions of the other agents according to the policy, form action \( a = a_i \cup a_{-i} \), and return from \( T(w_{last}, a) \) to trajectory \( tr' \). \( w_{last} \) is the last state of the world in trajectory \( tr \).

In track \( tr \), when policy profile \( \pi \) is used for all agents, the expected value (EV) of agent \( i \) is recorded as \( v_i(tr) \). For \( s_i \), EV is \( v_i(s_i) \), and for the entire game EV is \( v_i(\pi) \). EV of an action \( a_i \) in the trajectory is expressed as \( v_i(tr, a_i) \), and EV of an action \( a_i \) in the information-state is expressed as \( v_i(s_i, a_i) \). An \( \epsilon \)-equilibrium is a strategic arrangement in which no player can gain more \( \epsilon \) than the original benefit by unilaterally changing their strategy. Formally, for all agents \( i \), if \( v_i(\pi') \geq v_i(\pi, s_i, a_i) - \epsilon \) is satisfied, \( \pi' \) is an \( \epsilon \)-equilibrium.

3. Related Work

Due to the limitation of update strategy[4,6,9,29], tabular CFR if difficult to fully traverse for large games. An abstract scheme for a specific domain[10,14] can reduce the size of a game, but this is not applicable to many games, and they usually require specialized knowledge of a particular game.

The combination of neural networks and CFR was a natural solution to the problem of being difficult to handle large games. Neural Fictitious Self-Play (NFSP) [16] is the first deep reinforcement learning algorithm that learns Nash equilibrium of incomplete information game from self-play game. The policy gradient and actor critic methods have been shown to have similar properties when properly adjusted [22,39].

Deep CFR is a fully parameterized variant of CFR, which is better than NFSP in performance. SD-CFR[26] also eliminates the need for the average network in Deep CFR and shows better time and space complexity.

4. Review of Counterfactual Regret Minimization (CFR)

Counterfactual Regret Minimization (CFR) is an iterative algorithm. CFR can select synchronous updates or alternate updates. If the former is selected, CFR generates a new iteration policy \( \sigma_i' \) for all
players $i \in N$ for each iteration $t$. Conversely, player iterative strategies are updated alternately every two times.

To understand how CFR converge to a Nash equilibrium, we first define $r'_i(I,a)$ as the instantaneous regret value for any action $a \in A(I)$ of any player $i \in I$.

$$r'_i(I,a) = \pi^{\sigma'}_i(I)(v^{\sigma'}_i(I,a) - v^{\sigma'}_i(a))$$  \hspace{1cm} (1)

Where $v^{\sigma'}_i(I) = \sum_{i \in t} \pi^{\sigma'}_i(tr)^i u^{\sigma'}_i(tr)(\pi^{\sigma'}_i(I))^{-1}$ and $v^{\sigma'}_i(I,a) = \sum_{i \in t} \pi^{\sigma'}_i(tr)u^{\sigma'}_i(tr \rightarrow a)(\pi^{\sigma'}_i(I))^{-1}$. Instinctively, $r'_i(I,a)$ quantifies how many players $i$ can win if he always chooses $a$ in $I$ and reaches $I$ according to a $\sigma'$ game. In general, the regret of repetition $t$ is $R^T(i,a) = \sum_{i=1}^{T} r'_i(I,a)$. Then the repetition strategy for player $i$ can be derived from the following equation:

$$\sigma^T_i(I,a) = \left\{ \begin{array}{ll}
\frac{R^T_i(I,a) \left( \sum_{i \in A(I)} R^T_i(I,a) \right)^{-1}}{\sum_{i \in A(I)} R^T_i(I,a)} & \text{if } \sum_{i \in A(I)} R^T_i(I,a) > 0 \\
\sum_{i \in A(I)} R^T_i(I,a) & \text{otherwise} 
\end{array} \right.$$  \hspace{1cm} (2)

For most variants of CFR, the iterative policy profile $\sigma_i$ does not converge as $t$ grows. Mean strategy $\overline{\sigma}^T_i$ has been proven to be a strategy that converges to equilibrium. For all $I \in I$, and each $a \in A(I)$ it is defined as

$$\overline{\sigma}^T_i(I,a) = \sum_{i=1}^{T} \pi^{\sigma'}_i(I)\sigma^T_i(I,a)(\sum_{i=1}^{T} \pi^{\sigma'}_i(I))^{-1}.$$  \hspace{1cm} (3)

4.1. Discounted CFR (DCF)

Discounted CFR (DCF) has made some improvements to the $R^T(i,a)$ and $\overline{\sigma}^T_i$ equations. A special case of Deep CFR is linear CFR (LCFR), where the contribution of the instantaneous regret of iteration $t$ is weighted by $t$. That allow LCFR to become a two orders of magnitude faster than CFR in some big games.

4.2. Monte Carlo CFR (MC-CFR)

Regular CFR cannot be used in big games due to the need to cross the game tree. Monte Carlo CFR (MC-CFR) is one way to deal with this situation, in which each iteration crosses only part of the game tree. A sample $Q$ is used for a subset of a node $Q$ in the connected game tree based on how many $Q$ distribution each iteration. For sampled iterative information sets, $r'_i(I,a)$ equals $r'_i(I,a)$ of the probability of sampling $I$.

4.3. Deep CFR

The CFR approach is either not feasible for large games that require a complete game tree or not versatile when applying specific domain abstractions. Deep CFR approximates linear CFR by alternating player updating. This is sample-based, and have the advantage of not having to remember the regret table which can be applied to any two-player zero-sum game.

On each iteration, Deep CFR fitting a value network $\hat{D}_i$ for each player $i$ to approximate advantage, which is defined as $D^T_i(I,a) = R^T_{i,\text{linear}}(I,a)(\sum_{i=1}^{T} (t\pi^{\sigma'}_i(I)))^{-1}$, where $R^T_{i,\text{linear}}(I,a) = \sum_{i=1}^{T} (t\pi^{\sigma'}_i(I,a))$. In large games, Deep CFR divides $R^T_{i,\text{linear}}(I,a)$ by the total linear range $\sum_{i=1}^{T} (t\pi^{\sigma'}_i(I))$ to avoid the problem of learning difficulty due to the low probability of reaching a tree-branchings. This does not change the correctness of the result, but $\sum_{i=1}^{T} (t\pi^{\sigma'}_i(I))$ is the same for all $a \in A(I)$. 

3
Deep CFR has improved this point in order to select actions that have the greatest advantage in \( \sum_{a \in A(I)} D'_i(I,a) \leq 0 \). Deep CFR batches external sampling to obtain \( \hat{D} \) training data. All instantaneous regret values collected in \( N \) traverse are stored in the memory buffer \( B^t \). To simulate the behavior of linear CFR, it is necessary to weight the training loss between the \( \hat{D} \) prediction and the pity vector sampled at \( B^t \).

Once the training is complete, Deep CFR fits into another neural network \( \hat{S}_i(I,a) \) and approximates the linear mean strategy.

\[
\alpha^t_i(I,a) = \frac{D_i'(I,a)}{\left|A(I)\right|} \text{ if } \frac{D_i'(I,a)}{\left|A(I)\right|} > 0 \\
\alpha^t_i(I,a) \text{ otherwise}
\]

5. Description of SDRA

In this section, we will talking about SDRA, (S)ingle (D)eep (R)egret minimization with (A)dvantage Baselines. The main contribution of this paper is to effectively combine an external sample CFR using single deep neural network approximation with baseline.

Like Single Deep CFR, SDRA is trained from alternating CFR[31]. For iteration \( t \) of \( s \) of information-state, the parameterized advantage network \( \theta'_t \) is trained in \( s \) to obtain the estimated advantage \( \tilde{A}(s,a|\theta'_t) \) of external operation \( a \in A(s) \). \( \tilde{A}'_t \) is the alternative training network to \( R'_t \), and when training is complete (that is, when all the advantages are negative), it chooses the best action with probability 1.

5.1. Reducing uncertainty with baseline \( M \)

SDRA uses the basic \( \hat{M}_{i,t}(s'(tr)),a,\phi_t' \) network represented by \( \phi_t' \), where \( s' \) is a set of information-states for all players below \( tr \), \( s'(tr)=(s_1(tr),...,s_n(tr)) \). Initialize \( M'_t \) network with random weighting when \( t=0 \), and use \( M'_{t} \) to update \( M'_{t} \) when \( t>0 \). The target sample \( R'_t(tr) + \sum_{a \in A} \pi'_i(s'(tr),a),\tilde{Q}_i'(s'(tr'),a'_i) \) is updated to minimize the mean square error (MSE) obtained from \( T(h,a_1,\pi'_t) \).

Assuming that the externally sampled track records \( x \), the track \( tr \) on the baseline should be adjusted as follows:

\[
\approx_{v'_i} (tr,a_i| x) = \begin{cases} 
\tilde{Q}'_i(s'(tr'),a_i) - (v'_i(tr') - \tilde{Q}'_i(s'(tr'),a_i) - \epsilon)(\approx_i(s_1,a_i))^{-1} & \text{if } a_i = a'_i \\
\tilde{Q}'_i(s'(tr),a_i) & \text{otherwise}
\end{cases}
\]

\[
\approx_{v'_i} (tr| x) = \begin{cases} 
R_i(tr) & \text{if } tr = x \\
\sum_{a \in A} \pi'_i(tr,a_i)v'_i(tr,a_i|x) & \text{otherwise}
\end{cases}
\]

The advantage of instantaneous sampling is defined as

\[
A'_i(s,a_i) = \approx_{v'_i}(s_1,a_i) - \approx_{v'_i}(s_i)
\]

The expectation value of \( A'_i(s,a_i) \) is \( r'_i(s,a_i)(x'_i(s_1,a_i))^{-1} \).
6. Experimental Setup

Hyperparameters of SDRA are investigated and compared with SD-CFR and Deep CFR. We choose DDZ and PDK, two of the popular poker games in China, as indicators. For all the experiments, the mean values and standard deviations of three independent actions were statistic.

For each iteration, \( M^i \) trains a small set of 500 minibatches of 256 samples using an Adam optimizer [8] and with a learning rate of 0.005 for two games and the gradient norm to 1. SDRA uses the same training parameters as SD-CFR by default [26,28]. There are 400,000 samples in DDZ and 200,000 samples in PDK. Value network has two games with 300 batches 1024 samples and 1000 batches 256 samples respectively. For an average SDRA and DCFR network, add a buffer \( B^i \) of 200,000 samples to each player. The Average network is trained for 400 batches of 512 samples.

In DDZ, the number of SDRA sampling traverses in one iteration is 500, and SD-CFR is 300. In PDK, SDRA and SD-CFR are repeated 1,000 and 500 times respectively.

7. Experimental Result

First, figure 1 and figure 2 show the hyperparameters of SDRA and other algorithms. SDRA was found to be slightly better than the other algorithms in two games. This may be because \( M \) changes slightly with each iteration. Deep CFR performance will improve once the value network training is complete.

![Figure 1. Ablation studies in PDK. Left: Comparing for SDRA. Right: Different numbers of minibatch of baseline.](image-url)
8. Conclusion
In this study, we have introduced SDRA, an algorithm based on single deep counterfactual minimization and self-play deep reinforcement learning. In two-player zero-sum imperfect information games, the SDRA converges to ε-equilibrium. In practice, the SDRA performs slightly better in the RL algorithm than the existing baseline NFSP [16].

References
[1] Vitter, J. S. (1985). Random sampling with a reservoir. ACM Transactions on Mathematical Software (TOMS), 11(1), 37-57.
[2] Hansen, E. A., Bernstein, D. S., & Zilberstein, S. (2004, July). Dynamic programming for partially observable stochastic games. In AAAI (Vol. 4, pp. 709-715).
[3] Southey, F., Bowling, M. P., Larson, B., Piccione, C., Burch, N., Billings, D., & Rayner, C. (2012). Bayes' bluff: Opponent modelling in poker. arXiv preprint arXiv:1207.1411.
[4] Zinkevich, M., Johanson, M., Bowling, M. H., & Piccione, C. (2007, September). Regret Minimization in Games with Incomplete Information. In NIPS (Vol. 7, p. 1729).
[5] Van Seijen, H., Van Hasselt, H., Whiteson, S., & Wiering, M. (2009, March). A theoretical and empirical analysis of Expected Sarsa. In 2009 ieee symposium on adaptive dynamic programming and reinforcement learning. Nashville. (pp. 177-184).
[6] Lanctot, M., Waugh, K., Zinkevich, M., & Bowling, M. H. (2009, January). Monte Carlo Sampling for Regret Minimization in Extensive Games. In NIPS (pp. 1078-1086).
[7] Lanctot, M. (2013). Monte Carlo sampling and regret minimization for equilibrium computation and decision-making in large extensive form games. (doctoral dissertation).
[8] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
[9] Tammelin, O. (2014). Solving large imperfect information games using CFR+. arXiv preprint arXiv:1407.5042.
[10] Ganzfried, S., & Sandholm, T. (2014, June). Potential-aware imperfect-recall abstraction with earth mover's distance in imperfect-information games. In Proceedings of the AAAI Conference on Artificial Intelligence. Québec City. (Vol. 28, No. 1).
[11] Van Hasselt, H., Guez, A., & Silver, D. (2016, March). Deep reinforcement learning with double q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence. Phoenix. (Vol. 30, No. 1).
[12] Waugh, K., Morrill, D., Bagnell, J., & Bowling, M. (2015, February). Solving games with
functional regret estimation. In Proceedings of the AAAI Conference on Artificial Intelligence. Austin. (Vol. 29, No. 1).

[13] Bowling, M., Burch, N., Johanson, M., & Tammelin, O. (2015). Heads-up limit hold’em poker is solved. Science, 347(6218), 145-149.

[14] Brown, N., Ganzfried, S., & Sandholm, T. (2015, April). Hierarchical Abstraction, Distributed Equilibrium Computation, and Post-Processing, with Application to a Champion No-Limit Texas Hold' em Agent. In AAAI Workshop: Computer Poker and Imperfect Information.

[15] Wang, Z., Schaul, T., Hassel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016, June). Dueling network architectures for deep reinforcement learning. In International conference on machine learning. Phoenix. (pp. 1995-2003). PMLR.

[16] Heinrich, J., & Silver, D. (2016). Deep reinforcement learning from self-play in imperfect-information games. arXiv preprint arXiv:1603.01121.

[17] Lisy, V., & Bowling, M. (2016). Equilibrium approximation quality of current no-limit poker bots. arXiv preprint arXiv:1612.07547.

[18] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.

[19] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). Mastering the game of go without human knowledge. Nature, 550(7676), 354-359.

[20] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

[21] Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., ... & Bowling, M. (2017). Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. Science, 356(6337), 508-513.

[22] Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Pérolat, J., ... & Graepel, T. (2017). A unified game-theoretic approach to multiagent reinforcement learning. arXiv preprint arXiv:1711.00832.

[23] Brown, N., & Sandholm, T. (2018). Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. Science, 359(6374), 418-424.

[24] Jin, P., Keutzer, K., & Levine, S. (2018, July). Regret minimization for partially observable deep reinforcement learning. In International conference on machine learning Stockholm (pp. 2342-2351). PMLR.

[25] Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., ... & Silver, D. (2018, April). Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence. New Orleans. (Vol. 32, No. 01, pp. 1829-1836).

[26] Steinberger, E. (2019). Single deep counterfactual regret minimization. arXiv preprint arXiv:1901.07621.

[27] Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., ... & Bowling, M. (2020). The hanabi challenge: A new frontier for ai research. Artificial Intelligence, 280, 103216.

[28] Brown, N., Lerer, A., Gross, S., & Sandholm, T. (2019, May). Deep counterfactual regret minimization. In International conference on machine learning. Long Beach. (pp. 793-802). PMLR.

[29] Brown, N., & Sandholm, T. (2019, July). Solving imperfect-information games via discounted regret minimization. In Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu. (Vol. 33, No. 01, pp. 1829-1836).

[30] Brown, N., & Sandholm, T. (2019). Superhuman AI for multiplayer poker. Science, 365(6456), 885-890.

[31] Burch, N., Moravcik, M., & Schmid, M. (2019). Revisiting CFR+ and alternating updates. Journal of Artificial Intelligence Research, 64, 429-443.
[32] Hennes, D., Morrill, D., Omidshafiei, S., Munos, R., Perolat, J., Lanctot, M., ... & Tuyls, K. (2019). Neural replicator dynamics. arXiv preprint arXiv:1906.00190.

[33] Davis, T., Schmid, M., & Bowling, M. (2020, November). Low-Variance and Zero-Variance Baselines for Extensive-Form Games. In International Conference on Machine Learning. Vienna. (pp. 2392-2401). PMLR.

[34] Kovařík, V., Schmid, M., Burch, N., Bowling, M., & Lisý, V. (2019). Rethinking formal models of partially observable multiagent decision making. arXiv preprint arXiv:1906.11110.

[35] Farina, G., Kroer, C., & Sandholm, T. (2020, November). Stochastic regret minimization in extensive-form games. In International Conference on Machine Learning. Vienna. (pp. 3018-3028). PMLR.

[36] Li, H., Hu, K., Ge, Z., Jiang, T., Qi, Y., & Song, L. (2018). Double neural counterfactual regret minimization. arXiv preprint arXiv:1812.10607.

[37] Perolat, J., Munos, R., Lespiau, J. B., Omidshafiei, S., Rowland, M., Ortega, P., ... & Tuyls, K. (2020). From Poincaré Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization. arXiv preprint arXiv:2002.08456.

[38] Schmid, M., Burch, N., Lanctot, M., Moravcik, M., Kadlec, R., & Bowling, M. (2019, July). Variance reduction in monte carlo counterfactual regret minimization (VR-MCCFR) for extensive form games using baselines. In Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu. (Vol. 33, No. 01, pp. 2157-2164).

[39] Srinivasan, S., Lanctot, M., Zambaldi, V., Pérolat, J., Tuyls, K., Munos, R., & Bowling, M. (2018). Actor-critic policy optimization in partially observable multiagent environments. arXiv preprint arXiv:1810.09026.