TEMPERATURE REGRET MATCHING FOR IMPERFECT-INFORMATION GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual regret minimization (CFR) methods are effective for solving two player zero-sum extensive games with imperfect information. Regret matching (RM) plays a crucial role in CFR and its variants to approach Nash equilibrium. In this paper, we present Temperature Regret Matching (TRM), a novel RM algorithm that adopts a different strategy. Also, we consider not only the opponent's strategy under the current strategy but also the opponent's strategies of the several last iterations for updating the external regret of each iteration. Furthermore, we theoretically demonstrate that the update of TRM converges to Nash Equilibrium. Competitive results in imperfect-information games have verified its effectiveness and efficiency.

1 INTRODUCTION

Games in extensive form provide a mathematical framework for modeling the sequential decision-making problems with imperfect information. We focus on solving poker games, a common benchmark for two-player zero-sum imperfect-information games. For these games, the goal is to find a Nash Equilibrium (NE) (Nash, 1950).

The most popular algorithms to solve this problem are variants of counterfactual regret minimization (CFR) (Zinkevich et al., 2007; Lanctot et al., 2009; Gibson et al., 2012). In pariticular, the development of CFR+ (Tammelin, 2014; Tammelin et al., 2015) and LCFR Brown & Sandholm (2019) provides stronger baselines than vanilla CFR. CFR+ was used to solve heads-up limit Texas hold'em (Bowling et al., 2015) and heads-up no-limit Texas hold'em (HUNL) (Moravcik et al., 2017; Brown & Sandholm, 2018). It helps agents defeat poker professionals. LCFR accelerates the learning process and outperforms CFR+ in HUNL subgames and 5-card Goofspiel. These immediate counterfactual regrets are defined by the counterfactual rewards and can be iteratively minimized by regret matching (RM) (Blackwell, 1956; Hart & Mascolell, 2000; Abernethy et al., 2011) or AdaHedge (Freund & Schapire, 1997; De Rooij et al., 2014).

In this work, we first focus on RM in CFR. We prove that any strategies which are inversely positively to the external regret can help the CFR algorithm converge to an ε-equilibrium (Nesterov, 2005). Furthermore, we prove that any combinatorial regret matching methods converge to an ε-equilibrium. Based on the above conclusion, we propose Temperature Regret Matching (TRM), which is a method to change the matching weight of external regret according to the number of iterations.

Secondly, we consider that the strategy at this iteration must be able to win the last iteration because regret matching methods increase the action weight to obtain the expected return. Inspired by Optimistic CFR variants (Syrgkanis et al., 2015; Brown & Sandholm, 2019) which the last iteration is counted twice when calculating the strategy can lead to substantially faster convergence , we consider not only the opponent's strategy under the current strategy but also the opponent's strategy of the last iteration for the external regret of each iteration. Also, we give the reason for this regret matching method.

To summarize, the main contributions of this work are listed bellow in three-fold:

- We prove that any strategies which are inversely positively to the external regret can help the CFR algorithm converge to an ε-equilibrium. Furthermore, we prove that any combinatorial regret matching methods converge to an ε-equilibrium.
- We propose a new regret matching method Temperature Regret Matching (TRM) which adopts a different strategy to obtain the output strategies, which helps the average strategy converges faster.
- We consider not only the opponent's strategy under the current strategy but also the opponent's strategy of the last moment for the external regret of each iteration, which further improves the learning efficiency.

We empirically evaluate TRM in the vanilla CFR (Zinkevich et al., 2007), CFR+ (Tammelin, 2014) , LCFR (Brown & Sandholm, 2019), and Optimistic CFR on the standard benchmarks Leduc Hold'em. Extensive experimental analyses and comparisons demonstrate the effectiveness of Temperature Regret Minimization.

2 NOTATIONS AND PERLIMINARIES

In this section, we first introduce the notations and definitions of imperfect-information games in extensive form. Then we introduce Best Response (BR) and Nash Equilibrium (NE). Also, we introduce Regret Matching (RM), CFR , CFR+, Linear CFR, and Optimistic CFR.

2.1 NOTATIONS

Extensive games compactly model the decision-making problems with sequential interactions among multiple agents. In extension form games, there is a finitset of players: $\mathcal{P}$. An extension form game can be represented by a game tree H of histories, where a history h is a sequence of actions in the past. A player function P assigns a member of $\mathcal{P} \cup c$ to each non-terminal history, where c is the chance. $A(h)$ is the actions available at a history and $P(h)$ is the player who acts at this history. If $P(h) = c$ then chance determines the action taken after history h. If a player take an action a at history h and reach an new history h', we represent this as $h \cdot a = h'$. $Z \subseteq H$ are terminal histories in which no actions are available. For each player $i \in \mathcal{P}$, there is a payoff function $u_i : Z \to R$. $\triangle_i$ is the range of payoffs reachable by player i. Furthermore, $\triangle_i = \max_{z \in Z} u_i(z) - \min_{z \in Z} u_i(z)$ and $\triangle = \max_i \triangle_i$. If $\mathcal{P} = 1, 2$ and $u_1 + u_2 = 0$, the game is two-player zero-sum.

In imperfect-information games, imperfect information is denoted by information set(infoset) I_i for each player $i \in \mathcal{P}$. All states $h \in I_i$ are indistinguishable to i. Let $A = \max_h |A(h)|$. A strategy of player i is a function σ_i for player i in infoset I. The probability of a particular action a is denoted by $\sigma_i(I, a)$. Since all histories in an information set belonging to player i are indistinguishable, the strategies in each of them must be identical. For all $h \in I, \sigma_i(h) = \sigma_i(I)$ and $\sigma_i(h, a) = \sigma_i(I, a)$. We define σ_i to be a probability vector for player i over all available strategies in the game. A strategy profile $\sigma = \{\sigma_i|, \sigma_i \in \Sigma_i, i \in \mathcal{P}\}$ is a collection of strategies for all players. Σ_i is the set of all possible strategies for player i. The strategy of all players other than player i is presented as σ_{-i}. $u_p(\sigma_i, \sigma_{-i})$ is the expected payoff for i if player i plays according to σ_i and the other players play according to σ_{-i}.

Let $\pi^\sigma(h)$ be the probability of history h occurring if players choose actions according to σ. We can decompose $\pi^\sigma(h) = \Pi_{i \in N \{c\}} \pi^\sigma(h)$ into each player's contribution to this probability. Hence, $\pi_i^\sigma(h)$ is the probability that if player i plays according to σ if all players other than i, and chance always chose actions leading to h. Let $\pi_{-i}^\sigma(h)$ be the product of all players' contribution (including chance) except player i. In this paper we focus on perfect-recall games. Therefore, for $i = P(I)$ we define $\pi_i(I) = \pi_i(h)$ for $h \in I$.

2.2 BEST RESPONSE AND NASH EQUILIBRIUM

A best response to σ_i is a strategy $BR(\sigma_i)$ satisfies that

$$u_i(\sigma^i, BR(\sigma_i)) = \max_{\sigma'_{-i}} u_i(\sigma^i, \sigma'_{-i}). \tag{1}$$

Nash equilibrium is a strategy profile where everyone plays a best response. A Nash equilibrium is a strategy profile σ satisfies:

$$u_1(\sigma) \geq \max_{\sigma_1^{'} \in \Sigma_1} u_1(\sigma_1^{'}, \sigma_2) \qquad u_2(\sigma) \geq \max_{\sigma_2^{'} \in \Sigma_2} u_2(\sigma_1, \sigma_2^{'}). \tag{2}$$

An approximation of a Nash equilibrium or ϵ-Nash equilibrium in a two-player extensive game is a strategy profile σ satisfies:

$$u_1(\sigma) + \epsilon \geq \max_{\sigma_1^{'} \in \Sigma_1} u_1(\sigma_1^{'}, \sigma_2) \qquad u_2(\sigma) + \epsilon \geq \max_{\sigma_2^{'} \in \Sigma_2} u_2(\sigma_1, \sigma_2^{'}). \tag{3}$$

The ϵ-NE in an extensive game can be efficiently computed by regret minimization. The exploitability of a strategy (σ_1, σ_2) can be interpreted as the approximation error to the Nash equilibrium. The exploitability is defined as

$$\max_{\sigma^{'},^1} u_1(\sigma^{'},^1, \sigma^2) + \max_{\sigma^{'},^2} u_2(\sigma^1, \sigma^{'},^2). \tag{4}$$

2.3 Regret Matching(RM) and Counterfactual Regret Minimization (CFR)

Counterfactual regret minimization (CFR) is an equilibrium finding algorithm for extensive games that minimizes regret in each infoset (Zinkevich et al., 2007). Regret matching (RM) is the most popular option (Blackwell, 1956). Regret is an online learning concept that has triggered many powerful learning algorithms. These learning algorithms minimize some kinds of regrets, known as regret minimization algorithms such as regret matching and AdaHedge. To define this concept, we first consider repeatedly playing an extensive game. Let Σ_a be the set of valid actions. At each iteration, the player selects an action a_t and get a reward CFR makes frequent use of counterfactual value v, which is the expected utility of an infoset if a player i tries to reach it. Given a strategy profile σ and a strategy σ, the counterfactual value $v^\sigma(I)$ for an infoset I and the counterfactual value of a special action a in this infoset are defined as:

$$v^\sigma(I) = \sum_{h \in I} \left(\pi_{-i}^\sigma(h) \sum_{z \in Z} (\pi_\sigma(h, z) u_i(z)) \right), \tag{5}$$

$$v^\sigma(I, a) = \sum_{h \in I} \left(\pi_{-i}^\sigma(h) \sum_{z \in Z} (\pi_\sigma(h \cdot a, z) u_i(z)) \right). \tag{6}$$

We define σ^t as the strategy profile used on iteration t. The regret $r^t(I, a)$ for action a in infoaet I at iteration T is :

$$r^t(I, a) = v^{\sigma^t}(I, a) - v^{\sigma^t}(I). \tag{7}$$

The whole regret R^T in T iterations is:

$$R^T(I, a) = \sum_{t=1}^{T} r^t(I, a). \tag{8}$$

Let $R_+^T(I, a) = \max\{R^T(I, a), 0\}$. In RM, player i selects actions $a \in A(I)$ on each iteration T according to probabilities:

$$\sigma^T(I, a) = \begin{cases} \frac{R_+^T(I,a)}{\sum_{a' \in A(I)} R_+^T(I,a')}, & \sum_{a'} R_+^T(I, a') > 0 \\ \frac{1}{|A(I)|}, & otherwise. \end{cases} \tag{9}$$

According to RM in infoset I, the regret on interation T satisfies $R_i^T \leq \triangle \sqrt{|A(I)|} \sqrt{T}$ (Zinkevich et al., 2007). The time average strategy $\overline{\sigma}_T^i(I) = \frac{\sum_t \pi_{\sigma_t}^i(I) \sigma_t^i(I)}{\sum_t \pi_{\sigma_t}^i(I)}$ converges to an ϵ-NE. CFR+ is a variant of CFR with the two small changes. First, CFR+ set all negative regret to 0. Formally, CFR+ defines the regret-like value Q as $Q^T(I, a) = \max\{Q^{T+1}(I, a) + r^t(I, a), 0\}$ and uses it as $R_+^T(I, a)$. CFR+ uses strategy on iteration T according to Regret Matching+ (RM+) rather than RM. Second, CFR+ uses a weighted average strategy $\overline{\sigma^{'}}_T^i(I) = \frac{\sum_t t \pi_{\sigma_t}^i(I) \sigma_t^i(I)}{\sum_t t \pi_{\sigma_t}^i(I)}$ rather than using a average strategy as CFR. Linear CFR is a varient of CFR which multiplies the external regret by $\frac{t}{t+1}$ on iteration T. It means the iteration t regret has a weight $\frac{2t}{T^2+T}$ in iteration T. Optimistic CFR counts the regret and uses a modified regret $R_{mod}^T = \sum_{t=1}^T r^t(I, a) + 2r^T(I, a)$ rather than R^T.

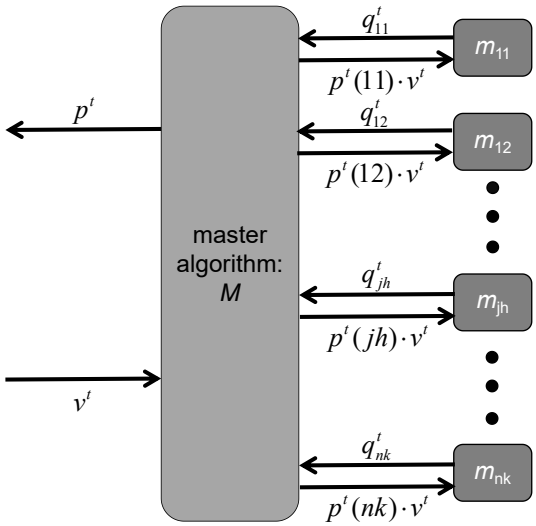


Figure 1: From swap regret to external regret.

3 TEMPERATURE REGRET MATCHING

In this section, we propose Temperature Regret Matching (TRM), an efficient RM method that adopts a different strategy to obtain the output strategies using a temperature perliminary. In all past variants of CFR, each iteration strategy is given by regret matching using external regret. We discuss using different RM methods in CFR when determining strategies. We first theoretically demonstrate that different regret matching methods converge to an ε-equilibrium. We give our regret matching methods.

$$\sigma^T(I,a) = \begin{cases} \frac{(R^T_+(I,a))^\beta}{\sum_{a'\in A(I)}(R^T_+(I,a'))^\beta}, & \sum_{a'} R^T_+(I,a') > 0, \\ \frac{1}{|A(I)|}, & otherwise; \end{cases} \tag{10}$$

in which $\beta > 0$ is a hyper-parameter. This regret matching method can converge to an ε-equilibrium (Cesa-Bianchi & Lugosi, 2006). Furthermore, when we set different β for different iteration regret matching, it can also converge to an ε-equilibrium. We demonstrate that different regret matching methods (in which we can change $beta$ during calculating output strategies) can converge to an ε-equilibrium. This part follows that if there is a no-external-regret algorithm, then there is a no-swap-regret algorithm. Our hope is that we can change β randomly during calculating strategies. Let n denote the number of actions. Suppose there are $n * k$ different no-external-regret algorithms $(M_1 = (m_{11}, m_{12}, ..., m_{1k}), M_2 = (m_{21}, m_{22}, ..., m_{2k}), ..., M_n = (m_{n1}, m_{n2}, ..., m_{nk}))$ with different RM parameter β. In which M_j is the set of algorithm for action j. The master algorithm M(shown as Fig. 1) is: 1) Receive distributions $q^t_1, ..., q^t_n$ over actions from $M_1, M_2, ..., M_n$. 2) Compute and outpur a consensus distribution p^t. 3) Receive a counterfactual vector v^t from the adversary. 4) Give algorithm m_{jh} the counterfactual vector $p^t(jh) \cdot v^t$. Let $\delta : A \to A$ be a switching function. The time-average expected value of the master algorithm M is :

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{h=1}^{k} p^t(ih)v^t(i). \tag{11}$$

The time-average expected value under a switching function $\delta : A \to A$ is :

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{h=1}^{k} p^t(ih)\cdot v^t(\delta(i)). \tag{12}$$

We need to prove that when $t \to \infty$, Eqn.(11) and Eqn.(12) should be equivalent. For a set algorithm M_j, actions are chosen according to its recommended distributions $q^1_{jh}, ..., q^T_{jh}$ and the expected values are $p^1(jh) \cdot c^1, ..., p^T(jh) \cdot c^T$. For an algorithm m_{jh}, the expected values are $p^T(jh) \cdot v^T$.

Algorithm M_j receives its time-average value is:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{h=1}^{k} q_{jh}^{t}(i)\left(p^{t}(jh)v^{t}(i)\right), \tag{13}$$

Because m_{jh} is a no-regret algorithm, the perceived value of each fixed action $a \in A$ should be smaller than:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{h=1}^{k} p^{t}(jh)v^{t}(a) + R_{jh}, \tag{14}$$

as $T \to \infty$, $R_j \to 0$. Now fix a switching function δ. According to Eqn.(13) and Eqn.(14), we can get:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{h=1}^{k} q_{jh}^{t}(i)p^{t}(jh)v^{t}(i) \leq \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{n}\sum_{h=1}^{k} p^{t}(jh)v^{t}(\delta(j)) + \sum_{j=1}^{n}\sum_{h=1}^{k} R_{jh}. \tag{15}$$

As $t \to \infty$, $\sum_{j=1}^{n}\sum_{h=1}^{k} R_{jh}$ should be close to 0, the right part of Eqn.(15) should be equal to Eqn.(12). So we can choose the splitting of the value vector v^t among the no-external-regret algorithms m_{jh}. The p^t for each action $i \in A$ and each iteration t should be choosen as:

$$p^{t}(ih) = \sum_{j=1}^{n}\sum_{h=1}^{k} q_{jh}^{t}(i)p^{t}(jh). \tag{16}$$

It means we can first ask m_1 algorithm for a recommended strategy σ_t then use the regret R^T ask m_2 algorithm for a recommended strategy. When an action with a higher regret, we need to choose it with higher probability. Due to $\beta > 0$, we give our temperature regret matching algorithm:

$$\sigma^{T}(I,a) = \begin{cases} \frac{(R_{+}^{T}(I,a))^{\alpha+\gamma t}}{\sum_{a'\in A(I)}(R_{+}^{T}(I,a'))^{\alpha+\gamma t}}, & \sum_{a'} R_{+}^{T}(I,a') > 0 \\ \frac{1}{|A(I)|}, & otherwise. \end{cases} \tag{17}$$

Here α, γ are two hyperparameters α should be close to 1 and γt should be close to 0. We give a temperature parameter γ to RM as a way to increase noise. When we update the strategies using TRM using T iterations, the average of the hyper-parameter $\beta = \alpha - \frac{T\gamma}{2}$ should equal to 1, which can be compared with RM under the same settings. This setting can help agents have a certain chance of producing more effective strategies.

4 A NEW REGRET UPDATING METHOD

In this part, we give our novel regret updating method. Optimistic CFR (Brown & Sandholm, 2019) counts the regret and uses a modified regret $R_{mod}^{T} = \sum_{t=1}^{T} r^{t}(I,a) + 2r^{T}(I,a)$ rather than R^T and leads to faster converge. Linear CFR calcluate the t iteration regret has a weight $\frac{2t}{T^2+T}$ after T iterations. However, the regret r^t gained in each iteration t is obtained through the virtual games between the current iteration of strategy $\left(\sigma_i^t, \sigma_{-i}^t\right)$ of player i and his/her opponent player $-i$. In this section, we consider not only the strategy of our opponent in the current iteration but also the strategy of our opponent in different previous iterations. When player $i's$ own strategy σ_i^t fights with all previous iterations of opponent strategies $\sigma_{-i}^1, \sigma_{-i}^2, ..., \sigma_{-i}^t$, player i gets t regrets in the current iterations, which is somewhat similar to LCFR's weight of t for the current regret. However, saving all past strategies will consume too many storage resources, and calculating t regret values will also consume too many computing resources. To reduce the computational storage complexity, we save the strategies of both sides of the last iteration of the game and update the regret values with the opponent strategies of this iteration and the last iteration. We redefine two regret in iteration t for player i, the regret $r_i^{\sigma_i^t, \sigma_{-i}^{t-1}}(I_i, a)$ for player i in iteration t using σ_i^t and the opponent player $-i$ using σ_{-i}^{t-1} is:

$$r_i^{\sigma_i^t, \sigma_{-i}^{t-1}}(I_i, a) = v_i^{\sigma_i^t, \sigma_{-i}^{t-1}}(I_i, a) - v_i^{\sigma_i^t, \sigma_{-i}^{t-1}}(I_i). \tag{18}$$

The regret $\sigma_i^t r_i^{\sigma_i^t,\sigma_{-i}^t}(I_i,a)$ for player i in iteration t using σ_i^t and the opponent player $-i$ using σ_{-i}^t is:

$$r_i^{\sigma_i^t,\sigma_{-i}^t}(I_i,a) = v_i^{\sigma_i^t,\sigma_{-i}^t}(I_i,a) - v_i^{\sigma_i^t,\sigma_{-i}^t}(I_i). \tag{19}$$

The regret in iteration $r_i^{\sigma_i^t}(I_i,a)$ is:

$$r_i^{\sigma_i^t}(I_i,a) = r_i^{\sigma_i^t,\sigma_{-i}^{t-1}}(I_i,a) + r_i^{\sigma_i^t,\sigma_{-i}^t}(I_i,a). \tag{20}$$

This way of calculating regret value has two advantages. First, in the traditional way of updating regret, player i increases the weight for actions with large counterfactual values to ensure that the expected return of the current strategy against the previous iteration of opponent strategy continues to increase, simultaneously, player $-i$ reduces the expected return of the current player in the same way. Expressed mathematically as follows:

$$u_i(\sigma_t^i,\sigma_{t-1}^{-i}) \geq u_i(\sigma_t^i,\sigma_t^{-i}) \geq u_i(\sigma_{t-1}^i,\sigma_t^{-i}). \tag{21}$$

When we save the strategies of the last iteration and updating the regret using last two strategies, the expected return of player i in $t, t+1$th interation satisfies that:

$$u_i(\sigma_t^i,\sigma_{t-1}^{-i}) \geq u_i(\sigma_t^i,\sigma_t^{-i}) \geq u_i(\sigma_{t-1}^i,\sigma_t^{-i}), \tag{22}$$

$$u_i(\sigma_{t+1}^i,\sigma_{t-1}^{-i}) \geq u_i(\sigma_{t+1}^i,\sigma_{t+1}^{-i}) \geq u_i(\sigma_{t-1}^i,\sigma_{t+1}^{-i}). \tag{23}$$

Second, when the CFR+/Linear CFR algorithm iterates many times, according to experience, each iteration of its output strategy is close to the Nash equilibrium. We consider playing against several opponents close to the Nash equilibrium, which can help the algorithm converge faster. An overall description of our updating and regret matching is shown in Algorithm 1.

Algorithm 1: Temperature Regret Matching and the new regret updating method.

for t=1 *to* T **do**

$v_i^{\sigma^t}(I_i) = \sum_{h\in I_i, h\sqsupseteq z, z\in Z} \pi_i^{\sigma^t}(h,z)\pi_{-i}^{\sigma^t}(z)u_i(z)$.

$r_i^{\sigma_i^t,\sigma_{-i}^{t-1}}(I_i,a) = v_i^{\sigma_i^t,\sigma_{-i}^{t-1}}(I_i,a) - v_i^{\sigma_i^t,\sigma_{-i}^{t-1}}(I_i)$.

$r_i^{\sigma_i^t,\sigma_{-i}^t}(I_i,a) = v_i^{\sigma_i^t,\sigma_{-i}^t}(I_i,a) - v_i^{\sigma_i^t,\sigma_{-i}^t}(I_i)$.

$R_i^t(I_i,a) = R_i^{t-1}(I_i,a) + r_i^{\sigma_i^t,\sigma_{-i}^{t-1}}(I_i,a) + r_i^{\sigma_i^t,\sigma_{-i}^t}(I_i,a)$.

$\sigma^T(I,a) = \begin{cases} \frac{(R_+^T(I,a))^{\alpha+\gamma t}}{\sum_{a'\in A(I_i)}(R_+^T(I,a'))^{\alpha+\gamma t}}, & \sum_{a'} R_+^T(I,a') > 0, \\ \frac{1}{|A(I)|}, & otherwise. \end{cases}$

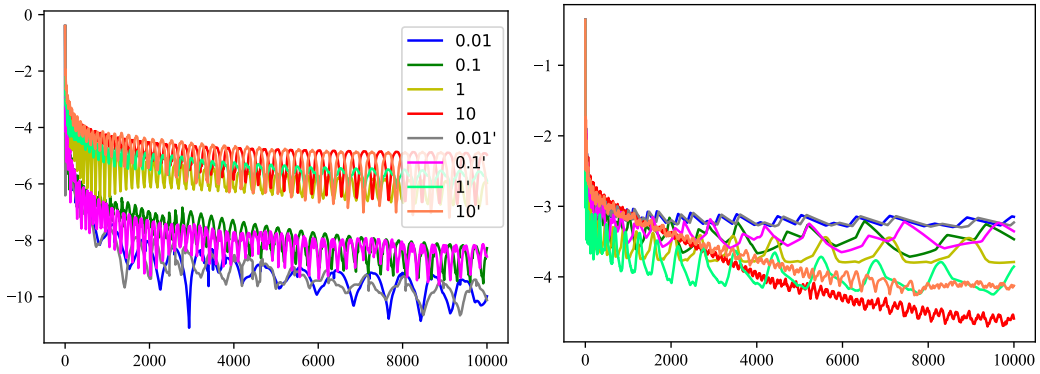
$S^t(I_i,a) = S^{t-1}(I_i,a) + \pi_i^{\sigma^t}(I_i)\sigma_i^t(I_i,a)$.

end

$\overline{\sigma_i}^T(I_i,a) = \frac{S^T(I_i,a)}{\sum_{a'\in A(I_i)} S^T(I_i,a')}$.

5 EXPERIMENTS

To verify the effectiveness of our proposed TRM algorithm and regret matching method, we evaluate their performances on some games such as Rock-Paper-Scissors, a modified form of Rock-Paper-Scissors in which the winner receives two points and the loser loses two points when either player chooses Scissors, Leduc Hold'em, and Big Leduc Hold'em against existing CFR variants. Leduc Hold'em a two-players IIG of poker introduced in Southey et al. (2005). In Leduc Hold'em, there is a deck of six cards that includes two suites, each with three ranks. The cards are often denoted by king, queen and jack. The game has a total of two rounds. Each round has a maximum of two raises per round. Each player gets a private card in the first round and the opponent's card is hidden. In the second round, another card is dealt with as a community card, and the information about this card is open to both players. If a player's private card is paired with the community card, that player wins the game; otherwise, the player with the highest private card wins the game. Both players bet



(a) Rock-Paper-Scissors (b) A modified form of Rock-Paper-Scissors

Figure 2: The **left** and **right** shows distances (log10) between the strategies and Nash equilibriumin the two games when $\beta \in \{0.01, 0.1, 1(RM), 10, a_{0.01}, a_{0.1}, a_1, a_{10}\}$. The results are reported by using in 10000 itearations. a' represents a value $\beta \in (0, 2a)$ that is randomly set in each iteration.

one chip into the pot before the cards are dealt. Moreover, a betting round follows at the end of each dealing round. There are four kinds of actions: fold (End this game, the other gets all the pot.), call (Increase his/her bet until both players have the same chips.), check (Do not action.), and bet (Add some chips to the pot.) in Leduc Hold'em. In Leduc Hold'em, the player may wager any amount of chips up to a maximum of that player's remaining stack. There is also no limit on the number of raises or bets in each betting round. The Big Leduc Hold'em has the same rules as Leduc Hold'em and uses a deck of twenty-four cards with twelve ranks. In addition to the larger size of the state space, BigLeduc allows a maximum of six instead of two raises per round.

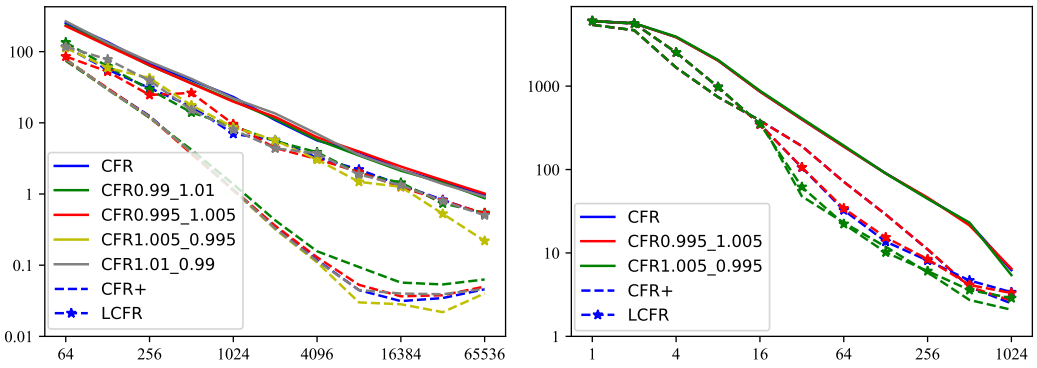
5.1 Ablation Studies

We take Rock-Paper-Scissors and the modified form of Rock-Paper-Scissors as the experimental environments for our ablation studies. We empirically evaluate our algorithm TRM against vanilla RM. We verify that when we set different β for different iterations regret matching, it can also converge to an ε-equilibrium. We set β to different values, *i.e.*, $\beta \in \{0.01, 0.1, 1(RM), 10, a_{0.01}, a_{0.1}, a_1, a_{10}\}$ to test the sensitivity of the algorithm. a_x represents a value $\in (0, 2x)$ that is randomly set in each iteration. We measure the distance between the strategies and Nash equilibrium ($(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ in Rock-Paper-Scissors and $(\frac{2}{5}, \frac{2}{5}, \frac{1}{5})$ in the modified form Rock-Paper-Scissors) of these algorithms. Figure 2 (left) shows the performance of different β in Rock-Paper-Scissors and Figure 2 (right) shows the performance of different β in the modified form Rock-Paper-Scissors. These results validate that RM converges to an ε-equilibrium setting different β for different iterations.

5.2 Poker Games

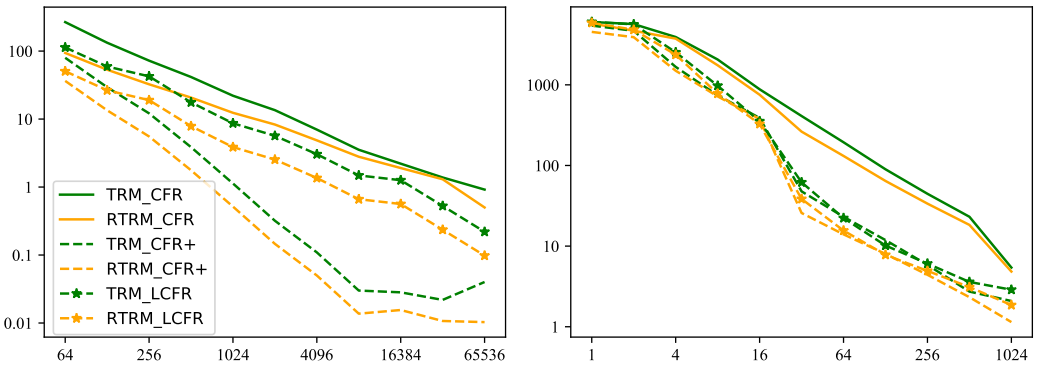
In this section, we empirically evaluate our algorithm against existing CFR variants. Since we do not know what Nash equilibrium strategies are, we measure the exploitability of these algorithms. Experiments are conducted on variants of two common benchmarks in imperfect-information game solving: Leduc Hold'em and Big Leduc Hold'em.

We run each algorithm on each Leduc Hold'em game for 65536 iteartions in Leduc Hold'em and 1024 iteartions in Big Leduc Hold'em. We empirically compare TRM-(CFR, CFR+, LCFR) with existing methods vanilla CFR (Zinkevich et al., 2007), CFR+ (Tammelin, 2014), and LCFR (Brown & Sandholm, 2019). Figure 3 (left) shows the performance of different α, γ in Leduc Hold'em and Figure 3 (right) shows the performance of different α, γ in Big Leduc Hold'em. We can see that the performance of TRM-CFR has a similar performance to CFR on Leduc Hold'em. This is because in the experiments of CFR on Leduc Hold'em, there are a large portion of histories with n average. The performance of TRM-(CFR+, LCFR) which $a_{1.005,0.095}$ is better than CFR+ and LCFR on both Leduc games.



(a) Leduc Hold'em (b) Big Leduc Hold'em

Figure 3: The **left** and **right** shows the exploitability (log10) of these algorithms when $\beta \in \{1, a_{0.99-1.01}, a_{0.995-1.005}, a_{1.005-0.995}, a_{1.01-0.09}\}$. The exploitability ($y$-axis) are reported in T itearations (x-axis, $T = 65536$ in Leduc and $T = 1024$ in Big Leduc). $a_{x,y}$ represents a value that is set as $\beta = x + \frac{t(y-x)}{T}$ in iteration t.



(a) Leduc Hold'em (b) Big Leduc Hold'em

Figure 4: The **left** and **right** shows the exploitability (log10) of TRM and RTRM in CFR, CFR+ and LCFR. The exploitability (y-axis) are reported in T itearations (x-axis, $T = 65536$ in Leduc and $T = 1024$ in Big Leduc).

We use the best-performance parameter $a_{1.005,0.095}$ in TRM to measure whether TRM and the new regret updating method (RTRM) can cooperate to enhance the performance further. We use the network in deep CFR (DCFR) (Brown et al., 2019) to save the startegies for players $i, -i$ in iteration $t-1$ to avoid consuming too many storage resources. Figure 4 (left) shows the performance of TRM and RTRM in Leduc Hold'em and Figure 4 (right) shows the performance in Big Leduc Hold'em. Compared to TRM, TRM with the new regret updating method algorithm learns faster.

6 Conclusion

In this work, we propose TRM, a novel regret matching method for CFR to solve imperfect information games and theoretically demonstrate that the update of TRM converges to Nash Equilibrium. TRM provides a mechanism to CFR methods to dynamically change the matching weight of external regret according to the number of iterations. Experimental results in some IIGs demonstrate that TRM is beneficial for CFR and its variants to accelerate convergence speed.

REFERENCES

Jacob Abernethy, Peter L Bartlett, and Elad Hazan. Blackwell approachability and no-regret learning are equivalent. In *Conference on Learning Theory*, volume 19, pp. 27–46, 2011.

David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. In *AAAI Conference on Artificial Intelligence*, pp. 1829–1836, 2019.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International Conference on Machine Learning*, pp. 793–802, 2019.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. 2006.

Steven De Rooij, Tim Van Erven, Peter Grunwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15(1):1281–1316, 2014.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. volume 55, pp. 119–139, 1997.

Richard G Gibson, Marc Lanctot, Neil Burch, Duane Szafron, and Michael Bowling. Generalized sampling and variance in counterfactual regret minimization. In *AAAI Conference on Artificial Intelligence*, pp. 1355–1361, 2012.

Sergiu Hart and Andreu Mascolell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. In *Neural Information Processing Systems*, pp. 1078–1086, 2009.

Matej Moravcik, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

Jr John F Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

Yu Nesterov. Excessive gap technique in nonsmooth convex minimization. *Siam Journal on Optimization*, 16(1):235–249, 2005.

Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: opponent modelling in poker. In *Uncertainty in Artificial Intelligence*, pp. 550–558, 2005.

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. In *Neural Information Processing Systems*, pp. 2989–2997, 2015.

Oskari Tammelin. Solving large imperfect information games using cfr+. *arXiv preprint arXiv:1407.5042*, 2014.

Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit texas hold'em. In *International Joint Conferences on Artificial Intelligence*, pp. 645–652, 2015.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Neural Information Processing Systems*, pp. 1729–1736, 2007.