Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Using Response Functions to Measure Strategy Strength

Trevor Davis and **Neil Burch** and **Michael Bowling**
{trdavis1,burch,mbowling}@ualberta.ca
Department of Computing Science
University of Alberta
Edmonton, AB, Canada T6G 2EG

Abstract

Extensive-form games are a powerful tool for representing complex multi-agent interactions. Nash equilibrium strategies are commonly used as a solution concept for extensive-form games, but many games are too large for the computation of Nash equilibria to be tractable. In these large games, exploitability has traditionally been used to measure deviation from Nash equilibrium, and thus strategies are aimed to achieve minimal exploitability. However, while exploitability measures a strategy's worst-case performance, it fails to capture how likely that worst-case is to be observed in practice. In fact, empirical evidence has shown that a less exploitable strategy can perform worse than a more exploitable strategy in one-on-one play against a variety of opponents. In this work, we propose a class of response functions that can be used to measure the strength of a strategy. We prove that standard no-regret algorithms can be used to learn optimal strategies for a scenario where the opponent uses one of these response functions. We demonstrate the effectiveness of this technique in Leduc Hold'em against opponents that use the UCT Monte Carlo tree search algorithm.

Introduction

Extensive-form games are a commonly-used, natural representation for sequential decision-making tasks. Their ability to model multiple agents, chance events, and imperfect information makes them applicable to a wide range of problems. In these games, Nash equilibrium strategies are often used as a solution concept. Efficient algorithms exist for finding Nash equilibria in two-player zero-sum games, but these fail to scale to the very large games that result from many human interactions (e.g., two-player limit Texas hold'em poker, which has approximately 10^{18} game states). In such large games, a variety of techniques are used to find a strategy profile which approximates a Nash equilibrium. In order to evaluate these techniques, researchers would like to be able to measure the similarity of the resulting approximation with a Nash equilibrium.

Traditionally, performance against a worst-case adversary, or exploitability, of a strategy is used as a proxy measure for this similarity. For examples, see Waugh et al. (2009a), Johanson et al. (2011), and Ganzfried, Sandholm, and Waugh (2012). In a two-player zero-sum game, a Nash equilibrium strategy is guaranteed to achieve maximal expected performance in the worst case. Thus, it is natural to regard strategies with low exploitability (better performance against a worst-case opponent) as being closer to a Nash equilibrium than strategies with high exploitability (worse performance against a worst-case opponent). However, as Ganzfried, Sandholm, and Waugh identified, there are issues with using exploitability as a measure of strategy strength or with trying to find a strategy that is optimal with respect to exploitability (2012).

Exploitability of a strategy is calculated by evaluating the utility of the strategy playing against an opponent using a best-response function. In this paper, we show how any response function can be used to define a real-valued function for evaluating strategies. Johanson et al. modified an equilibrium-finding algorithm to instead find a strategy which achieves maximal performance against a best-response opponent (2012). We extend that approach to alternative response functions with the algorithm CFR-f. We prove that CFR-f converges for an intuitive family of response functions. We further prove that CFR-f can be extended to non-deterministic response functions.

Using CFR-f, we can learn optimal static strategies for playing against adaptive opponents. We show the value of this technique by using it to learn strategies designed to beat opponents who are responding to us with the UCT algorithm. UCT is a Monte Carlo tree search algorithm which makes use of regret minimization techniques from the multi-armed bandit setting (Kocsis and Szepsvari 2006), and it has been shown to be effective in complex games such as Go (Gelly and Wang 2006). As a Monte Carlo algorithm, the strength of UCT directly correlates with the number of samples it has of the opponent's strategy. In a small poker game, we are able to generate strategies which have better performance when playing against UCT opponents than a Nash equilibrium against the same opponents.

Background

An **extensive-form game** is a formalism for modeling sequential decision-making tasks. It uses a game tree representation, where each node is a **history** $h \in H$, and each edge is a player action or chance event. A **player function**

$P(h)$ assigns which player acts next at history h, by mapping each history either to a player $i \in N$ or to c, a **chance player** which mixes over actions according to a predefined probability distribution. Each leaf of the tree is a **terminal history** $z \in Z$ at which each player i is assigned a utility by a **utility function** as $u_i(z)$. If the game satisfies $\sum_{i \in N} u_i(z) = 0$ for every terminal history z, then it is said to be **zero-sum**. The game is said to exhibit **imperfect information** if some actions or chance events are not observed by all players. In such games, histories are grouped into **information sets**, where histories in the same information set are indistinguishable by the player assigned to act at those nodes.

A (behavioral) **strategy** for player i, $\sigma_i \in \Sigma_i$, is a function which maps every information set at which i acts to a probability distribution over the actions i can play at the information set. A **strategy profile** $\sigma \in \Sigma$ is a set which contains one strategy σ_i for each player $i \in N$. The subset of σ which contains the strategies for all players except i is denoted σ_{-i}. The probability of reaching each terminal history is fully determined by a strategy profile, so we can write the expected utility for player i as $u_i(\sigma)$, or equivalently $u_i(\sigma_i, \sigma_{-i})$.

A **Nash Equilibrium** is a strategy profile in which no player has incentive to deviate to a strategy that is not part of the profile. Formally, σ is a Nash equilibrium if

$$u_i(\sigma) \geq u_i(\sigma'_i, \sigma_{-i}), \forall \sigma'_i \in \Sigma_i, \forall i \in N.$$

An **ε-equilibrium** is a strategy profile in which no player can gain more than ε in expected utility by deviating.

In two-player zero-sum games, a result known as the **minimax theorem** holds:

$$v_1 = \max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} u_1(\sigma_1, \sigma_2) = \min_{\sigma_1 \in \Sigma_1} \max_{\sigma_2 \in \Sigma_2} u_1(\sigma_1, \sigma_2).$$

v_1 is called the **game value** for player 1. Because the game is two-player and zero-sum, $v_2 = -v_1$ is the game value for player 2. If σ is a Nash equilibrium in a two-player zero-sum game, then $u_i(\sigma) = v_i$ for $i = 1, 2$.

In an iterated game, **regret** refers to the difference in how much an agent would have gained by playing a particular fixed strategy at all time steps, minus their actual observed utility. Given each player i plays strategy σ_i^t at time step t, the **average overall regret** for player i at time T is

$$R_i^T = \frac{1}{T} \max_{\sigma_i^* \in \Sigma_i} \sum_{t=1}^{T} \left(u_i(\sigma_i^*, \sigma_{-i}^t) - u_1(\sigma_i^t, \sigma_{-i}^t) \right)$$

Counterfactual Regret Minimization (CFR) is a state-of-the-art algorithm for finding ε-equilibria in extensive-form games (Zinkevich et al. 2007). CFR uses iterated self-play, and works by minimizing a form of regret at each information set independently.

Although CFR has been applied to games with approximately 10^{11} information sets (Jackson 2012), it requires memory linear in the number of information sets, and there exist large games of interest that cannot be solved by any equilibrium-finding algorithm with current computing resources. Such games are typically approached using **abstraction**, in which a mapping is established from the information sets in the large game to information sets in a smaller, **abstract game**. The abstract game is solved with a technique such as CFR, and the resulting strategy is mapped back to the full game. A full treatment of abstraction in extensive-form games, and particularly poker, can be found in (Johanson et al. 2013).

CFR-BR is a modified form of CFR for use in abstract games (Johanson et al. 2012). Instead of using self-play, CFR-BR uses the CFR algorithm in the abstraction for one player, while the other player is replaced with an opponent that plays a best response to the CFR player in the full game. Whereas running CFR in an abstract game converges to an equilibrium for that abstraction, CFR-BR will converge to the strategy that is least exploitable (in the full game) out of all the strategies that can be expressed in the abstract game.

Using Exploitability to Evaluate Strategies

A **best-response** strategy is a strategy that achieves maximal expected performance against a particular set of opponent strategies. σ_i is a best-response to σ_{-i} if

$$u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i}), \forall \sigma'_i \in \Sigma_i$$

In a two-player zero-sum game, the **exploitability** of a strategy is how much expected utility a best-response opponent can achieve above the game value:

$$\begin{aligned} \text{exploit}(\sigma_i) &= \max_{\sigma^*_{-i} \in \Sigma_{-i}} u_{-i}(\sigma_i, \sigma^*_{-i}) - v_{-i} \\ &= v_i - \min_{\sigma^*_{-i} \in \Sigma_{-i}} u_i(\sigma_i, \sigma^*_{-i}). \end{aligned}$$

In large games, the game value may be intractable to compute, so we instead consider the average exploitability of a strategy profile:

$$\begin{aligned} \text{exploit}(\sigma) &= \frac{1}{2} \left(\text{exploit}(\sigma_1) + \text{exploit}(\sigma_2) \right) \\ &= \frac{1}{2} \left(\max_{\sigma_2^* \in \Sigma_2} u_2(\sigma_1, \sigma_2^*) + \max_{\sigma_1^* \in \Sigma_1} u_1(\sigma_1^*, \sigma_2) \right). \end{aligned}$$

In a Nash equilibrium, each strategy must be a best-response to the rest of the profile (or else the player would have incentive to deviate to a best-response). Thus, a Nash equilibrium σ has $\text{exploit}(\sigma) = 0$. The difference in exploitability between strategy profiles can be used as a metric to compare them. Because we can use this metric to compare a strategy profile to a Nash equilibrium even when we know none of the action probabilities of the equilibrium, it is a useful tool for measuring strategy strength in domains where we can't compute an equilibrium. However, there are limitations to using the exploitability metric.

It is not possible to compute a best-response in all domains. While best-response computation is simpler than equilibrium computation, it is still intractable in many large games. Only recent algorithmic advances have allowed the computation of best-responses in limit Texas Hold'em poker (approximately 10^{18} game states), and computation still takes 76 cpu-days (Johanson et al. 2011). In much larger games, like the variant of no-limit Texas Hold'em poker which is used in the Annual Computer Poker Competition (approximately 10^{76} game states), computing a best-

response is hopeless. In addition, no efficient algorithm exists for computing a best-response in games exhibiting **imperfect recall**. In games with imperfect recall, players can forget information that they had previously known about the game state. Such games are commonly used when creating abstractions (Waugh et al. 2009b).

There are also issues with the quality of exploitability as a measure of strategy strength. Exploitability measures the worst-case performance of a strategy, but not how easy it is for the worst case to be found. In the large games we are most concerned with, computing a best response is likely to be too resource-intensive for the opponent to do online, even if she has access to our full strategy. If she doesn't have access to our strategy and must learn it during game play, the situation is even more bleak. Thus, as long as our strategy is private, it is not clear that we need to worry about opponents learning to maximally exploit it. However, some strategies might be easier to exploit than others. A strategy that loses the maximum whenever the opponent makes a particular action at her first information set will be evaluated the same under exploitability as a strategy that loses the same maximum, but only when the opponent plays one particular strategy. It seems clear that it will be easier for an intelligent opponent to learn to exploit the first strategy rather than the second one.

These drawbacks have been observed in practice, as empirical evidence suggests that exploitability is not a good measure of strategy strength in one-on-one play. Waugh performed an experiment with a pool of strategies in Leduc Hold'em (a toy poker domain played with six cards and two rounds of betting). Each pair of strategies was played against each other to determine the expected utility for each strategy, and then the strategies are ranked in two fashions: In the **total bankroll** ranking, strategies are ordered by their average expected utility against all other strategies, and in the **instant runoff** setting, the strategy with the worst expected utility is iteratively removed until one winner remains. Waugh found a correlation coefficient between exploitability and ranking of only 0.15 for total bankroll and 0.30 for instant runoff (2009). Johanson et al. evaluated several of the strategies submitted to the limit Texas Hold'em event at the 2010 Annual Computer Poker Competition. The winner of the competition by instant runoff ranking was more exploitable than three of the strategies it defeated, and tended to have better performance than these less exploitable strategies when playing against the other agents (Johanson et al. 2011). Johanson et al. found that a strategy produced with CFR-BR had worse one-on-one performance than a strategy produced with CFR in the same abstraction, despite the CFR-BR strategy being less exploitable (2012). Bard, Johanson, and Bowling found that when the size of an abstract game is varied for only one player's information sets, exploitability and one-on-one performance are inversely correlated in limit Texas Hold'em (2014).

Pretty-Good Responses

Given the limitations of exploitability, we propose new metrics for evaluating the strength of a strategy. In order to address the shortcomings of exploitability, these new metrics should be able to measure how difficult a strategy is to exploit. If σ is harder to exploit than σ', there must be some opponent that can effectively exploit σ' but not σ. We thus propose using a response function $f\colon \Sigma_1 \to \Sigma_2$ (without loss of generality, we will assume player 2 is exploiting player 1 for the remainder of this paper), so if the strategy being tested is σ_1, the opponent will play $f(\sigma_1)$. This response function naturally induces a real-valued function which we call the **response value function**. The response value function for f is $v_f(\sigma_1) = u_1(\sigma_1, f(\sigma_1))$. If f is an exploitive function, a higher value for $v_f(\sigma_1)$ implies that σ_1 does better against an opponent which tries to exploit it, so $v_f(\sigma_1)$ should directly correlate with some dimension of strategy strength for σ_1.

It is unclear if one response value function is sufficient to measure how difficult a strategy is to exploit. For instance, consider response functions f_1 and f_2 which both exploit the opponent, but f_1 is better at exploiting opponents that are harder to exploit. Consider applying these responses to strategies σ_1^1, σ_1^2, and σ_1^3 which have the same exploitability, but σ_1^1 is harder to exploit than σ_1^2, which is harder to exploit than σ_1^3. It might be the case that v_{f_1} cannot differentiate between σ_1^2 and σ_1^3 because it achieves the maximum against each strategy, and it might also be the case that v_{f_2} cannot differentiate between σ_1^1 and σ_1^2 because it cannot learn to exploit either strategy. In this case we need multiple response functions to fully evaluate the strategies. We thus propose that instead of replacing exploitability with one response function f, we instead use a set of response functions. The response functions all attempt to exploit their opponents, but with varying strength. On one end of the spectrum, we could have a response function that always plays a static strategy, and on the other, we could have a best-response function.

If $v_f(\sigma_1)$ correlates with some dimension of strategy strength, we would naturally like to be able to find a $\sigma_1 \in \Sigma_1$ that maximizes the value of $v_f(\sigma_1)$. Inspired by how CFR-BR minimizes the exploitability metric which arises from the best-response function, we propose a new algorithm CFR-f for generic response functions f. CFR-f is an iterated algorithm, where on iteration t the learning player (the **CFR-agent**) plays σ_1^t as specificed by the CFR algorithm. The other player (the **response-agent**) plays $f(\sigma_1^t)$. The CFR-f algorithm is not guaranteed to converge to the σ_1 which maximizes $v_f(\sigma_1)$ for every choice of f. We now present a family of response functions for which CFR-f will converge.

Definition 1. *A function $f\colon \Sigma_{-i} \to \Sigma_i$ is called a* **pretty-good response** *if $u_i(\sigma_{-i}, f(\sigma_{-i})) \geq u_1(\sigma_{-i}, f(\sigma'_{-i}))$ for all $\sigma_{-i}, \sigma'_{-i} \in \Sigma_{-i}$. f is a* **δ-pretty-good response** *if $u_i(\sigma_{-i}, f(\sigma_{-i})) + \delta \geq u_1(\sigma_{-i}, f(\sigma'_{-i}))$ for all $\sigma_{-i}, \sigma'_{-i} \in \Sigma_{-i}$.*

A pretty-good response is a function that maximizes the responder's utility when she correctly hypothesizes and responds to her opponents' actual strategies. Every pretty-good response f for player i can be associated with a subset of strategies $\Sigma_i^f \subseteq \Sigma_i$ such that $f(\sigma_{-i}) = \operatorname{argmax}_{\sigma_i \in \Sigma_i^f} u_i(\sigma_{-i}, \sigma_i)$. Any best-response function is a pretty-good response (where $\Sigma_i^f = \Sigma_i$). In addition, any re-

sponse function that returns a best response in some abstract game is also a pretty-good response in the full game.

If we are attempting to optimize with regards to v_f, considering only pretty-good response functions makes intuitive sense. In the two-player zero-sum case, if f is a pretty-good response, then $v_f(\sigma_1)$ will be a lower bound on our utility when we play against any opponent that uses f, even if that opponent can't fully observe our strategy. With other functions, no such guarantee holds. If an opponent using a response function f that is not a pretty-good response mistakenly believes we are playing σ_1', she could play $f(\sigma_1')$ and cause us to lose utility. We now consider what is required for CFR-f to converge to an optimal strategy with respect to v_f.

Definition 2. *Let $f\colon \Sigma_1 \to \Sigma_2$ be a response function and define $\sigma_2^t = f(\sigma_1^t)$ for $t = 1, ..., T$. f is called* **no-regret learnable** *if for every sequence of strategies $\sigma_1^1, \sigma_1^2, ..., \sigma_1^T \in \Sigma_1$ such that $R_1^T \leq \varepsilon$ (where $\varepsilon > 0$), we have that*

$$u_1(\overline{\sigma}_1^T, f(\overline{\sigma}_1^T)) + \varepsilon \geq \max_{\sigma_1^* \in \Sigma_1} u_1(\sigma_1^*, f(\sigma_1^*))$$

where $\overline{\sigma}_1^T$ is the mixed strategy that mixes equally between each of $\sigma_1^1, ..., \sigma_1^T$. f is called **no-regret δ-learnable** *if for every sequence of strategies $\sigma_1^1, \sigma_1^2, ..., \sigma_1^T \in \Sigma_1$ such that $R_1^T \leq \varepsilon$ (where $\varepsilon > 0$), we have that*

$$u_1(\overline{\sigma}_1^T, f(\overline{\sigma}_1^T)) + \varepsilon + \delta \geq \max_{\sigma_1^* \in \Sigma_1} u_1(\sigma_1^*, f(\sigma_1^*)).$$

Notice that if f is no-regret learnable, then $\overline{\sigma}_1^T$ is within ε of achieving the maximal value of v_f. Because CFR is a regret-minimizing algorithm, we know that R_1^T converges to zero if $\sigma_1^1, ..., \sigma_1^T$ are the strategies chosen by CFR. Thus if f is no-regret learnable, the average strategy $\overline{\sigma}_1^T$ produced by CFR-f will converge to the strategy achieving the maximal value of v_f. We show that pretty-good responses fulfill this condition.

Theorem 1. *Every $f\colon \Sigma_1 \to \Sigma_2$ that is a pretty-good response is no-regret learnable. Every $f\colon \Sigma_1 \to \Sigma_2$ that is a δ-pretty-good response is no-regret 2δ-learnable.*

Proof.

$$\begin{aligned}
\max_{\sigma_1^* \in \Sigma} u_1(\sigma_1^*, f(\sigma_1^*)) &= \frac{1}{T} \max_{\sigma_1^* \in \Sigma_1} \sum_{t=1}^{T} u_1(\sigma_1^*, f(\sigma_1^*)) \\
&\leq \frac{1}{T} \max_{\sigma_1^* \in \Sigma_1} \sum_{t=1}^{T} \left(u_1(\sigma_1^*, f(\sigma_1^t)) + \delta \right) \\
&\leq \frac{1}{T} \sum_{t=1}^{T} u_1(\sigma_1^t, f(\sigma_1^t)) + \varepsilon + \delta \\
&\leq \frac{1}{T} \sum_{t=1}^{T} \left(u_1(\sigma_1^t, f(\overline{\sigma}_1^T)) + \delta \right) + \varepsilon + \delta \\
&= u_1(\overline{\sigma}_1^T, f(\overline{\sigma}_1^T)) + \varepsilon + 2\delta
\end{aligned}$$

□

We have established a family of deterministic response functions which we can optimize our performance against. However, we would also like to handle non-deterministic response functions. Monte Carlo algorithms are good candidates for the exploitive response functions we discussed at the start of the section. As a Monte Carlo algorithm receives more samples of its opponent, its ability to exploit that opponent increases. Thus, if we use a set of Monte Carlo algorithms which have a varying number of opponent samples, they will also have varying exploitive strength against that opponent. In order to handle such algorithms, we must extend our ideas to non-deterministic response functions. Such a function maps a strategy σ_1 to a probability distribution over Σ_2. Equivalently, we can work with a probability distribution over response functions $F \in \Delta_{\{f\colon \Sigma_1 \to \Sigma_2\}}$. The idea of a response value function still holds for such distributions: $v_F(\sigma_1) = E_{f \sim F}[u_1(\sigma_1, f(\sigma_1))]$. The CFR-$f$ algorithm can be extended to a CFR-f algorithm which samples a response function $f_t \sim F$ on each iteration. We must also extend our notions of pretty-good response and no-regret learnable.

Definition 3. *Let $F \in \Delta_{\{f\colon \Sigma_{-i} \to \Sigma_i\}}$ be a probability distribution over response functions. We say that F is an* **expected pretty-good response** *if*

$$E_{f \sim F}[u_i(\sigma_{-i}, f(\sigma_{-i}))] \geq E_{f \sim F}[u_i(\sigma_{-i}, f(\sigma'_{-i}))] \quad (1)$$

for all $\sigma_{-i}, \sigma'_{-i} \in \Sigma_{-i}$. We say that F is an **expected δ-pretty-good response** *if*

$$E_{f \sim F}\left[u_i(\sigma_{-i}, f(\sigma_{-i}))\right] + \delta \geq E_{f \sim F}\left[u_i(\sigma_{-i}, f(\sigma'_{-i}))\right] \quad (2)$$

for all $\sigma_{-i}, \sigma'_{-i} \in \Sigma_{-i}$.

Definition 4. *Let $F \in \Delta_{\{f\colon \Sigma_1 \to \Sigma_2\}}$ be a probability distribution over response functions, and define $\sigma_2^t = f_t(\sigma_1^t)$ for $t = 1, ..., T$, where each $f_t \sim F$ is chosen independently. F is called* **no-regret learnable** *if for every sequence of strategies $\sigma_1^1, \sigma_1^2, ..., \sigma_1^T \in \Sigma_1$ such that $R_1^T \leq \varepsilon$, we have that*

$$E_{f \sim F}\left[u_1(\overline{\sigma}_1^T, f(\overline{\sigma}_1^T))\right] + \varepsilon \geq \max_{\sigma_1^* \in \Sigma_1} E_{f \sim F}\left[u_1(\sigma_1^*, f(\sigma_1^*))\right]$$

F is called **no-regret δ-learnable** *if we choose $f_1, ..., f_T \sim F$ independently and for every sequence of strategies $\sigma_1^1, \sigma_1^2, ..., \sigma_1^T \in \Sigma_1$ such that $R_1^T \leq \varepsilon$, we have that*

$$\begin{aligned}
E_{f \sim F}\left[u_1(\overline{\sigma}_1^T, f(\overline{\sigma}_1^T))\right] + \varepsilon + \delta \\
\geq \max_{\sigma_1^* \in \Sigma_1} E_{f \sim F}\left[u_1(\sigma_1^*, f(\sigma_1^*))\right]
\end{aligned}$$

Let $u_{\max} = \max_{\sigma_1 \in \Sigma_1, \sigma_2 \in \Sigma_2} u_1(\sigma_1, \sigma_2)$ be the maximum expected utility that player 1 can achieve, $u_{\min} = \min_{\sigma_1 \in \Sigma_1, \sigma_2 \in \Sigma_2} u_1(\sigma_1, \sigma_2)$ be the minimum expected utility that player 1 can achieve, and $\Delta_1 = u_{\max} - u_{\min}$ be the range of expected utilities. We now show that expected pretty-good responses are no-regret learnable.

Theorem 2. *Every F that is an expected δ-pretty-good response is no-regret $(2\delta + \gamma)$-learnable with probability at least $1 - 2\exp(-\frac{T^2\gamma^2}{2\Delta_1^2})$, where γ is a free parameter.*

Proof. Let $\sigma_1^* \in \operatorname{argmax}_{\sigma_1 \in \Sigma_1} E_{f \sim F}[u_1(\sigma_1, f(\sigma_1))]$. If we assume each of the following:

$$E_{f \sim F}\left[\frac{1}{T} \sum_{t=1}^{T} u_1(\sigma_1^*, f(\sigma_1^t))\right]$$

$$\leq \frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^*, f_t(\sigma_1^t)) + \frac{1}{2}\gamma \quad (3)$$

$$\frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^t, f_t(\sigma_1^t)) \leq E_{f\sim F}\left[\frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^t, f(\sigma_1^t))\right] + \frac{1}{2}\gamma \quad (4)$$

then it follows that F is no-regret $(2\delta+\gamma)$-learnable:

$$\begin{aligned}
&\max_{\sigma_1^*} E_{f\sim F}\left[u_1(\sigma_1^*, f(\sigma_1^*))\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} E_{f\sim F}\left[u_1(\sigma_1^*, f(\sigma_1^*))\right] \\
&\leq \frac{1}{T}\sum_{t=1}^{T} E_{f\sim F}\left[u_1(\sigma_1^*, f(\sigma_1^t))\right] + \delta \\
&= E_{f\sim F}\left[\frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^*, f(\sigma_1^t))\right] + \delta \\
&\leq \frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^*, f_t(\sigma_1^t)) + \delta + \frac{1}{2}\gamma \\
&\leq \frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^t, f_t(\sigma_1^t)) + \varepsilon + \delta + \frac{1}{2}\gamma \\
&\leq E_{f\sim F}\left[\frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^t, f(\sigma_1^t))\right] + \varepsilon + \delta + \gamma \\
&\leq E_{f\sim F}\left[\frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^t, f(\overline{\sigma}_1^T))\right] + \varepsilon + 2\delta + \gamma \\
&= E_{f\sim F}\left[u_1(\overline{\sigma}_1^T, f(\overline{\sigma}_1^T))\right] + \varepsilon + 2\delta + \gamma
\end{aligned}$$

Thus we can bound the overall probability that F is not no-regret $(2\delta+\gamma)$-learnable by the probability that either (3) or (4) is false. Because $f_1, ..., f_T$ are chosen independently, we can do this using Hoeffding's Inequality.

$$\Pr\left[E_{f\sim F}\left[\frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^*, f(\sigma_1^t))\right] - \frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^*, f_t(\sigma_1^t)) > \frac{1}{2}\gamma\right] \leq \exp\left(-\frac{T^2\gamma^2}{2\Delta_1^2}\right)$$

$$\Pr\left[\frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^t, f_t(\sigma_1^t)) - E_{f\sim F}\left[\frac{1}{T}\sum_{t=1}^{T} u_1(\sigma_1^t, f(\sigma_1^t))\right] \geq \frac{1}{2}\gamma\right] \leq \exp\left(-\frac{T^2\gamma^2}{2\Delta_1^2}\right)$$

The probability that either event is true is no more than their sum, which gives us the result. □

Even if F is not a pretty-good response in expectation, it can be no-regret learnable if there is only a low probability p that it is not a pretty-good response. In this case, however, CFR-f does not converge fully to the optimal strategy according to v_F, but is only guaranteed to converge to a strategy that is within $2p\Delta_1$ of optimal.

Theorem 3. *Let $F \in \Delta_{\{f:\,\Sigma_1\to\Sigma_2\}}$ be a probability distribution over response functions such that for any $\sigma_1, \sigma_1' \in \Sigma_1$, we have that $u_1(\sigma_1, f(\sigma_1)) \leq u_1(\sigma_1, f(\sigma_1'))$ with probability at least $1-p$ given that $f \sim F$. Then F is no-regret $(2p\Delta_1+\gamma)$-learnable with probability at least $1-2\exp(-\frac{T^2\gamma^2}{2\Delta_1^2})$, where γ is a free parameter.*

Proof.

$$\begin{aligned}
&E_{f\sim F}\left[u_1(\sigma_1, f(\sigma_1))\right] \\
&\leq E_{f\sim F}\left[u_1(\sigma_1, f(\sigma_1'))|u_1(\sigma_1, f(\sigma_1)) \leq u_1(\sigma_1, f(\sigma_1'))\right] \\
&\quad * \Pr\left[u_1(\sigma_1, f(\sigma_1)) \leq u_1(\sigma_1, f(\sigma_1'))\right] \\
&\quad + u_{\max}\left(1 - \Pr\left[u_1(\sigma_1, f(\sigma_1)) \leq u_1(\sigma_1, f(\sigma_1'))\right]\right) \\
&= E_{f\sim F}\left[u_1(\sigma_1, f(\sigma_1'))\right] \\
&\quad - E_{f\sim F}\left[u_1(\sigma_1, f(\sigma_1'))|u_1(\sigma_1, f(\sigma_1)) > u_1(\sigma_1, f(\sigma_1'))\right] \\
&\quad * \left(1 - \Pr\left[u_1(\sigma_1, f(\sigma_1)) \leq u_1(\sigma_1, f(\sigma_1'))\right]\right) \\
&\quad + u_{\max}\left(1 - \Pr\left[u_1(\sigma_1, f(\sigma_1)) \leq u_1(\sigma_1, f(\sigma_1'))\right]\right) \\
&\leq E_{f\sim F}\left[u_1(\sigma_1, f(\sigma_1'))\right] \\
&\quad - u_{\min}\left(1 - \Pr\left[u_1(\sigma_1, f(\sigma_1)) \leq u_1(\sigma_1, f(\sigma_1'))\right]\right) \\
&\quad + u_{\max}\left(1 - \Pr\left[u_1(\sigma_1, f(\sigma_1)) \leq u_1(\sigma_1, f(\sigma_1'))\right]\right) \\
&\leq E_{f\sim F}\left[u_1(\sigma_1, f(\sigma_1'))\right] + \Delta_1\left(1 - (1-p)\right) \\
&= E_{f\sim F}\left[u_1(\sigma_1, f(\sigma_1'))\right] + p\Delta_1
\end{aligned}$$

Thus F is an expected $p\Delta_1$-pretty-good response, and the result follows by Theorem 2. □

Using CFR-f, we can learn static strategies that are close to optimal for playing against certain adaptive opponents. However, there are widely used response agents that don't fit within the framework of pretty-good responses. We now show experimentally that CFR-f can generate empirically strong strategies against one such agent that has been used to construct effective strategies in several domains, but does not fit into the framework of the above theory.

Experimental Results

We tested the technique of CFR-f in the game of limit Leduc Hold'em. Leduc Hold'em is a small poker game, which is played with a deck containing two cards each of three different suits. Each player is randomly dealt one private card and there is a betting round. Then a community card is dealt publicly, and another betting round occurs. The player with the best two card poker hand using his private card and the public card wins the game. For full details, see Waugh et al. (2009a).

For the response function in CFR-f, we used the UCT algorithm, resulting in an algorithm we refer to as **CFR-UCT**. For each iteration t of CFR-UCT, we ran a CFR update for

the CFR-agent to create strategy σ_1^t, then we used UCT to train a response to σ_1^t. On each iteration the UCT-agent created an entirely new game tree, so the response depended only on σ_1^t. We gave the UCT-agent k iterations of the UCT algorithm, each of which correspond to one sample of σ_1^t, where k is a parameter of CFR-UCT. Through the UCT iterations, the UCT-agent formed an observed single-player game tree, which we then used to form a static response strategy $f_t(\sigma_1^t)$.

Because of the Monte Carlo nature of UCT, it will return a fully random response strategy when $k = 0$, and as k goes to infinity it will converge to a best-response. Empirically UCT transitions smoothly between these extremes as k increases (Kocsis and Szepsvari 2006). Because of this relationship, we can generate a collection of responses that exploit σ_1 to varying degrees by training UCT with various values of k. This makes UCT potentially useful as a source of response functions for evaluating strategy strength, and thus we would like to use CFR-UCT to generate strategies that are optimal against UCT opponents. However, UCT does not fit nicely into the pretty-good response framework, since we have no guarantees on the utility of the strategy it generates. We therefore have no theoretical guarantees on the quality of the strategy that CFR-UCT outputs.

We trained CFR-UCT(k) strategies for a range of k values. We also trained a ε-equilibrium with CFR, resulting in a strategy that is exploitable for .002 bets/game. Against each of these strategies, we used UCT to train counter-strategies with a variety of k values and played the strategies against the counter-strategies. Because UCT is a Monte Carlo algorithm and inherently noisy, we averaged the values over 100 independent runs of UCT.

Figure 1 shows the results of playing CFR-UCT(1000), CFR-UCT(10000), CFR-UCT(100000), and the ε-equilibrium against UCT counter-strategies trained using a range of k values. Figure 2 shows similar data, but now with more k values for CFR-UCT represented on the x-axis, and also shows how the CFR-UCT strategies do in one-on-one play against the ε-equilibrium and against a best response.

From the results we can see that for any $k_2 \leq k_1$, the strategy produced by CFR-UCT(k_1) will achieve higher value than an ε-equilibrium in one-on-one play against UCT(k_2). We have thus shown that using CFR-f, we can learn to do better against adaptive opponents than we would do by playing an optimal strategy for the game. In addition, we have shown that CFR-f can converge to an effective strategy against an opponent not covered by the pretty-good response theory. Despite being highly exploitable, the CFR-UCT strategies lose only a small amount in one-on-one play against an ε-equilibrium, and do not lose to the weaker UCT counter-strategies which are actively trying to exploit them, indicating that they are difficult to exploit, a strength not shown via the exploitability metric. The UCT(k_2) strategies where k_2 is much larger than the k_1 value used to train CFR-UCT(k_1) are able to successfully exploit the CFR-UCT(k_1) strategies, whereas UCT(k) strategies for smaller k are not able to do so, which lends credence to our conjecture that using a range of k values gives us a set of response functions which are able to exploit the opponent with varying strength.

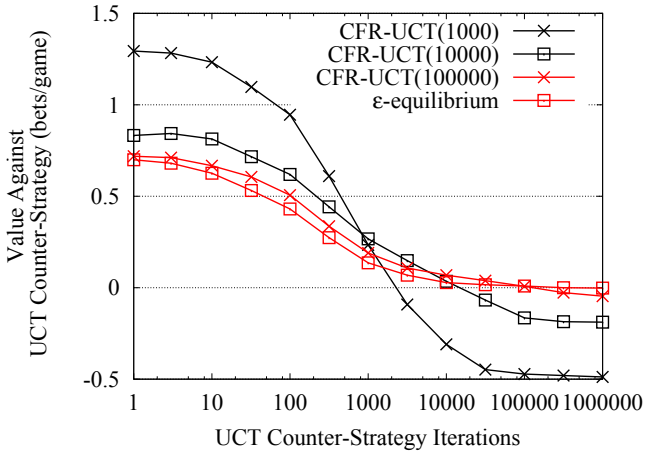

Figure 1: Performance of CFR-UCT strategies and an ε-equilibrium against UCT counter-strategies, as the counter-strategy uses more UCT iterations.

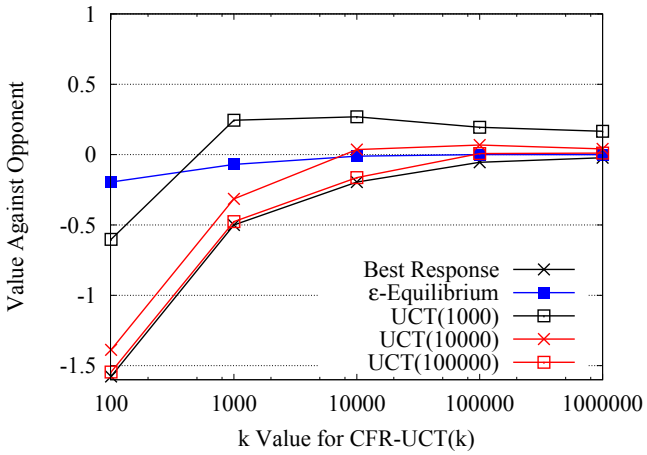

Figure 2: Performance of CFR-UCT(k) strategies against a variety of opponents as the k value is increased.

Conclusion

We have demonstrated that the exploitability metric leaves much to be desired as a measure of strategy strength. By using other metrics to complement it, we can measure not only a strategy's worst-case performance, but also how difficult the strategy is to exploit. We have shown how such metrics can arise from the response value functions induced by response functions, and we examined a family of such response functions that we can learn to optimize with regards to. We demonstrated the validity of our CFR-f algorithm for optimizing against an opponent that is adaptive and exploitive by using it to learn strategies to defeat UCT opponents in the domain of Leduc Hold'em poker. Our results also reinforced the notion that exploitability is incomplete as a measure of strategy strength, and that how difficult a strategy is to exploit is a key factor in its performance in actual one-on-one competition.

Acknowledgments

The authors would like to thank the members of the Computer Poker Research Group at the University of Alberta for their helpful input throughout this research. This research was supported by NSERC, Alberta Innovates Technology Futures, Alberta Innovates Centre for Machine Learning, and computing resources provided by Compute Canada.

References

Bard, N.; Johanson, M.; and Bowling, M. 2014. Asymmetricabstraction for adversarial settings. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. To appear.

Ganzfried, S.; Sandholm, T.; and Waugh, K. 2012. Strategy purification and thresholding: Effective non-equilibrium approaches for playing large games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Gelly, S., and Wang, Y. 2006. Exploration exploitation in Go: UCT for Monte-Carlo Go. In *Advances in Neural Information Processing Systems 19 (NIPS)*.

Jackson, E. 2012. Slumbot: An implementation of counterfactual regret minimization on commodity hardware. In *2012 Computer Poker Symposium*.

Johanson, M.; Waugh, K.; Bowling, M.; and Zinkevich, M. 2011. Accelerating best response calculation in large extensive games. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*.

Johanson, M.; Bard, N.; Burch, N.; and Bowling, M. 2012. Finding optimal abstract strategies in extensive-form games. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*.

Johanson, M.; Burch, N.; Valenzano, R.; and Bowling, M. 2013. Evaluating state-space abstractions in extensive-form games. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Kocsis, L., and Szepsvari, C. 2006. Bandit based Monte-Carlo planning. In *Proceedings of the 17th European Conference on Machine Learning (ECML)*.

Waugh, K.; Schnizlein, D.; Bowling, M.; and Szafron, D. 2009a. Abstraction pathologies in extensive games. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Waugh, K.; Zinkevich, M.; Johanson, M.; Kan, M.; Schnizlein, D.; and Bowling, M. 2009b. A practical use of imperfect recall. In *Proceedings of the 8th Symposium on Abstraction, Reformulation and Approximation (SARA)*.

Waugh, K. 2009. Abstraction in large extensive games. Master's thesis, University of Alberta, Edmonton, Alberta, Canada.

Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2007. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20 (NIPS)*.