Bayesian Opponent Exploitation in Imperfect-Information Games

Sam Ganzfried
Ganzfried Research
Miami Beach, Florida, 33139
sam@ganzfriedresearch.com
School of Computing and Information Sciences
Florida International University
Miami, Florida, 33199
sganzfri@cis.fiu.edu

Qingyun Sun
Department of Mathematics
Stanford University
Stanford, CA, 94305
qysun@stanford.edu

***Abstract*—Two fundamental problems in computational game theory are computing a Nash equilibrium and learning to exploit opponents given observations of their play (opponent exploitation). The latter is perhaps even more important than the former: Nash equilibrium does not have a compelling theoretical justification in game classes other than two-player zero-sum, and for all games one can potentially do better by exploiting perceived weaknesses of the opponent than by following a static equilibrium strategy throughout the match. The natural setting for opponent exploitation is the Bayesian setting where we have a prior model that is integrated with observations to create a posterior opponent model that we respond to. The most natural, and a well-studied prior distribution is the Dirichlet distribution. An exact polynomial-time algorithm is known for best-responding to the posterior distribution for an opponent assuming a Dirichlet prior with multinomial sampling in normal-form games; however, for imperfect-information games the best known algorithm is based on approximating an infinite integral without theoretical guarantees. We present the first exact algorithm for a natural class of imperfect-information games. We demonstrate that our algorithm runs quickly in practice and outperforms the best prior approaches. We also present an algorithm for a uniform prior.**

***Index Terms*—game theory; opponent modeling; imperfect information; Dirichlet prior; uniform prior; Bayesian approach**

I. Introduction

Imagine you are playing a game repeatedly against one or more opponents. What algorithm should you use to maximize your performance? The classic "solution concept" in game theory is the Nash equilibrium. In a Nash equilibrium σ, each player is simultaneously maximizing his payoff assuming the opponents all follow their components of σ. So should we just find a Nash equilibrium strategy for ourselves and play it in all the game iterations?

Unfortunately, there are some complications. First, there can exist many Nash equilibria, and if the opponents are not following the same one that we have found (or are not following one at all), then our strategy would have no performance guarantees. Second, finding a Nash equilibrium is challenging computationally: it is PPAD-hard and is widely conjectured that no polynomial-time algorithms exist [1]. These challenges apply to both extensive-form games (of both perfect and imperfect information) and strategic-form games, for games with more than two players and non-zero-sum games. While a particular Nash equilibrium may happen to perform well in practice,[1] there is no theoretically compelling justification for why computing one and playing it repeatedly is a good approach. Two-player zero-sum games do not face these challenges: there exist polynomial-time algorithms for computing an equilibrium [3], and there exists a game value that is guaranteed in expectation in the worst case by all equilibrium strategies regardless of the strategy played by the opponent (and this value is the best worst-case guaranteed payoff for any of our strategies). However, even for this game class it would be desirable to deviate from equilibrium to learn and exploit perceived weaknesses of the opponent; for instance, if the opponent has played Rock in each of the first 500 iterations of rock-paper-scissors, it seems desirable to put additional weight on paper beyond the equilibrium value of $\frac{1}{3}$.

Thus, learning to exploit opponents' weaknesses is desirable in all game classes. One approach would be to construct an opponent model consisting of a single mixed strategy that we believe the opponent is playing given our observations of his play and a prior distribution (perhaps computed from a database of historical play). This approach has been successfully applied to exploit weak agents in limit Texas hold 'em poker, a large imperfect-information game [4].[2] A drawback is that it is potentially not robust. It is very unlikely that the opponent's strategy matches this point estimate exactly, and we could perform poorly if our model is incorrect. A more robust approach, which is the natural one to use in this setting, is to use a Bayesian model, where the prior and posterior are full distributions over mixed strategies of the opponent, not single mixed strategies. A natural prior distribution, which has been

[1] An agent for 3-player limit Texas hold 'em computed by the counterfactual regret minimization algorithm (which converges to Nash equilibrium in certain games) performed well in practice despite a lack of theoretical justification [2].

[2] This approach used an approximate Nash equilibrium strategy as the prior and is applicable even when historical data is not available, though if additional data were available a more informed prior that capitalizes on the data would be preferable.

studied and applied in this context, is the Dirichlet distribution. The pdf of the Dirichlet distribution is the belief that the probabilities of K rival events are x_i given that each event has been observed $\alpha_i - 1$ times: $f(x, \alpha) = \frac{1}{B(\alpha)} \prod x_i^{\alpha_i - 1}$.[3] Some notable properties are that the mean is $E[X_i] = \frac{\alpha_i}{\sum_k \alpha_k}$ and that, assuming multinomial sampling, the posterior after including new observations is also Dirichlet, with parameters updated based on the new observations.

Prior work has presented an efficient algorithm for optimally exploiting an opponent in normal-form games in the Bayesian setting with a Dirichlet prior [5], which is essentially the fictitious play rule [6]. Given prior counts α_i for each opponent action, the algorithm increments the counter for an action by one each time it is observed, and then best responds to a model for the opponent where he plays each strategy in proportion to the counters. This algorithm would also extend directly to sequential games of perfect information, where we maintain independent counters at each opponent decision node; this would also work for games of imperfect information where the opponent's private information is observed after each round (so that we would know exactly what information set he took the observed action from). For all of these game classes the algorithm would apply to both zero and general-sum games, for any number of players. However, it would not apply to imperfect-information games where opponents' private information is not observed after play.

An algorithm exists for approximating a Bayesian best response in imperfect-information games, which uses importance sampling to approximate an infinite integral. This algorithm has been successfully applied to limit Texas hold 'em poker [7]. [4] However, it is only a heuristic approach with no guarantees. The authors state,

> "Computing the integral over opponent strategies depends on the form of the prior but is difficult in any event. For Dirichlet priors, it is possible to compute the posterior exactly but the calculation is expensive except for small games with relatively few observations. This makes the exact BBR an ideal goal rather than a practical approach. For real play, we must consider approximations to BBR."

However, we see no justification for the claim that it is possible to compute the posterior exactly in prior work, and there could easily be no closed-form solution. In this paper we present a solution for this problem, leading to the first exact optimal algorithm for performing Bayesian opponent exploitation in imperfect-information games. While the claim is correct that the computation is expensive for large games, we show that in a small yet realistic game it outperforms all prior approaches, which are based on sampling. Furthermore, we show that the computation can run extremely quickly even for large number of observations (though it can run into numerical instability). We also present general theory and an algorithm for another natural prior distribution (uniform over a polyhedron).

[3] $B(\alpha)$ is the beta function $B(\alpha) = \frac{\prod \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)}$, where $\Gamma(n) = (n-1)!$ is the gamma function.

[4] In addition to Bayesian Best Response, the paper also considers heuristic approaches for approximating several other response functions: Max A Posteriori Response and Thompson's Response.

II. Meta-algorithm

The problem of developing efficient algorithms for optimizing against a posterior distribution, which is a probability distribution over mixed strategies for the opponent (which are themselves distributions over pure strategies) seems daunting. We need to be able to compactly represent the posterior distribution and efficiently compute a best response to it. Fortunately, we show that our payoff of playing any strategy σ_i against a probability distribution over mixed strategies for the opponent equals our payoff of playing σ_i against the mean of the distribution. Thus, we need only represent and respond to the single strategy that is the mean of the distribution, and not to the full distribution. While this result was likely known previously, we have not seen it stated explicitly, and it is important enough to be highlighted so that it is on the radar of the AI community.

Suppose the opponent is playing mixed strategy σ_{-i} where $\sigma_{-i}(s_{-j})$ is the probability that he plays pure strategy $s_{-j} \in S_{-j}$. By definition of expected utility, $u_i(\sigma_i, \sigma_{-i}) = \sum_{s_{-j} \in S_{-j}} \sigma_{-i}(s_{-j}) u_i(\sigma_i, s_{-j})$. We can generalize this naturally to the case where the opponent is playing according to a probability distribution with pdf f_{-i} over mixed strategies: $u_i(\sigma_i, f_{-i}) = \int_{\sigma_{-i} \in \Sigma_{-i}} \left[f_{-i}(\sigma_{-i}) \cdot u_i(\sigma_i, \sigma_{-i}) \right]$. Let $\overline{f_{-i}}$ denote the mean of f_{-i}. That is, $\overline{f_{-i}}$ is the mixed strategy that selects s_{-j} with probability $\int_{\sigma_{-i} \in \Sigma_{-i}} \left[\sigma_{-i}(s_{-j}) \cdot f_{-i}(\sigma_{-i}) \right]$. Then we have the following:

Theorem 1.

$$u_i(\sigma_i, \overline{f_{-i}}) = u_i(\sigma_i, f_{-i}).$$

That is, the payoff against the mean of a strategy distribution equals the payoff against the full distribution.

Proof.

$$\begin{aligned}
& u_i(\sigma_i, \overline{f_{-i}}) \\
= & \sum_{s_{-j} \in S_{-j}} \left[u_i(\sigma_i, s_{-j}) \int_{\sigma_{-i} \in \Sigma_{-i}} \left[\sigma_{-i}(s_{-j}) \cdot f_{-i}(\sigma_{-i}) \right] \right] \\
= & \sum_{s_{-j} \in S_{-j}} \left[\int_{\sigma_{-i} \in \Sigma_{-i}} \left[u_i(\sigma_i, s_{-j}) \cdot \sigma_{-i}(s_{-j}) \cdot f_{-i}(\sigma_{-i}) \right] \right] \\
= & \int_{\sigma_{-i} \in \Sigma_{-i}} \left[\sum_{j \in S_{-j}} \left[u_i(\sigma_i, s_{-j}) \cdot \sigma_{-i}(s_{-j}) \cdot f_{-i}(\sigma_{-i}) \right] \right] \\
= & \int_{\sigma_{-i} \in \Sigma_{-i}} \left[u_i(\sigma_i, \sigma_{-i}) \cdot f_{-i}(\sigma_{-i}) \right] \\
= & u_i(\sigma_i, f_{-i})
\end{aligned}$$

□

Theorem 1 applies to normal and extensive-form games (with perfect or imperfect information), for any number of players (σ_{-i} could be a joint strategy profile for all opponents).

Now suppose the opponent is playing according a prior distribution $p(\sigma_{-i})$, and let $p(\sigma_{-i}|x)$ denote the posterior

probability given observations x. Let $\overline{p(\sigma_{-i}|x)}$ denote the mean of $p(\sigma_{-i}|x)$. As an immediate consequence of Theorem 1, we have the following corollary.

Corollary 1. $u_i(\sigma_i, \overline{p(\sigma_{-i}|x)}) = u_i(\sigma_i, p(\sigma_{-i}|x))$.

Corollary 1 implies the meta-procedure for optimizing performance against an opponent using p:

Algorithm 1 Meta-algorithm for Bayesian opponent exploitation

Inputs: Prior distribution p_0, response functions r_t

$M_0 \leftarrow \overline{p_0(\sigma_{-i})}$
$R_0 \leftarrow r_0(M_0)$
Play according to R_0
for $t = 1$ to T **do**
  $x_t \leftarrow$ observations of opponent's play at time step t
  $p_t \leftarrow$ posterior distribution of opponent's strategy given prior p_{t-1} and observations x_t
  $M_t \leftarrow$ mean of p_t
  $R_t \leftarrow r_t(M_t)$
  Play according to R_t

There are several challenges for applying Algorithm 1. First, it assumes that we can compactly represent the prior and posterior distributions p_t, which have infinite domain (the set of opponents' mixed strategy profiles). Second, it requires a procedure to efficiently compute the posterior distributions given the prior and the observations, which requires updating potentially infinitely many strategies. Third, it requires an efficient procedure to compute the mean of p_t. And fourth, it requires that the full posterior distribution from one round be compactly represented to be used as the prior in the next round. We can address the fourth challenge by using a modified update step:

$p_t \leftarrow$ posterior distribution of opponent's strategy given prior p_0 and observations $x_1, \ldots, x_t$.

We will be using this new rule in our main algorithm.

The response functions r_t (which return a strategy for ourselves that performs well against input opponents' strategies) could be standard best response, for which linear-time algorithms exist in games of imperfect information (and a recent approach has enabled efficient computation in extremely large games [8]). They could also be a more robust response, e.g., one that places a limit on the exploitability of our own strategy, perhaps one that varies over time based on performance (or a lower-variance estimator) [9], [10], [11]. In particular, the restricted Nash response has been demonstrated to outperform best response against agents in limit Texas hold 'em whose actual strategy may differ substantially from the exact model [9].

III. Robustness of the approach

It has been pointed out that, empirically, the approach described is not robust: if we play a full best response to a point estimate of the opponent's strategy we can have very high exploitability ourselves, and could perform very poorly if in fact we are wrong about our model [9]. This could happen for several reasons. Our modeling algorithm could be incorrect: it could make an incorrect assumption about the prior and form of the opponent's distribution. This could happen for several reasons. One reason is that the opponent could actually be changing his strategy over time (possibly either by improving his own play or by adapting to our play), in which case a model that assumes a static opponent could be predicting a strategy that the opponent is no longer using. The opponent could also have modified his play strategically in an attempt to deceive us by playing one way initially and then counter-exploiting us as we attempt to exploit the model we have formed from his initial strategy (e.g., the opponent initially starts off playing extremely conservatively, then switches to a more aggressive style as he suspects we will start to exploit his extreme conservatism). His initial strategy need not arise from deception: it is also possible that simply due to chance events (either due to his own randomization in his strategy or due to the states of private information selected by chance) the opponent has appeared to be playing in a certain way (e.g., very conservatively), and as he becomes aware of this conservative "image," naturally it occurs to him to modify his play by becoming more aggressive.

A second reason that we could be wrong in our opponent model other than our algorithm incorrectly modeling the opponents' dynamic approach is that our observations of his play are very noisy (due to both randomization in the opponent's strategy and to the private information selected by chance), particularly over a small sample. Even if our approach is correct and the opponent is in fact playing a static strategy according to the distribution assumed by the modeling algorithm, it is very unlikely that our actual perception of his strategy is precisely correct.

A third reason, of course, is that the opponent may be following a static strategy that does not exactly conform to our model for the prior and/or sampling method used to generate the posterior.

We would like an approach that is robust in the event that our model of the opponent's strategy is incorrect, whichever the cause may be. Prior work has considered a model where the opponent plays according to a model x_{-i} with probability p and with probability $1-p$ plays a nemesis to our strategy [9]. For carefully selected values of p (typically 0.95 or 0.99), they show that this can achieve a relatively high level of exploitation (similar to a full best response) with a significantly smaller worst-case exploitability. We note that, as described in Section II, Algorithm 1 can be integrated with any response function, not necessarily a full best response, and so r_t could be selected to be the Restricted Nash Response from prior work [9]. However, it seems excessively conservative to give the opponent credit for playing a full nemesis to our strategy; if we are relatively confident in our opponent model, then a more reasonable robustness criterion would be to explore performance as we allow the opponent's strategy to differ by a small amount from the predicted strategy (i.e., the opponent

is playing a strategy that is very close to our model, and not necessarily putting weight on a full nemesis).

Suppose we believe the opponent is playing x_{-i}, while he is actually playing x'_{-i}. Let M be the maximum absolute value of a utility to player i, and let N be the maximum number of actions available to a player. Let $\epsilon > 0$ be arbitrary. Then, if $|x_{-i}(j) - x'_{-i}(j)| < \delta$ for all j, where $\delta = \frac{\epsilon}{MN}$,

$$\begin{aligned}
& |u_i(\sigma^*, x_{-i}) - u_i(\sigma^*, x'_{-i})| \\
= & \left| \sum_j (x_{-i}(j) - x'_{-i}(j)) u_i(\sigma^*, s_{-j}) \right| \\
<= & \sum_j \left| \left(x_{-i}(j) - x'_{-i}(j)\right) u_i(\sigma^*, s_{-j}) \right| \\
<= & \sum_j \left(\left|x_{-i}(j) - x'_{-i}(j)\right| \cdot |u_i(\sigma^*, s_{-j})| \right) \\
<= & \sum_j \left(|x_{-i}(j) - x'_{-i}(j)| \cdot M \right) \\
< & M \sum_j \delta <= MN\delta = MN \cdot \frac{\epsilon}{MN} = \epsilon
\end{aligned}$$

This same analysis can be applied directly to show that our payoff is continuous in the opponent's strategy for many popular distance functions (i.e., for any distance function where one strategy can get arbitrarily close to another as the components get arbitrarily close). For instance this would apply to L1, L2, and earth mover's distance, which have been applied previously to compute distances been strategies within opponent exploitation algorithms [4]. Thus, if we are slightly off in our model of the opponent's strategy, even if we are doing a full best response we will do only slightly worse.

IV. Exploitation algorithm for Dirichlet prior

As described in Section I the Dirichlet distribution is the conjugate prior for the multinomial distribution, and therefore the posterior is also a Dirichlet distribution, with the parameters α_i updated to reflect the new observations. Thus, the mean of the posterior can be computed efficiently by computing the strategy for the opponent in which he plays each strategy in proportion to the updated weight, and Algorithm 1 yields an exact efficient algorithm for computing the Bayesian best response in normal-form games with a Dirichlet prior. However, the algorithm does not apply to games of imperfect information since we do not observe the private information held by the opponent, and therefore do not know which of his action counters we should increment. In this section we will present a new algorithm for this setting. We present it in the context of a representative motivating game where the opponent is dealt a state of private information and then takes publicly-observable action, and present the algorithm for the general setting in Section IV-C.

We are interested in studying the following two-player game setting. Player 1 is given private information state x_i (according to a probability distribution). Then he takes a publicly observable action a_i. Player 2 then takes an action after observing player 1's action (but not his private information), and both players receive a payoff. We are interested in player 2's problem of inferring the (assumed stationary) strategy of player 1 after repeated observations of the public action taken (but not the private information). Note that this setting is very general. For example, in poker x_i could denote the opponent's private card(s) and a_i denote the amount he bets, and in an ad auction x_i could denote his valuation (e.g., high or low), and a_i could denote the amount he bids [12].

A. Motivating game and algorithm

For concreteness and motivation, consider the following poker game instantiation of this setting, where we play the role of player 2. Let's assume that in this two-player game, player 1 is dealt a King (K) and Jack (J) with probability $\frac{1}{2}$, while player 2 is always dealt a Queen. Player 1 is allowed to make a big bet of \$10 (b) or a small bet of \$1 (s), and player 2 is allowed to call or fold. If player 2 folds, then player 1 wins the \$2 pot (for a profit of \$1); if player 1 bets and player 2 calls then the player with the higher card wins the \$2 pot plus the size of the bet.

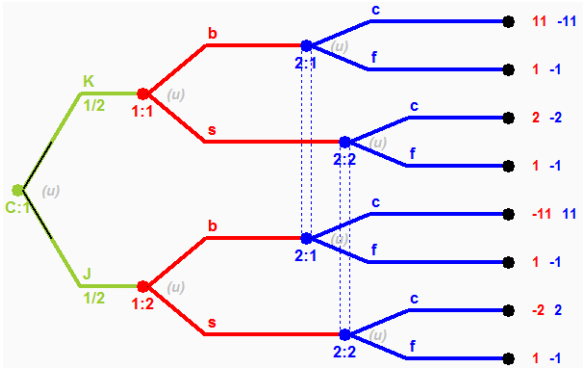

Fig. 1. Chance deals player 1 king or jack with probability $\frac{1}{2}$ at the green node. Then player 1 selects big or small bet at a red node. Then player 2 chooses call or fold at a blue node.

If we observe player 1's card after each hand, then we can apply the approach described above, where we maintain a counter for player 1 choosing each action with each card that is incremented for the selected action. However, if we do not observe player 1's card after the hand (e.g., if we fold), then we would not know whether to increment the counter for the king or the jack. To simplify analysis, we will assume that we never observe the opponent's private card after the hand (which is not realistic since we would observe his card if he bets and we call); we can assume that we do not observe our payoff either until all game iterations are complete, since that could allow us to draw inferences about the opponent's card. There are no known algorithms even for the simplified case of fully unobservable opponent's private information. We suspect that an algorithm for the case when the opponent's private information is sometimes observed can be constructed based on our algorithm, and we plan to study this problem in future work.

From analysis in the accompanying tech report [13], we are able to compute a closed-form expression for the expectation of the posterior probability that the opponent takes action b with a Jack given that we have just observed him take action b (the other quantities can be computed analogously), which is denoted by $P(b|O, J)$.

$$\frac{B(\alpha_{Kb}+1, \alpha_{Ks})B(\alpha_{Jb}+1, \alpha_{Js}) + B(\alpha_{Kb}, \alpha_{Ks})B(\alpha_{Jb}+2, \alpha_{Js})}{Z} \quad (1)$$

where the denominator Z is equal to

$$B(\alpha_{Kb}+1, \alpha_{Ks})B(\alpha_{Jb}+1, \alpha_{Js}) + B(\alpha_{Kb}, \alpha_{Ks})B(\alpha_{Jb}+2, \alpha_{Js})$$
$$+B(\alpha_{Kb}+1, \alpha_{Ks})B(\alpha_{Jb}, \alpha_{Js}+1) + B(\alpha_{Kb}, \alpha_{Ks})B(\alpha_{Jb}+1, \alpha_{Js}+1).$$

Note that the algorithm we have presented applies for the case where we play one more game iteration and collect one additional observation. However, it is problematic for the general case we are interested in where we play many game iterations, since the posterior distribution is not Dirichlet, and therefore we cannot just apply the same procedure in the next iteration using the computed posterior as the new prior. We will need to derive a new expression for $P(b|O, J)$ for this setting. Suppose that we have observed the opponent play action b for θ_b times and s θ_s times (in addition to the number of fictitious observations reflected in the prior α), though we do not observe his card. Then $P(b|O, J)$ equals

$$\frac{\sum_{i=0}^{\theta_b}\sum_{j=0}^{\theta_s} B(\alpha_{Kb}+i, \alpha_{Ks}+j)B(\alpha_{Jb}+\theta_b-i+1, \alpha_{Js}+\theta_s-j)}{Z} \quad (2)$$

The normalization term is

$$Z = \sum_i \sum_j [B(\alpha_{Kb}+i, \alpha_{Ks}+j)B(\alpha_{Jb}+\theta_b-i+1, \alpha_{Js}+\theta_s-j)$$
$$+B(\alpha_{Kb}+i, \alpha_{Ks}+j)B(\alpha_{Jb}+\theta_b-i, \alpha_{Js}+\theta_s-j+1)].$$

Details of the derivation are in the tech report.

Thus the algorithm for responding to the opponent is the following. We start with the prior counters on each private information-action combination, α_{Kb}, α_{Ks}, etc. We keep separate counters θ_b, θ_s for the number of times we have observed each action during play. Then we combine these counters according to Equation 2 in order to compute the strategy for the opponent that is the mean of the posterior given the prior and observations, and we best respond to this strategy, which gives us the same payoff as best responding to the full posterior distribution according to Theorem 1. There are only O(n^2) terms in the expression given by Equation 2, so this algorithm is efficient.

B. Example

Suppose the prior is that the opponent played b with K 10 times, played s with K 3 times, played b with J 4 times, and played s with J 9 times. Thus $\alpha_{Kb} = 10, \alpha_{Ks} = 3, \alpha_{Jb} = 4, \alpha_{Js} = 9$. Now suppose we observe him play b at the next iteration. Applying our algorithm using Equation 1 gives

$$p(b|O, J) = \frac{B(11,3)B(5,9) + B(10,3)(6,9)}{Z} = \frac{2.65209525e^{-7}}{Z}$$

$$p(s|O, J) = \frac{B(11,3)B(4,10) + B(10,3)(5,10)}{Z} = \frac{5.5888056e^{-7}}{Z}$$

$$\longrightarrow p(b|O, J) = \frac{2.65209525e^{-7}}{2.65209525e^{-7} + 5.5888056e^{-7}} = 0.3218210361.$$

So we think that with a jack he is playing a strategy that bets big with probability 0.322 and small with probability 0.678. Notice that previously we thought his probability of betting big with a jack was $\frac{4}{13} = 0.308$, and had we been in the setting where we always observe his card after gameplay and observed that he had a jack, the posterior probability would be $\frac{5}{14} = 0.357$.

An alternative "naïve" (and incorrect) approach would be to increment α_{Jb} by $\frac{\alpha_{Jb}}{\alpha_{Jb}+\alpha_{Kb}}$, the ratio of the prior probability that he bets big given J to the total prior probability that he bets big. This gives a posterior probability of him betting big with J of $\frac{4+\frac{4}{13}}{14} = 0.308$, which differs significantly from the correct value. It turns out that this approach is actually equivalent to just using the prior:

$$\frac{x + \frac{x}{x+y}}{x+y+1} \cdot \frac{x+y}{x+y} = \frac{x(x+y)+x}{(x+y+1)(x+y)} = \frac{x}{x+y}$$

C. Algorithm for general setting

We now consider the general setting where the opponent can have n different states of private information according to an arbitrary distribution π and can take m different actions. Assume he is given private information x_i with probability π_i, for $i = 1, \ldots, n$, and can take action k_i, for $i = 1, \ldots, m$. Assume the prior is Dirichlet with parameters α_{ij} for the number of times action j was played with private information i (so the mean of the prior has the player selecting action k_j at state x_i with probability $\frac{\alpha_{ij}}{\sum_j \alpha_{ij}}$). Assume that action k_{j^*} was observed in a new time step, while the opponent's private information was not observed. We now compute the expectation for the posterior probability that the opponent plays k_{j^*} with private information x_{i^*}.

$$\begin{aligned} & P(A = k_{j^*}|O, C = x_{i^*}) \\ = & \frac{\int \left[q_{k_j^*|x_i^*} \sum_{i=1}^n \left[\pi_i q_{k_{j^*}|x_i} \prod_{h=1}^m \prod_{j=1}^n q_{k_h|x_j}^{\alpha_{jh}-1} \right] \right]}{p(O) \prod_{i=1}^n B(\alpha_{i1}, \ldots, \alpha_{im})} \\ = & \frac{\sum_i \left[\pi_i \prod_j B(\gamma_{1j}, \ldots, \gamma_{nj}) \right]}{Z}, \end{aligned}$$

where $\gamma_{ij} = \alpha_{ij} + 2$ if $i = i^*$ and $j = j^*$, $\gamma_{ij} = \alpha_{ij} + 1$ if $j = j^*$ and $i \neq i^*$, and $\gamma_{ij} = \alpha_{ij}$ otherwise. If we denote the numerator by $\tau_{i^*j^*}$ then $Z = \sum_{i^*} \tau_{i^*j^*}$. Notice that the product is over n terms, and therefore the total number of terms will be exponential in n (it is O($m \cdot 2^n$)).

For the case of multiple observed actions, the posterior is not Dirichlet and cannot be used directly as the prior for the next iteration. Suppose we have observed action k_j θ_j times (in addition to the number of fictitious times indicated by the prior counts α_{ij}). We compute $P(q|O)$ analogously as

$$P(q|O) = \frac{\sum_{i=1}^{n}\left[\pi_i \sum_{\{\rho_{ab}\}} \prod_{h=1}^{m} \prod_{j=1}^{n} q_{k_h|x_j}^{\alpha_{jh}-1+\rho_{jh}}\right]}{p(O)\prod_{i=1}^{n} B(\alpha_{i1},\ldots,\alpha_{im})},$$

where the $\sum_{\{\rho_{ab}\}}$ is over all values $0 \leq \rho_{ab} \leq \theta_b$ with $\sum_a \rho_{ab} = \theta_b$ for each b, for $1 \leq a \leq n$, $1 \leq b \leq m$:

$$\sum_{\{\rho_{ab}\}} = \sum_{\rho_{1b}=0}^{\theta_b} \sum_{\rho_{2b}=0}^{\theta_b-\rho_{1b}} \cdots \sum_{\rho_{n-1,b}=0}^{\theta_b-\sum_{r=0}^{n-2}\rho_{rb}} \sum_{\rho_{nb}=\theta_b-\sum_{r=0}^{n-2}\rho_{rb}}^{\theta_b-\sum_{r=0}^{n-1}\rho_{rb}}.$$

The expression for the full posterior distribution is

$$P(q|O) = \frac{\sum_i \left[\pi_i \sum_{\{\rho_{ab}\}} \prod_h B(\alpha_{1h}+\rho_{1h},\ldots,\alpha_{nh}+\rho_{nh})\right]}{Z}$$

The total number of terms is $O\left(\left(\frac{(T+n)!}{n!T!}\right)^m\right)$, which is exponential in the number of private information states and actions, but polynomial in the number of iterations.

The following theorem shows an approach for computing products of the beta function that leads to an exponential improvement in the running time of the algorithm for one observation, and reduces the dependence on m for the multiple observation setting from exponential to linear, though the complexity still remains exponential in n and T for the latter. See tech report for full details [13].

Theorem 2. *Define $\gamma_j = \sum_{i=1}^{n} \gamma_{ij}$ and the empirical probability distribution $\hat{P}_j(i) = \frac{\gamma_{ij}}{\sum_{i=1}^{n}\gamma_{ij}} = \frac{\gamma_{ij}}{\gamma_j}$. Define the Gamma function $\Gamma(x) = \int_0^\infty x^{z-1}e^{-x}\,\mathrm{d}x$, for integer x, $\Gamma(x) = (x-1)!$. Now define the entropy of $\hat{P}_i$ as $E(\hat{P}_j) = -\sum_{i=1}^{n} \hat{P}_j(i)\ln\hat{P}_j(i)$. Then we have $\prod_{j=1}^{m} B(\gamma_{1j},\ldots,\gamma_{nj})$ equals*

$$\exp\left(\sum_{j=1}^{m}\left(-\gamma_j E(\hat{P}_j) - \frac{1}{2}(n-1)\ln(\gamma_j) + \sum_{i=1}^{n}\ln(P_j(i)) + d\right)\right).$$

Here d is a constant such that $\frac{1}{2}\ln(2\pi)n - 1 \leq d \leq n - \frac{1}{2}\ln(2\pi)$, where $\ln(2\pi) \approx 0.92$.

V. Algorithm for uniform prior distribution

Another prior that has been studied previously is the uniform distribution over a polyhedron. This can model the situation when we think the opponent is playing uniformly within some region of a fixed strategy, such as a specific Nash equilibrium or a "population mean" strategy based on historical data. Prior work has used this model to generate a class of opponents who are more sophisticated than opponents who play uniformly at random over the entire space [11]). For example, in rock-paper-scissors, we may think the opponent is playing a strategy uniformly out of strategies that play each action with probability within [0.31,0.35], as opposed to completely random over [0,1].

Let $v_{i,j}$ denote the jth vertex for player i, where vertices correspond to mixed strategies. Let p^0 denote the prior distribution over vertices, where $p^0_{i,j}$ is the probability that player i plays the strategy corresponding to vertex $v_{i,j}$. Let V_i denote the number of vertices for player i. Algorithm 2 computes the Bayesian best response in this setting. Correctness follows straightforwardly by applying Corollary 1 with the formula for the mean of the uniform distribution.

Algorithm 2 Algorithm for opponent exploitation with uniform prior distribution over polyhedron

Inputs: Prior distribution over vertices p^0, response functions r_t for $0 \leq t \leq T$

$M_0 \leftarrow$ strategy profile assuming opponent i plays each vertex $v_{i,j}$ with probability $p^0_{i,j} = \frac{1}{V_i}$
$R_0 \leftarrow r_0(M_0)$
Play according to R_0
for $t = 1$ to T **do**
  for $i = 1$ to N **do**
    $a_i \leftarrow$ action taken by player i at time step t
    for $j = 1$ to V_i **do**
      $p^t_{i,j} \leftarrow p^{t-1}_{i,j} \cdot v_{i,j}(a_i)$
    Normalize the $p^t_{i,j}$'s so they sum to 1
  $M_t \leftarrow$ strategy profile assuming opponent i plays each vertex $v_{i,j}$ with probability $p^t_{i,j}$
  $R_t \leftarrow r_t(M_t)$
  Play according to R_t

VI. Experiments

We ran experiments on the game described in Section IV-A. For the beta function computations we used the Colt Java math library.For our first set of experiments we tested our basic algorithm which assumes that we observe a single opponent action (Equation 1). We varied the Dirichlet prior parameters to be uniform in $\{1,n\}$ to explore the runtime as a function of the size of the prior (since computing larger values of the Beta function can be challenging). The results (Table I) show that the computation is very fast even for large n, with running time under 8 microseconds for $n = 500$. However, we also observe frequent numerical instability for large n. The second row shows the percentage of the trials for which the algorithm produced a result of "NaN" (which typically results from dividing zero by zero). This jumps from 0% for $n = 50$ to 8.8% for $n = 100$ to 86.9% for $n = 200$. This is due to instability of algorithms for computing the beta function. We used the best publicly available beta function solver, but perhaps there could be a different solver that leads to better performance in our setting (e.g., it trades off runtime for additional precision). Despite the cases of instability, the results indicate that the algorithm runs extremely fast for hundreds of prior observations, and since it is exact, it is the best algorithm for the settings in which it produces a valid output. Note that $n = 100$ corresponds to 400 prior observations on average since there are four parameters, and that the experiments in previous work used a horizon of 200 hands per match against an opponent [7].

We tested our generalized algorithm for different numbers of observations, using a fixed Dirichlet prior with all parameters equal to 2 as in prior work [7]. We observe (Table II) that the algorithm runs quickly for large numbers of observations, though again it runs into numerical instability for large values. As one example, it takes 19ms for $\theta_b = 101$, $\theta_s = 100$.

n	10	20	50	100	200	500
Time	0.0005	0.0008	0.0018	0.0025	0.0034	0.0076
NaN	0	0	0	0.0883	0.8694	0.9966

TABLE I

RESULTS OF MODIFYING DIRICHLET PARAMETERS TO BE U{1,N} OVER ONE MILLION SAMPLES. FIRST ROW IS AVERAGE RUNTIME IN MILLISECONDS. SECOND ROW IS PERCENTAGE OF THE TRIALS THAT OUTPUT "NAN."

n	10	20	50	100	200	500	1000
Time	0.015	0.03	0.36	2.101	10.306	128.165	728.383
NaN	0	0	0	0	0.290	0.880	0.971

TABLE II

RESULTS USING DIRICHLET PRIOR WITH ALL PARAMETERS EQUAL TO 2 AND θ_b, θ_s IN U{1,N} AVERAGED OVER 1,000 SAMPLES. FIRST ROW IS AVERAGE RUNTIME (MS), SECOND ROW IS % OF TRIALS PRODUCING "NAN."

We compared our algorithm against the three heuristics described in previous work [7]. The first heuristic Bayesian Best Response (BBR) approximates the opponent's strategy by sampling strategies according to the prior and computing the mean of the posterior over these samples, then best-responding to this mean strategy; Max A Posteriori Response heuristic (MAP) samples strategies from the prior, computes the posterior value for these strategies, and plays a best response to the one with highest posterior value; Thompson's Response samples strategies from the prior, computes the posterior values, then samples one strategy for the opponent from these posteriors and plays a best response to it. For all approaches we used a Dirichlet prior with the standard values of 2 for all parameters. For all the sampling approaches we sampled 1,000 strategies from the prior for each opponent and used these strategies for all hands against that opponent (as was done in prior work [7]). Note that one can draw samples x_i from a Dirichlet distribution by first drawing independent samples y_i from Gamma distributions each with density $\text{Gamma}(\alpha_i, 1) = \frac{y_i^{\alpha_i - 1} e^{-y_i}}{\Gamma(\alpha_i)}$ and then setting $x_i = \frac{y_i}{\sum_j y_j}$.We also tested a best response strategy that knows the actual mixed strategy of the opponent, not just a distribution over his strategies, as well as the Nash equilibrium strategy.[5] Note that the game has a value to us of -0.75, so negative values are not necessarily indicative of "losing."

Table III shows that our exact Bayesian best response algorithm (EBBR) outperforms the heuristic approaches, as expected since it is optimal when the opponent's strategy is drawn from the prior (though performance is very similar to BBR and not statistically distinguishable until 25 iterations). BBR performed best out of the sampling approaches, which is not surprising because it is trying to approximate the optimal approach while the others are optimizing a different objective. All of the sampling approaches outperformed just following the Nash equilibrium, and as expected all exploitation approaches performed worse than playing a best response to the opponent's actual strategy. Note that, against an opponent drawn from a Dirichlet distribution with all parameters equal to 2 and no further observations of his play, our best response would be to always call, which gives us expected payoff of zero. Thus for the initial column the actual value for EBBR when averaged over all opponents would be zero. Against this distribution the Nash equilibrium has expected payoff -0.375.

[5]Note that the Nash equilibrium for player 2 is to call a big bet with probability $\frac{1}{4}$ and a small bet with probability 1 (the equilibrium for player 1 is to always bet big with K and to bet big with probability $\frac{5}{6}$ with J).

Algorithm	Initial	10	25
EBBR	$\mathbf{-0.00003 \pm 0.0003}$	$\mathbf{-0.0004 \pm 0.0009}$	$\mathbf{0.0002 \pm 0.0008}$
BBR	-0.00003 ± 0.0003	-0.0004 ± 0.0009	-0.0065 ± 0.0008
MAP	-0.1649 ± 0.0002	-0.2025 ± 0.0007	-0.2664 ± 0.0007
Thompson	-0.2098 ± 0.0002	-0.2224 ± 0.0007	-0.2996 ± 0.0007
FullBR	0.4975 ± 0.0002	0.4971 ± 0.0006	0.4978 ± 0.0005
Nash	-0.3750 ± 0.0000	-0.3749 ± 0.0001	-0.3751 ± 0.0001

TABLE III

COMPARISON WITH ALGORITHMS FROM PRIOR WORK, FULL BEST RESPONSE, AND NASH EQUILIBRIUM USING DIRICHLET PRIOR WITH PARAMETERS EQUAL TO 2. SAMPLING ALGORITHMS USE 1000 SAMPLES. FOR INITIAL COLUMN WE SAMPLED 100 MILLION OPPONENTS FROM THE PRIOR, FOR 10 ROUNDS WE SAMPLED ONE MILLION, AND FOR 25 ROUNDS 500,000. RESULTS ARE AVERAGE WINRATE PER HAND OVER ALL OPPONENTS WITH 95% CONFIDENCE INTERVALS.

On the positive side the exploitation approaches (particularly EBBR and BBR) are able to significantly outperform the Nash equilibrium strategy when given access to a reliable prior distribution; however, none of them are able to improve over time as a result of additional observations (EBBR and BBR perform around the same with more observations while Thompson and MAP perform noticeably worse). This indicates that, for this setting at least, just observing the opponent's public action and not private information is not additionally useful in comparison to the performance variance and the noise introduced from sampling. In order to successfully learn beyond the prior in imperfect-information settings, algorithms will need access to some of the opponents' private information. Previous experiments had also shown that when the sampling approaches are played against opponents drawn from the prior, the winning rates converge, typically very quickly (even with access to the opponent's private information in certain hands that went to *showdown*): "The independent Dirichlet prior is very broad, admitting a wide variety of opponents. It is encouraging that the Bayesian approach is able to exploit even this weak information to achieve a better result." [7]

We also tested the effect of using only 10 samples of the opponent's strategy for the sampling approaches. The approaches would then have a noisier estimate of the opponent's strategy and should achieve lower performance against the actual strategy of the opponent, though run significantly faster.

Thompson and MAP performed very similarly using 10 vs. 1000 samples (these approaches essentially end up selecting a single strategy from the set of samples to be used as the model, and the results indicate that they are relatively insensitive to the number of samples used), but BBR performs significantly worse. While the performance between EBBR and BBR was statistically indistinguishable for 1000 samples, EBBR significantly outperforms BBR with 10 samples, particularly for more iterations. As before the sampling approaches seem to actually perform worse over time as the noise propagates,

Alg	Initial	10	25	100
EBBR	**.0001 ± .0003**	**-.0003 ± .0003**	**.0002 ± .0002**	**-.0014 ± .0005**
BBR	-.0662 ± .0003	-.0902 ± .0003	-.1634 ± .0002	-.3127 ± .0004
MAP	-.1699 ± .0002	-.2060 ± .0002	-.2657 ± .0001	-.3082 ± .0004
Thomp.	-.2118 ± .0002	-.2247 ± .0002	-.2844 ± .0001	-.3725 ± .0004
FullBR	.4976 ± .0002	.4973 ± .0002	.4975 ± .0001	.4969 ± .0003
Nash	-.3750 ± .0000	-.3750 ± .0000	-.3750 ± .0000	-.3750 ± .0001

TABLE IV

COMPARISON OF OUR ALGORITHM WITH ALGORITHMS FROM PRIOR WORK (BBR, MAP, THOMPSON), FULL BEST RESPONSE, AND NASH EQUILIBRIUM USING DIRICHLET PRIOR WITH PARAMETERS EQUAL TO 2. THE SAMPLING ALGORITHMS EACH USE 10 SAMPLES FROM THE OPPONENT'S STRATEGY (AS OPPOSED TO 1000 SAMPLES FROM OUR EARLIER ANALYSIS). FOR THE INITIAL COLUMN WE SAMPLED 100 MILLION OPPONENTS FROM THE PRIOR, FOR 10 AND 25 ROUNDS WE SAMPLED TEN MILLION, AND 300,000 FOR 100 ROUNDS.

while the performance of EBBR remains about the same. The dropoff of BBR is particularly significant. The results indicate that EBBR would be particularly preferable over the sampling approaches if the number of available samples is small (e.g., due to running time considerations) and as the number of game iterations increases (though eventually EBBR can run into numerical stability issues described earlier).

VII. CONCLUSION

One of the most fundamental problems in game theory is learning to play optimally against opponents who may make mistakes. We presented the first exact algorithm for performing exploitation in imperfect-information games in the Bayesian setting using the most well-studied prior distribution for this problem, the Dirichlet distribution. Previously an exact algorithm had only been presented for normal-form games, and the best previous algorithm was a heuristic with no guarantees. We demonstrated experimentally that our algorithm can be practical and that it outperforms the best prior approaches, though it can run into numerical stability issues for large numbers of observations.

We presented a general meta-algorithm and new theoretical framework for studying opponent exploitation. Future work can extend our analysis to many important settings. For example, we would like to study the setting when the opponent's private information is only sometimes observed (we expect our approach can be extended easily to this setting) and general sequential games where the agents can take multiple actions (which we expect to be hard, as indicated by the analysis in the tech report). We would also like to extend analysis for any number of agents. Our algorithm is not specialized for two-player zero-sum games (it applies to general-sum games); if we are able to compute the mean of the posterior strategy against multiple opponent agents, then best responding to this strategy profile is just a single agent optimization and can be done in time linear in the size of the game regardless of the number of opponents. While the Dirichlet is the most natural prior for this problem, we would also like to study other important distributions. We presented an algorithm for the uniform prior distribution over a polyhedron, which could model the situation where we think the opponent is playing a strategy from a uniform distribution in a region around a particular strategy, such as a specific equilibrium or a "population mean" based on historical data.

Opponent exploitation is a fundamental problem, and our algorithm and extensions could be applicable to many domains that are modeled as an imperfect-information games. For example, many security game models have imperfect information, e.g., [14], [15], and opponent exploitation in security games has been a very active area of study, e.g., [16], [17]. It has also been proposed recently that opponent exploitation can be important in medical treatment [18].

REFERENCES

[1] X. Chen and X. Deng, "Settling the complexity of 2-player Nash equilibrium," in *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, 2006.

[2] R. Gibson, "Regret minimization in games and the development of champion multiplayer computer poker-playing agents," Ph.D. dissertation, University of Alberta, 2014.

[3] D. Koller, N. Megiddo, and B. von Stengel, "Fast algorithms for finding randomized strategies in game trees," in *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC)*, 1994, pp. 750–760.

[4] S. Ganzfried and T. Sandholm, "Game theory-based opponent modeling in large imperfect-information games," in *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2011.

[5] D. Fudenberg and D. Levine, *The Theory of Learning in Games*. MIT Press, 1998.

[6] G. W. Brown, "Iterative solutions of games by fictitious play," in *Activity Analysis of Production and Allocation*, T. C. Koopmans, Ed. John Wiley & Sons, 1951, pp. 374–376.

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

[8] M. Johanson, K. Waugh, M. Bowling, and M. Zinkevich, "Accelerating best response calculation in large extensive games," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[9] M. Johanson, M. Zinkevich, and M. Bowling, "Computing robust counter-strategies," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2007, pp. 1128–1135.

[10] M. Johanson and M. Bowling, "Data biased robust counter strategies," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

[11] S. Ganzfried and T. Sandholm, "Safe opponent exploitation," *ACM Transactions on Economics and Computation (TEAC)*, 2015, special issue on selected papers from EC-12.

[12] P. Tang, Z. Wang, and X. Zhang, "Optimal commitments in auctions with incomplete information," in *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2016.

[13] S. Ganzfried and Q. Sun, "Bayesian opponent exploitation in imperfect-information games," *CoRR*, vol. abs/1603.03491, 2016. [Online]. Available: http://arxiv.org/abs/1603.03491

[14] J. Letchford and V. Conitzer, "Computing optimal strategies to commit to in extensive-form games," in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 2010.

[15] C. Kiekintveld, M. Tambe, and J. Marecki, "Robust Bayesian methods for Stackelberg security games (extended abstract)," in *Autonomous Agents and Multi-Agent Systems*, 2010.

[16] J. Pita, M. Jain, M. Tambe, F. Ordóñez, and S. Kraus, "Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition," *Artificial Intelligence Journal*, vol. 174, no. 15, pp. 1142–1171, 2010.

[17] T. H. Nguyen, R. Yang, A. Azaria, S. Kraus, and M. Tambe, "Analyzing the effectiveness of adversary modeling in security games," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2013.

[18] T. Sandholm, "Steering evolution strategically: Computational game theory and opponent exploitation for treatment planning, drug design, and synthetic biology," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2015, senior Member Blue Skies Track.