Estimating α-Rank by Maximizing Information Gain

Tabish Rashid[*],[1] **Cheng Zhang,**[2] **Kamil Ciosek**[2]

[1]University of Oxford
[2]Microsoft Research, Cambridge, UK
tabish.rashid@cs.ox.ac.uk, {cheng.zhang, kamil.ciosek}@microsoft.com

Abstract

Game theory has been increasingly applied in settings where the game is not known outright, but has to be estimated by sampling. For example, meta-games that arise in multi-agent evaluation can only be accessed by running a succession of expensive experiments that may involve simultaneous deployment of several agents. In this paper, we focus on α-rank, a popular game-theoretic solution concept designed to perform well in such scenarios. We aim to estimate the α-rank of the game using as few samples as possible. Our algorithm maximizes information gain between an epistemic belief over the α-ranks and the observed payoff. This approach has two main benefits. First, it allows us to focus our sampling on the entries that matter the most for identifying the α-rank. Second, the Bayesian formulation provides a facility to build in modeling assumptions by using a prior over game payoffs. We show the benefits of using information gain as compared to the confidence interval criterion of ResponseGraphUCB, and provide theoretical results justifying our method.

1 Introduction

Traditionally, game theory is applied in situations where the game is fully known. More recently, empirical game theory addresses the setting where this is not the case, instead, the game is initially unknown and has to be interacted with by sampling (Wellman 2006). One area in which this is becoming increasingly common is the ranking of trained agents relative to one another. Specifically, in the field of Reinforcement Learning game-theoretic rankings are used not just as a metric for measuring algorithmic progress (Balduzzi et al. 2018), but also as an integral component of many population-based training methods (Muller et al. 2020; Lanctot et al. 2017; Vinyals et al. 2019). In particular, for ranking, two popular solution concepts have recently emerged: Nash averaging (Balduzzi et al. 2018; Nash 1951) and α-rank (Omidshafiei et al. 2019).

In this paper, we aim to estimate the α-rank of a game using as few samples as possible. We use the α-rank solution concept for two reasons. First, it admits a unique solution whose computation easily scales to K-player games. Second, unlike older schemes such as Elo (Elo 1978), α-rank

is designed with intransitive interactions in mind. Because measuring payoffs can be very expensive, it is important to do it by using as few samples as possible. For example, playing a match of chess (Silver et al. 2017) can take roughly 40 minutes (assuming a typical game-length of 40 and up to 1 minute per move as used during evaluation), and playing a full game of Dota 2 can take up to 2 hours (Berner et al. 2019). *Our objective is thus to accurately estimate the α-rank using a small number of payoff queries.*

Rowland et al. (2019) proposed ResponseGraphUCB (RG-UCB) for this purpose, inspired by pure exploration bandit literature. RG-UCB maintains confidence intervals over payoffs. When they don't overlap, it draws a conclusion about their ordering, until all comparisons relevant for the computation of α-rank have been made. While this is provably sufficient to determine the true α-rank with a high probability in the infinite-α regime, their approach has two important limitations. First, since the frequentist criterion is indirect, relying on payoff ordering rather than the α-rank, the obtained payoffs aren't always used optimally. Second, it is nontrivial to include useful domain knowledge about the entries or structure of the payoff matrix.

To remedy these problems, we propose a Bayesian approach. Specifically, we utilize a Gaussian Process to maintain an epistemic belief over the entries of the payoff matrix, providing a powerful framework in which to supply domain knowledge. This payoff distribution induces an epistemic belief over α-ranks. We determine which payoff to sample by maximizing information gain between the α-rank belief and the obtained payoff. This allows us to focus our sampling on the entries that are expected to have the largest effect on our belief over possible α-ranks.

Contributions: Theoretically, we justify the use of information gain by showing a regret bound for a version of our criterion in the infinite-α regime. Empirically, our contribution is threefold. First, we compare to RG-UCB on stylized games, showing that maximizing information gain provides competitive performance by focusing on sampling the more relevant payoffs. Second, we evaluate another objective based on minimizing the Wasserstein divergence, which offers competitive performance while being computationally much cheaper. Finally, we demonstrate the benefit of building in prior assumptions.

---

2 Background

Games and α-Rank A game with K players, each of whom can play S_k strategies is characterized by its expected payoffs $M \in \mathcal{R}^{S_1 \times \ldots \times S_K}$ (Fudenberg and Tirole 1991). Letting $\mathcal{S} = S_1 \times \ldots \times S_K$ be the space of pure strategy profiles, the game also specifies a distribution over the payoffs associated with each player when $s \in \mathcal{S}$ is played. The α-rank of a game is computed by first defining an irreducible Markov Chain whose nodes are pure strategy profiles in $\mathcal{S}$. We denote the stochastic matrix defining this chain as C. The transition probabilities of the chain C are calculated as follows: Let $\sigma, \tau \in \mathcal{S}$ be such that τ only differs from σ in a single player's strategy and let $\eta = (\sum_{k=1}^{K}(|S_k| - 1))^{-1}$ be the reciprocal of the total number of those distinct τ. Let $M_k(\sigma)$ denote the expected payoff for player k when σ is played. Then, the probability of transitioning from σ to τ which varies only in player k's strategy is

$$C_{\sigma,\tau} = \begin{cases} \eta \frac{1-\exp(-\alpha(M_k(\tau)-M_k(\sigma)))}{1-\exp(-\alpha m(M_k(\tau)-M_k(\sigma)))} & \text{if } M_k(\tau) \neq M_k(\sigma), \\ \frac{\eta}{m} & \text{otherwise.} \end{cases}$$

$C_{\sigma,\upsilon} = 0$ for all υ that differ from σ in more than a single player's strategy, $C_{\sigma,\sigma} = 1 - \sum_{\tau \neq \sigma} C_{\sigma,\tau}$ to ensure a valid transition distribution, and $\alpha \geq 0, m \in \mathbb{N}^{>0}$ are parameters of the algorithm. We define the α-rank $r \in \mathcal{R}^{|\mathcal{S}|}$ as the unique stationary distribution of the chain C (Omidshafiei et al. 2019; Rowland et al. 2019) as $\alpha \to \infty$. In practice, a large finite value of α is used, or a perturbed version of the transition matrix C is used with an infinite α to ensure the resulting Markov Chain C is irreducible.

Single Population α-Rank In this paper we focus on the infinite-α regime and restrict our attention to the 2-player single population case of α-rank which differs slightly from above. Importantly, our method can be easily applied to multiple populations as described above in a straightforward way, but we focus on the single population case for simplicity. Let $S = S_1$ and $M(\sigma, \tau)$ denote the payoff when the first player plays σ and the second player plays τ. Note that $S_1 = S_2$ since the single population case considers a player playing a game against an identical player. In this particular setting, the α-rank $r \in \mathcal{R}^{|S|}$ and the perturbed transition matrix $C \in \mathcal{R}^{S \times S}$ is calculated as follows:

$$C_{\sigma,\tau} = \begin{cases} (|S|-1)^{-1}(1-\epsilon) & \text{if } M(\tau,\sigma) > M(\sigma,\tau), \\ (|S|-1)^{-1}\epsilon & \text{if } M(\tau,\sigma) < M(\sigma,\tau), \\ 0.5(|S|-1)^{-1} & \text{if } M(\tau,\sigma) = M(\sigma,\tau), \end{cases}$$

for $\sigma \neq \tau$. $C_{\sigma,\sigma} = 1 - \sum_{\tau \neq \sigma} C_{\sigma,\tau}$ again to ensure a valid transition distribution and ϵ is a small perturbation to ensure irreducibillity of the resulting chain. We abstract the above computation into the α-rank function $f : \mathcal{M} \to \mathcal{R}^{|S|}$, where $\mathcal{M}$ is the space of 2-player payoff matrices with S strategies for each player.

Wasserstein Divergence Let p and q be probability distributions supported on $\mathcal{X}$, and $c : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ be a distance. Define Π as the space of all joint probability distributions with marginals p and q. Wasserstein divergence (Villani 2008) with cost function c, is defined as:

$$\mathcal{W}_c(p,q) := \min_{\pi \in \Pi} \int_{\mathcal{X} \times \mathcal{X}} c(x,y) d\pi(x,y).$$

In this paper, we will utilize the Wasserstein distance between our belief distributions over α-rank, and so we set $\mathcal{X} = \Delta^{S-1}$, the $(S-1)$ probability simplex, and use $c(x,y) = \frac{1}{2}\|x - y\|_1$, i.e. the total variation distance. We will drop the suffix and denote this simply as $\mathcal{W}$.

3 Related Work

There are many methods related to the ranking and evaluation of agents in games. Elo (Elo 1978) and TrueSkill (Herbrich, Minka, and Graepel 2007; Minka, Cleven, and Zaykov 2018) both quantify the performance of an agent using a single number, which means they are unable to model *intransitive* interactions. Chen and Joachims (2016) extend TrueSkill to better model such interactions, while Balduzzi et al. (2018) do the same for Elo, improving its predictive power by introducing additional parameters. Balduzzi et al. (2018) also re-examines the use of Nash equilibrium, proposing to disambiguate across possible equillibria by picking the one with maximum entropy. However, it is well known that computing the Nash equilibrium is computationally difficult (Daskalakis, Goldberg, and Papadimitriou 2009) and only computationally tractable for restricted classes of games. In this paper, we focus on α-rank (Omidshafiei et al. 2019) since it has been designed with *intransitive* interactions in mind, it is computationally tractable for N-player games and shows considerable promise as a component of self-play frameworks (Muller et al. 2020).

Empirical Game Theory (Wellman 2006) is concerned with situations in which a game can only be interacted with through sampling. The most related work to ours investigates sampling strategies and concentration inequalities for the Nash equilibrium as opposed to the α-rank. Walsh, Parkes, and Das (2003) introduce Heuristic Payoff Tables (HPTs) in order to choose the samples that provide the most information about the currently chosen Nash equilibrium, where information is quantified as the reduction in estimated error. This differs from our approach both in the use of α-rank as opposed to the Nash equilibrium as our solution concept, and in the criterion used to select the observed payoff. Tuyls et al. (2020) provide concentration bounds for estimated Nash equilibria. Jordan, Vorobeychik, and Wellman (2008) find Nash equilibria from limited data by using information gain on distributions over strategies, a concept different from our information gain on distributions over ranks. We also utilize α-rank as the solution concept, rather than Nash equilibria.

Muller et al. (2020) utilise α-rank as part of a PSRO (Lanctot et al. 2017) framework. They do not use an adaptive sampling strategy for deciding which entries to sample, but are a natural application for applying our algorithm (and RG-UCB). Yang et al. (2019) introduce an approximate gradient-based algorithm which does not require access to the entire payoff matrix at once in order to compute α-rank.

Although their method does not require the entire payoff matrix at every iteration, it is not designed for operating in the same incomplete information setting that we explore in this paper since they assume every entry can be cheaply queried with no noise. Srinivas et al. (2009) prove regret bounds for Bayesian optimization with GPs. We use their concentration result to derive our bounds as well as as inspiration for our information gain criterion.

ResponseGraphUCB

Closest to our work is ResponseGraphUCB (RG-UCB) introduced by Rowland et al. (2019), which can be viewed as a frequentist analogue to our method which also operates in the infinite-α regime. RG-UCB first specifies an error threshold $\delta > 0$ and then samples payoffs until a stopping criteria determines the estimated α-rank is correct with probability at least $1 - \delta$. A key observation that RG-UCB relies on, is that in the infinite-α regime only the ordering between relevant payoffs is important. e.g. For pure strategy profiles σ and τ (with payoffs $M_k(\sigma)$ and $M_k(\tau)$ respectively) that are used in the computation of the Markov Chain transition probabilities, determining whether $M_k(\sigma) > M_k(\tau)$ or $M_k(\sigma) < M_k(\tau)$ is enough to know the transition probability accurately (their magnitude difference $|M_k(\sigma) - M_k(\tau)|$ is unimportant). RG-UCB maintains $(1 - \delta)$ confidence intervals for all values of $M_k(\sigma)$, and determines the ordering between σ and τ is correct when they do not overlap. A strategy profile is chosen to be sampled until all of its ordering are correctly determined. When all orderings are correctly determined the algorithm terminates.

Since the confidence intervals are constructed using frequentist concentration inequalities, we refer to RG-UCB as being a frequentist algorithm. In contrast, our Bayesian perspective provides a principled method for incorporating prior knowledge into our algorithm whereas it is much more difficult to encode modelling assumptions and prior knowledge with RG-UCB. The second important difference between our work and RG-UCB is that our information gain criterion is a direct objective, which selects the payoffs to sample based on how likely the received sample is to affect the α-rank. On the other hand, RG-UCB works indirectly, reducing uncertainty about the orderings between individual payoffs without considering their impact on the final α-rank, which makes it less efficient. Rowland et al. (2019) also theoretically justify the use of RG-UCB in the infinite-α regime by proving sample complexity results, whereas we provide asymptotic regret bounds for our approach which are commonly used to justify the sample efficiency of a Bayesian algorithm (Srinivas et al. 2009). Rowland et al. (2019) additionally provide a method for obtaining uncertainty estimates in the infinite-α regime, which is, however, not used as part of an adaptive sampling strategy.

4 Method

On a high level, our method works by maintaining an epistemic belief over α-ranks and selecting payoffs that lead to the maximum reduction in the entropy of that belief. Figure 1 provides a pictorial overview. In the middle of the figure, we maintain an explicit distribution over the entries of
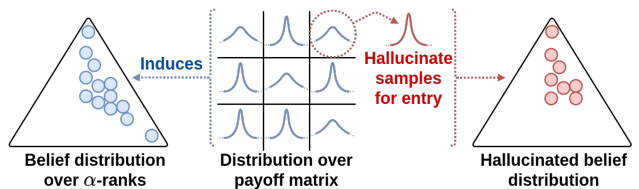

Figure 1: On the left, a belief over α-ranks is induced by a belief over the payoff matrix shown in the middle. A hallucinated belief distribution is shown on the right.

the payoff matrix. This payoff distribution induces a belief over α-ranks, shown on the left. When deciding which payoff to sample, we examine hypothetical belief states after sampling, striving to end up with a belief with the lowest entropy. One such hypothetical, or 'hallucinated' belief is shown on the right. We now describe our method formally, first describing the probabilistic model and then the implementation.

Payoffs: Ground Truth and Belief We denote the unknown true payoff matrix as $M_\star$. To quantify our uncertainty about what this true payoff is, we employ a Gaussian Process M, which also allows us to encode prior knowledge about payoff dependencies. Our framework is sufficiently general to allow for other approaches such as Bayesian Matrix Factorization (Salakhutdinov and Mnih 2008) or probabilistic methods for Collaborate Filtering (Su and Khoshgoftaar 2009) to be used. We choose to use Gaussian Processes due to their flexibility in encoding prior knowledge and modelling assumptions, and their ubiquity throughout literature.

The GP models noise in the payoffs as $\tilde{M} = M + \epsilon$, where $\epsilon \sim \mathcal{N}(0, I\sigma_A^2)$. When interacting with the game sequentially, the received payoffs are assumed to be generated as $m_t = M_\star(a_t) + \epsilon'_t$. Here, ϵ'_t are i.i.d. random variables with support on the interval $[-\sigma_A, \sigma_A]$. While it may at first seem surprising that we use Gaussian observation noise in the GP model, while assuming a truncated observation noise for the actual observation, this does not in fact affect our theoretical guarantees. We provide more details in Section 6. We denote the history of interactions at time t by H_t. Because of randomness in the observations, H_t is a random variable. The sequence of random variables $H_1, H_2, \ldots$ forms a filtration. We use the symbol h_t to denote particular realization of history so that $h_t = a_1, m_1, \ldots, a_{t-1}, m_{t-1}$.

Belief over α-ranks Our explicit distribution over the entries of the payoff matrix M induces an implicit belief distribution over the α-ranks. For all valid α-ranks r, $P(r) = P(M \in f^{-1}(r))$ where f^{-1} denotes the pre-image of r under f. In other words, the probability assigned to an α-rank r is the probability assigned to its pre-image by our belief over the payoffs. Since r is represented implicitly, we cannot query its mass function directly. Instead, we access r via sampling. This is done by first drawing a payoff from $m \sim M$ and then computing the resulting α-rank $f(m)$.

Picking Payoffs to Query At time t, we query the payoff that provides us with the largest information gain about the α-rank. Formally,

$$a_t = \arg\max_a \mathbb{I}(r\,;\,(\tilde{M}_t(a), a) \mid H_t = h_t)$$

$$= \arg\max_a \mathbb{H}\left(r \mid H_t = h_t\right)$$

$$- \mathop{\mathbb{E}}_{\tilde{m}_t \sim \tilde{M}_t(a)} \left[\mathbb{H}\left(r \mid H_t = h_t, A_t = a, \tilde{M}_t(a) = \tilde{m}_t\right) \right] \quad (1)$$

$$= \arg\min_a \mathop{\mathbb{E}}_{\tilde{m}_t \sim \tilde{M}_t(a)} \left[\mathbb{H}\left(r \mid H_t = h_t, A_t = a, \tilde{M}_t(a) = \tilde{m}_t\right) \right]$$
$$\quad (2)$$

In Equation (1), $\mathbb{H}\left(r \mid H_t = h_t\right)$ is the entropy of our current belief distribution over α-ranks, which does not depend on a and can be dropped from the maximization, producing Equation (2). The expectation in (2) has an intuitive interpretation as the expected negative entropy of our *hallucinated* belief, i.e. belief obtained by conditioning on a sample $\tilde{m}_t$ from the current model. In essence, we are pretending to receive a sample for entry a, and then computing what our resulting belief over α-ranks will be. By picking the entry as in (2), we are picking the entry whose sample will lead to the largest reduction in the entropy of our belief over α-ranks in expectation.

Algorithm 1 αIG algorithm. αIG(NSB) and αIG(Bin) variants differ in entropy estimator (Line 7).

1: **for** $t = 1, 2, \ldots T$ **do**
2:     **for** $a = 1, 2, \ldots |\mathcal{S}|$ **do**
3:         **for** $i = 1, 2, \ldots N_e$ **do**
4:             $\tilde{m}_t \sim \tilde{M}_t(a)$ ▷ 'Hallucinate' a payoff.
5:             Obtain hallucinated posterior payoff:

$$P(\hat{M}_t | H_t = h_t, A_t = a, \tilde{M}_t(a) = \tilde{m}_t)$$

6:             $D = \{r_1, \ldots, r_{N_b}\}$, where $r_i \sim f(\hat{M}_t)$ i.i.d.
7:             $\hat{h}_a^i = $ ESTIMATE-ENTROPY(D)
8:         **end for**
9:         $\hat{h}_a = \frac{1}{N_e} \sum_{i=1}^{N_e} \hat{h}_a^i$
10:     **end for**
11:     Query payoff $a_t = \arg\min_a \hat{h}_a$ ▷ Implements Eq. (2).
12: **end for**

Implementation Our algorithm, which we refer to as αIG, is summarized in Algorithm 1. At a high-level, αIG selects an action/payoff to query at each timestep (Line 1). In order to select a payoff to query as in Equation (2), we must approximate the expectation for each payoff (Line 2). In Line 4, we use our epistemic model to obtain a 'hallucinated' outcome $\tilde{m}_t$, as if we received a sample from selecting payoff a at timestep t. In Line 5, we condition our epistemic model on this 'hallucinated' sample $\tilde{m}_t$ in order to obtain our 'hallucinated' posterior over payoffs $\hat{M}_t$. In Line 7, we empirically estimate the entropy of the resulting induced belief distribution over α-ranks. To approximate the expectation in (2), we average out entropy estimates obtained from N_e different possible hallucinated payoffs in Line 9. Finally, in Line

11, we use these estimates to perform query selection as in (2) to select a payoff to query at timestep t.

Our algorithm depends on an entropy estimator ESTIMATE-ENTROPY, used in Line 7. We present results for 2 different entropy estimators: simple binning and NSB. The simple binning estimator estimates the entropy using a histogram. For comparison, we also used NSB (Nemenman, Shafee, and Bialek 2002), an entropy estimator designed to produce better estimates in the small-data regime.

Computational Requirements The main computational bottleneck of our algorithm is the calculations of α-rank in Line 6 of Algorithm 1. In order to perform query selection as in (2), we must compute the α-rank $|\mathcal{S}| \times N_e \times N_b$ times. For our experiments on the 4x4 Gaussian game this results in $16 \times 10 \times 500 = 80,000$ computations of α-rank (setting $N_e = 10, N_b = 500$), to select a payoff to query. Relative to ResponseGraphUCB, our method thus requires significantly more computation in order to select a payoff to query. However, in Empirical Game Theory, it is commonly assumed that obtaining samples from the game is very computationally expensive (which is true in many potential practical applications (Berner et al. 2019; Silver et al. 2017; Vinyals et al. 2019)). The increased computation required by our method to select a payoff to sample should then have a negligible impact to the overall computation time required, but the increased sample efficiency could potentially lead to large speed-ups.

We perform two simple optimizations when deploying the algorithm in practice. To save computational cost, we observe the same payoff N_r times in Line 11 rather than once, similar to rarely-switching bandits (Abbasi-yadkori, Pál, and Szepesvári 2011). Moreover, the number of samples N_b we can use to estimate the entropy is limited due to the computational cost of computing α-rank. In order to obtain better differentiation between the entropy of beliefs arising from sampling different payoffs, we heuristically perform conditioning in Line 5 N_c times. See Appendix for a more detailed discussion on this.

5 Query Selection by Maximizing Wasserstein Divergence

While the query objective proposed in (2) is backed both by an appealing intuition and a theoretical argument (see Section 6), it can be expensive to evaluate due to the cost of accurate entropy estimation. To address this difficulty, we also investigate an alternative involving the Wasserstein distance. The objective we consider is

$$\arg\max_a \mathbb{E}_{\tilde{m}_t \sim \tilde{M}_t}[\mathcal{W}(P(r|H_t = h_t),$$

$$P(r|H_t = h_t, A_t = a, \tilde{M}_t(a) = \tilde{m}_t))]. \quad (3)$$

Since the computation of Wasserstein distance from empirical distributions can be achieved by solving a linear program (Bonneel et al. 2011), Equation (3) naturally lends itself to being approximated via samples. In our implementation, we use POT (Flamary and Courty 2017) to approximate this distance.
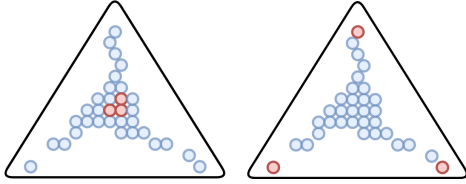
Figure 2: Diagram depicting the current belief (Blue) and 2 different hallucinated beliefs (Red). We are assuming a discrete distribution over α-ranks, where the belief is uniform across the relevant circles.

The Wasserstein distance is built on the notion of cost, which allows a practitioner the opportunity to supply additional prior knowledge. In our case, since α-ranks are probability distributions, a natural way to measure accuracy is to use the total variation distance, which corresponds to setting the cost to $c(x, y) = \frac{1}{2}\|x - y\|_1$. On the other hand, in cases where we are interested in finding the relative ordering of agents under the α-rank, an alternative cost such as the Kendall Tau metric (Fagin et al. 2006) could be used. While we emphasize the ability of the Wasserstein divergence to work with any cost, we leave the empirical study of non-standard costs for future work.

It is important to note that the objective in (3) is qualitatively different to the information gain objective proposed in (2). Figure 2 provides a diagram illustrating a major difference between the two objectives. The entropy for both belief distributions shown in red is the same. In contrast, the Wasserstein distance in (3) between the current belief in blue and the hallucinated belief in red is much smaller for the distribution on the left compared to the distribution on the right.

6 Theoretical Results

Notions of Regret We quantify the performance of our method by measuring regret. Our main analysis relies on Bayesian regret (Russo and van Roy 2018), defined as

$$J_t^B = 1 - \mathbb{E}_{h_t}\left[P(r = r_\star | H_t = h_t)\right], \tag{4}$$

where we used $r_\star$ to denote the α-rank with the highest probability under r at time t. In (4), the expectation is over realizations of the observation model. Since J_t^B, like all purely Bayesian notions, does not involve the ground truth payoff, we need to justify its practical relevance. We do this by benchmarking it against two notions of frequentist regret. The first measures how accurate the probability we assign to the ground truth $r_{GT} = f(M_\star)$ is

$$J_t^F = 1 - \mathbb{E}_{h_t}\left[P(r = r_{GT} | H_t = h_t)\right]. \tag{5}$$

The second measures if the mean of our payoff belief, which we denote M_μ, evaluates to the correct α-rank

$$J_t^M = 1 - \mathbb{E}_{h_t}\left[\delta\Big[f(M_\mu) = r_{GT}\Big]\right], \tag{6}$$

where the symbol $\delta\Big[\text{predicate}\Big]$ evaluates to 1 or 0 depending on whether the predicate is true or false. In Section 7, we empirically conclude that these three notions of regret are closely coupled in practice, changing at a comparable rate.

Regret Bounds As an intermediate step before discussing information gain on the α-ranks, we first analyze the behavior of a query selection rule which maximizes information gain over the payoffs.

$$\pi_{\text{IGM}}(a | H_t = h_t) = \arg\max_a \mathbb{I}(\tilde{M}_t \,;\, (\tilde{M}_t(a), a) \mid H_t = h_t). \tag{7}$$

The following result shows that using sampling strategy π_{IGM} for T timesteps leads to a decay in regret of at least $Te^{\mathcal{O}(-\sqrt[3]{\Delta^2 T})}$, proving it will incur no regret as $T \to \infty$.

Proposition 1 (Regret Bound For Information Gain on Payoffs). *If we select actions using strategy π_{IGM}, the regret at timestep T is bounded as*

$$J_T^B \le J_T^F = 1 - \mathbb{E}_{h_T}\left[P(r = r_{GT} | H_T = h_T)\right] \le Te^{g(T)} \tag{8}$$
$$\text{where } g(T) = \mathcal{O}(-\sqrt[3]{\Delta^2 T}).$$

The proof, and an explicit form of g are found in supplementary material. We now proceed to our second result, where we maximize information gain on the α-ranks directly. Consider a querying strategy that is an extension of (1) to T-step look-ahead, defined as

$$\pi_{\text{IGR}} = \arg\max_{a_1,\dots,a_T} \mathbb{I}(r \,;\, (\tilde{M}_1(a_1), a_1), \dots, (\tilde{M}_T(a_T), a_T)). \tag{9}$$

We quantify regret achieved by π_{IGR} in the proposition below.

Proposition 2 (Regret Bound For Information Gain on Belief over α-Ranks). *If we select actions using strategy π_{IGR}, regret is bounded as*

$$J_T^B = 1 - P(r = r_\star | H_T = h_T) \to 0 \;\; as \;\; T \to \infty.$$

Proposition 2 provides a theoretical justification for querying the strategies that maximize information gain on the α-ranks. A more explicit regret bound (similar to Proposition 1) and the proof are provided in Appendix. In practice, to avoid the combinatorial expense of selecting action sequences using π_{IGR}, we use the greedy query selection strategy in equation (1). While the regret result above does not carry over, this idealized setting at least provides some justification for information gain as a query selection criterion.

7 Experiments

In this section, we describe our results on synthetic games, graphing the Bayesian regret J_t^B described in Section 6. We also justify the use of Bayesian regret, showing that it is highly coupled with the ground truth payoff. We benchmark two versions of our algorithms, αIG (Bins) and αIG (NSB), which differ in the employed entropy estimator. We compare to three baselines: **RG-UCB**, a frequentist bandit algorithm (Rowland et al. 2019), **Payoff**, which maximizes the information gain about the payoff distribution, and **Uniform**, which selects payoffs uniformly at random. **RG-UCB** represents the current SOTA in this domain, **Payoff** represents the performance of a Bayesian method that does not take into account the structure of the mapping between payoffs and α-ranks, and **Uniform** provides a point of reference
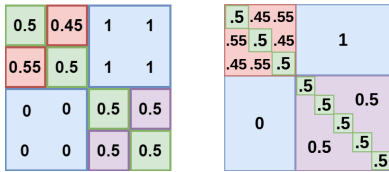
Figure 3: Payoff matrices for 2 Good, 2 Bad (Left) and 3 Good, 5 Bad (Right). Best viewed in color.

as the simplest/most naive method[1]. A detailed explanation of the experimental setup[2] and details on the used hyperparameters are included in Appendix.

Good-Bad Games To investigate our algorithm, we study two environments whose payoffs are shown in Figure 3. We start with the relatively simple environment with 4 agents. Figure 3 (Left) shows the expected payoffs, which we can interpret as the win-rate. Samples are drawn from a Bernoulli distribution with the appropriate mean. We refer to the environment as '2 Good, 2 Bad' since agents 1 and 2 are much stronger than the other 2 agents, winning 100% of the games against them. Since the ordering between agents 3 and 4 has no effect on the α-rank, gathering samples to determine this ordering (highlighted in Purple) does not affect the belief distribution over α-ranks. Furthermore, since we treat this as a 1-population game, the entries highlighted in Green where each agent plays against themselves do not affect the α-rank. Entries that are necessary to determine the ordering between agents 1 and 2 are the most relevant for the α-rank and are highlighted in Red. Since agent 2 is slightly better than agent 1, the true α-rank is $(0, 1, 0, 0)$. However, it can be difficult to determine the correct ordering between agents 1 and 2 without drawing many samples from these entries. The game thus provides a model for the common scenario of agents with clustered performance ratings.

Focusing on Relevant Payoffs Figure 4 presents the behavior of our method and RG-UCB on this task. As expected, RG-UCB splits its sampling between the Red entries and the Purple entries, whereas our method concentrates its sampling much more significantly on the relevant entries, determining the ordering between agents 1 and 2. This is because, in contrast to our method, RG-UCB aims to correctly determine the ordering between **all** entries used in the calculating of α-rank, irrespective of whether they matter for the final outcome.

Wasserstein Payoff Selection Does Well Comparing the Wasserstein Criterion with Information Gain payoff section, we can see that it enjoys better concentration of the sampling on the Red entries, and improved performance towards the end of training. Appendix provides a more detailed analysis of this.

---

[1] We do not include Uniform on the regret graphs, since there is no reasonable value we could compute for it.

[2] Code is available at github.com/microsoft/InfoGainalpharank.

Bayesian and Frequentist Regret Go Down Figure 5 shows the resulting performance of the methods on this task, measured by the regret. Due to the relative simplicity of the game, there is limited benefit to our method over RG-UCB, but there is a clear benefit over more naive methods that systematically or uniformly sample the entries. We can see that the Bayesian regret J_t^B and Frequentist regrets J_t^F and J_t^M are highly correlated, providing empirical justification for minimizing J_t^B and validating that our method is concentrating on the ground truth.

Comparing Entropy Estimators We also investigate a larger scale version of 2 Good, 2 Bad with 3 good and 5 bad agents. Figure 6 shows the results, demonstrating a clear benefit for our method using the Binning estimator for the Information Gain or the Wasserstein objective. The performance of the NSB entropy estimator is not surprising given the significantly larger nature of this task compared to '2 Good, 2 Bad'. A necessary part of the NSB estimator is an upper-bound on the total number of atoms in the distributions, for which we only have a crude approximation that grows exponentially with the size of the payoff matrix. Figure 7 shows the proportion of entries sampled for αIG (Bins), the Wasserstein objective, and RG-UCB. Once again, RG-UCB spends a significant part of its sampling budget determining the ordering between agents that do not have an effect on the α-rank of the game (in this task agents 3 to 8). In contrast, our methods concentrate their sampling on the Red entries that determine the payoffs between the top 3 agents, and hence the true α-rank. In general, our algorithm does not depend as much on accurate estimates on entropy but on identifying the distribution with a lowest entropy, for which the NSB estimator isn't tuned.

Incorporating Prior Knowledge A large benefit of our Bayesian-based approach is the ability to incorporate prior knowledge and modelling assumptions into our model in a principled manner. To demonstrate the benefits, we incorporate the following prior knowledge into both our algorithm and RG-UCB: 1) $M(\sigma, \tau) + M(\tau, \sigma) = 1$. 2) Entries in their respective blocks are equal to each other (except for the top left block). A detailed description of the setup is included in Appendix. Figure 8 compares the performance of αIG, αWass, and RG-UCB on 3 Good, 5 Bad when utilizing this prior knowledge. We can see that our approach significantly outperforms RG-UCB on this task, further demonstrating the importance of our direct information gain objective. The results also show significantly improved sample efficiency over the results in Figure 6, demonstrating that our αIG and αWass are able to efficiently take advantage of the prior knowledge supplied.

8   Conclusions

We described αIG, an algorithm for estimating the α-rank of a game using a small number of payoff evaluations. αIG works by maximizing information gain. It achieves competitive sample efficiency and allows a way of building in prior knowledge about the payoffs.
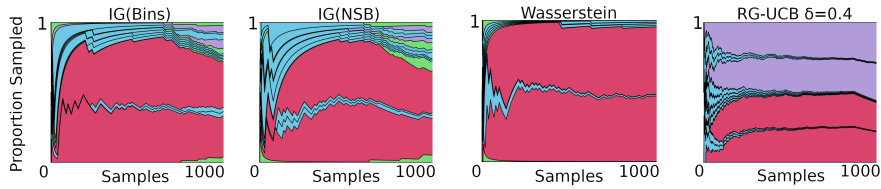
Figure 4: Proportion of entries sampled on 2 Good, 2 Bad for different methods and objectives.
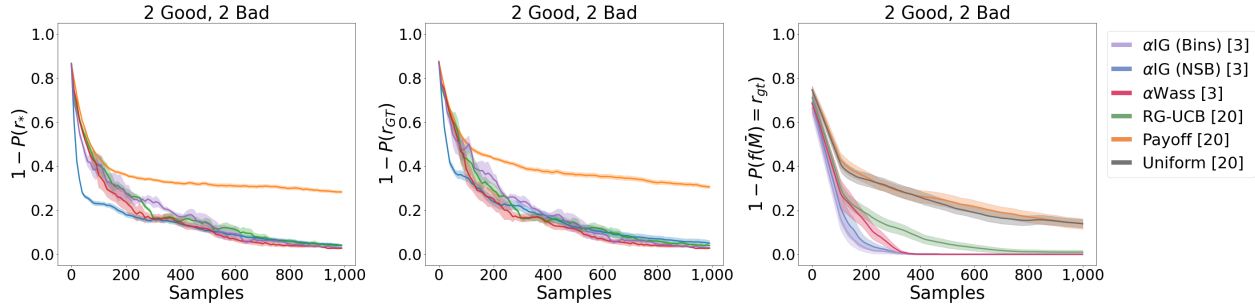

Figure 5: Results for 2 Good, 2 Bad. Graphs show the mean and standard error of the mean over multiple runs (shown in brackets) of 10 repeats each.
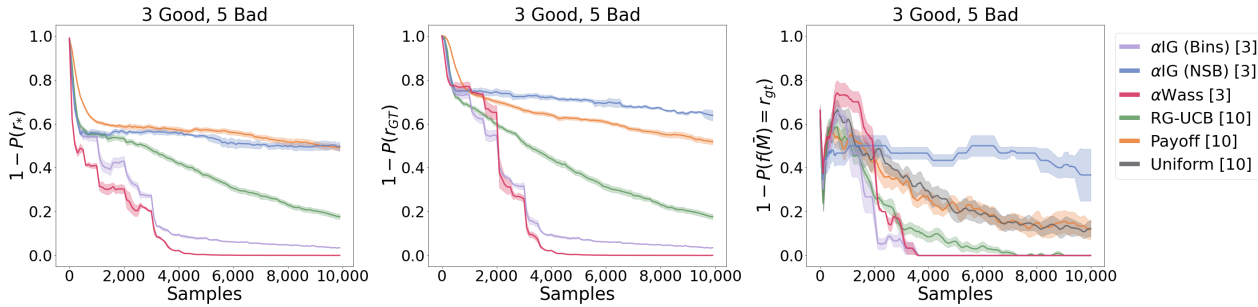

Figure 6: Results for 3 Good, 3 Bad. Graphs show the mean and standard error of the mean over multiple runs (shown in brackets) of 10 repeats each.
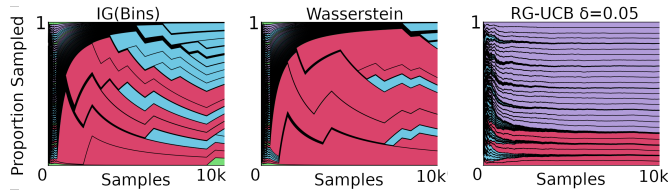

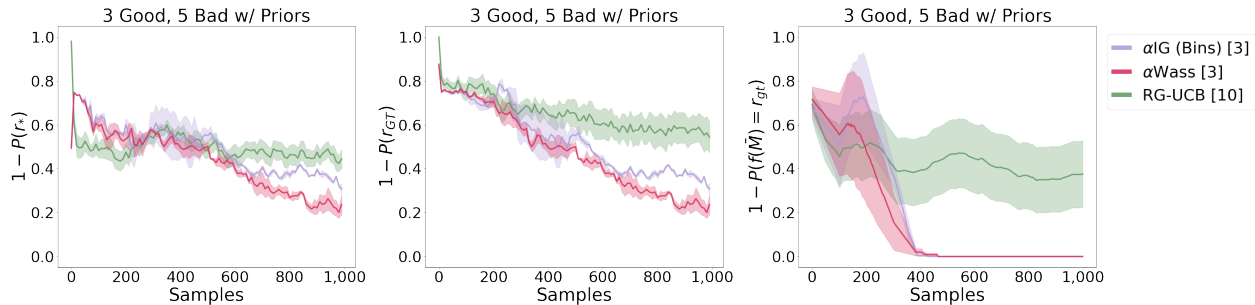Figure 7: Proportion of entries sampled on 3 Good, 5 Bad.

Figure 8: Results on 3 Good, 5 Bad when incorporating prior knowledge into the models.

5679

Acknowledgements

References

Abbasi-yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved Algorithms for Linear Stochastic Bandits. In Shawe-Taylor, J.; Zemel, R. S.; Bartlett, P. L.; Pereira, F.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 24*, 2312–2320. Curran Associates, Inc. URL http://papers.nips.cc/paper/4417-improved-algorithms-for-linear-stochastic-bandits.pdf.

Balduzzi, D.; Tuyls, K.; Perolat, J.; and Graepel, T. 2018. Re-evaluating evaluation. In *Advances in Neural Information Processing Systems*, 3268–3279.

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Debiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv preprint arXiv:1912.06680* .

Bonneel, N.; Van De Panne, M.; Paris, S.; and Heidrich, W. 2011. Displacement interpolation using Lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, 1–12.

Chen, S.; and Joachims, T. 2016. Modeling intransitivity in matchup and comparison data. In *Proceedings of the ninth acm international conference on web search and data mining*, 227–236.

Daskalakis, C.; Goldberg, P. W.; and Papadimitriou, C. H. 2009. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing* 39(1): 195–259.

Elo, A. E. 1978. *The rating of chessplayers, past and present*. Arco Pub.

Fagin, R.; Kumar, R.; Mahdian, M.; Sivakumar, D.; and Vee, E. 2006. Comparing partial rankings. *SIAM Journal on Discrete Mathematics* 20(3): 628–648.

Flamary, R.; and Courty, N. 2017. POT Python Optimal Transport library. URL https://pythonot.github.io/. PyPI version 0.7.0.

Fudenberg, D.; and Tirole, J. 1991. Game theory, 1991. *Cambridge, Massachusetts* 393(12): 80.

Herbrich, R.; Minka, T.; and Graepel, T. 2007. TrueSkill™: a Bayesian skill rating system. In *Advances in neural information processing systems*, 569–576.

Jordan, P. R.; Vorobeychik, Y.; and Wellman, M. P. 2008. Searching for approximate equilibria in empirical games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, 1063–1070.

Lanctot, M.; Zambaldi, V.; Gruslys, A.; Lazaridou, A.; Tuyls, K.; Pérolat, J.; Silver, D.; and Graepel, T. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, 4190–4203.

Minka, T.; Cleven, R.; and Zaykov, Y. 2018. Trueskill 2: An improved bayesian skill rating system. *Tech. Rep.* .

Muller, P.; Omidshafiei, S.; Rowland, M.; Tuyls, K.; Perolat, J.; Liu, S.; Hennes, D.; Marris, L.; Lanctot, M.; Hughes, E.; Wang, Z.; Lever, G.; Heess, N.; Graepel, T.; and Munos, R. 2020. A Generalized Training Approach for Multi-agent Learning. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=Bkl5kxrKDr.

Nash, J. 1951. Non-cooperative games. *Annals of mathematics* 286–295.

Nemenman, I.; Shafee, F.; and Bialek, W. 2002. Entropy and inference, revisited. In *Advances in neural information processing systems*, 471–478.

Omidshafiei, S.; Papadimitriou, C.; Piliouras, G.; Tuyls, K.; Rowland, M.; Lespiau, J.-B.; Czarnecki, W. M.; Lanctot, M.; Perolat, J.; and Munos, R. 2019. α-rank: Multi-agent evaluation by evolution. *Scientific reports* 9(1): 1–29.

Rowland, M.; Omidshafiei, S.; Tuyls, K.; Perolat, J.; Valko, M.; Piliouras, G.; and Munos, R. 2019. Multiagent Evaluation under Incomplete Information. In *Advances in Neural Information Processing Systems*, 12270–12282.

Russo, D.; and van Roy, B. 2018. Learning to optimize via information-directed sampling. *Operations Research* 66(1): 230–252.

Salakhutdinov, R.; and Mnih, A. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning*, 880–887.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815* .

Srinivas, N.; Krause, A.; Kakade, S. M.; and Seeger, M. 2009. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995* .

Su, X.; and Khoshgoftaar, T. M. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009.

Tuyls, K.; Perolat, J.; Lanctot, M.; Hughes, E.; Everett, R.; Leibo, J. Z.; Szepesvári, C.; and Graepel, T. 2020. Bounds and dynamics for empirical game theoretic analysis. *Autonomous Agents and Multi-Agent Systems* 34(1): 7.

Villani, C. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575(7782): 350–354.

Walsh, W. E.; Parkes, D. C.; and Das, R. 2003. Choosing samples to compute heuristic-strategy Nash equilibrium. In *International Workshop on Agent-Mediated Electronic Commerce*, 109–123. Springer.

Wellman, M. P. 2006. Methods for empirical game-theoretic analysis. In *AAAI*, 1552–1556.

Yang, Y.; Tutunov, R.; Sakulwongtana, P.; and Ammar, H. B. 2019. α α-Rank: Practically Scaling α-Rank through Stochastic Optimisation. *arXiv preprint arXiv:1909.11628* .