Evaluating State-Space Abstractions in Extensive-Form Games

Michael Johanson and Neil Burch and Richard Valenzano and Michael Bowling
University of Alberta
Edmonton, Alberta
{johanson,nburch,valenzan,mbowling}@ualberta.ca

ABSTRACT

Efficient algorithms exist for finding optimal policies in extensive-form games. However, human-scale problems are typically so large that this computation remains infeasible with modern computing resources. State-space abstraction techniques allow for the derivation of a smaller and strategically similar abstract domain, in which an optimal strategy can be computed and then used as a suboptimal strategy in the real domain. In this paper, we consider the task of evaluating the quality of an abstraction, independent of a specific abstract strategy. In particular, we use a recent metric for abstraction quality and examine imperfect recall abstractions, in which agents "forget" previously observed information to focus the abstraction effort on more recent and relevant state information. We present experimental results in the domain of Texas hold'em poker that validate the use of distribution-aware abstractions over expectation-based approaches, demonstrate that the new metric better predicts tournament performance, and show that abstractions built using imperfect recall outperform those built using perfect recall in terms of both exploitability and one-on-one play.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems—*Games*

General Terms

Algorithms

Keywords

Economic paradigms::Game theory (cooperative and non-cooperative); Learning and Adaptation::Multiagent Learning

1. INTRODUCTION

Realistic multiagent settings involve complex, sequential interactions between agents with different perspectives regarding the underlying state of the world. A general model for such settings is the extensive-form game with imperfect information. While state-of-the-art techniques for approximating Nash equilibria in extensive-form games [21, 8] have made remarkable progress [15, 11], the size of most real-world settings is beyond the capability of current solvers. For example, a common benchmark of progress is the domain of computer poker. Current solution techniques have found approximate equilibria in poker-like games with as many as 88 billion decision points [9], which is still four orders of magnitude smaller than the smallest poker game played by humans. The ubiquitous approach to handling such human-scale domains is *abstraction* [2, 16, 5], where strategically similar decision points for the players are grouped to construct an abstract game that is tractably sized for current solution techniques. The solution of the abstract game is then employed in the original game.

While even simple abstraction techniques have been found to be empirically effective [21], their success is not guaranteed. Waugh *et al.* [18] gave surprising examples of **abstraction pathologies** where strict refinements of abstractions can result in abstract strategy equilibria that are more exploitable in the real game. While there is little theory to guide the construction of abstractions, Gilpin and Sandholm [6] presented three methods for empirically comparing abstraction methodologies: one-on-one comparison, versus-equilibrium comparison, and versus-best-response comparison. While these remained the best-practice approach for abstraction evaluation, each of these methods has conceptual drawbacks: possible intransitivities, infeasible computation, and not being well correlated with actual performance (respectively). Johanson *et al.* [10] recently presented the **CFR-BR** algorithm, which computes the best Nash approximation strategy that can be represented in a given abstraction. This represents a new, fourth method for evaluating abstraction methodologies: comparing the representation power of an abstraction by how well it can approximate an unabstracted Nash equilibrium. In this paper, we will examine the efficacy of this new approach for evaluating abstractions, and use it to evaluate several abstraction methodologies in the poker domain. We show that not only does it have many desirable conceptual properties (e.g., transitivity and computational tractability), it is also empirically well-correlated with the in-game performance of abstract game equilibria. We demonstrate all of this through a series of abstraction evaluation experiments. In particular, we repeat the Gilpin and Sandholm experiments that concluded that expectation-based abstractions are weaker than distribution-aware abstractions[1]. We also use this technique

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

[1] Gilpin and Sandholm refer to their abstraction technique as being *potential-aware*, as it is distribution-aware and can also represent how quickly a hand may change over time.

to validate the efficacy of **imperfect recall** abstractions, in which an agent forgets information known in past decisions to refine its representation of its current state. Such abstractions are empirically effective [19, 12], but previous research has not shown a conclusive advantage. Finally, we present for the first time the abstraction methodology employed by Hyperborean, one of the top competitors in the Annual Computer Poker Competition.

2. BACKGROUND

Extensive-form games. Extensive-form games are an intuitive formalism for representing the interaction between agents and their environment. These interactions are represented by a tree, in which nodes represent game states and edges represent actions taken by one of the agents, $i \in N$, or **chance**, c. The root of the tree represents the start of the interaction, and actions are taken until a leaf, i.e., **terminal node** is reached. Each terminal node $z \in Z$ assigns a utility to each player i, $u_i(z)$. In **imperfect information** games, agents may not be able to observe some of the actions taken by chance or the other agents. In the poker setting we use the terms **private** and **public** to refer to actions visible to only one agent or to all agents, although in general other types of actions are possible. Each set of game states that are indistinguishable by the acting agent is called an **information set**. When some actions are not observed, an agent perceives the game not as a tree of game states, but as a tree of information sets. A **perfect recall** game has the natural property that each agent remembers the exact sequence of its past observations and actions leading to each decision.

A **behavioral strategy** (or simply a **strategy**) for each player i, σ_i, maps each of player i's information sets to a probability distribution over the legal actions. A **strategy profile** σ is a tuple containing a strategy for each player. Let σ_{-i} refer to the strategies of player i's opponents. Given a strategy profile σ, we denote each player's **expected utility** by $u_i(\sigma)$. Given the opponents' strategies σ_{-i}, a **best response** for player i is the strategy that maximizes utility against σ_{-i}, where $b_i(\sigma_{-i})$ is the utility of the best response strategy when played against σ_{-i}. A strategy profile σ is called an **ϵ-Nash equilibrium** if $\forall i \in N, b_i(\sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \leq \epsilon$. When $\epsilon = 0$, the profile is called a **Nash equilibrium**. In two-player repeated games where the agents alternate positions, each agent has one strategy for each position and their **exploitability** is their utility (averaged over all positions) against a worst-case adversary who, in each position, uses a best-response to the agent. In two-player zero-sum games, a Nash equilibrium has an exploitability of 0 and thus cannot lose, on expectation, to any adversary.

Poker is a canonical example of stochastic imperfect information extensive-form games. In this paper we will focus on two-player limit Texas hold'em, which is one of the variants played in the Annual Computer Poker Competition. The game begins with each player being given a **hand** of two **private cards** that only they can see or use. The players' actions are to **bet** or **call**, placing or matching wagers that their hand will be the strongest at the end of the game, or to **fold** to concede the game. This is followed by chance revealing an additional three **public cards** that both players can see and use, and an additional round of betting actions. After two additional such rounds in which one public card is revealed and the players act, the game is over and the player with the strongest hand made of their private cards and the public cards wins the wagers. Poker is a repeated game in which two agents will play a long series of such games with the overall goal of having the highest total winnings.

Counterfactual Regret Minimization. Counterfactual Regret Minimization (CFR) is a state-of-the-art algorithm for **solving** extensive-form games (*i.e.*, approximating a Nash equilibrium strategy) and has been widely used in the poker domain [21, 9]. Although it is only proven to converge to a Nash equilibrium in two-player zero-sum perfect recall games, in practice it appears robust when these constraints are violated as it has been successfully applied to multi-player games [14], non-zero-sum games [12], and imperfect recall games [19]. CFR is an iterative self-play algorithm. Each player starts with an arbitrary strategy. On each iteration, the players examine every decision, and for each possible action compare the observed value of their current policy to the value they could have achieved by making that action instead. This difference is the **regret** for playing an action, and the accumulated regret is used to determine the strategy used on the next iteration. In the limit, the average strategies used by the players will converge to a Nash equilibrium. CFR is efficient in both time and memory, requiring space which is linear in the number of actions across all information sets. While it has been applied to games with up to 8.8×10^{10} information sets [9], the computation remains intractable for domains as large as two-player limit Texas hold'em, which has 3.2×10^{14} information sets.

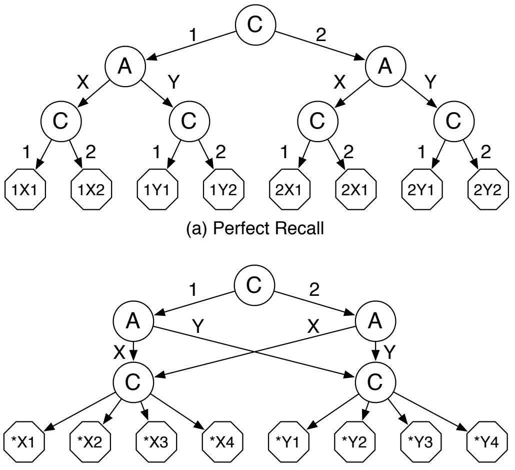
State-space abstraction. A **state space abstraction** is a many-to-one mapping between the game's information sets and the information sets in a smaller, artificially constructed abstract game. An agent using abstraction only observes its abstract game information set, and its strategy for that information set is used for all of the real information sets mapped to it. The goal is to construct a game small enough that an optimal strategy can be found through an algorithm such as CFR, and the resulting strategy can be used to choose actions in the original game, where it is hoped to closely approximate a Nash equilibrium strategy. The success of this approach relies on both the size of the abstract game (a larger and finer-grained abstract game can lose less information) and the domain features used to decide which information sets can be mapped together.

The earliest uses of state-space abstraction in poker involved the construction of abstract chance events, called **bins** by Shi and Littman [16], **buckets** by Billings *et al.* [2], and **signals** by Gilpin and Sandholm [5], by grouping together chance events that are similar according to a metric. As the players' actions were left unabstracted, the abstract game resembles the real game except with a coarsened representation of the chance events. A common metric used in this early work is a player's **expected hand strength ($E[HS]$)**. In the final round when all public cards have been revealed, a player's **hand strength (HS)** is the probability that their hand is stronger than a uniform randomly sampled opponent hand. In the earlier rounds, expected hand strength ($E[HS]$) is the expectation of hand strength over all possible rollouts of the remaining public cards. A related metric, **expected hand strength squared ($E[HS^2]$)**, computes the expectation of the squared hand strength values, and assigns a relatively higher value to hands with the potential to improve such as flush-draws or straight-draws.

These **expectation-based** metrics can be used to create abstract chance events in a number of different ways, such as bucketing based on expert-chosen ranges of $E[HS]$ values [2], bucketing based on $E[HS]$ ranges chosen so as to contain an equal number of hands (called **percentile bucketing**) [21], or by merging hands whose $E[HS]$ values differ by less than a threshold [4]. Additionally, two abstraction techniques can be **nested** by applying one and then subdividing by the other. For example, an abstraction might split the possible hands into five buckets by percentile $E[HS^2]$ and further split each into two percentile $E[HS]$ buckets, giving ten buckets overall. The Percentile nested $E[HS^2]$ / $E[HS]$ abstraction technique has been well studied by researchers [12, 19] and was used by Hyperborean in the Annual Computer Poker Competitions from 2007 to 2009.

Gilpin *et al.* showed that expectation-based metrics have difficulty distinguishing between hands that have the potential to improve and those that do not, and that this difference is strategically important [7]. High potential hands are called **drawing hands**, which might be weak initially but have the possibility to become very strong given fortunate chance outcomes later in the game. Expectation-based abstraction techniques place these hands into buckets along with hands that have a similar $E[HS]$ values and no likelihood of improving. Abstracting these strategically distinct hands together loses information, as an abstracted agent must choose one strategy to handle both cases. While the $E[HS^2]$ metric was designed to address this fault, it was only a partial solution. Gilpin *et al.* addressed this shortcoming through a multi-pass k-means abstraction technique in which the final round is clustered by $E[HS]$ and each earlier round was clustered by L_2 distance over histograms showing the probability of transitioning to the next round's buckets [7]. In later work, Gilpin and Sandholm compared these **distribution-aware** abstractions to expectation-based abstractions and found that expectation-based abstractions yielded stronger strategies in small abstractions, but are surpassed as more buckets are made available [6].

Imperfect Recall. **Imperfect recall** is a relaxation of perfect recall in which agents may "forget" some of the information that it has observed. It is not typically a property of a real domain (as humans cannot be forced to forget their observations), but is instead an optional property that can be used for abstract games. When creating an imperfect recall abstraction agents can be forced to discard old observations that are no longer strategically important, thus merging the real information sets that differed in this observation. This means that an agent may be able to distinguish two information sets early in a game, but not distinguish their descendant information sets later in the game. They will perceive the game as a directed acyclic graph instead of as a tree.



(a) Perfect Recall

(b) Imperfect Recall

Figure 1: Perfect recall and imperfect recall games.

An example of equal-sized perfect recall and imperfect recall abstractions in a poker-like game is shown in Figure 1. This game starts with a chance event, C, which deals the player a private card. Each abstraction coarsens that information and maps it to a bucket, 1 or 2, indicating that the card is in the top or bottom half of all possible cards. At the action node, A, the players take a sequence of actions, X, or Y, which is followed by a chance node at which a public card is revealed. This is where the two abstractions differ. In the perfect recall abstraction, the agent must remember its sequence of observations: 1 or 2, X or Y. The new chance information is coarsened by the perfect recall abstraction, and the agent receives one of two new buckets depending on their earlier observation. The sequences 1-1, 1-2, 2-1, and 2-2 represent different sets of chance events, and can have overlapping ranges according to metrics such as $E[HS]$: a weak hand that becomes strong may score higher than a strong hand that becomes weak. In the imperfect recall abstraction, only the players' action sequence, X or Y, is remembered while the original chance bucket, 1 or 2, is forgotten. The 1X and 2X paths merge, as do 1Y and 2Y. The second chance node is coarsened to one of four buckets, 1 to 4, representing the strength of its private card combined with the public card. These four buckets can be constructed to form non-overlapping ranges of $E[HS]$. If this second chance event makes the first less significant (*i.e.* if the agent's previous strength is not very important, as is the case in poker), then the imperfect recall representation may provide more useful information.

The use of imperfect recall abstractions in the poker domain was first proposed by Waugh *et al.* [19]. As they noted, imperfect recall presents several theoretical challenges: there is no guarantee that a Nash equilibrium for an imperfect recall game can be represented as a behavioral strategy (Nash's celebrated theorem only guarantees that a mixed strategy equilibrium exists), and no proof that CFR (or other efficient algorithms) will converge towards such a strategy if one exists. Recent work by Lanctot *et al.* has shown that CFR will converge in a class of imperfect recall games; however, this class does not include the abstractions typically used in poker [13]. However, CFR remains well-defined in imperfect recall abstractions and can be used to generate abstract strategies that can be used in the real game. Waugh *et al.* [19] showed that a small improvement was possible in two-player limit Texas hold'em, as imperfect recall discarded less relevant earlier observations and allowed new domain features to be used along with $E[HS]$.

3. EVALUATING ABSTRACTIONS

With many options available for constructing abstractions and no theory to guide these choices, progress has only been established through empirical evaluation. This involves creating abstract games, solving them, and evaluating the strategy in the real game. Gilpin and Sandholm [6] codified the possibilities for evaluating the resulting strategy, and thus evaluating the abstraction methodology itself. They described three approaches: in-game performance against other agents, in-game performance against an unabstracted Nash equilibrium, and exploitability in the real game.

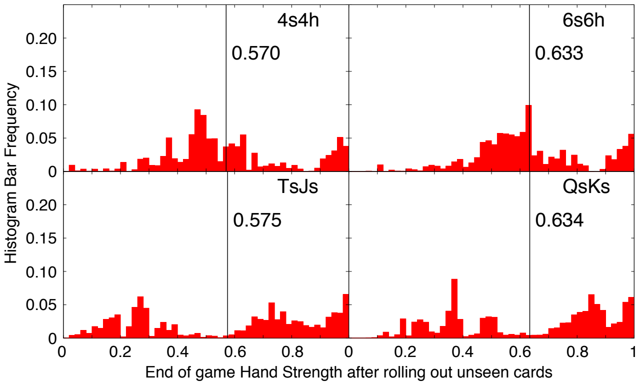
While these evaluation methods measure qualities we want, each involves a potentially serious drawback. In one-on-one play, it is possible to find intransitivities where strategy A defeats B, which defeats C, which defeats A. A weaker form of intransitivity occurs when A defeats B, but B defeats C by more than A defeats C. It is not clear what to conclude in such cases. In one-on-one play against a Nash equilibrium, many strategies of varying exploitability may tie. Even more problematic is that generating an unabstracted equilibrium strategy in human-scale domains is intractable. Finally, while measuring the exploitability of abstract strategies directly addresses the goal of approximating a Nash equilibrium, recent research has shown that abstract game equilibria may not be the abstract strategies with the lowest real game exploitability [18, 12]. In addition, both Waugh *et al.* [17, p.30 and p.52] (in a toy game) and Johanson *et al.* [12] (in Texas Hold'em), found that exploitability does not correlate well with one-on-one performance.

Johanson *et al.* recently presented CFR-BR: a CFR variant that, in perfect recall abstractions, converges towards an abstract strategy with the lowest real game exploitability [10]. These strategies are not abstract game equilibria, as are found by CFR, but instead are the closest approximations to a real game equilibrium that can be represented within an abstraction. In practice, the exploitability of these CFR-BR strategies is as little as $\frac{1}{3}$ of those found via CFR. While CFR-BR's convergence is only proven for perfect recall abstractions, in practice the same degree of improvement is shown in imperfect recall games. CFR-BR requires repeated traversals of the real game tree, and may not be tractable in large domains where abstraction enables CFR. However, calculating a strategy's exploitability also requires a real game tree traversal (although an efficient traversal may be possible [12]), and in such large games one-on-one performance may remain the only viable evaluation.

Johanson *et al.* also demonstrated that CFR-BR could be used for evaluating abstractions by measuring the closest approximation to a Nash equilibrium that can be represented by the abstraction [10]. In this paper, we will broadly apply the CFR-BR technique for the first time to compare new and existing abstraction techniques. Our experiments will evaluate two abstraction choices that have been raised by recent publications: the effectiveness of expectation-based as opposed to distribution-aware abstractions, as proposed by Gilpin and Sandholm [7], and the use of perfect recall as opposed to imperfect recall, as proposed by Waugh *et al.* [19]. We will also present for the first time the abstraction technique and distance metrics used by the Hyperborean agent in the Annual Computer Poker Competition since 2010.

4. ABSTRACTION AS CLUSTERING

To eliminate the need for direct human expert knowledge when creating an abstraction, the abstraction generation problem will be considered as a clustering problem. Given a target number of clusters (*i.e.* buckets) k and a distance function between information sets, a clustering algorithm can be used to partition the real information sets into the buckets that form the information sets of the abstract game.



	Earth mover's Distance				E[HS] Distance			
	4s4h	6s6h	TsJs	QsKs	4s4h	6s6h	TsJs	QsKs
4s4h		3.101	6.212	5.143		0.063	0.005	0.064
6s6h	3.101		6.462	5.286	0.063		0.015	0.001
TsJs	6.212	6.462		3.103	0.005	0.015		0.059
QsKs	5.143	5.286	3.103		0.064	0.001	0.059	

Figure 2: (top) Hand Strength histograms for four poker hands at the start of the game. (bottom) Earth mover's and E[HS] distances.

Using a clustering algorithm allows the abstraction designer to focus on two aspects of the task: designing a distance metric that represents the strategic similarity of two information sets, and choosing the total number of clusters on each round, k_i, so that the total number of information sets in the resulting abstract game is small enough to solve.

In practice, the number of information sets to be clustered can be very large, making the use of many clustering algorithms computationally intractable. In the poker domain, for example, the final round of Texas hold'em has 2,428,287,420 canonical combinations of public and private cards to be grouped into between one thousand (a small abstraction) and one million (a large abstraction) clusters or more. To make this large clustering problem tractable, we use a k-means implementation that uses the triangle inequality to reduce the number of distance function calls [3]. Multiple restarts and the k-means++ initialization [1] are also used to improve the quality of the clustering.

As in previous work in the limit Texas hold'em poker domain, the abstractions used in our experiments will only merge information sets on the basis of having similar chance events. This approach leaves the players' actions unabstracted and reduces the abstraction generation task to that of finding clusters of similar private and public cards. In the remainder of this section, we will present two new distance metrics for the poker domain that capture strategic similarities that were not handled by earlier expectation-based approaches, and describe how imperfect recall can be used to reallocate the distribution of buckets throughout the game.

Hand Strength Distributions. In Section 2, we described the expected hand strength metric. In the final round of the game, hand strength measures the probability of winning against a randomly sampled opponent hand, given the public cards. Earlier in the game, $E[HS]$ measures the expectation of hand strength over all possibilities for the remaining public cards. Thus, $E[HS]$ summarizes the *distribution* over possible end-game strengths into a single expected value.

As noted by Gilpin and Sandholm [6], this single value is unable to distinguish hands with differing potential to improve. Consider Figure 2(top), which shows the distributions over the final round hand strength of four Texas hold'em poker hands in the first round of the game. Each distribution is discretized into a histogram with values ranging from 0 (a guaranteed loss) to 1 (a guaranteed win). The height of each bar indicates the probability of the remaining public cards resulting in that hand strength, and the vertical black line and label shows $E[HS]$.

Note that the top and bottom histograms have different distribution shapes: 4♠4♡ and 6♠6♡ have most of their weight near their $E[HS]$ values, while T♠J♠ and Q♠K♠ have almost no weight near $E[HS]$ as the unrevealed cards will make this hand either strong or weak. This difference is an indication that the top and bottom rows are strategically distinct: the bottom row has high potential, while the top row does not. However, when comparing the columns of hands we find almost identical $E[HS]$ values. As such, expectation-based approaches would merge within each column, whereas merging along each row may be better.

This suggests the use of a *distribution-aware* similarity metric such as earth mover's distance [20] to compare two hand strength distributions. Earth mover's distance measures the minimum work required to change one distribution into another by moving probability mass. In one-dimensional discrete distributions such as these hand strength distributions, it can be efficiently computed with a single pass over the histogram bars. Unlike alternative distance metrics such as L_2 or Kolmogorov-Smirnov, earth mover's distance measures not only the difference in probability mass, but also how far that mass was moved. In Figure 2(bottom), the earth mover's distance and difference in $E[HS]$ for four hands are listed. In partitioning these four hands into two clusters, earth mover's distance would merge the rows (similar distribution shapes) while $E[HS]$ would merge the columns (similar expected values).

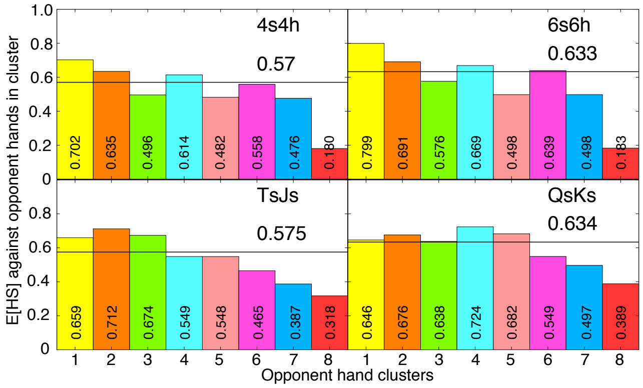
In Texas hold'em poker, hand strength histograms can be precomputed for every combination of private and public cards in the first three rounds, and earth mover's distance provides a candidate distance function for comparing them. After all of the public cards are revealed in the final round, each histogram would be a single impulse at the corresponding hand strength value, and earth mover's distance and the difference in hand strength values would be equivalent.

Opponent Cluster Hand Strength. Our second new distance metric addresses a different aspect of $E[HS]$. The 'hand strength' component of $E[HS]$ measures the probability of winning against a uniform randomly sampled opponent hand at the end of the game, and this provides one summary feature. However, we can also consider our probability of winning against multiple subsets or distributions of possible opponent hands, and thereby generate additional features. While any number of overlapping or non-overlapping subsets could be used, in this work we will partition the 169 starting hands into eight non-overlapping subsets, which we call **opponent clusters**[2]. These were formed by clustering the hands using the earth mover's distance metric on the first round, and are presented in Table 1.

Instead of using a single $E[HS]$ value, we will now compute eight values measuring the hand strength against hands drawn uniform randomly from each opponent cluster. For example, the eighth **Opponent Cluster Hand Strength (OCHS)** feature measures the probability of winning against an opponent hand sampled from the set of top pairs. For each game round, we can precompute a vector of OCHS features to describe a hand's strength. The L_2 distance between two vectors is then used as a distance metric. Figure 3 shows an example with the four first-round hands from Table 2 and the L_2 distances between their vectors. OCHS provides a richer representation of strength than $E[HS]$, which can itself be derived from the vector.

		Suited												
		2	3	4	5	6	7	8	9	T	J	Q	K	A
Unsuited	2	4	1	1	1	1	1	2	2	2	4	4	4	6
	3	1	6	1	1	1	1	2	2	3	4	4	6	6
	4	1	1	6	1	1	2	2	2	3	4	4	6	6
	5	1	1	1	6	3	3	3	3	3	4	4	6	6
	6	1	1	1	1	7	3	3	3	3	4	5	6	6
	7	1	1	1	2	3	7	3	3	5	5	5	6	7
	8	1	1	2	2	3	3	8	3	5	5	5	6	7
	9	2	2	2	2	3	3	3	8	5	5	5	7	7
	T	2	2	2	2	3	3	3	5	8	5	7	7	7
	J	2	2	4	4	4	4	5	5	5	8	7	7	7
	Q	4	4	4	4	4	4	5	5	5	5	8	7	7
	K	4	4	4	6	6	6	6	6	7	7	7	8	7
	A	6	6	6	6	6	6	6	7	7	7	7	7	8

Table 1: Eight hand clusters used for the OCHS features.



	OCHS L_2 Distance				E[HS] Distance			
	4s4h	6s6h	TsJs	QsKs	4s4h	6s6h	TsJs	QsKs
4s4h		0.066	0.095	0.104		0.063	0.005	0.064
6s6h	0.066		0.117	0.105	0.063		0.015	0.001
TsJs	0.095	0.117		0.098	0.005	0.015		0.059
QsKs	0.104	0.105	0.098		0.064	0.001	0.059	

Figure 3: (top) OCHS values for four poker hands at the start of the game. (bottom) OCHS L_2 and E[HS] distances.

Perfect and Imperfect Recall. We will now describe how clustering can be used to form abstractions with perfect and imperfect recall. A perfect recall abstraction is created hierarchically by solving many small clustering problems. To start, the first round of the game is clustered into k_1 clusters. In the second round, perfect recall requires that information sets may only be clustered together if they share the same sequence of observations. This means that we must solve k_1 independent clustering problems, each of which only includes those chance events that are descendents of chance events clustered together in the first round. Although each of these independent clustering problems could assign a different number of clusters, in our experiments we use the same constant k_2 for each. The hierarchical abstraction generation continues until the final round in which we have to solve $k_1 \cdot \ldots \cdot k_{n-1}$ clustering problems, into k_n clusters each, for a total of $k_1 \cdot \ldots \cdot k_n$ clusters in the final round.

When creating an imperfect recall abstraction, we simply cluster all of the chance events without considering their predecessors' clusters on earlier rounds. Solving one large

[2] Our use of these eight clusters was an engineering decision to limit the memory required for the precomputed tables; other choices may be even more effective.

clustering problem is more computationally difficult than solving many small ones. However, the larger number of clusters may allow for a more accurate clustering, as there will not be a need for clusters with similar features that differ only by their history.

The key constraint when making an abstraction is not the number of buckets either in each round or overall, but the total number of information sets in the resulting game, as this determines the memory required to solve it. In imperfect recall abstractions it is possible to change the distribution of buckets throughout the game, dramatically increasing the number of buckets in early rounds, without changing the overall number of information sets. We demonstrate this effect in Table 2. The 'Action Sequences' columns describe only the players' actions and not the chance events, and shows the number of action sequences leading to a choice inside the round and continuing to the next round. The next three sections describe nearly equally sized abstractions. PR-10-10-10-10 uses perfect recall, while IR-10-100-1,000-10,000 and IR-169-9,000-9,000-9,000 use imperfect recall. For each abstraction, the table lists the number of buckets and the number of information sets (buckets times decision points) in the abstraction in that round. The final row shows the total number of information sets.

The PR-10-10-10-10 and IR-10-100-1,000-10,000 abstract games are exactly the same size and use the same total number of buckets on each round: either through multiple small perfect recall clusterings, or in one large imperfect recall clustering. The IR-169-9,000-9,000-9,000 abstraction changes the distribution of buckets, shrinking the final round to 9,000 buckets and removing 5.67 million final round information sets. Due to the multiplying effect of the number of action sequences that reach the final round, removing one fourth-round bucket allows for the addition of 9 third-round buckets, 81 second-round buckets, or 567 first-round buckets. In this way, we can decrease the number of fourth-round buckets by 10% to get an abstraction that is lossless in the first round (*i.e.* it has 169 buckets) and has 9,000 buckets in the second and third rounds. Note that this type of redistribution is not possible when using perfect recall, as the larger number of buckets early in the game need to be remembered until the final round: having 169 buckets in the first round would allow only four buckets on each subsequent round.

5. RESULTS

We can now begin our empirical investigation of abstraction techniques, using the domain of two-player limit Texas hold'em poker. In this paper, we have described three abstraction techniques that are applicable to the first three rounds: Percentile Hand Strength (PHS), k-means earth mover (KE), and k-means OCHS (KO). We have two choices of abstraction techniques to use on the final round: Percentile Hand Strength (PHS) and k-means OCHS (KO). Each combination of an early-game and end-game technique can be used to form a different abstraction. Additionally, we can consider abstractions that use Perfect Recall (PR) and Imperfect Recall (IR), resulting in $2 \times 3 \times 2 = 12$ abstractions. An abstraction (or agent) named IR-KE-KO uses imperfect recall, k-means earth mover to abstract the first three rounds, and k-means OCHS to abstract the final round. Each abstraction will be of the sizes listed in Table 2: either Perfect Recall 10-10-10-10, or Imperfect Recall 169-9000-9000-9000 with a lossless first-round abstraction. In the first three rounds, PHS abstractions will use nesting to partition first by $E[HS^2]$ and then by $E[HS]$. Perfect recall PHS will use $5 \times 2 = 10$ buckets and imperfect recall PHS will use $150 \times 60 = 9000$ buckets. On the final round $E[HS^2]$ ranks hands in the same order as $E[HS]$, and so PHS uses a single partition into 10 or 9000 buckets.

We begin our evaluation of these abstraction styles and distance metrics with the first evaluation technique suggested by Gilpin and Sandholm: one-on-one performance between abstract game Nash equilibrium strategies [6]. For each abstraction, a parallel implementation of the Public Chance Sampled CFR algorithm (PCS) [11] was run for 4 days on a 48-core 2.2 GHz AMD computer[3]. Each pair of strategies was then played against each other for 10 million hands of duplicate poker to obtain statistically significant results with a 95% confidence interval of 1.1 mbb/g. The crosstable of this match is shown in Table 3. We find that every imperfect recall agent, regardless of abstraction technique, outperformed every perfect recall agent. Comparing each imperfect recall agent against its perfect recall equivalent (*i.e.*, IR-KE-KO to PR-KE-KO) we find that the imperfect recall agent also had a higher expected value against each opponent. Overall, the IR-KE-KO agent was undefeated and additionally scored the highest against each adversary. Ranked by average performance, the IR-KO-KO and IR-KE-PHS agents placed second and third.

Gilpin and Sandholm's third abstraction evaluation technique is to calculate the real game exploitability of abstract game Nash equilibrium strategies. In the CFR column of Table 4, we present the exploitability of the same CFR strategies used in the one-on-one crosstable. Note that the results are inconsistent: neither perfect recall or imperfect recall shows a clear advantage. Notably, the two KE-KO strategies are almost exactly tied, despite the fact that IR-KE-KO was considerably stronger in the crosstable. As described earlier, recent work by Waugh *et al.* [18] and Johanson *et al.* [12] has shown that abstract game Nash equilibria are rarely the least exploitable strategies representable in an abstraction, making this method of evaluating abstractions inconclusive.

The recently developed CFR-BR algorithm provides a more reliable metric [10]. In each abstraction, a parallel implementation of CFR-BR was run for 8 days on the same computer used to generate the CFR strategies[4]. The exploitability of these CFR-BR strategies is presented in Table 4, and the results are much more consistent with the one-on-one performance presented in Table 3. IR-KE-KO, IR-KO-KO, and IR-KE-PHS are once again ranked first, second and third. With the exception of PHS-PHS, the imperfect recall agents are also less exploitable than their perfect recall equivalents. Johanson *et al.* note that CFR-BR strategies tend to lose slightly when played against their more exploitable PCS equivalents [10, Fig. 8], and so the CFR strategies' one-on-one performance is of more interest. The outcomes of playing the CFR-BR agents against each other are very similar to those of the CFR agents in Table 3.

In Table 2, we showed that imperfect recall allows us to decrease the number of buckets in later rounds of the game

[3] Johanson *et al.* found that applying PCS to 10-bucket PR-PHS-PHS for 10^5 seconds was sufficient for near convergence [11, Figure 3c].

[4] Johanson *et al.* found that this time, 3.3×10^7 seconds, was sufficient for near convergence using PR-PHS-PHS and IR-KE-KO [10, Figures 6 and 7].

Round	# Action Sequences		PR 10-10-10-10		IR 10-100-1,000-10,000		IR 169-9,000-9,000-9,000	
	Inside	Continuing	# Buckets	# Infosets	# Buckets	# Infosets	# Buckets	# Infosets
1	8	7	10	80	10	80	169	1,352
2	7*10	7*9	10*10	700	100	700	9,000	630,000
3	7*9*10	7*9*9	10*10*10	630,000	1,000	630,000	9,000	5,670,000
4	7*9*9*10		10*10*10*10	56,700,000	10,000	56,700,000	9,000	51,030,000
Total				57,330,780		57,330,780		57,331,352

Table 2: Computing the number of information sets in three nearly equally sized Texas hold'em abstractions.

		Perfect Recall						Imperfect Recall						Mean
		PHS-PHS	PHS-KO	KE-PHS	KE-KO	KO-PHS	KO-KO	PHS-PHS	PHS-KO	KE-PHS	KE-KO	KO-PHS	KO-KO	
PR	PHS-PHS		-0.9	-1.9	-2.7	14.3	14.5	-8.8	-14.8	-21.1	-25.7	-19.1	-22.4	-8.1
	PHS-KO	0.9		-0.3	-2.3	14.2	13.5	-8.7	-14.7	-21.0	-26.3	-17.6	-22.2	-7.7
	KE-PHS	1.9	0.3		-2.4	15.2	14.5	-6.5	-13.0	-18.0	-24.2	-16.1	-21.6	-6.4
	KE-KO	2.7	2.3	2.4		16.9	16.3	-6.6	-12.5	-18.6	-24.8	-15.8	-21.0	-5.3
	KO-PHS	-14.3	-14.2	-15.2	-16.9		-1.69	-25.7	-30.3	-34.9	-39.9	-32.6	-37.3	-23.9
	KO-KO	-14.5	-13.5	-14.5	-16.3	1.69		-25.1	-30.2	-34.8	-39.6	-31.8	-37.2	-23.3
IR	PHS-PHS	8.8	8.7	6.5	6.6	25.7	25.1		-5.5	-15.5	-18.9	-10.8	-15.0	1.4
	PHS-KO	14.8	14.7	13.0	12.5	30.3	30.2	5.5		-9.6	-14.6	-6.0	-10.4	7.3
	KE-PHS	21.1	21.0	18.0	18.6	34.9	34.8	15.5	9.6		-6.7	2.6	-3.8	15.1
	KE-KO	25.7	26.3	24.2	24.8	39.9	39.6	18.9	14.6	6.7		9.3	2.8	21.2
	KO-PHS	19.1	17.6	16.1	15.8	32.6	31.8	10.8	6.0	-2.6	-9.3		-7.2	11.9
	KO-KO	22.4	22.2	21.6	21.0	37.3	37.2	15.0	10.4	3.8	-2.8	7.2		17.8

Table 3: Average performance in games between abstract strategies generated by Public Chance Sampled CFR. Results are in milli-big-blinds/game (mbb/g) over a 10 million hand duplicate match with a 95% confidence interval of 1.1 mbb/g.

	CFR		CFR-BR	
	PR	IR	PR	IR
PHS-PHS	288.942	358.188	89.632	94.841
PHS-KO	289.152	318.197	90.371	85.275
KE-PHS	281.63	339.872	90.720	80.557
KE-KO	282.856	282.395	84.039	64.820
KO-PHS	335.902	355.881	105.389	88.546
KO-KO	330.319	291.574	105.523	73.091

Table 4: Exploitability of CFR-BR and CFR strategies. Results are measured in milli-big-blinds/game and are exact.

Abstraction	CFR-BR Exploitability
PR KE-KO 10-10-10-10	84.039
IR KE-KO 10-100-1000-10000	89.7975
IR KE-KO 169-9000-9000-9000	64.820

Table 5: Effect of redistributing buckets in an abstraction.

in return for many more buckets in earlier rounds, without increasing the size of the game. In Table 5, we revisit this decision and also consider an IR KE-KO 10-100-1,000-10,000 abstraction. We find that the imperfect recall agent is more exploitable than its perfect recall equivalent, while the redistributed agent shows a significant decrease.

We can also measure the exploitability of CFR-BR strategies as a response to abstraction size, to investigate if these abstraction techniques improve at different rates. For this experiment, we consider five sizes of four abstractions: PR-PHS-PHS and IR-PHS-PHS, PR-KE-KO and IR-KE-KO. The perfect recall abstractions branch to 5, 6, 8, 10 and 12 buckets on each round, and the imperfect recall abstractions have a lossless first round and 570, 1175, 3700, 9000 and 18630 buckets on later rounds.

The CFR-BR exploitability results for these abstractions are presented in Figure 4 as a log-log plot. Comparing the slope of each curve, we find that IR-KE-KO and PR-KE-KO are steeper than PR-PHS-PHS and IR-PHS-PHS, indicating that their advantage increases with the abstraction size. The combination of abstraction techniques presented in this paper, imperfect recall with redistribution and the KE and KO techniques, is less exploitable at all tested abstraction sizes.

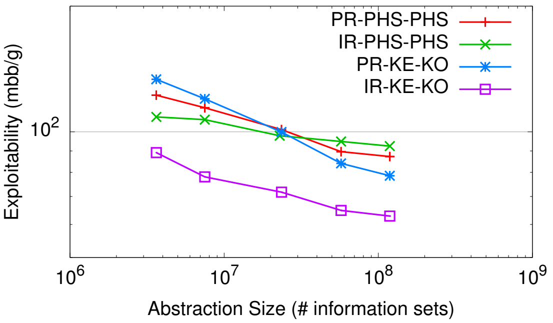


Figure 4: Exploitability of CFR-BR strategies in four abstractions as the abstraction size is varied.

6. DISCUSSION

Recent research towards state-space abstraction in the poker domain has raised two issues: the effectiveness of distribution-aware as compared to expectation-based approaches (as described by Gilpin and Sandholm [6]) and the practical uses of imperfect recall (as described by Waugh *et al.* [19]). The discovery that the exploitability of abstract game Nash equilibrium strategies was not an accurate measure of an abstraction's ability to represent a real Nash equilibrium has left these issues unresolved. Our goal in these experiments was to use the recently developed CFR-BR technique to survey these abstraction choices and evaluate them more precisely.

Gilpin and Sandholm's investigation showed that while agents in expectation-based abstractions are more effective in small abstract games, the distribution-aware agents match and surpass them as the number of buckets is increased. Figure 4 shows that our experiment matches their result: the steeper slope of the PR-KE-KO line as compared to PR-PHS-PHS shows that the distribution-aware metric makes better use of the available buckets as the abstraction size increases. In addition, the one-on-one crosstable in Table 3 shows that the distribution-aware agents using the k-means earth mover's abstractions outperformed the expectation-based agents.

We now turn to imperfect recall. In one-on-one performance, every imperfect recall agent, regardless of its abstraction features, outperformed every perfect recall agent. In terms of exploitability, aside from IR-PHS-PHS, every CFR-BR agent using imperfect recall except was found to be less exploitable than its perfect recall equivalent. While CFR-BR is not theoretically guaranteed to converge to a least exploitable strategy in an imperfect recall game, our results provide an upper bound: the least exploitable IR-KE-KO strategy is exploitable for at most 64.820 mbb/g, far less than the least exploitable perfect recall agent. While Waugh *et al.* found that imperfect recall and additional features provided a small advantage, we have shown a significant improvement while using the same domain features.

Finally, the CFR and CFR-BR results presented in Table 4 support Johanson *et al.*'s proposed use of CFR-BR to evaluate strategies instead of measuring the exploitability of abstract game Nash equilibria. The CFR results are inconsistent, showing no clear advantage for perfect or imperfect recall, and ordering the agents differently than the one-on-one crosstable. While there is no guarantee that the one-on-one results and exploitability should agree, the CFR-BR strategies are both far less exploitable in all cases, show an advantage for imperfect recall, and rank the top three agents in the same order as the one-on-one results. Using CFR-BR to evaluate abstractions based on their ability to approximate an unabstracted Nash equilibrium provides a more consistent metric than the previous approaches.

7. CONCLUSION

Historically, state-space abstraction techniques in extensive-form games have been evaluated by computing optimal abstract strategies and comparing their one-on-one performance and exploitability. A recently published technique, CFR-BR, directly finds the abstract strategy with the lowest real game exploitability, providing a more consistent measure of an abstraction's quality. Using this technique, we evaluated two abstraction choices in the poker domain: expectation-based as opposed to distribution-aware distance metrics, and imperfect recall abstractions. Our findings on distribution-aware techniques support those of Gilpin and Sandholm: distribution-aware distance metrics provide a clear advantage once the abstract game is large enough. We also demonstrated a clear improvement in one-on-one performance and exploitability through the use of imperfect recall abstractions, and demonstrated that imperfect recall abstractions can contain less exploitable strategies than equal sized perfect recall strategies.

Acknowledgements

We would like to thank Mihai Ciucu, Eric Jackson, Mengliao Wang, and the members of the University of Alberta Computer Poker Research Group. This research was supported by NSERC and Alberta Innovates Technology Futures, and was made possible by the computing resources provided by WestGrid, Réseau Québécois de Calcul de Haute Performance, and Compute/Calcul Canada.

8. REFERENCES

[1] D. Arthur and S. Vassilvitskii. *k*-means++: The advantages of careful seeding. In *SODA*, 2007.

[2] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *IJCAI*, 2003.

[3] C. Elkan. Using the triangle inequality to accelerate *k*-means. In *ICML*, 2003.

[4] A. Gilpin and T. Sandholm. A competitive texas hold'em poker player via automated abstraction and real-time equilibrium computation. In *AAAI*, 2006.

[5] A. Gilpin and T. Sandholm. Lossless abstraction of imperfect information games. *Journal of the ACM*, 54(5), 2007.

[6] A. Gilpin and T. Sandholm. Expectation-based versus potential-aware automated abstraction in imperfect information games: An experimental comparison using poker. In *AAAI*, 2008.

[7] A. Gilpin, T. Sandholm, and T. B. Sørensen. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of texas hold'em poker. In *AAAI*, 2007.

[8] S. Hoda, A. Gilpin, J. Peña, and T. Sandholm. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2):494–512, 2010.

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

[10] M. Johanson, N. Bard, N. Burch, and M. Bowling. Finding optimal abstract strategies in extensive-form games. In *AAAI*, 2012.

[11] M. Johanson, N. Bard, M. Lanctot, R. Gibson, and M. Bowling. Efficient Nash equilibrium approximation through Monte Carlo counterfactual regret minimization. In *AAMAS*, 2012.

[12] M. Johanson, K. Waugh, M. Bowling, and M. Zinkevich. Accelerating best response calculation in large extensive games. In *IJCAI*, pages 258–265, 2011.

[13] M. Lanctot, R. Gibson, N. Burch, and M. Bowling. No-regret learning in extensive-form games with imperfect recall. In *ICML*, 2012.

[14] N. A. Risk and D. Szafron. Using counterfactual regret minimization to create competitive multiplayer poker agents. In *AAMAS*, 2010.

[15] T. Sandholm. The state of solving large incomplete-information games, and application to poker. *AI Magazine*, 31(4):13–32, 2010.

[16] J. Shi and M. L. Littman. Abstraction methods for game theoretic poker. In *Computers and Games*, 2000.

[17] K. Waugh. Abstraction in large extensive games. Master's thesis, University of Alberta, 2009.

[18] K. Waugh, D. Schnizlein, M. Bowling, and D. Szafron. Abstraction pathology in extensive games. In *AAMAS*, 2009.

[19] K. Waugh, M. Zinkevich, M. Johanson, M. Kan, D. Schnizlein, and M. Bowling. A practical use of imperfect recall. In *SARA*, 2009.

[20] Wikipedia. Earth mover's distance — Wikipedia, the free encyclopedia, 2013.

[21] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione. Regret minimization in games with incomplete information. In *NIPS*, 2008.