Mach Learn (2009) 74: 159–189
DOI 10.1007/s10994-008-5091-5

Effective short-term opponent exploitation in simplified poker

Finnegan Southey · Bret Hoehn · Robert C. Holte

Received: 13 October 2005 / Revised: 7 October 2008 / Accepted: 9 October 2008 /
Published online: 14 November 2008
Springer Science+Business Media, LLC 2008

Abstract Uncertainty in poker stems from two key sources, the shuffled deck and an adversary whose strategy is unknown. One approach to playing poker is to find a pessimistic game-theoretic solution (i.e., a Nash equilibrium), but human players have idiosyncratic weaknesses that can be exploited if some model or counter-strategy can be learned by observing their play. However, games against humans last for at most a few hundred hands, so learning must be very fast to be useful. We explore two approaches to opponent modelling in the context of Kuhn poker, a small game for which game-theoretic solutions are known. Parameter estimation and expert algorithms are both studied. Experiments demonstrate that, even in this small game, convergence to maximally exploitive solutions in a small number of hands is impractical, but that good (e.g., better than Nash) performance can be achieved in as few as 50 hands. Finally, we show that amongst a set of strategies with equal game-theoretic value, in particular the set of Nash equilibrium strategies, some are preferable because they speed learning of the opponent's strategy by exploring it more effectively.

Keywords Game-playing · Opponent modelling · Experts · Bayesian · Poker

1 Introduction

Poker is a game of imperfect information against an adversary with an unknown, stochastic strategy. It represents a tough challenge to artificial intelligence research. Game-theoretic approaches seek to approximate the Nash equilibrium (i.e., maximin) strategies of the game (Koller and Pfeffer 1997; Billings et al. 2003; Gilpin and Sandholm 2005; Zinkevich et al.

Electronic supplementary material The online version of this article (http://dx.doi.org/10.1007/s10994-008-5091-5) contains supplementary material, which is available to authorized users.

F. Southey (✉)
Google, 1600 Amphitheater Pkwy., Mountain View, CA 94043, USA
e-mail: workfinn@lucubratio.org

B. Hoehn · R.C. Holte
Dept. of Computing Science, University of Alberta, 349 Athabasca Hall, Edmonton, AB,
T6G 2E8 Canada

2007a, 2007b), but this represents a pessimistic worldview where we assume some form of optimality in our opponent. Human players have weaknesses that can be exploited to obtain winnings higher than the game-theoretic value of the game. Learning by observing their play allows us to exploit their idiosyncratic weaknesses. The learning can be done either directly, by estimating a model of their strategy and then computing an appropriate response, or indirectly, by identifying an effective counter-strategy.

Several factors render this difficult in practice. First, real-world poker games like Texas Hold'em have huge game trees and the strategies involve many parameters (e.g., two-player, limit Texas Hold'em has $\sim 10^{18}$ parameters; Billings et al. 2003). The game also has high variance, stemming from the shuffled deck and one or both players playing stochastic strategies. Strategically complex, the aim in poker is not simply to win but to maximize winnings by enticing a weakly-positioned opponent to bet or a strongly-positioned opponent to fold. Decisions during a hand must be made with imperfect information because we cannot see our opponent's cards. A further aggravation of this uncertainty arises when one player folds. The opponent's cards are never observed, leaving us with only a partial observation even after the hand has been played out. Finally, we cannot expect a large amount of data when playing human opponents. We may play 200 hands or less against a given opponent and must learn how they can be exploited quickly in order to gain by their weaknesses.

This research explores how rapidly one can gain an advantage by observing opponent play given that only a small number of hands will be played in total. The aim here is not to develop new, specialized algorithms but rather to assess how established learning techniques perform. Can these algorithms improve over game-theoretic performance by quickly learning and exploiting human weaknesses? How should one play while learning? To address these questions, two standard learning approaches are studied: *maximum a posteriori parameter estimation* (*parameter learning*), and an "experts" method derived from Exp4 (Auer et al. 1995) (*strategy learning*). Both will be described in detail.

While most existing poker opponent modelling research focuses on real-world games (Korb and Nicholson 1999; Billings et al. 2004; Zinkevich et al. 2007a), we systematically study a simpler version, reducing the game's intrinsic difficulty to show that, even in what might be considered the simplest case, the problem is still hard in the sense that one cannot expect to converge to a near-maximal exploitation with so little data. We start by assuming that the opponent's strategy is fixed. Tracking a non-stationary strategy is a hard problem and learning to exploit a fixed strategy is clearly the first step. We also limit complexity by considering the game of Kuhn poker (Kuhn 1950), a tiny game for which complete game-theoretic analysis is available. Finally, we evaluate learning in a two-phase manner—the first phase exploring and learning while the second phase switches to pure exploitation based on what was learned. Note that we do not propose this fixed switching point as an actual strategy for play, but rather as a readily comprehensible experimental methodology. We use this simplified framework to demonstrate the following points:

- Learning to maximally exploit an opponent in a small number of hands is not feasible.
- A substantial advantage can nonetheless be attained rapidly, making short-term learning a winning proposition.
- Finally, we observe that, amongst the set of Nash strategies for the learner, the exploration inherent in some strategies facilitates faster learning compared with other members of the set.

The material presented here extends an earlier paper on this research (Hoehn et al. 2005) by presenting a wider range of results (including a very complete set in a supplemental online appendix), full derivations of our estimators, and more detailed descriptions of algorithms and experimental methodology. Related studies can be found in Hoehn (2006).

2 Kuhn poker

2.1 Rules and notation

Kuhn poker (Kuhn 1950) is a very simple, two-player game (P1—Player 1, P2—Player 2). The deck consists of three cards (J—Jack, Q—Queen, and K—King). There are two actions available: *bet* and *pass*. The value of each bet is 1. In the event of a *showdown* (players have matched bets), both players reveal their card and the player with the higher card wins the pot (the King is highest and the Jack is lowest). In the event of a *fold* (one player increases the standing bet and the other declines to match the bet), the folding player loses and the players' cards are not revealed. A single *hand* proceeds as follows:

- Both players initially put an ante of 1 into the pot.
- Each player is dealt a single card which they keep private and the remaining card is unseen by either player.
- After the deal, P1 has the opportunity to bet or pass.
  - If P1 bets in round one, then in round two P2 can:
    - pass (fold) and forfeit the pot to P1, or
    - bet (call P1's bet) and the game then ends in a showdown.
  - If P1 passes (checks) in round one, then in round two P2 can:
    - pass (check) and go to a showdown, or
    - bet, in which case there is a third action where P1 can
      - bet (call P2's bet) and go to showdown, or
      - pass (fold) and forfeit to P2.

It is useful to summarize a hand so we introduce the following notation

$$\langle \text{P1 card}\rangle\langle \text{P2 card}\rangle\langle \text{P1 action}\rangle\langle \text{P2 action}\rangle\cdots$$

where the cards are one of "J", "Q", "K", or "?" (the last meaning that the card has not been observed). The actions for P1 are "b" or "p" and for P2 they are "B" or "P" (we use upper vs. lowercase to distinguish the players more readily). For example, the string "QJpBb" means that P1 held a Queen, P2 held a Jack, and the actions were "P1 passes", "P2 bets", and "P1 bets". In the case where one player folds, players cannot observe each other's cards. It is useful to record observations of this kind, for example, "K?bP" means that P1 held a K and observed the actions "P1 bets" and "P2 passes". Because this game folded, P2's card was not observed by P1.

Kuhn poker is an attractive choice for study because, as the following discussion will elaborate, it has been analyzed in detail and can be completely characterized analytically. We can therefore compare empirical results to theoretical ideals easily. The game captures several essential qualities of real-world poker games: decision making with partial observations, bluffing, and information lost due to folded hands. Finally, the game is small enough to visualize some results in a fairly direct manner.

2.2 Analysis of the game

Figure 1 shows the game tree we consider in this work. The top row of nodes shows all possible combinations of cards held by the players (e.g., J|Q means P1 holds the Jack and P2 holds the Queen). P1's value for each outcome is indicated in the leaf nodes. The game is a *zero-sum* game (whatever one player gains, the other loses) so P2's values are simply the negation of P1's. Branches correspond to the available actions. Note that the dominated

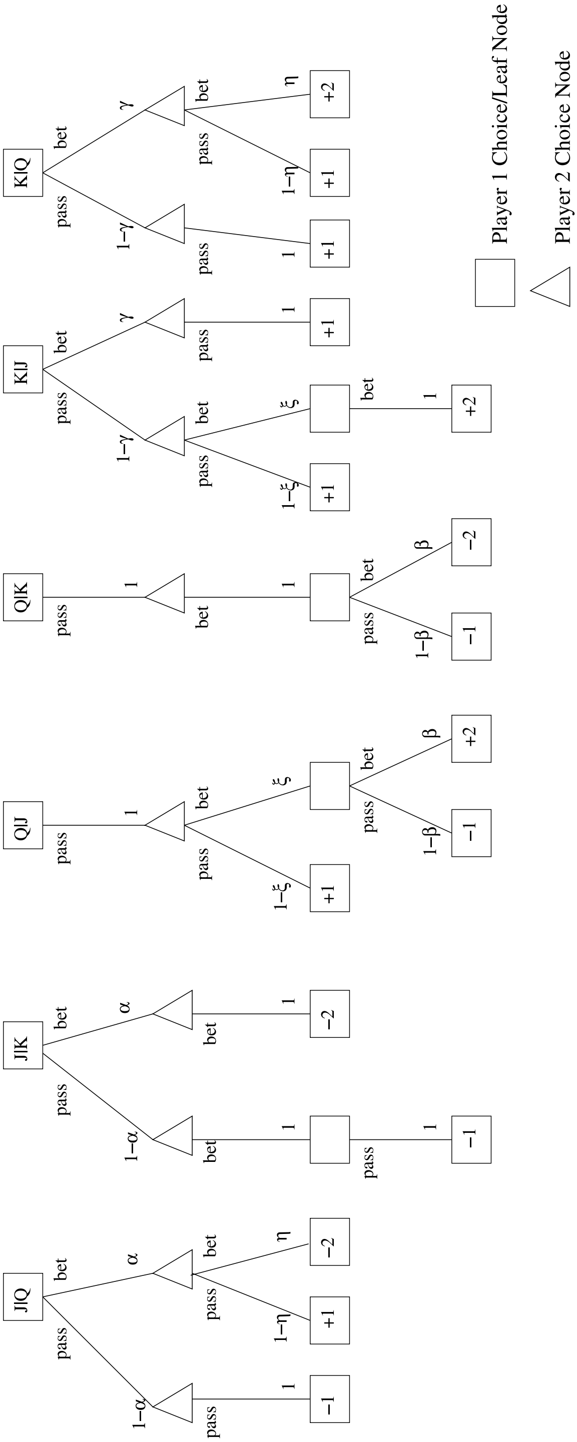

Fig. 1 Kuhn poker game tree with dominated strategies removed

strategies have been removed from this tree already. Informally, a dominated strategy is one for which there exists an alternative strategy that offers better value against at least one opponent strategy and equal value against all others.[1]

We eliminate these obvious sources of universally suboptimal play but note that strategies remain that play suboptimally against specific opponents. Dominated strategies often correspond to obvious mistakes such as folding when guaranteed to win, whereas the remaining suboptimal strategies typically err in failing to achieve an ideal mixture of actions (e.g., bluffing too often in a particular situation). We have eliminated obvious errors because our goal is to develop learning techniques that will challenge the best human and machine players rather than simply exploit poor play by weak players.

In Fig. 1, branches where alternatives are available are labeled with a parameter. The game has a well-known parametrization (Kuhn 1950) in which P1's strategy can be summarized by three parameters (α, β, γ), and P2's by two parameters (η, ξ). The decisions governed by these parameters are shown in Fig. 1. The meaning of the decisions governed by each parameter are

- $\alpha =$ probability that P1 bets in the first round when holding the Jack
- $\beta =$ probability that P1 calls in the third round when holding the Queen
- $\gamma =$ probability that P1 bets in the first round when holding the King
- $\eta =$ probability that P2 calls a P1 bet when holding the Queen
- $\xi =$ probability that P2 bluffs by betting when holding a Jack after P1 passes in the first round

If players play only non-dominated strategies, the expected value to P1 of any pair of opposing strategies (i.e., any pair of (α, β, γ) and (η, ξ)) is given by

$$EV(\alpha, \beta, \gamma, \eta, \xi) = \frac{1}{6}\left[\eta(-3\alpha + \gamma) + \xi(-1 + 3\beta - \gamma) + \alpha - \beta\right] \tag{1}$$

An important concept in game theory is that of a *Nash equilibrium*. Again informally, a Nash equilibrium is a pair of strategies, one for each player, such that neither player gains by unilaterally deviating from their own Nash strategy. So long as one player plays a Nash strategy, the other cannot, by playing some non-Nash strategy, improve on the expected value of playing a Nash strategy themselves.[2] Kuhn determined that the set of Nash strategies for P1 has the form $(\alpha, \beta, \gamma) = (\gamma/3, (1 + \gamma)/3, \gamma)$ for $0 \leq \gamma \leq 1$. Thus, there is a continuum of Nash strategies for P1 governed by a single parameter. There is only one Nash strategy for P2, $\eta = 1/3$ and $\xi = 1/3$; all other P2 strategies can be exploited by P1. If either player plays a Nash strategy (and neither plays dominated strategies), then P1 expects to lose at

[1] Dominance can be *strong* (the dominating strategy is always a better choice, regardless of the opponent's actions) or *weak* (the dominating strategy is better for one or more opponent strategies and equal for the rest). For example, the P2 strategy of passing when holding the Queen and faced with a P1 pass weakly dominates the strategy of betting in the same scenario. Similarly, the P1 strategy of betting in the first round when holding the Queen is weakly dominated by the strategy to pass and call a P2 bet (if such a bet is made). This latter dominance is not immediately obvious but is arrived at by iterative elimination of dominated strategies (Fudenberg and Levine 1998).

[2] In two-player, zero-sum games such as poker, all Nash strategies are interchangeable. That is, each player has a set of Nash strategies and any pairing from those two sets forms a Nash equilibrium. Furthermore, all pairings give the same expected value to the players. In more general games, this is not always the case; only specific pairs of strategies form equilibria and the equilibria may have different expected values.

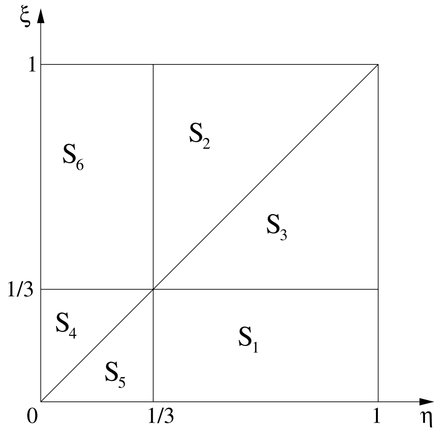

Fig. 2 Partition of P2 strategy space by maximal P1 strategies

a rate of $-1/18$ per hand.[3] Thus P1 can only hope to win in the long run if P2 is playing suboptimally and P1 deviates from playing Nash strategies to exploit errors in P2's play. Our discussion focuses on playing as P1 and exploiting P2, so, unless specified otherwise, all observations and results are from this perspective.

For any given P2 strategy, there is a set of corresponding P1 strategies that maximally exploit P2. Each such maximally exploitive strategy is called a *best response* (Fudenberg and Levine 1998). Furthermore, the set of best responses includes at least one *pure strategy*, a deterministic strategy in which all probabilities are zero or one. Thus, in any attempt to exploit an opponent, we would ideally use a strategy that is a best response to that opponent. Note that this applies equally to P2 exploiting P1.

In a game of this size, it is possible to characterize the space of strategies in great detail. In order to better systematize our study, we have analytically derived boundaries within the P2 strategy space that allow us to ensure that our study thoroughly covers all interesting regions of that space (Hoehn et al. 2005). The two-dimensional ($\eta \times \xi$) strategy-space for P2 can be partitioned into the 6 regions shown in Fig. 2. Within each region, one of P1's pure strategies gives maximal value to P1 (i.e., the pure strategy is a best response to all opponent strategies in that region). For P2 strategies corresponding to points on the lines dividing the regions, the adjacent P1 best responses achieve the same value. The intersection of the three dividing lines is the Nash strategy for P2. Therefore, to maximally exploit P2, it is sufficient to identify the region in which P2's strategy lies and play the corresponding P1 pure strategy. Note that there are 8 pure strategies for P1, written as (α, β, γ): $S_0 = (0, 0, 0)$, $S_1 = (0, 0, 1)$, $S_2 = (0, 1, 0), \ldots, S_7 = (1, 1, 1)$. Two of these ($S_0$ and S_7) are not a best response to any P2 strategy, so we need only consider the remaining six. In brief, the partitioning was derived by substituting the various P1 pure strategies into the expected value equation and then determining the ranges over η and ξ for which one pure strategy gives greater value than all others.[4] Partitioning of the strategy space is not required for any of our algorithms but has served to guide our choice of opponents and assist in understanding results.

[3] This is true because, in Kuhn poker, all non-dominated strategies are *essential* (i.e., have a non-zero probability of being played as part of some Nash strategy). In two player, zero-sum games, if one player plays a Nash strategy and the other plays some mixture of essential strategies, then they will always obtain the value of the equilibrium (von Neumann and Morgenstern 1947).

[4] We would like to thank Valeriy Bulitko for the analysis that produced this partitioning (Hoehn et al. 2005).

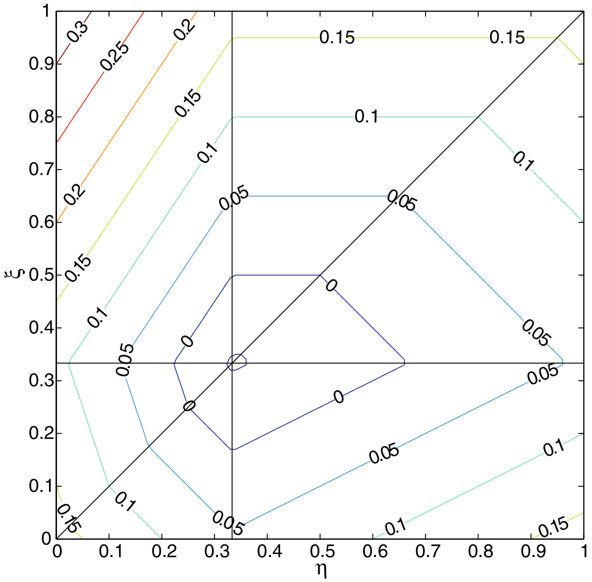

Fig. 3 Contour plot of exploitability of P2 over P2 strategy space

This natural division of P2's strategy space was used to obtain a selection of suboptimal opponents for our study. Six opponent strategies were created by selecting a point at random from each of the six regions. Written as pairs (η, ξ), they are $O_1 = (0.25, 0.67)$, $O_2 = (0.75, 0.8)$, $O_3 = (0.67, 0.4)$, $O_4 = (0.5, 0.29)$, $O_5 = (0.25, 0.17)$, $O_6 = (0.17, 0.2)$. Experiments were run against these six opponents. It is not necessary to run experiments against a Nash strategy opponent since the value against such an opponent is guaranteed regardless of the strategy P1 might play (excepting dominated strategies). Any attempt to exploit a Nash opponent can neither benefit nor harm either player.

Other experiments were run against randomly sampled opponents that all share the same level of exploitability. Figure 3 shows the exploitability of P2 over its strategy space. It is a contour plot of the expected value to P1 if P1 plays the appropriate best response at every point in P2 strategy space. This expected value is the best P1 can possibly achieve and therefore represents the maximum exploitability for every P2 strategy. Note that within a small area around the P2 Nash strategy, P1's expected value is less than 0. P1 can never hope to win against a P2 opponent playing within this area, although P1 can attempt to minimize the loss. Beyond the zero contour, exploiting the opponent is a winning proposition. Our randomly sampled opponents can be thought of as samples from some contour on this plot. This means that they behave differently, but all have the same value when challenged by the appropriate best response.

Experiments were also run where P2 is modelling P1. However, our discussion will be in terms of P1 modelling P2 for the sake of simplicity.

3 Parameter learning

The first approach we consider for exploiting the opponent is to directly estimate the parameters of their strategy and play a best response to that strategy. We start with a prior over the opponent's strategy and compute the *maximum a posteriori* (MAP) estimate of those parameters given our observations. This is a form of Bayesian parameter estimation, a typical approach to learning and therefore a natural choice for our study.

In general poker games a hand either results in a showdown, in which case the opponent's cards are observed, or a fold, which leaves the opponent's cards uncertain (we only get to observe their actions, our own cards, and any public cards). However, in Kuhn poker, the small deck and dominated strategies conspire in certain cases to make the opponent's cards obvious despite their folding. For example, if P1 holds the Jack and observes the sequence *bet–pass*, we can conclude that P2 must hold the Queen. Examination of Fig. 1 shows that if P2 holds the King, then passing on a bet would be a dominated strategy. Because we have precluded dominated strategies, we can therefore conclude that P2 holds the Queen despite never seeing it. Thus, certain folding observations (but not all) contain as much information as a showdown.

Parameter estimation in Kuhn poker is quite straightforward because in no case does the estimate of any single player parameter depend on an earlier decision governed by some other parameter belonging to that player. The task of computing a posterior distribution over opponent strategies for arbitrary poker games is more complicated and is discussed in a separate paper (Southey et al. 2005). For the present study, the dominated strategies and small deck again render the task relatively simple.

Once the opponent's strategy has been estimated, a best response must be computed. In general, this is achieved via the *expectimax* algorithm (Michie 1966; Russell and Norvig 1995), which involves a traversal of the game tree. However, since Kuhn poker has been thoroughly analyzed and the opponent strategy space partitioned as shown in Fig. 2, we simply determine the region in which the estimate lies and lookup the corresponding pure strategy. While we discuss the issue of scaling to larger games more fully in Sect. 9, we note here that the partitioning of the opponent's strategy space is simply a convenience in this particular case, and that computing (or approximating) an expectimax response in larger games need not be prohibitively expensive.

We have chosen to compute a response to the MAP estimate of the opponent's strategy, essentially assuming that the most probable opponent strategy is the only strategy they could be playing. In general, it would be more robust to consider all possible strategies the opponent might be playing weighted according to the posterior distribution over those strategies. If, for example, the posterior distribution gives high probability to multiple strategies that differ significantly (e.g., the posterior distribution is multimodal), the MAP approach may fixate on the wrong strategy. Computing the *Bayesian best response* to an opponent's play in poker has been explored previously for larger poker games (Southey et al. 2005). In this work we consider only the MAP estimate because our choice of prior distribution (see Sect. 3.1) means that the posterior distribution will be unimodal, so the MAP estimate is unlikely to be significantly deceptive.

It should be mentioned here that MAP estimation of the opponent's strategy followed by play of a best response is a generalization of a classic approach to learning in games known as *fictitious play* (Brown 1951) (see Fudenberg and Levine 1998 for a lengthy discussion of fictitious play). In fictitious play, the learner assumes their opponent's strategy is stationary. Each round, a *maximum likelihood* estimate of the opponent's strategy is computed and a best response is played against it. This is identical to our own procedure when a uniform prior is used ($Beta(0, 0)$). Note that the term "fictitious play" arises from its original conception as a means to compute maximin strategies by self-play. In this scenario, two instances of this algorithm repeatedly play against each other. It can be shown that each player's behavior, averaged over repeated games, will converge to a maximin strategy (i.e., a Nash strategy in two player zero-sum games) (Robinson 1951). Our experiments with different priors include the uniform prior.

3.1 Priors

For our prior we use a *Beta* distribution, which gives a probability distribution over a single parameter that ranges from 0 to 1. When modelling P2, therefore, we require two *Beta* distributions to characterize our prior belief of how they play. Each *Beta* distribution is characterized by two parameters, $\theta \geq 0$ and $\omega \geq 0$. The distribution can be understood as pretending that we have observed the opponent's binary decisions several times in the past, and that we observed θ choices one way and ω choices the other way. Thus, low values for θ and ω (e.g., $\mathit{Beta}(1,1)$) represent a weak prior, easily overcome by subsequent observations. Larger values (e.g., $\mathit{Beta}(10,10)$) represent a much stronger belief. We offer a somewhat more detailed discussion of *Beta* distributions in the Appendix.

A poorly chosen prior (i.e. a bad model of the opponent) that is weak may not cost us much because it will be quickly overwhelmed by observations. However, a good prior (i.e. a close model of the opponent) that is too weak may be thwarted by unlucky observations early in the game that belie the opponent's true nature. We examine the effects of the prior in a later section. The default prior, unless otherwise specified, is $\mathit{Beta}(1,1)$ for both η and ξ (i.e., the most probable P2 strategy is assumed to be $\eta = 0.5$ and $\xi = 0.5$, pretending we have observed, for each parameter, two decisions (one *bet* and one *pass*) governed by that parameter). Note that this prior was selected, before any experimentation, as a natural first choice if the experimenter was not previously familiar with other logical choices, such as the Nash strategy $(\frac{1}{3}, \frac{1}{3})$. The use of this prior is a very common default choice in Bayesian parameter estimation and other statistical methods where it is sometimes referred to as the *Laplacian correction* (e.g., Margineantu and Dietterich 2002).

It should be noted that playing a best response to the prior strategy itself would have different expected values against different opponents. Therefore, the payoff rates and winnings that will be shown in plots for the experimental results have different values against different opponents even in the first few steps of the game, despite the fact that the same initial strategy is used. We make this observation to explain what, at first, might seem like an inconsistency in our experiments. We present results for a variety of priors in Sect. 5.2.

3.2 Maximum a posteriori for Kuhn poker

We will now derive the MAP parameter estimate for P2, using our *Beta* prior. We want to find the most probable settings for parameters η and ξ given a set of observations O. More formally, we need to solve

$$\arg\max_{\eta,\xi} P(\eta,\xi|O)$$

Using Bayes rule in the standard way, we can get the following proportionality

$$P(\eta,\xi|O) \propto P(O|\eta,\xi)P(\eta,\xi)$$

where $P(\eta,\xi)$ is our prior. Our maximization then, is simply

$$\arg\max_{\eta,\xi} P(\eta,\xi|O) = \arg\max_{\eta,\xi} P(O|\eta,\xi)P(\eta,\xi)$$

We assume the two parameters are independent and each follows some *Beta* distribution, so $P(\eta,\xi) = P(\eta)P(\xi) = \mathit{Beta}(\theta_\eta,\omega_\eta)\mathit{Beta}(\theta_\xi,\omega_\xi)$.

3.2.1 Probability of Kuhn observations given parameters

$P(O|\eta,\xi)$ is the probability of the observations given parameters η and ξ. There are only a small number of possible observations in Kuhn poker, and only a subset of these actually depend on P2's strategy. Most of these observations are quite straightforward. For example, the probability of a single observation of $KQbB$ (i.e., P1 holds K, P2 holds Q, and the betting sequence is *bet–bet*), is $\frac{1}{6}\gamma\eta$. Probabilities such as these can easily be derived by examination of the tree in Fig. 1. In this case, there is a $\frac{1}{6}$ probability of KQ being dealt, and the betting sequence is the product of the two players' parameters γ and η. However, from P1's perspective, γ is a known constant, and so in our maximization, we can drop terms involving P1's parameters and the constants from dealing cards, leaving us with the parameter η. Therefore, the probability of observing the $KQbB$ game N times is proportional to η^N.

A complication arises when one of the players folds. In such a case, P1 does not get to observe P2's card. This can be addressed by marginalizing out the unknown quantity, which consists of summing the probabilities of the observation given every card that P2 could have held. In Kuhn poker, there can only be two such cards.

For example, suppose P1 observes $K?bP$ (i.e., P1 held K, P2's card is unknown because P2 folded, and the betting sequence was $bet-pass$). Two possible hands might have been played ($KJbP$ and $KQbP$). We must therefore sum the probabilities of these two, $P(K?bP)=P(KJbP)+P(KQbP)=\frac{1}{6}\gamma+\frac{1}{6}\gamma(1-\eta)\propto 2-\eta$. Therefore, the probability of observing the $K?bP$ game N times is proportional to $(2-\eta)^N$. A full discussion of handling uncertainty due to folding in poker can be found in Southey et al. (2005).

One additional subtlety relates to dominated strategies. The observation $J?bP$ would appear to be ambiguous because P2 folded. However, if P2 holds K, the strategy of passing is dominated by betting. Since we assume that no player plays dominated strategies, we can then infer that any observation of $J?bP$ was due to P2 holding Q. P1 has certain information about P2's card, even though it was not observed. We therefore write this sequence $J(Q)bP$ to show that it is not truly unknown.[5]

By examining the tree in this fashion, we can derive expressions for every possible observation. We omit a detailed account as they are all similar to the preceding examples. The final expression is then

$$\begin{aligned}P(O|\eta,\xi)\propto{}&\eta^{N_{JQbB}+N_{KQbB}}(1-\eta)^{N_{J(Q)bP}}(2-\eta)^{N_{K?bP}}\\&\xi^{N_{QJpBb}+N_{KJpBb}}(1-\xi)^{N_{QJpP}+N_{KJpP}}(1+\xi)^{N_{Q?pBp}}\end{aligned}$$

where each subscripted N is the number of times each corresponding observation was made.

The *Beta* priors simply correspond to additional counts, pretending we have made past observations. Therefore our objective simply becomes

$$\begin{aligned}P(O|\eta,\xi)P(\eta)P(\xi)\propto{}&\eta^{N_{JQbB}+N_{KQbB}+\theta_\eta}(1-\eta)^{N_{J(Q)bP}+\omega_\eta}(2-\eta)^{N_{K?bP}}\\&\xi^{N_{QJpBb}+N_{KJpBb}+\theta_\xi}(1-\xi)^{N_{QJpP}+N_{KJpP}+\omega_\xi}(1+\xi)^{N_{Q?pBp}}\end{aligned}$$

In order to simplify the discussion that follows, we summarize these counts as follows

$$A=N_{JQbB}+N_{KQbB}+\theta_\eta \qquad D=N_{QJpBb}+N_{KJpBb}+\theta_\xi$$

[5] Equivalently, we could note that the sequence $JKbP$ has probability 0, and then apply the summation over $JKbP$ and $JQbP$ as described for the more general folding case.

$$B = N_{J(Q)bP} + \omega_\eta \qquad E = N_{QJpP} + N_{KJpP} + \omega_\xi$$

$$C = N_{K?bP} \qquad F = N_{Q?pBp}$$

giving us the more readable objective

$$P(O|\eta,\xi)P(\eta)P(\xi) \propto \eta^A(1-\eta)^B(2-\eta)^C\xi^D(1-\xi)^E(1+\xi)^F$$

We now need to find the maximum of this objective. Note that the two parameters η and ξ occur independently, so we can maximize them independently.[6] We will now show the closed form maximization of each.

3.2.2 Maximum a posteriori for η

We maximize by setting the derivative of the posterior probability for η to zero.

$$\begin{aligned}\frac{\partial P(\eta,\xi|O)P(\eta)P(\xi)}{\partial\eta} &\propto A\eta^{A-1}(1-\eta)^B(2-\eta)^C - B\eta^A(1-\eta)^{B-1}(2-\eta)^C \\ &\quad -C\eta^A(1-\eta)^B(2-\eta)^{C-1} \\ &= A(1-\eta)(2-\eta) - B\eta(2-\eta) - C\eta(1-\eta) \\ &= 2A + (-3A-2B-C)\eta + (A+B+C)\eta^2\end{aligned}$$

This expression is quadratic so we need only find the roots to obtain the maximum. Using the negative root, which ensures a value in [0, 1], the estimate for η is then

$$\hat{\eta} = \frac{-b_\eta - \sqrt{b_\eta^2 - 4a_\eta c_\eta}}{2a_\eta}$$

where $a_\eta = A + B + C$, $b_\eta = -3A - 2B - C$, and $c_\eta = 2A$.

3.2.3 Maximum a posteriori for ξ

A similar derivation for ξ gives

$$\hat{\xi} = \frac{-b_\xi - \sqrt{b_\xi^2 - 4a_\xi c_\xi}}{2a_\xi}$$

where $a_\xi = -D - E - F$, $b_\xi = -E + F$, and $c_\xi = D$.

It should be noted that such convenient closed forms for MAP estimates of strategies are very rare in the space of possible Hold'em style pokers. Even in Kuhn poker, we encounter difficulty when we try to apply the same approach to having P2 model P1 (see Sect. 3.3 for more details on this). In general, even slightly more complicated games make the estimation problem substantially more difficult. We will discuss this broader issue in greater detail in Sect. 9.

[6] This independence is due in the first place to the structure of the game itself, which does not couple these two parameters, and in the second place, to our prior, which assumes independence. A different prior could conceivably couple these two parameters.

3.3 P2 modelling P1

The MAP estimators for P2 modelling P1's parameters have no convenient closed form solution that we have been able to compute (and there may be none). We therefore use the approximate estimators described in Hoehn (2006). The exact formulation is a bit lengthy so we omit it here. In brief, the estimators are computed by identifying each information set corresponding to the application of a P1 parameter and then considering the corresponding information sets for P2's view of the game. In several of these, P2 does not always obtain information about P1's holdings, due to folding. In such cases, there are two possible states for P1's holdings, depending on which card they were dealt. Since the overall number of occasions where such an information set was encountered by P1 is known to P2, the estimators make the assumption that exactly half of those occasions correspond to each possible card. For example, if P2 holds the King k times and observes the opponent holding the Queen q times, the Jack j, and does not observe the opponent's card u times (note that $k = q + j + u$), then the opponent is assumed to have held unobserved Queens $k/2 - q$ times and unobserved Jacks $k/2 - j$ times. Using these assumed counts for the unobserved opponent holdings, MAP estimates are then computed independently for each parameter in each such situation. Having now multiple estimates for each parameter, they are combined by a weighted sum, with weights proportional to the number of data points observed by the corresponding estimator.

There are a few tricky details related to the handling of very small numbers of observations or observations inconsistent with a perfectly even dealing of cards (see Hoehn 2006, Sect. 3.2.2 and particularly Eq. 3.6 for details) but, broadly, this estimation procedure can be thought of as an average of independent estimators, with the strong assumption that the cards were dealt exactly according to the mean of the distribution over cards. As the number of hands played grows large, this approximation is expected to behave like the true MAP estimate. Experiments described in Hoehn (2006) used a similar approximation for P1 modelling P2 and compared the learning results with the true MAP estimate. These showed little difference between the two approaches and that they quickly converge to near identical results as the number of hands played increases. This provides some evidence that the approximation has value, albeit in a simpler context. We therefore present the interested reader with results using this approximation for P2 modelling P1 in the supplemental, online appendix, with a caveat regarding any conclusions that may be drawn from them.

3.4 Nash equilibria and exploration

In two player, zero-sum games, Nash equilibrium strategies guarantee a certain minimum value regardless of the opponent's strategy. As such, they are "safe" strategies in the sense that they minimize the worst-case loss. As mentioned above, the Nash strategies for P1 in Kuhn poker guarantee an expected value of $-1/18$,[7] and thus can only guarantee a loss. Against a given P2 strategy, some non-Nash P1 strategy could be better or worse than Nash. There are no guarantees. So, even though any Nash strategy is a losing proposition for P1, it may be better than the alternatives against an unknown opponent. It therefore makes sense to consider adopting a Nash strategy until an opponent model can be learned. Then the best means of exploiting that model can be tried.

[7] Recall that all strategies we consider are *essential*, so the guarantee is for an exact value rather than just a minimum.

In many games, and more particularly in the case of Kuhn poker's P1, there are multiple Nash strategies. We explore the possibility that some of these strategies allow for faster learning of an opponent model than others. The existence of such strategies means that even though they guarantee identical game-theoretic values, some strategies may be more useful than others against exploitable opponents.

Another interesting approach is to mix essential strategies so as to maximize exploration, regardless of the cost. For this, we employ a "balanced" exploration strategy, ($\alpha = 1, \beta = 1, \gamma = 0.5$), that forces as many showdowns as possible and equally explores P2's two parameters. This exploratory strategy has a minimum winning rate of -0.417, which is more than 7 times worse than a Nash strategy. Therefore, the information it gains can come at a substantial cost. Of course, it is possible that against a particular opponent, the balanced strategy is a good response and exploits that opponent effectively. The experiments presented in Sect. 5.3 show how this tradeoff between safety and exploration plays out in practice.

Finally, we will note that one might choose to play non-essential strategies in order to obtain more information.[8] We have explored this possibility briefly, omitting the results here. While one can gain some information by playing a particular dominated strategy in Kuhn poker (P2 passing when holding the King in round 2), experiments showed that the cost of playing this strategy vs. the information gained was a poor tradeoff (Hoehn 2006). However, in other forms of poker one might gain useful information by playing dominated strategies (e.g., calling an opponent to a showdown in order to observe their holdings in a situation where the only sensible choice from an immediate winnings perspective is to fold).

4 Strategy learning

The other learning approach we examine here is what we will call *strategy learning*. We can view a strategy as an *expert* that recommends how to play the hand. In experts-based learning, a set of experts is used, each making its recommendation and the final decision being made by a *master* program. A *score* is kept for each expert, tracking how good its past decisions would have been. The master program selects its actions by considering the scores of the various experts. Favor is given to the experts in proportion to their past success. There are many specific variations on this basic approach, intended to handle the different features of specific problem domains.

Taking the six pure strategies shown in Fig. 2 plus a single Nash strategy ($\alpha = \frac{1}{6}, \beta = \frac{1}{2}, \gamma = \frac{1}{2}$) as our experts, we use a variation of the Exp4 algorithm (Auer et al. 1995) to control play by these experts. Exp4 is a *bounded-regret* algorithm designed for partially-observable games, based on earlier work using experts for perfect information games (Freund and Schapire 1996, 1999). It mixes exploration and exploitation in an online fashion to ensure that it cannot be trapped by a deceptive opponent. Exp4 has two parameters, a learning rate $\rho > 0$ and an exploration rate $0 \le \psi \le 1$ ($\psi = 1$ is uniform random exploration with no online exploitation).

As formulated by Auer et al., Exp4 only handles games with a single decision. However, for Kuhn poker, a sequence of decisions is sometimes necessary. This slightly complicates matters because a strategy specifies a distribution over actions in every possible situation. For any single observed hand, however, we will only observe a subset of the possible decision points. The exact subset depends on chance events (i.e., cards being dealt) and on the

[8] Note that, in general, dominated strategies are a subset of the non-essential strategies. In Kuhn poker specifically, all non-essential strategies are dominated.

opponent's actions. Therefore, two strategies that give differing recommendations in unobserved parts of the game might agree on the set of actions taken during this particular hand. Since either strategy could have produced the observed actions, it makes sense to award each expert some score, proportional to the probability with which they would have behaved as observed. We call this algorithm *sequential Exp4* (see Algorithm 1 for details). A closely related algorithm has been analyzed in Zinkevich (2004).

Algorithm 1 Sequential Exp4

```
1. Given $K$ strategies (experts), $\sigma_1 \cdots \sigma_K$, initialize the *scores* for each strategy to zero: $s_i = 0, 1 \le i \le K$
2. For $t = 1, 2, \ldots$ until the match ends:
   (a) Let the probability of playing the $i$th strategy for hand $t$ be
       $$p_i(t) = (1 - \psi)\frac{(1+\rho)^{s_i(t)}}{\sum_{j=1}^{K}(1+\rho)^{s_j(t)}} + \frac{\psi}{K}$$
   (b) Randomly select a strategy $\sigma_z$ from the set of $K$ experts with probability proportional to the $p_i$.
   (c) Play according to $\sigma_z$.
   (d) Observe the resulting sequence of actions $\mathbf{a}$ and the hand's winnings $w$ (scaled so that $w \in [0, 1]$).
   (e) Compute the probability for each strategy of generating the observed sequence of $d$ actions, $q_i(t) = P(\mathbf{a}|\sigma_i) = \prod_{j=1}^{d} P(a_j|\sigma_i)$
   (f) Compute new scores
       $$s_i(t+1) = s_i(t) + q_i(t)\frac{w}{\sum_{j=1}^{K} p_j(t)q_j(t)}, \quad 1 \le i \le K$$
```

Exp4 makes very weak assumptions regarding the opponent so that its guarantees apply very broadly. In particular, it assumes a non-stationary opponent that can decide the payoffs in the game at every round. This is a much more powerful opponent than our assumptions dictate (a stationary opponent and fixed payoffs). Along with updating all agreeing experts, a further modification was made to the basic algorithm in order to improve its performance in our particular setting.

A simple improvement, intended to mitigate the effects of small sample sizes, is to replace the single score (s_i) for each strategy with multiple scores, depending on the card they hold. We also keep a count of how many times each card has been held. So, instead of just s_i, we have per-card scores $s_{i,J}$, $s_{i,Q}$, and $s_{i,K}$, and card counters $c_{i,J}$, $c_{i,Q}$, and $c_{i,K}$. We then update the score specific to the card held during the hand and increment the corresponding counter. We now compute the expert scores for Algorithm 1's probabilistic selection as follows: $s_i = \frac{1}{3}s_{i,J}/c_{i,J} + \frac{1}{3}s_{i,Q}/c_{i,Q} + \frac{1}{3}s_{i,K}/c_{i,K}$. This avoids erratic behavior if one card shows up disproportionately often by chance (e.g. the King 10 times and the Jack only once). Naturally, such effects vanish as the number of hands grows large but we are specifically concerned with short-term behavior. We are simply using the sum of estimated expectations instead of estimating the expectation of a sum, in order to reduce variance.

In all experiments reported here, $\rho = 1$ and $\psi = 0.75$, values determined by experimentation to give good results. Recall that we are attempting to find out how well it is *possible* to do, so this parameter tuning is consistent with our objectives.

5 Experimental results

We conducted a large set of experiments using both learning methods to answer various questions. In particular, we are interested in how quickly learning methods can achieve better than Nash equilibrium (i.e., winning rate $\geq -1/18$) or breakeven (i.e., winning rate ≥ 0) results for P1, assuming the opponent is exploitable to that extent. In the former case, P1 is successfully exploiting an opponent to reduce losses, while in the latter case P1 can actually win if enough hands are played. However, we aim to play well in short matches, making asymptotic winning rates of limited interest. Most of our results focus on the total winnings over a small number of hands (typically 200, although other numbers are considered).

In our experiments, P1 plays an exploratory strategy up to hand t, learning during this period. P1 then stops learning and switches strategies to exploit the opponent. In parameter learning, unless specified otherwise, the "balanced" exploratory strategy described earlier is used throughout the first phase. In the second phase, a best response is computed to the estimated opponent strategy and that is "played" (in practice, having both strategies, we compute the exact expected winning rate using Eq. 1). For strategy learning, sequential Exp4 is run in the first phase, attempting some exploitation as it explores, since it is an online algorithm. In the second phase, the highest rated expert plays the remaining hands.

We are chiefly interested in the number of hands after which it is effective to switch from exploration to exploitation. Our results are expressed in two kinds of plot. The first kind is a *payoff rate plot*—a plot of the expected payoff rate versus the number of hands before switching, showing the rate at which P1 will win **after** switching to exploitation. Such plots serve two purposes; they show the long-term effectiveness of the learned model, and also how rapidly the learner converges to maximal exploitation.

The second kind of plot, a *total winnings plot,* is more germane to our goals. It shows the expected total winnings versus the number of hands before switching, where the player plays a fixed total number of hands (e.g. 200). This is a more realistic view of the problem because it allows us to answer questions such as: if P1 switches at hand 50, will the price paid for exploring be offset by the benefit of exploitation? It is important to be clear that the x-axis of both kinds of plot refers to the number of hands before switching to exploitation.

All experiments were run against all six P2 opponents selected from the six regions in Fig. 2. Results were also run for randomly generated opponents, all with the same maximum exploitability. In these *fixed exploitability* experiments, a maximum exploitation rate, τ, is fixed for the experiment and a new opponent is randomly generated every trial such that a best response for each opponent wins at rate τ. This allow us to average results across a large set of opponent strategies without introducing variance due to different levels of exploitability.

Only representative results are shown here due to space constraints. The remaining results are available in the supplemental online appendix. The supplemental online appendix also contains results for P2 modelling P1 (see Sect. 3.3 for related comments). Results were averaged over 30,000 trials for both parameter learning and strategy learning. The single opponent in the figures that follow is O_6, unless otherwise specified, and is typical of the results obtained for the six opponents. Similarly, results are for parameter learning unless otherwise specified, and consistent results were found for strategy learning, albeit with overall lower performance.

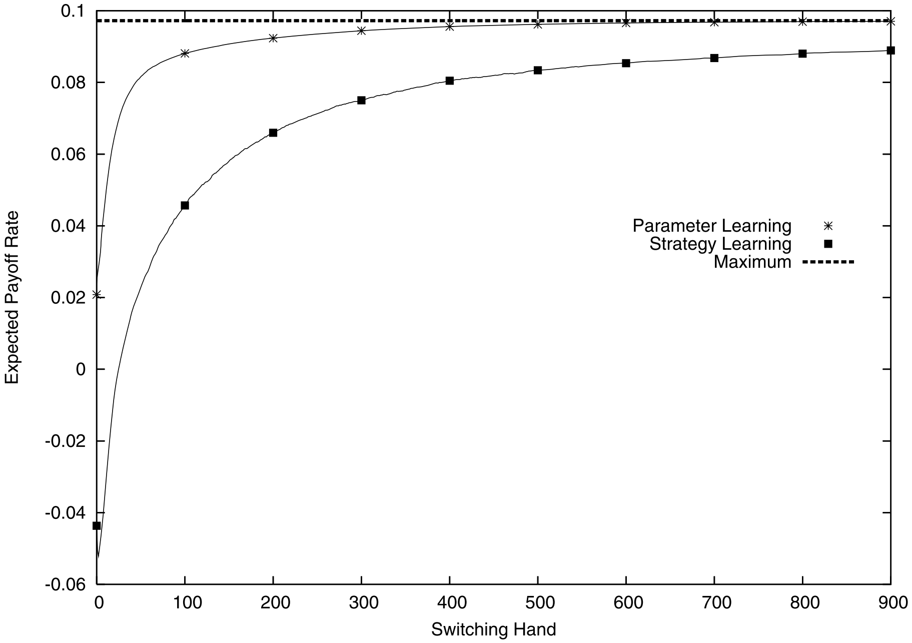

Fig. 4 Convergence Rate Study: Expected payoff rate vs. switching hand for parameter and strategy learning against O_6

5.1 Convergence rate study

This study addresses the question of how quickly the two learning approaches converge to optimal exploitation of the opponent (i.e., the true best response). Figure 4 shows the expected payoff rate plot of the two learning methods against a single opponent. The straight line near the top shows the maximum exploitation rate for this opponent (i.e. the value of a best response to P2's strategy). It takes 400 hands for parameter learning to almost converge to the maximum and strategy learning does not converge within 900 hands. Results for other opponents are generally similar or worse (O_2 is a notable exception), requiring several hundred hands for near-convergence. This shows that, even in this tiny game against a stationary opponent, one cannot expect to achieve maximal exploitation in a small number of hands, at least with these standard methods and probably most related variations. The possibility of maximal exploitation in larger games can reasonably be ruled out on this basis and we must adopt more modest goals for opponent modelling. Figure 5 shows the same study, but averaged over random opponents with fixed exploitability $\tau = 0.055$. The results here are very similar to the single opponent, but we also show the results for the Nash exploration parameter learning ($\gamma = 1$).

5.2 Parameter learning prior study

In any Bayesian parameter estimation approach, the choice of prior is clearly important. Here we present a comparison of various priors against a single opponent ($O_6 = (0.17, 0.2)$). Expected total winnings are shown for five priors. Each of these is characterized by two *Beta* distributions and we note the most probable parameter setting under that prior in parentheses.

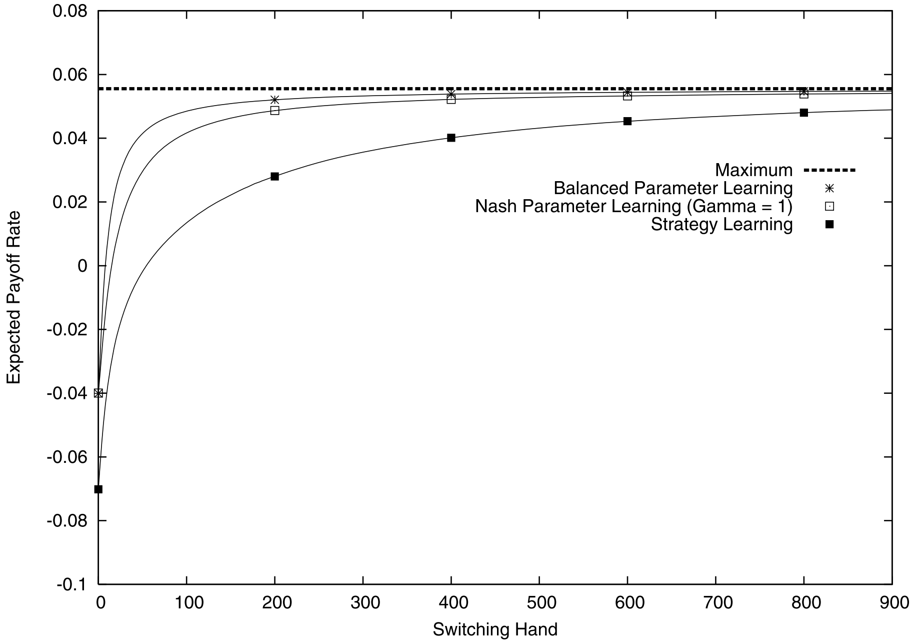

Fig. 5 Convergence Rate Study: Expected payoff rate vs. switching hand for parameter and strategy learning against randomly generated opponents exploitable at a rate $\tau = 0.055$

- a weak, default prior of $[Beta(1, 1), Beta(1, 1)]$ (0.5, 0.5)
- a weak, bad prior of $[Beta(1.4, 0.6), Beta(1, 1)]$ (0.7, 0.5)
- a strong, default prior of $[Beta(10, 10), Beta(10, 10)]$ (0.5, 0.5)
- a strong, bad prior of $[Beta(14, 6), Beta(10, 10)]$ (0.7, 0.5)
- an *uninformed* prior (no prior at all)

The weak priors assume 2 fictitious observations of each parameter and the strong priors assume 20 observations each. The "bad" prior is so called because it is quite distant from the real strategy of this opponent. The uninformed prior has no fictitious observations; MAP estimation with such a prior is known as *maximum likelihood* estimation. Figure 6 shows that the weak and uninformed priors clearly do better than the strong, allowing for fast adaptation to the correct opponent model. The strong priors perform much more poorly, especially the strong bad prior. It is also worth noting that after 50 hands, the bad weak prior is scarcely inferior to the default weak prior, so our poor early choice does not hurt us much. While very close on O_6, the weak default prior and uninformed prior each outperform the other on some of the opponents, making no clear choice between them obvious.

5.3 Nash exploration study

Figure 7 shows the expected total winnings for parameter learning when various Nash strategies are played by the learner during the learning phase. The strategies with larger γ values are typically stronger, more effectively exploring the opponent's strategy during the learning phase. This advantage is true across almost all opponents we tried, with the behavior of $\gamma = 0$ a noteworthy exception in that on some opponents it is the best performer, while

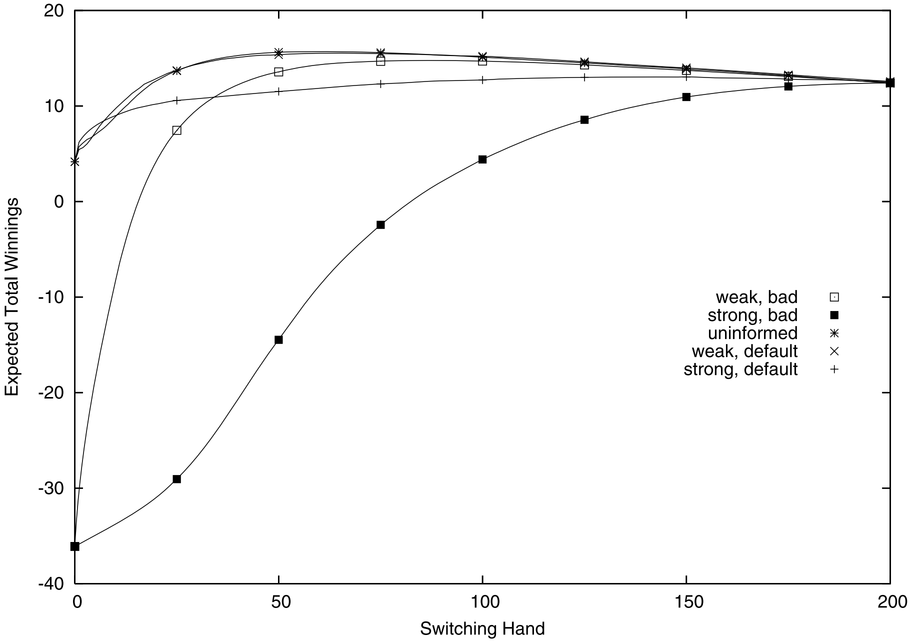

Fig. 6 Prior Study: Four different priors for parameter learning against O_6

on others it is the worst. This is because the Nash strategy with $\gamma = 0$ always passes while holding the King or the Jack and therefore never makes any observations related to the opponent's η parameter.

5.4 Learning method comparison

Figure 8 directly compares strategy and parameter learning (both balanced and Nash exploration ($\gamma = 0.75$)), all against a single opponent. Balanced parameter learning outperforms strategy learning substantially for this opponent. Over all opponents, either the balanced or the Nash parameter learner is the best, and strategy learning is worst in most cases (a notable exception is strategy learning against opponent O_3 and for O_1 its results are not far from the winner, Nash parameter learning). Figure 9 shows the same study averaged over random opponents with maximum exploitability $\tau = 0.055$. Here we see some differences. Most notably, the Nash exploration is a much safer choice for late switches. Balanced exploration gives best results with early switches but pays a heavy penalty for switching late. Switching at around 50 hands is a safe choice for either of the parameter learners. Strategy learning remains an overall loser, but is more robust to late switching than balanced exploration.

5.5 Game length study

This study is provided to show that our total winnings results are robust to games of varying length. While most of our results are presented for games of 200 hands, it is only natural to question whether different numbers of hands would have different optimal switching points. Figure 10 shows overlaid total winnings plots for 50, 100, 200, and 400 hands using parameter learning. The lines are separated because the possible total winnings is different for

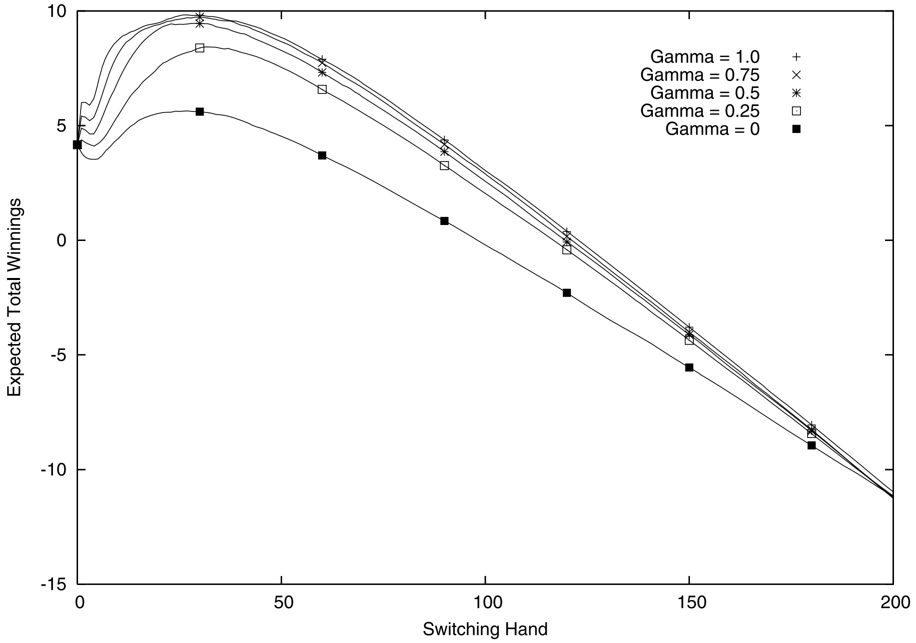

Fig. 7 Nash Exploration Study: Expected total winnings vs. switching hand for parameter learning using various Nash strategies for exploration against O_6

differing numbers of hands. The important observation to make is that the highest value regions of these curves are fairly broad, indicating that switching times are flexible. Moreover, the peak regions of the various curves overlap substantially. Thus, switching at hand 50 is a reasonable choice for all of these game lengths, offering close to the best possible total winnings in all cases. This means that even if we are unsure, *a priori*, of the number of hands to be played, we can be confident in our choice of switching time, at least over the range of 50 to 400. Moreover, this result is robust across our range of opponents. A switch at hand 50 works acceptably in all cases.

5.6 Proportion above Nash study

If a Nash strategy is known, then any deviation from that strategy must be justified by the hope that it successfully exploits the opponent. Otherwise, one is better off sticking to the "safe" strategy. While the opponent modelling algorithms have been seen to do well on average, how often is the attempt to exploit a losing proposition? We attempt to answer this question by plotting the proportion of trials in which the opponent modeller's total winnings equal or exceed the expected total winnings for playing a Nash strategy, versus the switching hand. This proportion is the frequency with which the attempt to exploit at least did not hurt us, and possibly was beneficial. It gives some notion of how damaging the variance can be. Figure 11 shows this experiment against a single opponent, O_6. The results show that, around the 50 hand switching point, over 80% of trials of balanced parameter learning and only slightly less than 80% of trials of Nash exploration parameter learning achieve at least the expected winnings of a Nash strategy. Strategy learning fares the worst but still performs

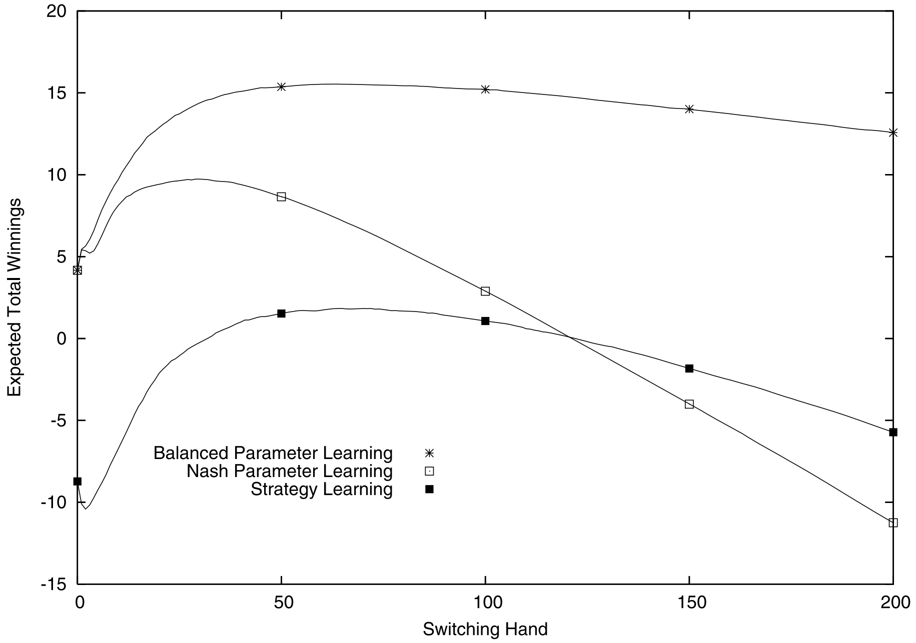

Fig. 8 Learning Method Comparison: Expected total winnings vs. switching hand for both parameter learning and strategy learning against O_6

at least as well as the expected Nash strategy winnings in almost 70% of the trials at its best switching point.

6 Non-monotonic learning curves

Most of the payoff rate curves in our parameter-learning studies are like Fig. 5, with the expected payoff rate of the estimated model increasing monotonically as more hands are played. This makes sense intuitively; even though chance events in the deal of the cards or the opponent's stochastic play can sometimes mislead, on average each hand gives representative information about the opponent, which accumulates to produce increasingly accurate estimates of the opponent's strategic parameters.

However, some payoff curves in our studies exhibit a different, non-monotone behavior. For example, Fig. 12 shows three convergence rate curves for P1 modelling P2 when P2 is playing the strategy $O_2 = (\eta, \xi) = (0.75, 0.8)$, P1 is using the Balanced Explore strategy for exploration and P1's initial estimate of P2's parameters is $(\eta, \xi) = (0.5, 0.5)$. The different curves result from P1 having different strengths given to this initial estimate, with "weak" and "strong" being defined exactly as in the Prior study in Sect. 5.2. The uninformed curve is based on P1 abandoning its initial estimate of a parameter as soon as it has any observed data on which to base its estimate. For example, when P1, with the uninformed prior, first sees P2 bet with the Jack, it will immediately change its estimate of ξ to be 1.0. As can be seen in Fig. 12, the payoff rate curves for the "weak" and uninformed priors are not monotonically increasing; they decrease very sharply during the first few hands of play and

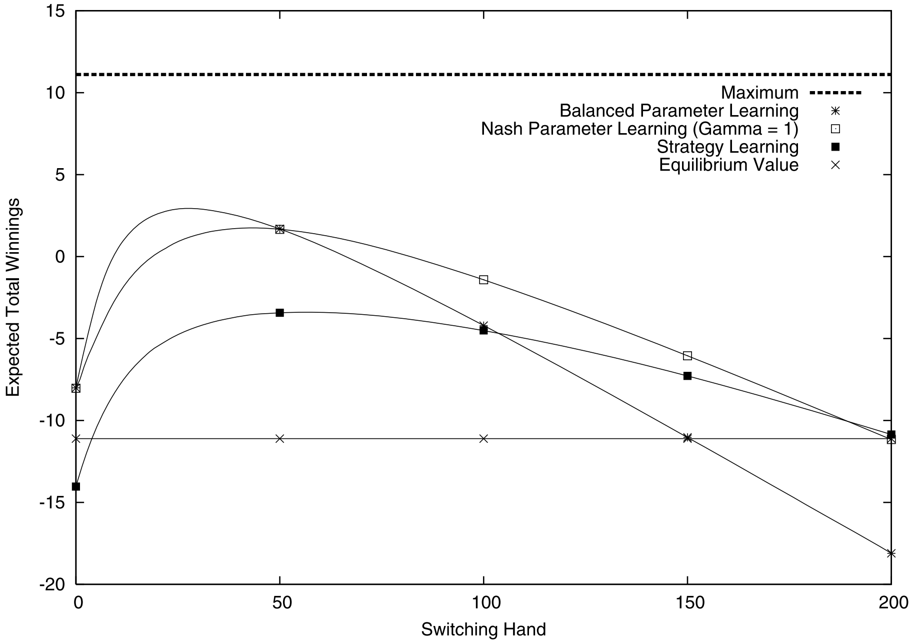

Fig. 9 Learning Method Comparison: Expected total winnings vs. switching hand for both parameter learning and strategy learning against randomly generated opponents exploitable at a rate $\tau = 0.055$

only become monotonically increasing after roughly 10 hands. These curves are averages over 30,000 trials, so this effect is systematic.

The explanation of this phenomenon is as follows. Although the default values for η and ξ are not especially close to their actual values, in P2's strategy space they are on the boundary of regions S2 and S3 (see diagram 2), which means P1's best response given these default values is almost the best possible response to O_2. This can be seen in Fig. 12 by how close the expected payoff is at hand 0 to the maximum possible expected payoff. As long as the data gathered during play keeps P1's estimates in regions S2 and S3, its expected payoff will remain at or above the initial value. However, if the first few hands of play are not representative of O_2's play (e.g. O_2 does not bet with the Jack even though it has a probability of 0.8 of doing so) P1's estimate will move out of regions S2 and S3 and its expected payoff will plummet from roughly $+0.09$ to less than -0.1. Figure 13 shows the percentage of trials on which this happened. After playing 9 hands, P1's model had an expected payoff of -0.1 or less on almost 8% of the trials when the default strength was "weak" and on more than 25% of the trials when the prior was uninformed. From hand 10 onwards (hand 5 for uninformed) this percentage decreases monotonically, causing the expected payoff rate curve to increase monotonically.

This phenomenon is important because it reflects a fundamental limit on how quickly an adaptive system can acquire a trustworthy model of its opponent. If the strength of the default setting is sufficiently great to avoid this non-monotonicity when the default produces a good best response, it will be detrimental to learning when the default is not good, as shown in Fig. 6. Thus, whatever strength is assigned to the default, it will take roughly 15 hands of play to be sure the learned model is not badly wrong.

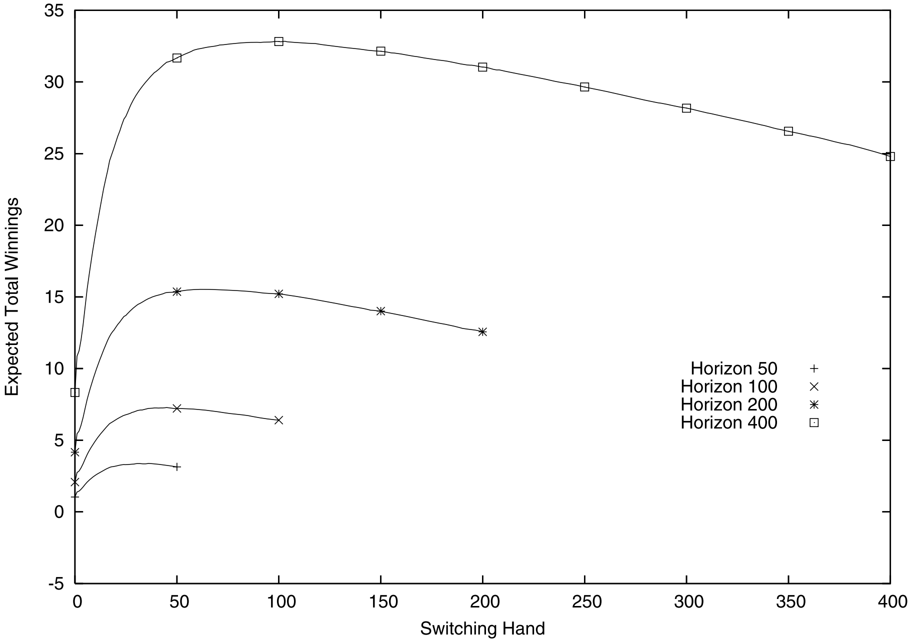

Fig. 10 Game Length Study: Expected total winnings vs. switching hand for game lengths of 50, 100, 200, and 400 hands of parameter learning against O_6

7 Learning against a non-stationary opponent

One issue that has not been explored so far in this paper is the modelling of non-stationary opponents, a complex issue for which much research remains to be done. In this section we present an initial study in which each player is modelling the other. The main purpose of this experiment is to highlight the complexities of this situation rather than offer a comprehensive treatment.

In this experiment P1 is a parameter learner (using either the Balanced Exploration data-collection strategy or a Nash strategy ($\gamma = 0.75$)) and P2 is a strategy learner (ComponentAverageExp3 with 5 experts, described below). An H-hand match is played, with P1 switching from exploration to exploitation at hand t (and not changing his model or counter-strategy from that point on), and P2 using the ComponentAverageExp3 method to select and re-weight its strategies continually throughout the match. P2 is therefore non-stationary, because the probability with which it selects each expert strategy changes from hand to hand to reflect the success of the strategy against P1. We expect this ability to continually adapt will give P2 a great advantage over the simple "explore-then-exploit" P1.

P2 uses 5 experts $(\eta, \xi) = \{(1/3, 1/3), (0, 0), (0, 1), (1, 0), (1, 1)\}$, and the parameter settings $\rho = 1$ and $\psi = 0.30$. Because the experts initially have equal weight, P2 will initially appear to be playing the mixed strategy $(\eta, \xi) = (0.47, 0.47)$, the average η and ξ values of its experts. Because P2 uniformly chooses between its strategies 30% of the time ($\psi = 0.30$), P2's effective η and ξ values cannot be less than 0.141 or greater than 0.841. This range of values allow P2 to play strategies in any region of the P2 strategy space, and to heavily exploit P1's non-Nash exploration strategy and P1's play after it switches to playing a best response to its model of P2.

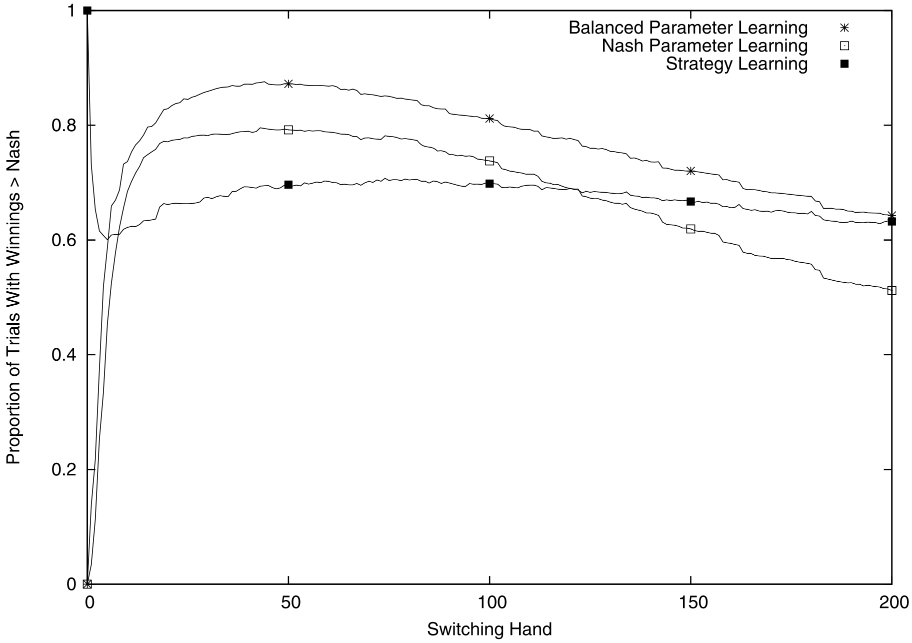

Fig. 11 Proportion Above Nash Strategy Study: Proportions of trials with winnings higher than Nash vs. switching hand for parameter and strategy learning against O_6

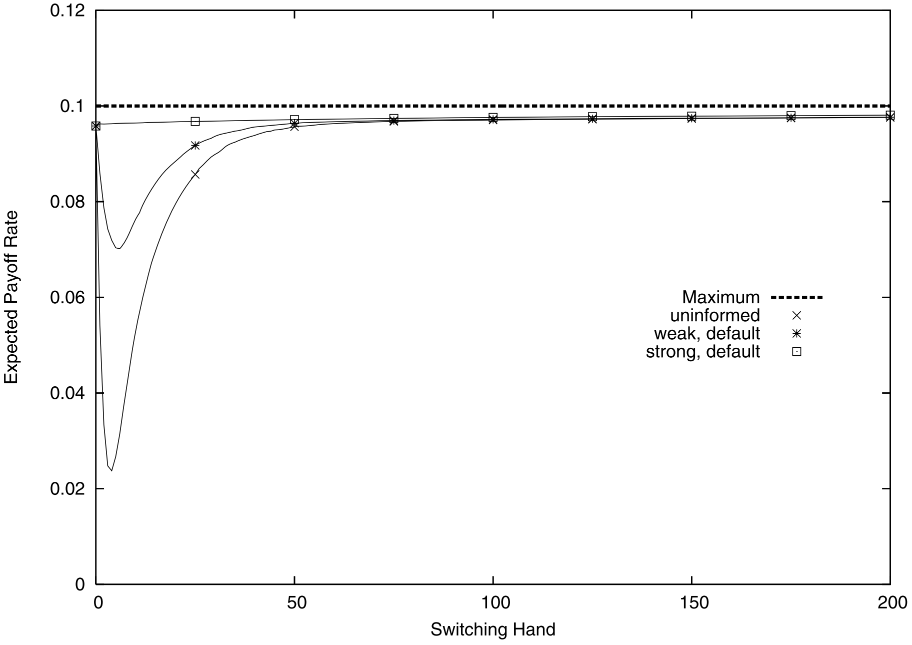

Fig. 12 Expected Payoff Rate for Balanced Explore (P1) versus O_2 for different strengths of priors

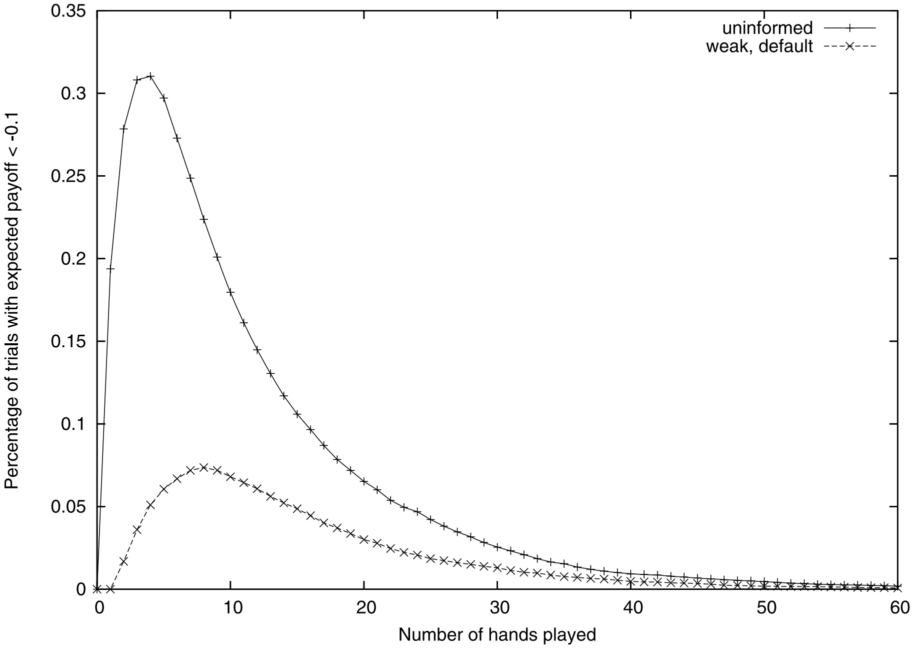

Fig. 13 Percentage of trials on which P1 had a badly wrong model of O_2

Figure 14 shows the total expected winnings for P1 over a H = 200 hand match as a function of the time at which P1 switches from the data gathering phase to the exploitation phase. The curve marked with the asterisks shows P1 using the "Balanced" exploration strategy while the curve marked with the squares shows P1 using a Nash data-gathering strategy. Each point is an average of 30,000 200-hand matches. The dashed horizontal line shows the winnings of a static Nash strategy over a 200-hand match.

P1's initial parameter estimates, $(\eta, \xi) = (0.5, 0.5)$, are almost perfectly correct for P2's initially uniformly weighted experts. If P2 were a static opponent, P1 would do very well to switch at time 0; it would have an expected payoff rate of -0.011, five times better than the payoff rate of the static Nash Strategy. But P2 is not static, and if P1 switches at time 0, P1 will be playing a fixed strategy throughout the match, giving P2 all 200 hands to shift weight towards, and play, the expert that is best against this P1 strategy. As a consequence, if P1 switches at time 0 its total winnings are somewhat worse than playing a static Nash strategy.

The curves exhibit the 3-phase form that is often seen when P1 has good initial estimates of stationary opponents. In the first phase (switching hand in the approximate range 1–15 for Balanced Exploration, 1–20 for Nash), total winnings drops sharply. This is caused by the randomness of the cards dealt and the stochastic play by both P1 and P2, which result in there being a significant number of short hand sequences that mislead P1 into choosing to play strategies S4 or S5, which will be highly exploited by P2 without P2 having to change its initial expert weightings.

In the second phase (switching hand in the range 16–45 for Balanced, 21–150 for Nash), enough hands have been played that P1 is reliably away from the "disaster zone" that caused the steep initial decline. P1's total winnings improve steadily through this phase.

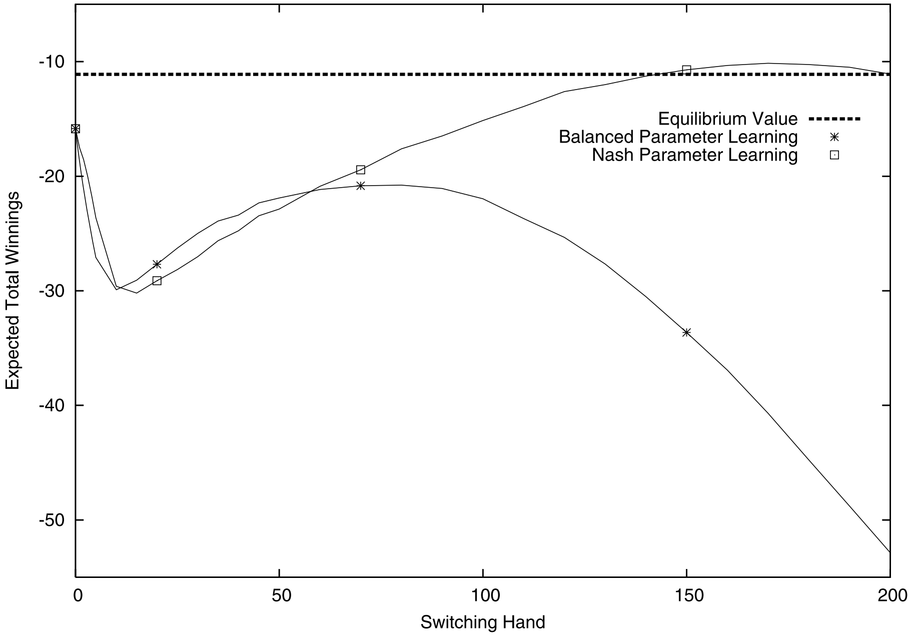

Fig. 14 Priors = (0.5, 0.5)

Against a stationary opponent, the second phase ends once P1 has learned a very good (if not quite optimal) model of P2. Further exploration is detrimental to total winnings once the incremental improvement in the model gained by further exploration is outweighed by the reduction in the number of trials in which the learned model can be exploited, the classic exploration-exploitation tradeoff.

Against a non-stationary opponent, the second phase ends for much more complex reasons. First of all, there is pressure for P1 to switch from exploration to exploitation simply because P2 is modelling P1 while P1 explores. If P1 explores for too long, P2 will learn, and play, a best response to P1's exploration strategy and P1 will be paying an inordinately large price for the information it is gathering. This is why the second phase for Balanced exploration is so much shorter than the second phase for Nash exploration: the former strategy is easily identifiable and highly exploitable, whereas the latter is hard for P2 to learn, because of its high stochasticity, and minimally exploitable. On the other hand, there are also reasons for P1 to extend its second phase against a non-stationary opponent. The longer P1 continues to explore the more strongly P2 will believe P1 is playing according to the exploration strategy and the slower P2 will be to respond to P1's switch to exploitation; in addition, P2 will have less time to exploit P1's learned model. The interplay between all these factors determines the optimal switching point for P1.

8 Related work

Poker has received increasing attention in recent years, with a recent emphasis on opponent modelling. For large scale games, both frequentist (Billings et al. 2004) and Bayesian (Korb and Nicholson 1999) approaches have been tried, but each of these has omitted some aspect of the full game in their model. A recent crop of results has appeared for small scale games attempting, as in this paper, to approach the problem with small models that are easier to analyze. Powers and Shoham examine the problem of computing a best response given only samples from an opponent's strategy, exploring both oblivious opponents and opponents that are aware of the information they have revealed (Powers and Shoham 2003). They also present criteria for learning in games and an algorithm achieving those criteria in two player, perfect information games against a stationary opponent (Powers and Shoham 2005).[9] Geoff Gordon has examined a class of no-regret algorithms on a game of one-card poker (Gordon 2005). Finally, Poland and Hutter have conducted a study similar to ours in spirit, empirically evaluating a probabilistic modelling approach and an experts-based approach on 2 × 2 matrix games rather than poker (Poland and Hutter 2005).

9 Scaling to larger games

The study of a small game like Kuhn poker allows exact comparisons to theoretical ideals, a thorough study by empirical means, and detailed analysis of results by hand. However, the question naturally arises whether the opponent modelling approaches discussed here will behave similarly on larger games, including real-world poker games. In particular, are large-scale versions of these algorithms likely to achieve the rapid learning required? Game theory research like Kuhn's is typically limited to small games for which analytical methods can be applied. At the other end of the spectrum, research on real-world games like chess and poker typically attacks the full version of the games with approximate algorithms and empirical studies.

There are two obvious ways in which Kuhn poker can be scaled up,

1. broadening: adding cards to the deck, thereby increasing branching factors, and,
2. deepening: adding more rounds of betting or allowing more bets per round, thereby increasing depth.

In other Hold'em-style pokers that have been studied, including research games such as Leduc Hold'em (Southey et al. 2005) and Rhode Island Hold'em (Shi and Littman 2001), and real-world games like limit Texas Hold'em, both dimensions have been increased. Compared to Kuhn poker, Leduc Hold'em has six cards instead of three and two rounds of betting instead of one. A typical set of rules for limit Texas Hold'em uses the full deck of 52 cards and four betting rounds with as many as three or four bets allowed per round. Increasing either aspect of the game increases the size of the game tree and therefore the number of decision points (*information sets* in game theory parlance) for both players. To fully describe the strategy of a player, one must specify the probabilities of the actions at every information set. This set of parameters grows very quickly as the number of cards and rounds increases.

[9] The Powers and Shoham algorithm simplifies to something similar to our Nash exploration parameter learning in the case of two player, zero-sum games. It starts by using the BullyMixed strategy during an initial exploration phase. In two-player, zero-sum games, the BullyMixed strategy is equivalent to a Nash strategy. The exact path of the algorithm depends on the relationship between the variance in the game's value on each round, variance in the distribution of opponent actions, and a set of fixed parameters, but it is likely to switch from its Nash exploration to playing an ϵ-best response against an estimate of the opponent strategy. While their setting is perfect information games and they have no prior over opponent strategies, our approach of Nash exploration followed by playing best responses to a MAP estimate of the opponent based on imperfect observations is conceptually very similar.

The challenge to opponent modelling is immediate. In terms of direct parameter learning, we must estimate a large set of parameters with very little data. With increased deck size, the number of parameters within each information set grows, adding to the uncertainty of folded hands. In terms of the indirect, expert-based, strategy learning approach, the set of possible experts increases dramatically. We must evaluate the performance of a large set of experts based on only a small number of hands.

For short-term opponent modelling to have any hope, we must rely on the existence of some correlation between the decisions made in different parts of the game tree. This is not an unreasonable belief in many cases, as we would expect reasonable players to behave similarly in similar situations. A good player will mix their strategies in order to confuse opponents but there must still be some consistency in their play, dictated by the nature of the game (highly erratic or nonsensical play will not perform well).

Research has explored two ways of addressing this question of correlation. One is to take the game itself and identify "similar" situations, for some chosen definition of similarity. The game can then be simplified and strategies generated with respect to this new game. Another approach is to make assumptions about the nature of the strategies themselves. In our parameter learning, this corresponds to a prior over strategies, while in strategy learning it corresponds to the selection of experts.

9.1 Abstraction

The simplification of games is typically referred to as *abstraction* in the related research. Abstraction reduces the size of a game tree by merging nodes together and treating them as equivalent. By extension, abstraction is a means to reduce the number of information sets (distinguishable decision points) in the game by grouping their nodes together. Information sets can be grouped together into equivalence classes, essentially assuming that strategies can be well-modelled by assuming the players will use the same (possibly randomized) strategy at all information sets within a class. Each equivalence class becomes an information set in the new, abstracted game. Information sets in poker consist of all cards revealed to the player and all actions taken by all players so far. Clearly, these can be grouped together in arbitrary ways. However, most work on abstraction has looked at grouping information sets together based on cards rather than actions. In this case, distinct sequences of actions are treated as distinct, but differences amongst cards are partially ignored.

One common abstraction for cards is to compute the *all-in-equity* for a set of cards (Shi and Littman 2001; Billings et al. 2003; Southey et al. 2005). This number can be thought of as the proportion of the pot the player can expect to win given the cards they hold and imagining all possible future cards that might appear and all cards that might be held by the opponent. In two player games specifically, it is the probability of winning plus half the probability of a tie. The actions taken so far in the game are ignored by this metric, so it is clearly a gross simplification. Nevertheless, it does provide a convenient scalar estimate in the range of [0, 1] for the "strength" of the player's hand. This range can then be discretized into, for example, 10 intervals, effectively grouping cards into one of ten card equivalence classes. These card equivalence classes, combined with the action history, make up the information sets in the abstracted game.

One example of abstraction over actions is found in the VexBot program (Billings et al. 2004). VexBot is akin to maximum likelihood, estimating the opponent's strategy from frequency counts for actions and observed opponent cards. However, in the presence of very little data, estimates will be scattered amongst the huge number of information sets. To achieve some generalization, VexBot tracks estimates over several different abstractions that group together situations based on similarities in actions. For example, one such grouping considers a set of situations to be the same if the opponent made an equal number of bets/raises. This essentially ignores the player's own actions, and the specific order of the opponent's actions. VexBot then combines the estimates from the different abstractions to guide its play.

Finally, Gilpin and Sandholm have worked extensively on automatically generating abstractions of game trees with specific application to poker (Gilpin and Sandholm 2006, 2007a, 2007b; Gilpin et al. 2007). These methods even allow for the discovery of abstracted games with Nash equilibria that correspond directly to the equilibria in the original game. Approximate methods that do not preserve equilibria provide an avenue for even smaller abstractions.

9.2 Correlation in strategies

Abstraction is essentially a hard decision about correlation in the original game; under an abstraction, two information sets from the original game are either completely correlated or completely independent. This has the attractive property that all subsequent reasoning is performed with respect to a smaller game. However, it is clearly a very strong assumption about the nature of opponents. Another, smoother approach is to recognize that strategies at information sets may be more weakly correlated. For example, a player holding two tens might be expected to have similar behavior when holding a pair of nines, only somewhat less cautious (e.g., more likely to bet).

In Bayesian parameter learning, such correlations can be captured in the prior over strategies, giving higher probabilities to strategies that reflect correlations between information sets supposed to be similar. The priors used in our work here on Kuhn poker do not capture any such correlation, since each parameter uses a *Beta* distribution as a prior. However, in the work described in Southey et al. (2005), an expert-designed prior was used. Characterized by ten parameters, the prior would generate strategies that correlate the behavior between betting rounds and over similar classes of cards. Thus, an overall tendency toward aggressive betting would be reflected over a range of strong hands, and would be pursued round after round.

In strategy learning, as games grow larger, enumerating all the pure strategies for use as experts quickly becomes infeasible. It is therefore necessary to select a smaller set of strategies, or to work with classes of strategies for which effective algorithms can be found that consider large numbers. In the simplest case, experts can be sampled from a prior similar to that used by parameter learning. Another approach is to sample strategies from a prior, compute a best response for each sampled strategy, and use those as experts. More recently, the research in Johanson et al. (2007) explores the question of generating robust strategies for use in opponent modelling, dealing directly with large abstractions of Texas Hold'em.

9.3 Learning in larger games

The closed-form MAP estimates used here for parameter learning in Kuhn poker are not feasible for larger games. One alternative is to use the popular *expectation-maximization* (EM) algorithm (Dempster et al. 1977). In Southey et al. (2005), we instead used Monte Carlo sampling from the prior. Over a fixed sample, one can simply track the sampled strategy with highest posterior probability as an approximation to the true MAP strategy. While those experiments did not use the same two-phase, explore/exploit framework presented here, we can make some observations about the convergence of the posterior distribution.

In Southey et al. (2005), experiments on Leduc and abstracted Texas Hold'em were run over 200 hands against fixed opponents using 1000 strategies sampled from the prior. In each experiment, the sampled strategy with the highest posterior probability after 200 hands was recorded, which we will here denote $\hat{\sigma}_{\mathrm{MAP}}$. This means that $\hat{\sigma}_{\mathrm{MAP}}$ is the "best fit" to the opponent amongst the sampled strategies. For Leduc Hold'em, the relative posterior probability of $\hat{\sigma}_{\mathrm{MAP}}$ (i.e., the proportion of posterior probability attributed to it) was about 0.05 after 50 hands, 0.15 after 100 hands, 0.4 after 150 hands, and 0.78 after 200 hands. This shows that the eventual best fit strategy had substantial mass on it (much more than the uniform weighting initially on the samples), even after 50 hands. The results are still more striking in Texas Hold'em, where the relative posterior probability of $\hat{\sigma}_{\mathrm{MAP}}$ was about 0.12 after 50 hands, 0.3 after 100 hands, 0.6 after 150 hands, and 0.9 after 200 hands. This is an even more rapid convergence to the best fit. While this does not necessarily imply strong play, which is heavily influenced by the prior from which the sampled strategies are drawn, it does demonstrate fast learning. This leads us to believe that parameter learning methods, at least, will scale to offer some short-term benefit. Further study is required to determine exactly how much can be achieved with different priors.

Results are not available for strategy learning, but there are similarities to the parameter learning case. Strategy learning relies on a weaker signal to inform the choice of strategy, looking only at performance against the opponent and not at the specific actions taken by the opponent. However, there are several ways to reduce the variance in the estimates obtained by observing performance (e.g., attempting to separately account for randomness introduced by the cards rather than by a stochastic opponent). Strategy learning also has the advantage that it does not require that our set of strategies contain one that is similar to the opponent, but only that it contain one that is effective against the opponent. In larger games, where we can consider only a comparatively small subset of the possible strategies, this advantage may become important.

10 Conclusions

This work shows that learning to maximally exploit an opponent, even a stationary one in a game as small as Kuhn poker, is not generally feasible in a small number of hands. However, the learning methods explored are capable of showing positive results in as few as 50 hands, so that learning to exploit is typically better than adopting a pessimistic Nash strategy. Furthermore, this 50 hand switching point is robust to game length and opponent. Future work includes non-stationary opponents, a wider exploration of learning strategies, and larger games. Both approaches can scale up, provided the number of parameters or experts is kept small (abstraction can reduce parameters and small sets of experts can be carefully selected). Also, the exploration differences amongst equal-valued strategies (e.g., Nash in two player, zero-sum games) deserves more attention. It may be possible to more formally characterize the exploratory effectiveness of a strategy. We believe these results should encourage more opponent modelling research because, even though maximal exploitation is unlikely, fast opponent modelling may still yield significant benefits.

Acknowledgements Thanks to the Natural Sciences and Engineering Research Council of Canada and the Alberta Ingenuity Centre for Machine Learning for funding, and the University of Alberta poker group for their insights. Particular thanks to Martin Zinkevich and Michael Bowling for their insights.

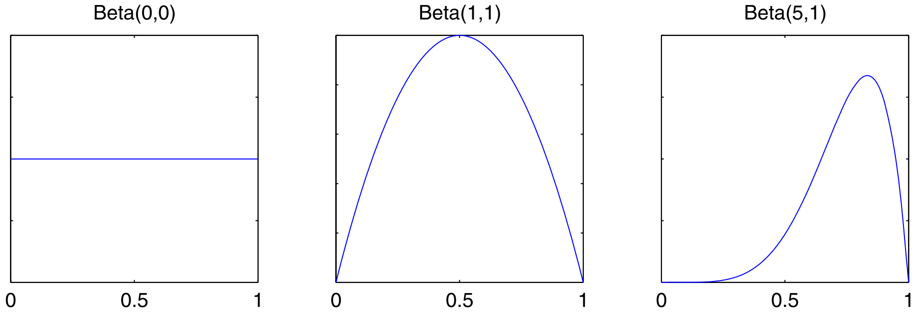

Fig. 15 Example *Beta* distributions.

Appendix: Beta distribution

A *Beta* distribution gives a probability distribution over a single probability (a value in [0, 1]). As such, it is a useful prior for single parameters in probabilistic models. A *Beta* distribution is characterized by two parameters, θ and ω. The probability assigned to a parameter value by a *Beta* distribution is

$$P_{Beta(\theta,\omega)}(x) = x^{\theta}(1-x)^{\omega}\frac{\Gamma(\theta+\omega)}{\Gamma(\theta)\Gamma(\omega)}$$

where the ratio of Gamma functions is simply a normalizing constant. A single probability is a distribution over two events. *Beta* distributions can be understood as "pretending" that we have observed several events in the past, and that we observed θ of one event and ω of the other. Figure 15 shows three examples of *Beta* distributions. Note how we can obtain the uniform distribution with *Beta*(0, 0) or distributions showing the impact of past "pretended" evidence.[10]

References

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (1995). Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proceedings of the 36th annual symposium on foundations of computer science* (pp. 322–331).

Billings, D., Burch, N., Davidson, A., Holte, R. C., Schaeffer, J., Schauenberg, T., & Szafron, D. (2003). Approximating game-theoretic optimal strategies for full-scale poker. In *Proceedings of 18th international joint conference on artificial intelligence (IJCAI-2003)* (pp. 661–675).

Billings, D., Davidson, A., Schauenberg, T., Burch, N., Bowling, M., Holte, R., Schaeffer, J., & Szafron, D. (2004). Game tree search with adaptation in stochastic imperfect information games. In *Computers and games'04* (pp. 21–34). Berlin: Springer.

Brown, G. W. (1951). Iterative solution of games by fictitious play. In T. C. Koopmans (Ed.), *Activity analysis of production and allocation* (pp. 374–376). New York: Wiley.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, *39*, 1–38.

Freund, Y., & Schapire, R. (1996). Game theory, on-line prediction and boosting. In *Proceedings of the 9th annual conference on computational learning theory (COLT-96)* (pp. 325–332).

[10] Some formulations of the *Beta* distribution offset the distribution's parameters by -1 (i.e., $P_{Beta(\theta,\omega)}(x) = x^{\theta-1}(1-x)^{\omega-1}\frac{\Gamma(\theta+\omega)}{\Gamma(\theta)\Gamma(\omega)}$). We choose to use the above version, which offers a more direct connection to the interpretation of "pretended" observations.

Freund, Y., & Schapire, R. E. (1999). Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, *29*, 79–103.

Fudenberg, D., & Levine, D. K. (1998). *The theory of learning in games*. Cambridge: MIT Press.

Gilpin, A., & Sandholm, T. (2005). Optimal Rhode Island Hold'em poker. Intelligent systems demonstration described. In *Proceedings of the 20th national conference on artificial intelligence (AAAI-05)* (pp. 1684–1685).

Gilpin, A., & Sandholm, T. (2006). A competitive Texas Hold'em poker player via automated abstraction and real-time equilibrium computation. In *Proceedings of the 21st national conference on artificial intelligence (AAAI-06)* (pp. 1007–1013).

Gilpin, A., & Sandholm, T. (2007a). Better automated abstraction techniques for imperfect information games with application to Texas Hold'em poker. In *Proceedings of the 6th international joint conference on autonomous agents and multiagent systems (AAMAS-07)* (p. 192).

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

Gilpin, A., Sandholm, T., & Sørensen, T. B. (2007). Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas Hold'em poker. In *Proceedings of the 22nd national conference on artificial intelligence (AAAI-07)*.

Gordon, G. (2005). *No-regret algorithms for structured prediction problems* (Draft version).

Hoehn, B. (2006). *The effectiveness of opponent modelling in a small imperfect information game*. Master's thesis, University of Alberta, Dept. of Computing Science.

Hoehn, B., Southey, F., Holte, R. C., & Bulitko, V. (2005). Effective short-term opponent exploitation in simplified poker. In *Proceedings of the 20th national conference on artificial intelligence (AAAI-2005)* (pp. 783–788).

Johanson, M., Zinkevich, M., & Bowling, M. (2007). Computing robust counter-strategies. In *Advances in neural information processing systems 20 (NIPS 2007)* (pp. 721–728).

Koller, D., & Pfeffer, A. (1997). Representations and solutions for game-theoretic problems. *Artificial Intelligence*, *94*(1), 167–215.

Korb, K., Nicholson, A., & Jitnah, N. (1999). Bayesian poker. In *Proceedings of the 15th conference on uncertainty in artificial intelligence (UAI-1999)* (pp. 343–350).

Kuhn, H. W. (1950). A simplified two-person poker. *Contributions to the Theory of Games*, *1*, 97–103.

Margineantu, D. D., & Dietterich, T. G. (2002). Improved class probability estimates from decision tree models. In D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, & B. Yu (Eds.), *Lecture notes in statistics: Vol. 171. Nonlinear estimation and classification* (pp. 169–184). New York: Springer.

Michie, D. (1966). Game-playing and game-learning automata. In *Advances in programming and non-numerical computation* (pp. 183–196). Elmsford: Pergamon. Chap. 8.

Poland, J., & Hutter, M. (2005). *Universal learning of repeated matrix games* (Tech. rep. IDSIA-18-05). IDSIA.

Powers, R., & Shoham, Y. (2003). *Computing best response strategies via sampling* (Tech. rep.). Stanford.

Powers, R., & Shoham, Y. (2005). New criteria and a new algorithm for learning in multi-agent systems. In L.K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems 17 (NIPS-2004)* (pp. 1089–1096).

Robinson, J. (1951). An iterative method of solving a game. *The Annals of Mathematics, 2nd Series*, *54*, 296–301.

Russell, S., & Norvig, P. (1995). *Artificial intelligence: a modern approach* (1st ed.). Upper Saddle River: Prentice Hall.

Shi, J., & Littman, M. (2001). Abstraction models for game theoretic poker. In *Computers and games (CG-2000)* (pp. 333–345).

Southey, F., Bowling, M., Larson, B., Piccione, C., Burch, N., Billings, D., & Rayner, C. (2005). Bayes' bluff: opponent modelling in poker. In *Proceedings of the 21st conference on uncertainty in artificial intelligence (UAI-2005)* (pp. 550–558).

von Neumann, J., & Morgenstern, O. (1947). *The theory of games and economic behavior* (2nd ed.). Princeton: Princeton University Press.

Zinkevich, M. (2004). *Theoretical guarantees for algorithms in multi-agent settings*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.

Zinkevich, M., Bowling, M., & Burch, N. (2007a). A new algorithm for generating equilibria in massive zero-sum games. In *Proceedings of the 22nd national conference on artificial intelligence (AAAI 2007)* (pp. 788–793).

Zinkevich, M., Bowling, M., Johanson, M., & Piccione, C. (2007b). Regret minimization in games with incomplete information. In *Advances in neural information processing systems 20 (NIPS 2007)* (pp. 721–728).