Active Sensing for Opponent Modeling in Poker

Adam Eck and Leen-Kiat Soh
Department of Computer Science and Engineering
University of Nebraska-Lincoln
256 Avery Hall, Lincoln, NE, 68588, USA
{aeck, lksoh}@cse.unl.edu

Abstract
One approach to designing an intelligent agent capable of winning competitive games such as Texas hold’em poker is to use opponent modeling to learn about an opponent’s behavior, then exploit that knowledge to maximize long term winnings. However, opponent modeling can suffer from several problems, including slow convergence due to a lack of a priori knowledge, noisy or dynamic opponent behavior, and possible reverse exploitation through the “get taught and exploited” problem. One potential solution to these problems is active sensing, whereby an agent actively decides how and when to gather information to revise its knowledge, rather than simply relying on whatever observations are produced during its routine actions. In this paper, we describe an active sensing agent for Texas hold’em poker introduced into the 2011 ACPC, which finished the best of known opponent modeling agents in the heads up limit competition. In an empirical evaluation, we demonstrate the benefits of active sensing, including improved winnings and less uncertainty. We also describe how active sensing can be extended to other types of opponent modeling agents, and identify key areas of improvement for our agent.

Introduction
One popular, valuable domain commonly used for artificial intelligence (AI) research is competitive game playing agents. This includes the card game Texas Hold’em poker (hereafter referred to simply as poker), which is the focus of AAAI’s Annual Computer Poker Competition (ACPC). Poker is a valuable application for AI due to the complex environment properties of this domain, including (1) partial observability and incomplete information, e.g., unknown opponent cards and strategies, (2) stochastic and dynamic behavior, e.g., random card dealings and changing opponent behavior, and (3) uncertainty, e.g. in the agent’s beliefs, the opponent’s actions, and potential winnings.

Within poker, one interesting subproblem common to other applications of AI is opponent modeling. For example, agents use information collected during games about their opponents’ behavior to predict future opponent actions (e.g., Billings et. al, 2002) or their chosen strategy (e.g., Southey et. al, 2005) in order to calculate the agent’s best response to the current game situation. With opponent modeling, an agent can exploit its knowledge of the opponent to potentially achieve higher winnings, in contrast to more conservative (but often successful) approaches, such as playing Nash equilibrium strategies (e.g., Zinkevich et. al, 2007) that maximize worst-case winnings against any opponent.

However, opponent modeling is a challenging problem for several reasons. First, the ground truth of opponent behavior is hidden and must be estimated based on observations. These observations might be noisy, especially if the opponent plays to deceive the agent (e.g., by “bluffing” where it pretends it is likely to win when instead it has poor cards). Further, opponent behavior might change over time, in which case older information is no longer relevant. Acting on an incorrect model can worsen the agent’s chances of winning, defeating the purpose of opponent modeling.

Moreover, an opponent who knows that an agent models its behavior can use this information to exploit the modeling agent. Specifically, an opponent modeling agent is susceptible to the “get taught and exploited” problem (Sandholm, 2007), where the opponent plays one strategy for a while, then switches strategies to one that exploits the expected behavior of an agent that modeled and counters the opponent’s earlier gameplay. For example, the opponent might play very conservatively at first and fold often, causing the agent to loosen up and place more risky bets even if it doesn’t expect to have the highest cards because it believes the opponent will eventually give up the hand. Then, the opponent switches to a more aggressive strategy, gambling more on hands it has a reasonable chance of winning to take more money from the now loose agent.

With human poker players, one approach to overcome these challenges to opponent modeling is to sacrifice some hands (and winnings) simply for the sake of gathering additional information in the hopes of building a more accurate opponent model. For example, a player might choose to stay in hands with small losses in the hopes of winning larger hands later in the game using newly acquired information. Further, a player might periodically play hands it is unlikely to win, both to gather more information and to throw off the opponent’s modeling of her behavior.

Within the AI community, such behavior is called active sensing (or active perception). In active sensing, an agent makes explicit decisions about its sensing in order to improve its information gathering (Weyns et. al, 2004), instead of relying on whatever information it might observe from actions taken for non-sensing purposes. Active sensing has been popularly applied to domains such as user
Within competitive game playing domains such as poker, controlling information gathering for opponent modeling can benefit from active sensing in several ways. First, similar to the behavior of human players, active sensing could potentially improve the quality of opponent modeling and overall gameplay. That is, by staying in additional hands for the sole purpose of gathering more information about an opponent’s strategy, the agent sacrifices some small losses to potentially better exploit the opponent later on and win more money. Further, active sensing can help overcome the “get taught and exploited” problem by choosing to perform additional sensing in response to any perceived inaccuracy of the agent’s opponent model. For example, if the agent observes that its opponent behavior deviates from its learned model, the agent can choose to gather additional information about the opponent to improve its model and reverse the exploitation of the agent.

In this paper, we describe a particular agent (named GBR) entered into the 2011 ACPC that utilized active sensing in order to improve its opponent modeling. Specifically, the agent modeled the relationship between opponent actions and their hand strength in order to better estimate its own likelihood of winning each particular hand. When the agent was more uncertain about its beliefs over opponent hand strength, it was more likely to stay in hands to collect more information to build better exploitable beliefs in the future. This agent placed the highest of the known opponent modeling agents in the two player heads-up limit competition, as well as 13th and 14th overall out of 19 teams in the most popular two-player limit competition under the bankroll and instant run-off measures, respectively (ACPC, 2011c).

In the following sections, we will first provide background on poker and active sensing. Next, we describe our agent’s methodology, including the opponent modeling and active sensing components. Then, we demonstrate the advantages of active sensing with an experiment against several baseline and dynamic agents. Next, we outline avenues of future work, including addressing several possible pitfalls and challenges observed with our approach, and extending active sensing to other opponent models. Finally, we conclude with a brief summary of the paper.

Background and Related Work

Texas Hold’em Poker

In Texas Hold’em poker, two or more players compete during hands, during which players bet money and try to increase the amount won with the hand. Each hand begins with each player dealt two private cards. Then, a round of

1 Based on the descriptions of agent behavior provided by the 2011 ACPC contestants (ACPC, 2011b)
One interesting area of research in poker has been to develop approaches to handle noisy, incorrect, incomplete information stored in an opponent model, especially while the model is learned online during a game of poker. After all, if an agent is using an opponent model to guide its play, its performance will only be as good as the opponent model used. Recent approaches generally rely on an equilibrium strategy as a fallback plan, either computing a response to an opponent which plays a mixture of the modeled behavior and an equilibrium strategy (Johanson and Bowling, 2009) or playing an equilibrium strategy while modeling the opponent, then exploiting the opponent model only after a specific number of hands of been played (Ganzfried and Sandholm, 2011).

In this paper, we present an opponent modeling approach that combines hand strength probabilities similar to early opponent modeling approaches (without enumerating all possible opponent hands, e.g., Billings et. al, 1998; 2002) and more popular recent Bayesian modeling (e.g., Baker and Cowling, 2007). Instead of relying on equilibrium strategies to overcome weaknesses in the opponent model, we instead aim to directly improve the quality of opponent models over time using active sensing to potentially better maximize agent winnings.

**Active Sensing**

To improve opponent modeling in poker, we turn to the field of active sensing. Active sensing is the process through which an agent actively manages its sensing behavior by choosing sensing actions to perform based on their observational value (e.g., accuracy, uncertainty reduction), rather than passively relying on whatever observations happen to be produced by the environment (Weyns et. al, 2004). Addressing agent sensing from this perspective enables the agent to explicitly consider the benefits and costs of sensing actions, allowing it to proactively aim to maximize its sensing performance, as opposed to reactively rely on (potentially) suboptimal observations. These decisions can be in response to missing information required for incoming tasks, to address deficiencies (e.g., low accuracy or confidence) found in the agent’s knowledge, or other needs for new information.

For example, Boutilier (2002) and Doshi and Roy (2008) both used active sensing to perform user preference elicitation where an agent chose sensing actions to best refine the agent’s beliefs about the user’s preferences in order to provide intelligent user support. Similarly, Williams and Young (2007) studied the problem of determining user goals in a dialog management system, again through prompting the user for information. Further, Guo (2003) considered the active classification problem where agents chose sensing actions to gather information used to classify some object in the environment. More recently, Spaan et. al (2010) considered how to coordinate active sensing between multiple agents. Specifically, they considered a system composed of fixed position video cameras and mobile robots responsible for monitoring a public space for events like fires or people moving in the area.

In this section, we propose an approach using active sensing to enhance opponent modeling in poker, summarized in Figure 1. First, we describe how the agent models its opponent and its own hand using hand strength estimates. Second, we describe how we use active sensing to improve opponent modeling by gathering additional information when it is most beneficial to do so, in order to improve the quality of the opponent model and potentially achieve greater overall winnings. As described previously, an agent following this approach was our entry (called GBR) into the 2011 ACPC.

**Hand Strength-Based Opponent Modeling**

In our approach, the agent makes decisions about its best response to the current hand using hand strength estimations. We measure hand strength as the estimated probability that the agent will win the current hand. We calculate this measure as the percentage of hands won during Monte Carlo simulation of the known private and community cards against random dealings of both opponent cards and unrevealed community cards (using the Poker Prophesier library (JavaFlair, 2011) in our ACPC entry).

However, since this measure considers only the known cards, it neglects information based on the opponent’s behavior that could be used to refine the agent’s prediction of whether or not it will win the current hand (especially since hand strength is calculated over all possible opponent hands, only one of which the opponent actually holds). To capture this information, we model the opponent’s behavior to estimate opponent hand strength given their actions using Bayesian modeling. Specially, we model the likelihood $P(a|s)$ that the opponent takes action $a$ based on its hand strength $s$, then use Bayes rule to compute the posterior likelihood $P(s|a)$ that the opponent’s hand strength is a certain value, based on its actions.

To model the opponent’s behavior, we use a frequentist
counting approach. Specifically, the agent maintains counts \( c(s, a) \) corresponding to the number of times it observed the opponent performing each action \( a \) for each possible hand strength \( s \) per round. Since hand strength is continuous over \([0, 1]\), we discretize this value into a set \( S \) of equally sized bins to make our model tractable. Using the \( c(s, a) \) values, the agent can then compute the likelihood that its opponent will take an action conditioned on a given hand strength as follows:

\[
P(a|s) = \frac{c(s, a)}{\sum_{a' \in A} e^{c(s, a')}}
\]  

where \( A \) is the set of possible actions (i.e., raise, call, fold).

These counts are updated after each hand when the opponent’s hand strength can be known for certain. Specifically, once the opponent reveals its cards at the end of the hand, its hand strength can be calculated using Monte Carlo simulation since all of the information necessary for such calculations are known (both private and community cards). However, if the opponent does not reveal its cards (e.g., the hand did not reach the final round), then the expected hand strength is estimated as \( |S| - hs + 1 \), where \( hs \) is the agent’s own estimated hand strength when a particular action was taken by the opponent. While this value might be inaccurate, it is the best the agent can calculate given its known information.

Then, during a hand of poker, the agent can exploit its opponent model to estimate the opponent’s hand strength to make better decisions about its own action. Specifically, the agent uses Bayes rule to update its estimate of the opponent’s hand strength based on its observed action:

\[
P'(s|a) = P(s|a) = \frac{P(a|s)P(s)}{P(a)} = \frac{P(a|s)P(s)}{\sum_{s' \in S} P(a|s')P(s')}
\]

For each hand, the agent initializes its beliefs over opponent hand strength \( P(s) \) with a simple distribution where half of the mass is centered at \(|S| - hs + 1\) to exploit its knowledge of its own hand strength, while the rest of the mass is uniformly distributed over the other possible hand strengths since no knowledge is known favoring any other hand strength. In the future, we plan to investigate other distributions for estimating initial opponent hand strength.

Please note that this Bayesian procedure for calculating \( P(s|a) \) closely follows other forms of Bayesian opponent modeling, such as the work of Baker and Cowling (2007) who also modeled opponents based on the observed actions. However, recall (c.f., Background) that their approach assumed that opponents ignored their cards and always played the same strategy, while we assume an opponent that varies its behavior based on its private cards and probability of winning.

Finally, given an opponent model, the agent now has a better approach for choosing an action to perform than simply considering its own hand strength. Specifically, given an opponent’s most recent action \( a \) (or none if the agent is the first to act in the hand), the agent chooses an action to exploit its opponent using the procedure presented in Figure 2. This procedure bases its decision on how similar the agent’s own hand strength (\( hs \)) is to the estimate of the opponent’s \( (os) \): folding if the agent believes it is in a significantly worse position to win, raising if it believes it is in a significantly better position, and calling if the agents’ hands are similar.

**Entropy-based Exploration for Active Sensing**

One problem with opponent modeling (and agent learning in general) is that it takes many interactions with the opponent before an adequate model is learned. Thus, if an agent always plays the best response in a hand, it might take many hands to learn an adequate model. This occurs because the best response to a potentially losing hand is to fold, which will (1) happen frequently in a game of poker, and (2) fail to gather sufficient information about the opponent to adequately model the opponent’s behavior. If the number of hands in a game is fixed, longer learning times result in fewer hands in which the agent can exploit its learned model, resulting in fewer winnings.

To overcome this problem, we propose to enhance opponent modeling by using an active sensing approach where an agent chooses to stay in hands – even when it thinks those are losing hands – to continue to collect information when it can most benefit in order to boost its opponent modeling. This improvement in opponent modeling is valuable to the agent because it enables the agent to potentially achieve more long term winnings due to faster and potentially higher quality opponent modeling, but at the cost of some losses during active sensing.

To tradeoff between costs and benefits, we perform active sensing by modeling the problem as one of exploration vs. exploitation. In our approach, the agent decides to “explore” the opponent’s behavior to improve its model by choosing to gather more information when it is uncertain about the opponent, while other times “exploiting” the opponent model to maximize its winnings by following the previously described best response to the hand.

We control the amount of exploration by considering the agent’s uncertainty in its’ beliefs about its opponent. Specifically, when the agent has a high uncertainty in its opponent hand strength estimates \( P \), it recognizes that its opponent model is also uncertain about the opponent’s actions in the current situation. Thus, the agent has more to gain by staying in and exploring the opponent’s behavior to update its model, as opposed to situations where it has more certainty and its model is already adequate. Factoring in uncertainty allows active sensing to decide when it is more appropriate to gather additional information, rather than staying in too many hands too long. Note that

```
chooseAction(a, P)
1. Calculate hand strength hs
2. Update opponent hand strength distribution P
3. Calculate os = E[P] over all s ∈ S
4. Choose action act
   a. If hs > os + 1, act = Raise
   b. Else if hs < os – 1, act = Fold
   c. Else act = Call
5. Return act
```

*Figure 2: chooseAction Procedure*
chooseAction(a, P)

5. If \( \text{act} = \text{Fold} \)
   a. Choose \( \text{rand} \sim U(0, 1) \)
   b. If \( \text{rand} < H(P)\epsilon, \text{act} = \text{Call} \)
6. Return \( \text{act} \)

Figure 3: Active Sensing Update to chooseAction Procedure

one could argue that an agent should fold when it has high uncertainty in its opponent hand strength estimates to my- optically diminish its losses. However, this represents the situation when the agent most needs to improve its model, and folding prevents the agent from gathering necessary information that could most benefit the agent later on.

In our approach, the agent calculates its uncertainty as the entropy in the probability distribution over opponent hand strengths:

\[
H(P) = -\sum_{s \in S} P(s) \log_{\text{base}} P(s) \tag{3}
\]

where the base of the logarithm was chosen to normalize the value between \([0, 1] \). Entropy is an appropriate measure of uncertainty because it is lowest when the probability mass is centered on one point and highest when the probability values are homogenous. Entropy measures have been proposed elsewhere in the active sensing literature to guide agent sensing (e.g., Araya-Lopez et. al, 2010).

Using the calculated entropy value, the agent performs a weighted \( \epsilon \)-greedy method of exploration: it chooses to (1) explore and stay in the hand with probability \( H(P)\epsilon \), or (2) exploit its model with probability \( 1 - H(P)\epsilon \). Thus, the more uncertain the agent is, the more likely it is to continue gathering information to better inform its model. Likewise, the more certain the agent is, the less need it has to update the opponent model. Of course, the agent should not stay in too many hands to continue gathering information, so setting \( \epsilon \) appropriately small prevents the likelihood of exploration from being too large. We will empirically evaluate various \( \epsilon \) values later in this paper.

Putting active sensing together with the rest of the agent’s reasoning, we update the decision procedure followed in Figure 2 by replacing line 5 with the steps shown in Figure 3. Here, the agent only explores if its best response is to fold, otherwise it is already collecting more information about the opponent’s behavior. Further, if it does choose to explore, it only uses the call action to minimize the amount of additional money it must bet to stay in the hand to minimize losses during active sensing.

**Experimental Setup**

**Poker Playing Agents**

To demonstrate the benefits of active sensing in opponent modeling in poker, we conduct an empirical study competing opponent modeling agents with and without active sensing against various types of agents. Specifically, we vary the \( \epsilon \) parameter to our agent to represent different maximum amounts of active sensing. We categorize the opponents in our study into the following groups:

<table>
<thead>
<tr>
<th>Table 1: Mixed Strategy Action Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strategy</strong></td>
</tr>
<tr>
<td>Aggressive</td>
</tr>
<tr>
<td>Optimistic</td>
</tr>
<tr>
<td>Conservative</td>
</tr>
<tr>
<td>Tactic</td>
</tr>
<tr>
<td>Weak</td>
</tr>
<tr>
<td>Medium</td>
</tr>
<tr>
<td>Strong</td>
</tr>
</tbody>
</table>

1. **Mixed strategies**, representing agents that choose actions based on their hand strengths, conforming to three different styles of play: **Conservative**, **Optimistic**, and **Aggressive**. These agents follow mixture model probabilities for actions shown in Table 1, where the agent’s hand strength determines a probability distribution *(i.e., “tactic”)* from which the agent chooses an action at random.

2. **Fixed action strategies**, representing agents that always perform the same actions regardless of their hands: **Raise**, **Call**, and **CheckFold** (which calls when it can stay in for free, else it folds). These approaches are commonly used in poker agent evaluation (e.g., Ganzfried and Sandholm, 2011).

3. **Deceptive strategies**, representing agents trying to catch an opponent modeler in the “get taught and exploited” problem by changing strategies during the game: **Switch**, which changes its strategy once after a long period from Conservative to Aggressive.

4. **Adaptive strategies**, representing agents that adapt their play over time based on learned behavior: **QL**, which uses Q-Learning (Watkins, 1989) to learn the expected winnings of the hand for each action based on its current hand strength *(with a discounted learning rate)*, then chooses actions using Softmax (Vermorel and Mohri, 2005).

**Evaluation Metrics and Parameters**

To evaluate advantages and disadvantages of active sensing, we consider the following measures of performance:

1. **Agent winnings**, measured as the average number of big blinds won per hand by the agent over the opponent.

2. **Belief uncertainty**, measured as the average entropy *(Eq. 3)* of the agent’s belief about its opponent’s hand strength \( P \), used to determine the need for and directly control the likelihood of performing active sensing.

We expect active sensing agents to achieve higher winnings due to more knowledge about the opponent and thus better exploitation late in games. However, increased sensing could cause fewer winnings if the agent over senses and stays in too many hands after it has formed an adequate model of the opponent. Thus, setting the \( \epsilon \) value appropriately is important, as discussed previously. We also expect active sensing agents to achieve the lowest belief uncertainty, due to the increase in information used to refine the knowledge in the opponent model.
We follow the same experimental setup used for the two-player heads-up limit competition in the 2011 ACPC (ACPC, 2011d), including the same software, with parameters shown in Table 2. We compete each opponent against variants of our approach twice per random seed – once in each position of the hand to eliminate bias from the random dealing of cards. After each hand, each agent’s chips are reset (i.e., we use Doyle’s game). However, we use 10 different random seeds per match to further reduce the variance in our results, rather than only one as in the ACPC.

### Results

#### Effect of Active Sensing on Agent Winnings

We now investigate how active sensing affected the opponent modeling agent’s winnings against the various opponents. We present the agent’s winnings with 95% confidence intervals in Table 3, measured as the average number of big blinds won per hand. From these results, we make several important observations.

First, we observe that against almost every opponent (except Conservative), an opponent modeling agent with active sensing enabled (ε > 0.0) achieved higher winnings than the opponent modeling agent without active sensing (ε = 0.0). Thus, the choice of staying in extra hands the agent otherwise would have folded in order to gather more information about the opponent paid off, in spite of the additional cost of not folding. Therefore, active sensing can be beneficial for better exploiting the opponent.

Second, we observe that the amount of active sensing that achieved the best performance varied greatly based on the opponent. For most of the mixed strategies (Aggressive, Optimistic) that randomly choose actions depending on hand strength, a small amount of active sensing was optimal (ε = 0.1). We believe this was due to the agent directly modeling the decision behavior of the opponent, and thus less additional sensing was necessary to adequately capture opponent behavior in the model. In contrast, for the fixed action strategies (Raise, Call, Check Fold), more active sensing (ε = 0.4) led to better agent winnings. Here, the opponents did not choose actions based on their hand strength, so the agent’s model did not reflect the decision process of its opponent. Therefore, we hypothesize that in general, less active sensing is necessary the closer the opponent model reflects the opponent’s actual reasoning, whereas more active sensing would be useful if a general opponent model were used that might not reflect the actual reasoning of the opponent. This result also has implications on how to better design a framework to strategically decide when to apply active sensing reasoning. We intend to further investigate this as future work.

Additionally, we also observe that for the more dynamic opponents – both deceptive (Switch) and adaptive (QL) strategies – more active sensing was beneficial (ε = 0.3 or 0.4) and resulted in greater agent winnings. This is noteworthy for the deceptive Switch agent since lower active sensing was more beneficial against either of its two component strategies (Conservative and Aggressive), as observed in the previous paragraph. Moreover, recall that QL learns and adapts from its experiences when playing against our active sensing agent. This means that although our agent’s active sensing decisions had cumulative impact on its opponent’s behavior, the active sensing agent seemed to still maintain its cost-effectiveness. In fact, against QL, the best active sensing agent (ε = 0.3) achieved statistically significantly (with α = 0.05) greater winnings than the opponent modeling agent without active sensing (ε = 0.0). Overall, against opponents who change strategies either to confuse the agent’s modeling (e.g., as in the “get taught and exploited” problem [Sandholm, 2007]) or to adapt to the agent’s behavior, more active sensing can be beneficial to help the agent itself adapt to its opponent. In the future, we plan on testing against a wider range of deceptive and adaptive agents to better understand exactly how much active sensing is beneficial, as well as when is too much and would result in worsened agent performance.

#### Effect of Active Sensing on Belief Uncertainty

Next, we investigate how active sensing affected the belief uncertainty of the agent while it was choosing how to play its hand. We present the uncertainty in the agent’s beliefs about opponent hand strength with 95% confidence intervals in Table 4, measured as the average entropy (Eq. 3) of the agent’s beliefs $P$. From these results, we again make several important observations.

First, we observe that against every opponent, active sensing led to the least amount of uncertainty (i.e., most certainty) in the agent’s beliefs during the hands of poker. Thus, gathering additional information through active sensing did in fact lead to better beliefs developed from opponent modeling, which led to the greater winnings previously observed. Therefore, as in similar environments where agents rely on environment monitoring, active sensing is beneficial to opponent modeling in competitive games for improving the agent’s knowledge through better sensing.

Second, we observe that unlike with agent winnings, more active sensing generally was better than less active sensing. This was true for all opponents except against Aggressive and QL, against whom the optimal level of active sensing was still relatively high (ε = 0.2 or 0.3). This is expected, but what is most interesting is that the optimal amount of active sensing for maximizing agent winnings did not correspond to the optimal amount for minimizing agent uncertainty. This reinforces our earlier concern that an agent can perform more active sensing than necessary and the amount of active sensing should be properly tuned by either the developer a priori or by the

---

**Table 2: Experimental Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Hands</td>
<td>3000</td>
</tr>
<tr>
<td>Chip Count</td>
<td>1000</td>
</tr>
<tr>
<td>Blind Size</td>
<td>5/10</td>
</tr>
<tr>
<td>$\epsilon$ Values</td>
<td>0.0, 0.1, 0.2, 0.3, 0.4</td>
</tr>
<tr>
<td>Switch Period</td>
<td>1000 hands</td>
</tr>
</tbody>
</table>
when the agent could decide to
ible hand
-pan, so more information
wa-
T
e opponent's hand strength
oker. For
n-
not
know the ground truth of opponent hand strength if a hand
to play its current hand. First, recall that an agent can only
know the ground truth of opponent hand strength if a hand
reaches a showdown. Otherwise, the agent must estimate
the ground truth based on its own hand strength as
$|S| - hs + 1$. However, the opponent’s hand strength in
reality might be any possible value, so the model updates
for non-showdown hands could be incorrect. In the future,
we plan on addressing this pitfall by considering alternative
methods for updating the model for non-showdown
hands or only relying on showdowns for model updates.

Second, the distribution of hand strengths for all possible
hands in Texas hold’em poker is Gaussian with most of
its mass around a 50% chance of winning. Thus, most
hands have a strength near the mean, so more information
is collected during modeling for hand strengths near the
mean of the distribution and less information is collected
for the tails of the distribution. Hence, when a hand closer
to the tail is encountered, the agent will not have an ade-
quate model to correctly estimate its opponent’s hand
strength. However, this presents another opportunity for
guided information gathering with active sensing – the
agent should sense more when it believes that the oppo-
nent’s hand strength might be one it does not have much
information for yet. We intend to explore other types of
active sensing for our model in the future.

### Future Work

#### Addressing Pitfalls to Current Model

As described previously, our opponent model might not
have been as effective as expected in helping the agent
estimate its opponent’s hand strength when choosing how
to play its current hand. First, recall that an agent can only
know the ground truth of opponent hand strength if a hand
reaches a showdown. Otherwise, the agent must estimate
the ground truth based on its own hand strength as
$|S| - hs + 1$. However, the opponent’s hand strength in
reality might be any possible value, so the model updates
for non-showdown hands could be incorrect. In the future,
we plan on addressing this pitfall by considering alternative
methods for updating the model for non-showdown
hands or only relying on showdowns for model updates.

Extending Active Sensing

Aside from our approach, active sensing could also be ben-
eficial to other forms of opponent modeling in poker. For
example, suppose the agent modeled the opponent by de-
determining what strategy the opponent was playing from
amongst a set of possible strategies (e.g., Southey et. al,
2005; Baker and Cowling, 2007) rather than modeling op-
ponent hand strength. Then the agent could decide to
gather more information when it was most uncertain about

<table>
<thead>
<tr>
<th>Opponent</th>
<th>$\epsilon = 0.0$</th>
<th>$\epsilon = 0.1$</th>
<th>$\epsilon = 0.2$</th>
<th>$\epsilon = 0.3$</th>
<th>$\epsilon = 0.4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggressive</td>
<td>0.3951 ± 0.0331</td>
<td>0.4091 ± 0.0335</td>
<td>0.3570 ± 0.0338</td>
<td>0.3755 ± 0.0343</td>
<td>0.3756 ± 0.0349</td>
</tr>
<tr>
<td>Optimistic</td>
<td>0.3486 ± 0.0272</td>
<td>0.3755 ± 0.0275</td>
<td>0.3638 ± 0.0277</td>
<td>0.3513 ± 0.0279</td>
<td>0.3445 ± 0.0285</td>
</tr>
<tr>
<td>Conservative</td>
<td>0.2258 ± 0.0177</td>
<td>0.2093 ± 0.0176</td>
<td>0.2116 ± 0.0181</td>
<td>0.2190 ± 0.0184</td>
<td>0.2072 ± 0.0184</td>
</tr>
<tr>
<td>Raise</td>
<td>1.7219 ± 0.0839</td>
<td>1.7176 ± 0.0842</td>
<td>1.7051 ± 0.0844</td>
<td>1.7079 ± 0.0849</td>
<td>1.7302 ± 0.0854</td>
</tr>
<tr>
<td>Call</td>
<td>0.5780 ± 0.0216</td>
<td>0.5840 ± 0.0216</td>
<td>0.5911 ± 0.0216</td>
<td>0.5961 ± 0.0216</td>
<td>0.6015 ± 0.0217</td>
</tr>
<tr>
<td>CheckFold</td>
<td>0.2996 ± 0.0052</td>
<td>0.3028 ± 0.0053</td>
<td>0.3074 ± 0.0053</td>
<td>0.3123 ± 0.0054</td>
<td>0.3151 ± 0.0054</td>
</tr>
<tr>
<td>QL</td>
<td>0.3700 ± 0.0272</td>
<td>0.4023 ± 0.0274</td>
<td>0.4133 ± 0.0287</td>
<td>0.4405 ± 0.0285</td>
<td>0.4079 ± 0.0298</td>
</tr>
<tr>
<td>Switch</td>
<td>0.2652 ± 0.0269</td>
<td>0.2582 ± 0.0269</td>
<td>0.2631 ± 0.0273</td>
<td>0.2567 ± 0.0279</td>
<td>0.2824 ± 0.0283</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opponent</th>
<th>$\epsilon = 0.0$</th>
<th>$\epsilon = 0.1$</th>
<th>$\epsilon = 0.2$</th>
<th>$\epsilon = 0.3$</th>
<th>$\epsilon = 0.4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggressive</td>
<td>0.8135 ± 0.0005</td>
<td>0.8133 ± 0.0005</td>
<td>0.8117 ± 0.0005</td>
<td>0.8097 ± 0.0005</td>
<td>0.8100 ± 0.0005</td>
</tr>
<tr>
<td>Optimistic</td>
<td>0.8092 ± 0.0005</td>
<td>0.8097 ± 0.0005</td>
<td>0.8092 ± 0.0005</td>
<td>0.8080 ± 0.0005</td>
<td>0.8074 ± 0.0005</td>
</tr>
<tr>
<td>Conservative</td>
<td>0.8337 ± 0.0006</td>
<td>0.8322 ± 0.0006</td>
<td>0.8314 ± 0.0006</td>
<td>0.8298 ± 0.0006</td>
<td>0.8280 ± 0.0006</td>
</tr>
<tr>
<td>Raise</td>
<td>0.8296 ± 0.0004</td>
<td>0.8294 ± 0.0003</td>
<td>0.8286 ± 0.0003</td>
<td>0.8288 ± 0.0003</td>
<td>0.8284 ± 0.0003</td>
</tr>
<tr>
<td>Call</td>
<td>0.8058 ± 0.0003</td>
<td>0.8055 ± 0.0003</td>
<td>0.8053 ± 0.0003</td>
<td>0.8050 ± 0.0003</td>
<td>0.8047 ± 0.0003</td>
</tr>
<tr>
<td>CheckFold</td>
<td>0.8898 ± 0.0008</td>
<td>0.8877 ± 0.0008</td>
<td>0.8854 ± 0.0008</td>
<td>0.8829 ± 0.0008</td>
<td>0.8812 ± 0.0008</td>
</tr>
<tr>
<td>QL</td>
<td>0.8113 ± 0.0006</td>
<td>0.8132 ± 0.0006</td>
<td>0.8109 ± 0.0006</td>
<td>0.8120 ± 0.0006</td>
<td>0.8144 ± 0.0005</td>
</tr>
<tr>
<td>Switch</td>
<td>0.8223 ± 0.0005</td>
<td>0.8219 ± 0.0005</td>
<td>0.8201 ± 0.0005</td>
<td>0.8192 ± 0.0005</td>
<td>0.8190 ± 0.0004</td>
</tr>
</tbody>
</table>

Table 3: Agent Winnings (Average Big Blinds Won per Hand)
Note: Bold Signifies the Best (Highest) $\epsilon$ Value per Opponent

Table 4: Belief Uncertainty (Average Entropy in Agent’s Belief About Opponent Hand Strength)
Note: Bold Signifies the Best (Lowest) $\epsilon$ Value per Opponent

Agent during gameplay. Moreover, a strategy that only
optimizes belief uncertainty is not viable, as some tradeoff
with the costs must be incorporated. In the future, we plan
to investigate how to better adapt active sensing.

Finally, we observe that in general, the agent’s beliefs
were pretty uncertain. Recall (Eq. 3) that we scale uncer-
tainty into a range between $[0, 1]$ and the higher the value,
the more uncertain the agent is. From Table 4, we observe
that in all cases, the average uncertainty was greater than
0.8. Certainly, more uncertainty is expected early in a
hand since the agent has not observed many opponent ac-
tions to refine its beliefs about opponent hand strength.
However, we note that even for hands that reach show-
down (the longest possible hand with the most refinements
of beliefs), the average final uncertainty value (not report-
ed) was never less than 0.77 for any $\epsilon$ value against any
opponent. Thus, it appears that our opponent modeling
was not as successful as intended in predicting opponent
hand strength, which was important in the agent’s reason-
ing (Figure 2) and calculating a best response. Therefore,
there is certainly room for improvement in our opponent
modeling, which could both further boost agent winnings
and enable even more effective active sensing.
the opponent’s strategy, similar to our approach, by considering the entropy in the posterior distribution over opponent strategies. Moreover, if the agent computed a distance function from the current model to computed strategies (e.g., Ganzfried and Sandholm, 2011), the agent could choose to gather more information when the distances between models were all significantly large.

Moreover, active sensing could also be used in a more directed manner to improve an existing opponent model. For example, when the agent recognizes it has little information about the expected behavior of the opponent for a specific scenario (e.g., after a strong community card is revealed), it could be more willing to stay in. Similarly, it could choose actions other than call to intentionally gather specific information, such as how the opponent responds to a raise after the agent was slow-playing its hand. Finally, active sensing could also be extended to be more cost-aware, choosing to only gather information when the risks of current losses are exceeded by potential future winnings.

Conclusions

In conclusion, we have introduced active sensing as a method to boost opponent modeling in competitive games such as Texas hold’em poker. We described an approach using active sensing to increase information gathering in an actual agent called GBR entered into the 2011 ACPC, which placed the best of known opponent modeling agents. Using a brief empirical investigation against various types of opponent agents, we demonstrated the benefits of active sensing: both higher winnings and more certain beliefs about the opponent. However, we also identify several challenges to our existing opponent model, which we intend to investigate as future work. Finally, we also outline how active sensing can be extended beyond our present use, which we believe is another interesting avenue for artificial intelligence research and could potentially yield better opponent modeling agents for poker.

Acknowledgements

This research was supported by the National Science Foundation through an NSF Graduate Research Fellowship under grant DGE-0548501 and was completed utilizing the Holland Computing Center of the University of Nebraska. We thank L.D. Miller for his contributions to this research.

References


Williams, J.D. and Young, S. 2007. Partially observable Markov decision processes for spoken dialog systems. Computer Speech and Language, 21, 393-422.