Data Biased Robust Counter Strategies

Michael Johanson, Michael Bowling

November 14, 2012
Computer Poker Research Group

Created Polaris - the world’s strongest program for playing Heads-Up Limit Texas Hold’em Poker

July 2008: Went to Las Vegas, played against six poker pros, won the 2nd Man-Machine Poker Championship

Won several events in the 2008 AAAI Computer Poker Competition

Research goals:

Solve very large extensive form games

Learn to model and exploit opponent’s strategy
In this talk, we present a technique for dealing with three types of model uncertainty:

- The opponent / environment changes after we model it
- The model is more accurate in some areas than others
- The model’s prior beliefs are very inaccurate
Texas Hold’em Poker

- Our domain: 2-player Limit Texas Hold’em Poker
  - Zero-Sum Extensive form game
  - Repeated game (Hundreds or thousands of short games)
  - Hidden information (Can’t see opponent’s cards)
  - Stochastic elements (Cards are dealt randomly)
  - Goal: Win as much money as possible

- RL interpretation:
  - POMDP (when opponent’s strategy is static)
  - Some properties of world are known
    - Probability distribution at chance nodes
  - Don’t know exactly what state you are in (because of opponent’s cards)
  - Transition probabilities at opponent choice nodes are unknown
  - Payoffs at terminal nodes are unknown
There are lots of ways to play games like poker. Two are well known:

- **Nash Equilibrium**
  - Minimizes worst-case performance
  - Doesn’t try to exploit opponent’s mistakes

- **Best Response**
  - Maximizes performance against a specific static opponent
  - Doesn’t try to minimize worst-case performance
  - Problem: requires the opponent’s strategy

**Goals:**

- Observe the opponent, build a model, and use it instead of the opponent’s strategy
- Bound worst-case performance
  - Model could be inaccurate
  - Opponent could change
Types of Strategies

Performance against a static opponent, in millibets per game

- Game Theory: Nash equilibrium. Low exploitiveness, low exploitability
- Decision Theory: Best response. High exploitiveness, high exploitability
Types of Strategies

Performance against a static opponent, in millibets per game

- Mixture: Linear tradeoff of exploitiveness and exploitability
Types of Strategies

Performance against a static opponent, in millibets per game

- **Exploitation of Opponent (mb/g)**
- **Worst Case Exploitability (mb/g)**
- **Mixture**
- **Restricted Nash Response**

- **Restricted Nash Response:** Much better than linear tradeoff
Restricted Nash Response

- Proposed by Johanson, Zinkevich and Bowling (Computing robust counter-strategies, NIPS 2007)

Choose a value $p$ and play an unusual game:
- With probability $p$, opponent is forced to play according to a static strategy
- With probability $1 - p$, opponent is free to play as they like

- $p = 1$: Best response
- $p = 0$: Nash equilibrium
- $0 < p < 1$: Different tradeoffs between exploiting model and being robust to any opponent!

This provably generates the best possible counter-strategies to the opponent
Restricted Nash Response

Performance against model of Orange

Exploitation of Opponent (mb/g)

Worst Case Exploitability (mb/g)

(0)

(0.5)

(0.7)

(0.8)

(0.9)

(0.93)

(0.97)

(0.99)

(1)
Goals:

- Observe the opponent, build a model, and use it instead of the opponent’s strategy
- Bound worst-case performance
  - Model could be inaccurate
  - Opponent could change
Frequentist Opponent Models

- Observe 100,000 to 1 million games played by the opponent.
- Do frequency counts on actions taken at information sets.
- Model assumes opponent takes actions with observed frequencies.
- Need a default policy when there are no observations.
  - Poker: Always-Call.
Problems with Restricted Nash Response

Problem 1: Overfitting to the model

![Graph showing exploitation vs. exploitability](image-url)
Problems with Restricted Nash Response

Problem 2: Requires a lot of training data
Data Biased Response

- Restricted Nash Response had two problems:
  - Model wasn’t accurate in states we never observed
  - Model was more accurate in some states than in others
- We need a new approach. We’d like to only use the model wherever we have reason to trust it
- New approach: use model’s accuracy as part of the restricted game
Let's set up another restricted game. Instead of one $p$ value for the whole tree, we'll set one $p$ value for each choice node, $p(i)$.

- More observations $\rightarrow$ more confidence in the model $\rightarrow$ higher $p(i)$
- Set a maximum $p(i)$ value, $P_{\text{max}}$, that we vary to produce a range of strategies
Data Biased Response

Three examples:

- 1-Step: \( p(i) = 0 \) if 0 observations, \( p(i) = P_{\text{max}} \) otherwise
- 10-Step: \( p(i) = 0 \) if less than 10 observations, \( p(i) = P_{\text{max}} \) otherwise
- 0-10 Linear: \( p(i) = 0 \) if 0 observations, \( p(i) = P_{\text{max}} \) if 10 or more, and \( p(i) \) grows linearly in between

By setting \( p(i) = 0 \) in unobserved states, our prior is that the opponent will play as strongly as possible
DBR doesn’t overfit to the model

RNR and several DBR curves:

![Graph showing multiple curves for RN, 1-Step, 10-Step, and 0-10 Linear, with axes labeled Exploitation (mb/h) on the y-axis and Exploitability (mb/h) on the x-axis. The graph compares different models and their performance in terms of exploitation and exploitability.]
DBR works with fewer observations

0-10 Linear DBR curve:
Conclusion

- **Data Biased Response technique:**
  - Generate a range of strategies, trading off exploitation and worst-case performance
  - Take advantage of observed information
  - Avoid overfitting to parts of the model we suspect are inaccurate
Future directions

- Extend to single-player domains
  - Can overfitting be reduced by assuming a slightly adversarial environment in unobserved / underobserved areas?
- More rigorous method for setting $p$ from the observations