

OPPONENT EXPLOITATION

CS 15-888 Computational Game Solving

Tuomas Sandholm

Traditionally two approaches to tackling games

- **Game theory approach**
 - Safe in 2-person 0-sum games
 - Doesn't maximally exploit weaknesses in opponent(s)
- **Opponent modeling/exploitation**
 - Needs prohibitively many repetitions to learn in large games (loses too much during learning)
 - Crushed by game theory approach in Texas Hold'em
 - *Get-taught-and-exploited problem* [Sandholm, *Artificial Intelligence*, 2007]

Let's hybridize the two approaches

- Start playing based on game theory approach
- As we learn opponent(s) deviate from equilibrium, start adjusting our strategy to exploit their weaknesses
 - Requires no prior knowledge about the opponent
 - Adjust more in points of the game for which more data is now available

Deviation-Based Best Response (DBBR) algorithm

(generalizes to multi-player games)

- Main idea:
 - We'd like to conservatively assume that the opponent is playing the best (*i.e.*, least exploitable) strategy that is consistent with our observations of his play
 - The obvious way to accomplish this would be to add linear constraints to the LP for finding an equilibrium that force the opponent model to conform with our observations. However, that would not be practical for real-time play in large games
 - To obtain a practical algorithm, we must find a faster way of constructing an opponent model from our observations. DBBR constructs the model by noting deviations of our opponent's observed action frequencies from equilibrium frequencies

Deviation-Based Best Response (DBBR) algorithm

(generalizes to multi-player games)

- Compute an (approximate) equilibrium
- Maintain counters of opponent's play throughout the match
- **for** $n = 1$ **to** |public histories|
 - Compute opponent posterior action probabilities at n using a Dirichlet prior
 - Compute opponent posterior bucket probabilities (opponent so far using Bayes rule)
 - Compute model of opponent's strategy at n
- **Return** best response to the opponent model

$$\alpha_{n,a} = \frac{p_{n,a}^* \cdot N_{prior} + c_{n,a}}{N_{prior} + \sum_{a'} c_{n,a'}}$$

Many ways to define opponent's "best" strategy that is **consistent with bucket probabilities**

- L_1 distance to the precomputed equilibrium strategy => LP
- L_2 distance to the precomputed equilibrium strategy => QP
- Custom weight-shifting algorithm => greedy
- ...

L1 case in full detail

minimize
$$\sum_{b \in B_n} \sum_{a \in A_n} [\beta_{n,b} \cdot |x_{n,b,a} - \sigma_{n,b,a}^*|]$$

subject to
$$\sum_{b \in B_n} [\beta_{n,b} \cdot x_{n,b,a}] = \alpha_{n,a} \text{ for all } a \in A_n$$

$$\sum_{a \in A_n} x_{n,b,a} = 1 \text{ for all } b \in B_n$$

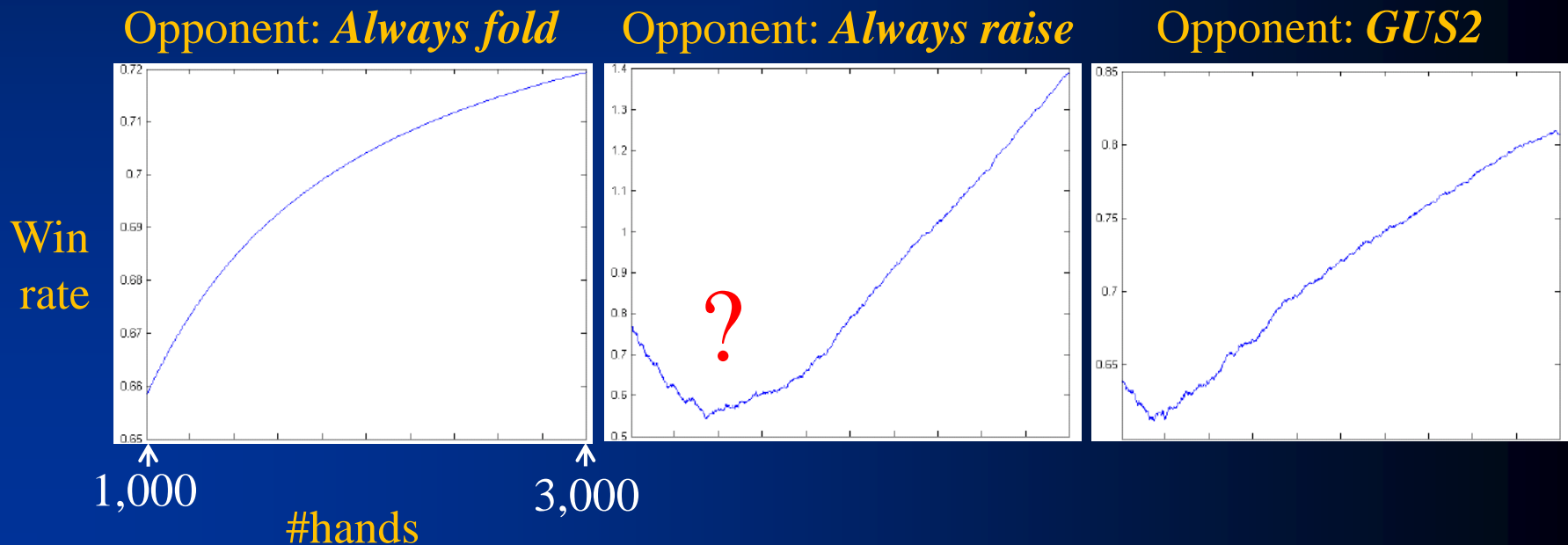
$$0 \leq x_{n,b,a} \leq 1 \text{ for all } a \in A_n, b \in B_n$$

Idea behind the custom weight-shifting algorithm

- *E.g.*, suppose the opponent is only raising 30% of the time when first to act, while σ^* raises 50% of the time in that situation
- Instead of doing a full L1 or L2-minimization explicitly, we could use the following heuristic algorithm:
 - Sort all buckets by how often the opponent raises with them under σ^* , then greedily keep removing buckets from his raising range until the weighted sum (using the $\beta_{n,b}$'s as weights) equals 30%
 - This is a simple greedy algorithm, which can be run significantly more efficiently in practice than the L1 and L2-minimization procedures, which repeatedly use CPLEX at runtime

Experiments on opponent exploitation

- Significantly outperforms game-theory-based base strategy in 2-player limit Texas Hold'em against
 - trivial opponents
 - weak opponents from AAAI computer poker competitions
- Don't have to turn this on against strong opponents
- Of the opponent model selection variants, L_2 and weight shifting had selective superiority, L_1 performed worse



Opponent-exploitation approaches using “ ϵ -safe best response”

- “ **ϵ -safe best response**”: [McCracken & Bowling, *AAAI-04 Fall Symposium*]
 - Of the ϵ -safe strategies, pick the one that does best against an opponent model
 - Tested on Rock-Paper-Scissors; doesn’t scale to large games
- “**Restricted Nash response (RNR)**” [Johanson, Zinkevich & Bowling, *NIPS-07*]
 - Modified game is formed in which the opponent is forced to act according to an opponent model with probability p , and is free to play w.p. $(1 - p)$. Solved using CFR
 - Works well if opponent’s strategy is known, but opponent model is typically formed through a limited number of observations, and may be incomplete (it cannot predict the opponent’s strategy in some states) or inaccurate
 - RNR performs poorly under such circumstances [Johanson & Bowling, *AISTATS-09*]
 - Overfits to opponent model: with too little data, can make a strategy both less exploitive and more exploitable
 - Very sensitive to training against the target opponent
- “**Data biased responses (DBR)**” [Johanson & Bowling, *AISTATS-09*]
 - Like RNR but opponent’s strategy is p_I -constrained on a per-information set I basis, and depends on our confidence in the accuracy of the model. Again, solved using CFR
 - Less overfitting
 - More effective in 2-player limit Texas Hold’em than RNR
 - Assumes a call action if there were no observations about opponent’s strategy
 - Assumes cards are shown after every hand unlike DBBR

Online implicit agent modelling

[Bard, Johanson, Burch & Bowling, AAMAS-13]

- Offline, compute a small number of strong strategies
 - Use RNR if training opponents' strategies are known; DBR if we have only samples from them
 - As a side effect, DBR computes robust strategies that mimic the data. At each info set, the mimic will (with some probability based on the amount of data available) choose its play so as to prevent exploitation by the DBR strategy. This mimic strategy behaves increasingly like the agent that produced the data as more observations are available
 - (There is no need for finding such a strategy when using an RNR with an explicit opponent model: the best possible mimic in such a case is the model itself, which is already available.)
 - Too big a portfolio can be worse. They used a greedy algorithm to add strategies to the portfolio—assuming an oracle chooses the best strategy in the portfolio against each mimic
- Online, use no-regret learning (they used Exp4) to choose among them
 - In estimating the strategies' EVs, use unbiased variance-reduction techniques
 - These techniques imagine alternate observations that remain consistent with the observed actions taken by other agents (e.g., alternate private cards or alternate actions that would end the game)
 - Details that enable that
 1. Since we're mixing extensive-form strategies instead of a distribution over single actions, we must average the strategies' action sequence probabilities
 2. We force the weight of each expert to be at least some minimum value. This guarantees that the acting strategy has non-zero probability on every action sequence played by some response in the portfolio

Safe opponent exploitation ?

- Definition. *Safe* strategy achieves at least the value of the (repeated) game in expectation
- Is safe exploitation possible (beyond selecting among stage-game equilibrium strategies)?
- This work applies also to extensive-form games, not just repeated games

When can opponent be exploited safely?

- ~~Opponent played an (iterated weakly) dominated strategy?~~

R is a gift
but not iteratively weakly dominated

	L	M	R
U	3	2	10
D	2	3	0



- ~~Opponent played a strategy that isn't in the support of any eq?~~

R isn't in the support of any equilibrium
but is also not a gift

	L	R
U	0	0
D	-2	1

- Definition.** We received a *gift* if opponent played a strategy such that we have an equilibrium strategy for which the opponent's strategy isn't a best response
- Theorem.** Safe exploitation is possible iff the game has gifts
- E.g., rock-paper-scissors doesn't have gifts

Exploitation algorithms

1.  Risk what you've won so far
 2.  Risk what you've won so far in expectation (over nature's & own randomization), i.e., risk the gifts received
 - Assuming the opponent plays a nemesis in states where we don't know
- ...
- **Theorem.** A strategy for a 2-player 0-sum game is safe iff it never risks more than the gifts received according to #2
 - Can be used to make any opponent model / exploitation algorithm safe
 - No prior (non-eq) opponent exploitation algorithms are safe
 - #2 experimentally better than more conservative safe exploitation algs
 - Suffices to lower bound opponent's mistakes
 - This type of safe exploitation is actually an equilibrium refinement: it is still a Nash equilibrium of the repeated (or extensive-form) game, and not all Nash equilibria of the repeated (or extensive-form) game accomplish this

Current & future research on opponent exploitation

- Understanding exploration vs exploitation vs safety
- In DBBR, what if there are multiple equilibria or near-equilibria?
- Application to other games
- Exploiting the opponent's myopia -- e.g., in treater vs. disease games in medicine [Sandholm *AAAI-15*; Kroer & Sandholm *IJCAI-16*, *AAAI-18*, *Artificial Intelligence 2020*]