Raising the Stakes

Jonathan Schaeffer
Department of Computing Science
University of Alberta

jonathan@cs.ualberta.ca
GAMES Research Group

- Largest AI research group using games
- Classic Games
  - Chess, checkers, go, hex
  - Poker, hearts, spades
- Commercial games
  - Role-playing (Bioware)
  - Sports (Electronic Arts)
  - Real-time strategy (Relic)
Thank You

- Darse Billings
- Michael Bowling
- Neil Burch
- Aaron Davidson
- Rob Holte
- Morgan Kan
- Bryce Larson
- Carmelo Piccione
- Terrance Schauenberg
- Finnegan Southey
- Duane Szafron
Darse Billings

- Poker player
- Philosopher
- Computer scientist
- Perpetual graduate student
Poker as a Test-bed for AI

- Multiple agents (typically 10)
- Imperfect information (unknown cards)
- Stochastic (shuffled deck)
- Risk management (betting)
- Partial observability
- Opponent modeling

Unlike most other games-applied research, poker has many properties of real-world applications.
Applications

- Adversarial opponent modeling
- Negotiation
- Auctions
- Real-time strategy games

- Most of the techniques described here have been used for real-time planning in (military) strategy games
Texas Hold'em

- Hundreds of poker variants
- Most strategically complex variant widely played
- *No-Limit* Texas Hold’em used to determine the champion in the annual World Series of Poker
- Our research: *Limit* Texas Hold’em
  - 10-player (ring game)
  - 2-player (heads up)
The Goal
The Real Goal

Chris Ferguson
Stu Ungar
Annie Duke
Texas Hold'em Rules

- 2-10 players
- Four rounds of play
  - Pre-flop: Each player gets two private cards; bet
  - Flop: Three community cards; bet
  - Turn: One community card; bet
  - River: One community card; bet
- Betting:
  - fold (lose all you money)
  - check/call (match what is in the pot)
  - bet/raise (increase amount in the pot)
- Best five card poker hand wins
Texas Hold'em Example

Bets $20

Alice

Bill

Chris

Pot: $110

J♠ 8♣ 4♠ J♦

B

T♠ 9♠

Poki

Dave
Obstacle 1: Feeling Lucky?
Obstacle 2: Does He Have It?
Obstacles (3)

- Imperfect Information
  - Few data points
  - Probabilistic strategies are essential
  - Need to infer what the opponent is likely to have (modulo bluffing)

- Opponent Modeling
  - Every opponent has a different style
  - Opponents change style frequently
  - One cannot be predictable!
Building a Strong AI
1991-2005

- Rule-based (bad)

- Simulations (better but weak)
  - Loki -> Poki (1997-present)

- Game theory (good)
  - PsOpti (2001-present)

- Learning and adaptation (best?)
  - Vexbot (2003-present)
  - Bayesbot (2005-present)
Rule-based Strategy

- Need a *real* domain expert (which we had)
- Poker knowledge is fragile
  - Covered common cases reasonably well
  - Unable to properly recognize and handle exceptions
- Poker expert quickly becomes the bottleneck, applying Band-Aid solutions to fix problems
- Easily exploitable by an experienced player
- Loki could win at a good rate in Internet games
- The most knowledge-intensive strategy tried
Simulations Strategy

- For each betting decision (f/c/r):
  - “Deal” cards to the opponent
  - Simulate hand to end of game
  - Gather statistics on each decision

- Less reliance on expert knowledge
  - Search is knowledge!

- Reliant on having a good model of the opponent

- Stronger than Loki in the laboratory, but not that much stronger against humans

- Less expert knowledge -> better play
Game Theory

- Nash equilibrium
- Find an “optimal” answer
- Assumes both sides play perfectly
- Recipe for drawing matches against strong players
Two-player Limit Hold'em

O(10^{18})

1,624,350
9 of 19
17,296
9 of 19
45
9 of 19
44
19

Initial
Flop
Turn
River
Bets
Bets
Bets
Bets

Pre-flop: 2 private cards
Flop: 3 community cards
Turn: 1 community card
River: 1 community card
Optimal Play

- Solve a linear programming problem for two-player Hold’em

- An axis is all possible *information sets* for each player
  - An information set is a set of states in the search tree where the player could be
  - Can’t know for sure because of hidden information
Information Sets

Chance

Q/I

Q/K

J/K

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r

f c r
Abstraction

Full game tree $T$

Abstract game tree $T^*$

Solve (LP)

too big to solve

Strategy For $T$

Strategy For $T^*$

abstraction (lossy)

reverse mapping
Pseudo-optimal Solution

- Pretend that you and your opponent can have only one of 6 types of hands
  - fantastic, very strong, strong, pretty good, so-so, weak
- This new game is 100 billion times simpler than real poker
- This new game looks like poker… and use its solution to play real poker.
Bucketing

- Reduce branching factor at chance nodes
- Overlaying "strategically similar" sub-trees

Original Bucketing

Transition Probabilities

Next Round Bucketing
Abstraction Models

Texas Hold'em
O(10^{18})

1,624,350
Initial
w^2 (36)

9 of 19
Bet Sequence
7 of 15

17,296
Flop
x^2 (36)

9 of 19
Bet Sequence
7 of 15

45
Turn
y^2 (36)

9 of 19
Bet Sequence
7 of 15

44
River
z^2 (36)

19
Bet Sequence
15

Abstract Pre-flop Model
O(10^7)

Abstract Post-flop Model
O(10^7)
Solving the LP

- Generates a system of equations
  - 10,000 to 100,000 entries per player
  - Sparse; 100 non-zero entries per row/column
  - Needs 24 hours to solve on an Athlon 1800 with 4 GB of RAM using CPLEX

- Use 7 categories of hands? Gets too big very quickly!
PsOpti2 versus "the Count"
PsOpti

Pro:
- Quantum leap in playing ability
- Poker knowledge used in the abstraction (pre-processing), not in the program (during play)
- Improved version

Con
- It takes a while, but strong players can identify weaknesses and exploit them
- Program is oblivious to opponent's strategy
- Optimal is not maximal
"You have a very strong program. Once you add opponent modeling to it, it will kill everyone."

Gautum Rao
Rock, Paper, Scissors

- 1st & 2nd International RoShamBo Championships
- Deceptively simple problem that illustrates the differences between optimal and maximal play
locaine Powder
Adaptive Play

- Numerous standard machine learning attempts (genetic algorithms, neural nets, decision trees, etc.) but with no notable success
- Gather data on opponent decisions and attempt to learn a model of their play
Context Tree (1)

- For each opponent, maintain a context tree of all possible betting sequences
  - In a given betting context, keep track of how often the opponent did a f/c/r
- To compute a betting decision, search the tree and compute an expected value
- Need:
  - Assign a value to the leaf nodes
  - Assign f/c/r frequencies to betting nodes
Use a variation of Expectimax
- Value of a node is the sum of the weighted value of each of the children

For opponent's decisions, use a mixed strategy

For program's decisions, use a maximum or mixed strategy

Miximix and Miximax algorithms
Using a Context Tree
Obstacles

- Few data points for a given betting context
  - Treat "similar" scenarios the same way, allowing one to abstract multiple contexts into one context

- Leaf node data is sparse
  - Use empirical data when we have it, else infer a "reasonable" distribution based on the betting sequence

- Default seeding of the tree
  - Defaults from a strong player or from PsOpti don't work; they are each skewed to a particular style
Results

- Vexbot handily defeats PsOpti and all our other bots by large margins... over 40,000 hands
- Results against humans are mixed, ranging from brilliant to mediocre
- Vexbot needs to learn quickly and start winning within 50 hands
Bayesbot

- Use a different model of a game based on probability distributions
- Accelerate learning by classification
- For each player (parameter space):
  - Level 1 (4 options):
    - Betting frequency, folding frequency, bluffing frequency, trapping frequency
  - Level 2 (4 options):
    - Preflop, flop, turn, river
  - Level 3 (4 options):
    - Bet level (bet, raise, re-raise, cap betting)
Bayesbot

Algorithm
- Start with a random model
- Play a hand
- Use Bayesian inference to a move to a model that better approximates the data seen
- Play a hand with the new model and repeat step above

Simplifications
- Subset of the complete abstraction model
- Use 10 values to approximate distribution

Early stages, but results are very promising
Conclusions

- Poker is a challenging AI domain (if only I had realized this in 1991)
- Optimal strategy is a misleading concept; maximal play will win more money
- Adaptive bots hold the key to world-class poker play
- Current program might be an even-money bet against a top player
Play Online!

http://games.cs.ualberta.ca/poker/
Better Yet... Buy it :) 

http://poki-poker.com
http://poker-academy.com
Publicity!