POKER AGENTS

University of Nebraska Lincoln

LD Miller & Adam Eck

April 14 & 19, 2011

Motivation

- Classic environment properties of MAS
  - Stochastic behavior (agents and environment)
  - Incomplete information
  - Uncertainty

- Application Examples
  - Robotics
  - Intelligent user interfaces
  - Decision support systems

Motivation

- Popular environment: Texas Hold'em poker
  - Enjoyed by users
  - Interaction with agents
  - Many solutions

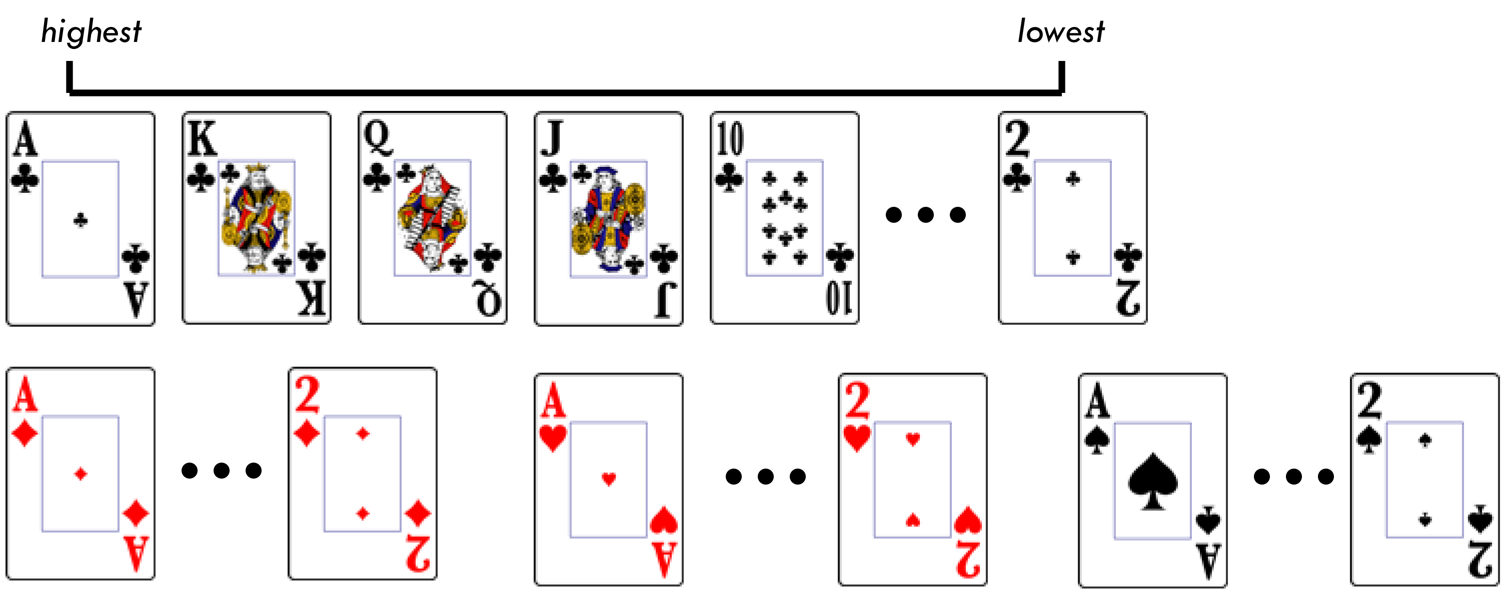
- Annual Computer Poker Challenge (ACPC)
  - Held with AAAI conference
  - Existing game framework
  - Competition!

Overview

- Background
- Methodology
- Results
- Conclusions

Background | Texas Hold'em Poker

- Variant of poker developed in Robstown, Texas in early 1900s
- Played with 52 card deck



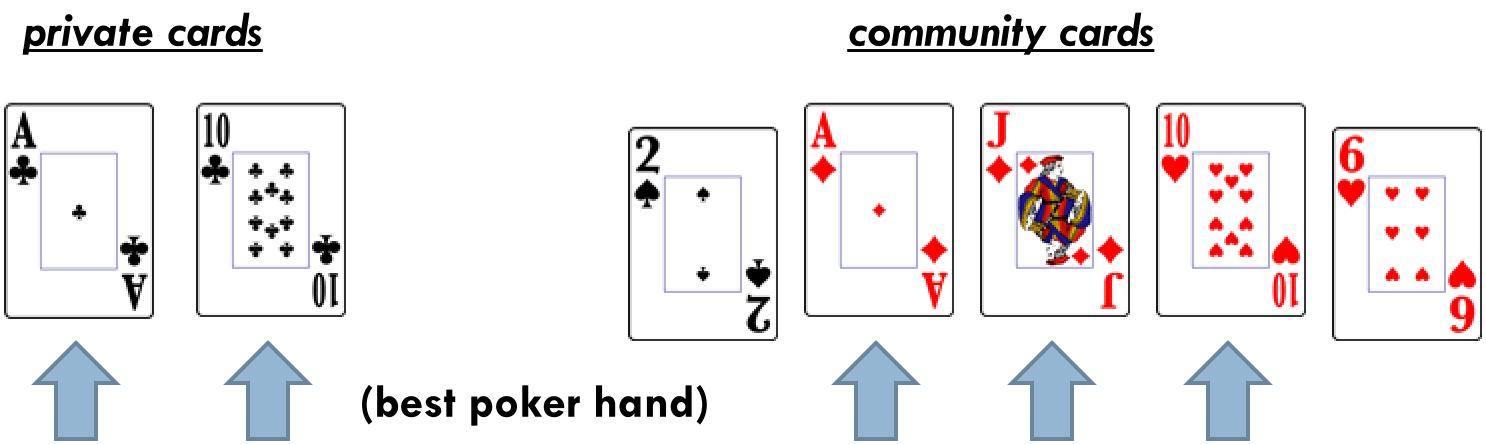
Background | Texas Hold'em Poker

- Ranking of poker hands

Source: http://www.learn-texas-holdem.com/

Background | Texas Hold’em Poker

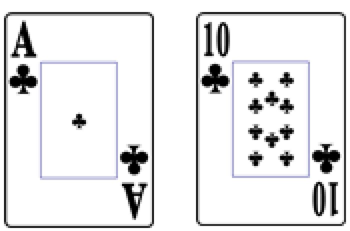
- Uses both 2 **private** and 5 **community** cards
- Construct the best possible poker hand out of 5 cards (use 3-5 community)



Background | Texas Hold'em Poker

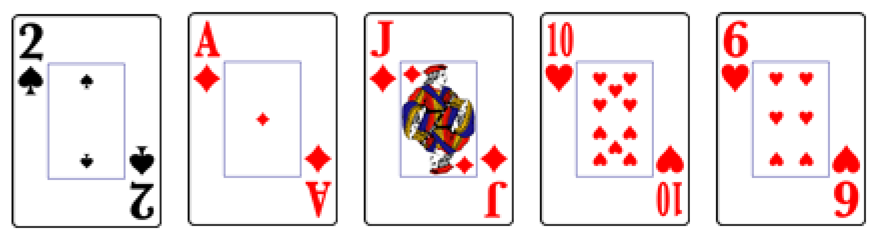
- Games consist of 4 different **steps**
- Actions: bet (check, raise, call) and fold
  - Bets can be **limited** or unlimited

private cards



(1) **pre-flop**

community cards



(2) **flop** (3) **turn** (4) **river**

Background | Texas Hold'em Poker

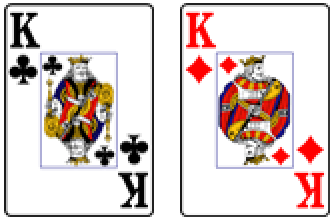
- Significant worldwide popularity and revenue
  - World Series of Poker (WSOP) attracted **63,706** players in 2010 (WSOP, 2010)
  - Online sites generated estimated **$20 billion** in 2007 (Economist, 2007)

- Has fortuitous mix of strategy and luck
  - Community cards allow for more accurate modeling
  - Still many "outs" or remaining community cards which defeat strong hands

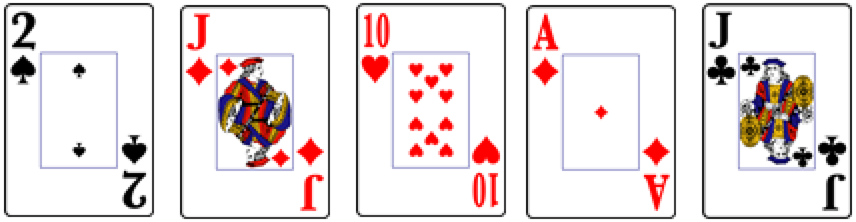
Background | Texas Hold'em Poker

- Strategy depends on **hand strength** which changes from step to step!
  - Hands which were strong early in the game may get weaker (and vice-versa) as cards are dealt

private cards



community cards



raise! **raise!** **check?** **fold?**

Background | Texas Hold'em Poker

- Strategy also depends on **betting behavior**
- Three different types (Smith, 2009):
  - Aggressive players who often bet/raise to force folds
  - Optimistic players who often call to stay in hands
  - Conservative or "tight" players who often fold unless they have really strong hands

Methodology | Strategies

- Problem: provide basic strategies that simulate betting behavior types
  - Must include hand strength
  - Must incorporate stochastic variance or “gut feelings”
    - Action: fold/call with high/low hand strength

Methodology | Strategies

- Solution 1: use separate mixture models for each type
  - All three models use the **same** set of three tactics for weak, medium, and strong hands
    - Each tactic uses a different probability distribution for actions (raise, check, fold)
  - However, each model has a **different** idea what hand strength constitutes a weak, medium, and strong hand!

Methodology | Strategies

- Solution 2: Probability distributions
  - Hand strength measured using Poker Prophesier (http://www.javaflair.com/pp/)

(1) Check hand strength for tactic

Behavior	Weak	Medium	Strong
Aggressive	[0…0.2)	[0.2…0.6)	[0.6…1)
Optimistic	[0…0.5)	[0.5…0.9)	[0.9…1)
Conservative	[0…0.3)	[0.3…0.8)	[0.8…1)

(2) "Roll" on tactic for action

Tactic	Fold	Call	Raise
Weak	[0…0.7)	[0.7…0.95)	[0.95…1)
Medium	[0…0.3)	[0.3…0.7)	[0.7…1)
Strong	[0…0.05)	[0.05…0.3)	[0.3…1)

Methodology | Meta-strategies

- Problem: basic strategies are very simplistic
  - Little emphasis on **deception**
  - Don’t **adapt** to opponent
- Consider four meta-strategies
  - Two as baselines
  - Two as active AI research

Methodology | Deceptive Agent

- Problem 1: Agents don't explicitly **deceive**
  - Reveal strategy every action
  - Easy to model

- Solution: alternate strategies periodically
  - Conservative to aggressive and vice-versa
  - Break opponent modeling (concept shift)

Methodology | Explore/Exploit

- Problem 2: Basic agents don't **adapt**
  - Ignore opponent behavior
  - Static strategies

- Solution: use reinforcement learning (RL)
  - Implicitly model opponents
  - Revise action probabilities
  - **Explore** space of strategies, then **exploit** success

Methodology | Explore/Exploit

- RL formulation of poker problem
  - State s: hand strength
    - Discretized into 10 values
  - Action a: betting behavior
    - Fold, Call, Raise
  - Reward R(s,a): change in bankroll
    - Updated after each hand
    - Assigns same reward to all actions in a hand

Methodology | Explore/Exploit

- Q-Learning algorithm
  - Discounted learning
  - Single-step only

- Explore/Exploit balance
  - Choose actions based on expected reward
  - Softmax
    - Probabilistic matching strategy
    - Used by humans (Daw et. al, 2006)
    - Roulette selection

$$P(a|s) = \frac{e^{\frac{R(s,a)}{T}}}{\sum_{a' \in A} e^{\frac{R(s,a')}{T}}}$$

Methodology | Active Sensing

- Opponent modeling
  - Another approach to **adaptation**
  - Want to understand and predict opponent's actions
  - **Explicit** rather than implicit (RL)

- Primary focus of previous work on AI poker
  - Not proposing a new modeling technique
    - Adapt existing techniques to basic agent design
  - Vehicle for fundamental agent research

Methodology | Active Sensing

- Opponent model = knowledge
  - Refined through observations
    - Betting history, opponent's cards
  - Actions produce observations
    - **Information is not free**
- Tradeoff in action selection
  - Current vs. future hand winnings/losses
  - Sacrifice vs. gain

Methodology | Active Sensing

- Knowledge representation
  - Set of Dirichlet probability distributions
    - Frequency counting approach
    - Opponent state s° = their estimated hand strength
    - Observed opponent action a°

$$P(a|s^o) = \frac{c(s^o, a^o)}{\sum_{a^{o\prime} \in A} c(s^o, a^{o\prime})}$$

- Opponent state
  - Calculated at end of hand (if cards revealed)
  - Otherwise 1 – s
    - Considers all possible opponent hands

Methodology | Active Sensing

- Challenge: how to choose actions?
  - Goal 1: Win current hand
  - Goal 2: Win future hands (good modeling)
  - Goals can be conflicting

- Another exploration/exploitation problem!
  - Explore: learn opponent model
  - Exploit: use model in current hand

Methodology | Active Sensing

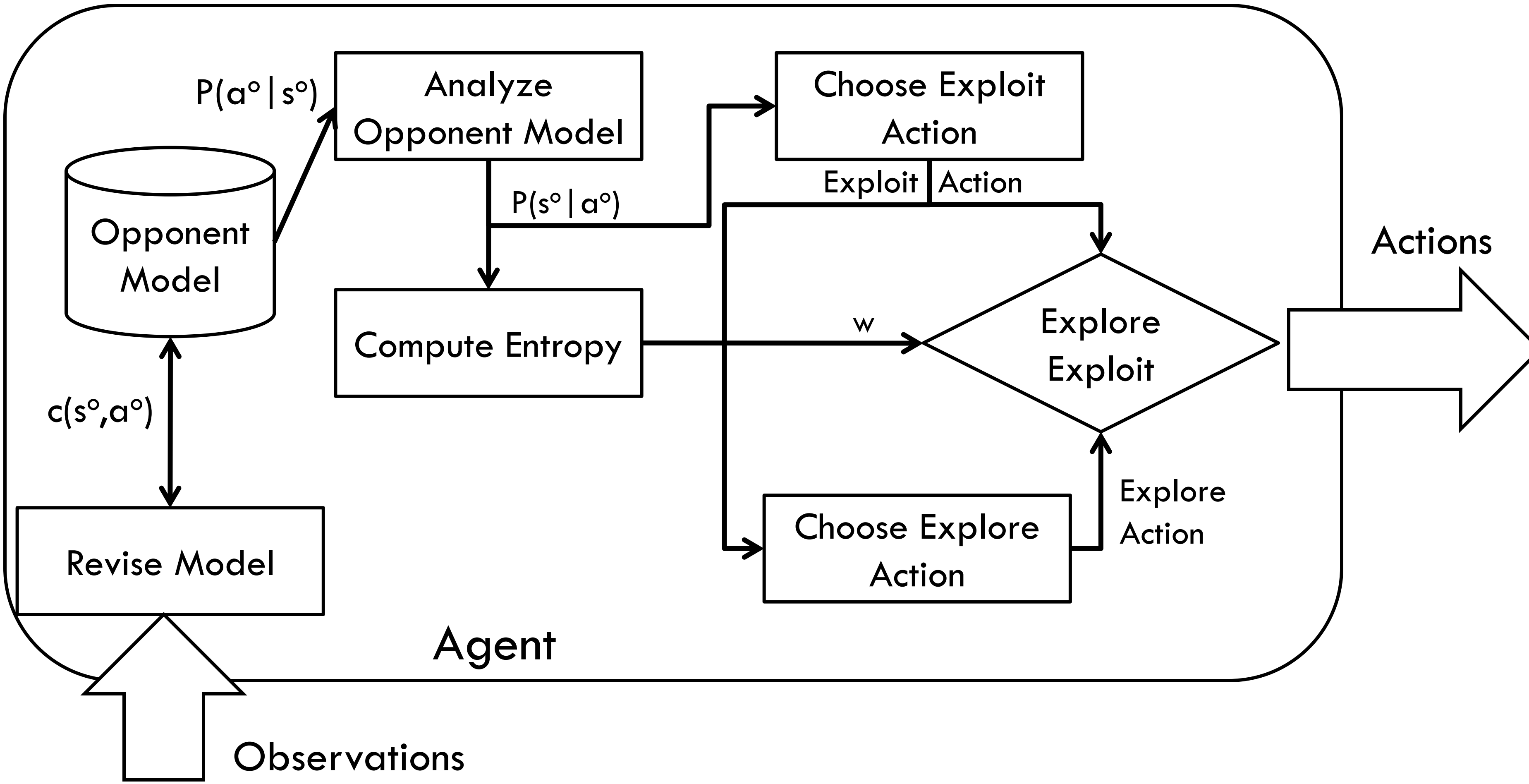
- Exploitation
  - Use opponent actions to revise hand strength model
    - Have $P(a^\circ | s^\circ)$
    - Estimate $P(s^\circ | a^\circ)$
    - Use Bayes rule
      - $P(s^\circ | a^\circ) = P(s^\circ | a^\circ)\, P(a^\circ) / P(s^\circ)$
  - Action selection
    - Raise if our hand strength >> $E[P(s^\circ | a^\circ)]$
    - Call if our hand strengh $\approx E[P(s^\circ | a^\circ)]$
    - Fold if our hand strength << $E[P(s^\circ | a^\circ)]$

Methodology | Active Sensing

- Use adaptive ε-greedy approach
  - Explore with probability $w * \varepsilon$
  - Exploit with probability $1 - w * \varepsilon$

- Control adaptive exploration through w
  - w = entropy of $P(s^\circ | a^\circ)$
  - High when probabilities most similar
    - High uncertainty
  - Low when probabilites diverse
    - Low uncertainty

Methodology | Active Sensing



Methodology | BoU

- Problem 1: Current strategies (basic and EE) focus only on hand strength
  - No thought given to other “features” such as betting sequence, pot odds, etc.
  - No thought given to previous hands against same opponent
- Such a myopic approach limits the reasoning capability for such agents
- Solution 1: Strategy should consider entire “session” including all the above features

Methodology | BoU

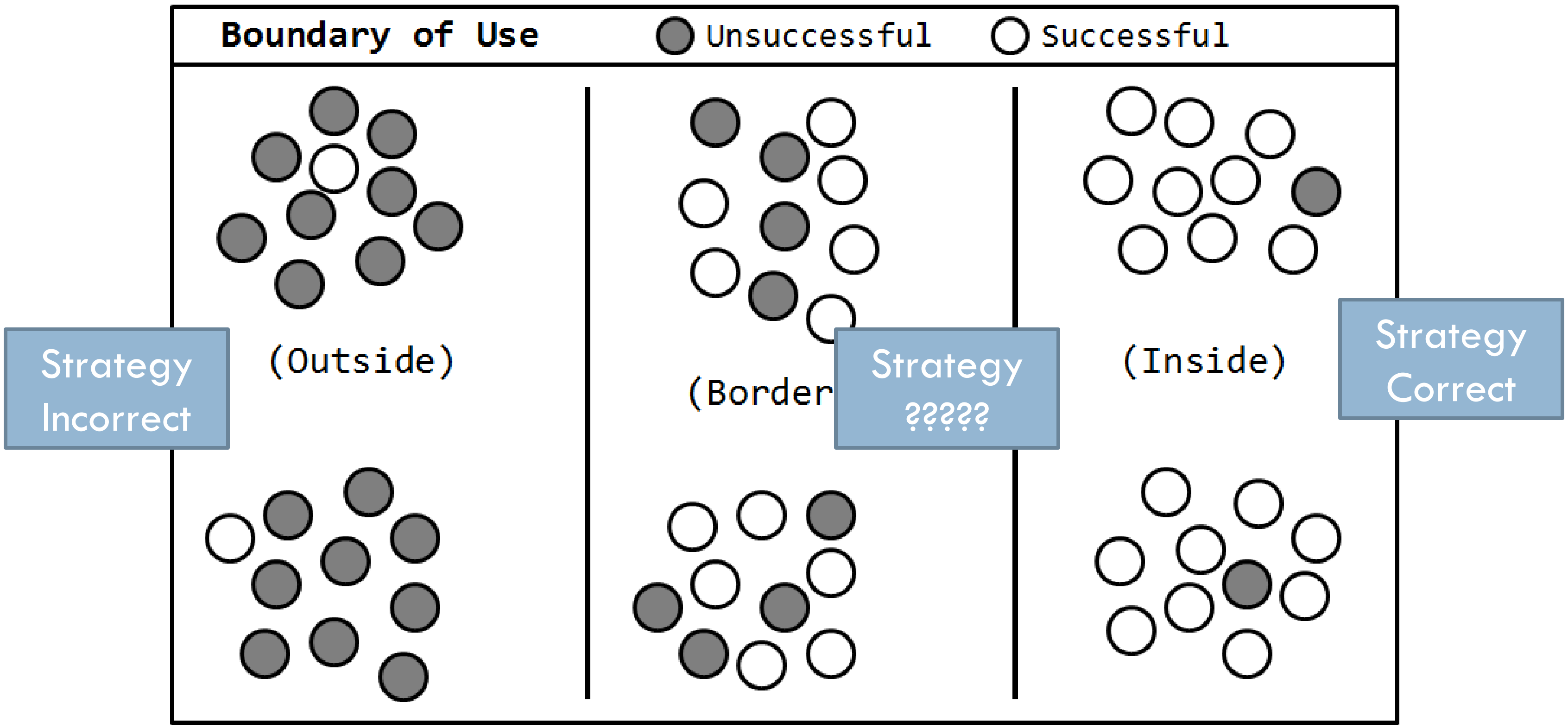
- Problem 2: Different strategies may only be effective against certain opponents
  - Example: Doyle Brunson has won 2 WSOP with 7-2 off suit—**worst** possible starting hand
  - Example: An aggressive strategy is detrimental when opponent knows you are aggressive
- Solution 2: Choose the "correct" strategy based on the **previous** sessions

Methodology | BoU

- Approach 2: Find the Boundary of Use (BoU) for the strategies based on previously collected sessions
  - BoU partitions sessions into three types of regions (successful, unsuccessful, mixed) based on the session outcome
  - Session outcome—**complex** and **independent** of strategy
- Choose the correct strategy for new hands based on region membership

Methodology | BoU

- BoU Example



- Ideal: All sessions inside the BoU

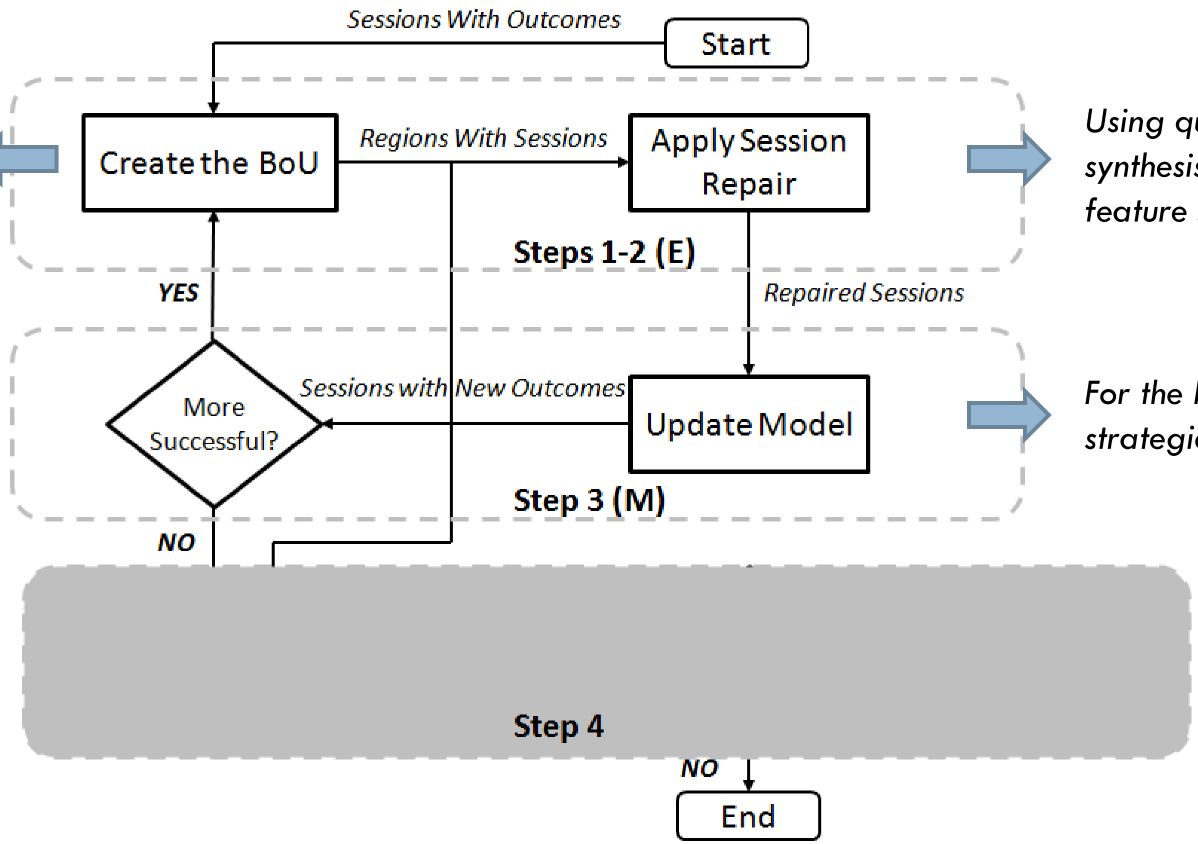
Methodology | BoU

- Approach 2. Improve the BoU using focused refinement (on mixed regions)
  - Repair session data to make it more beneficial for choosing the strategy
    - Active learning
    - Feature selection
  - Update the strategies chosen (based on the "repaired" sessions) which may change outcome

Methodology | BoU

- BoU Framework

Based on previous poker sessions



Using query synthesis and feature selection

For the basic strategies

Methodology | BoU

- Challenges (to be addressed)
  - How do we determine numeric outcomes?
    - Amount won/lost per hand
    - Correct action taken for each step
  - How do we assign region types to numeric outcomes?
    - Should a session with +120 outcome and a session with +10 both be in **successful** region?
  - How do we update outcomes using the strategies?
    - Say we switch from conservative to aggressive so the agent would not have folded
    - How do we **simulate** the rest of the hand to get the session outcome?

Methodology | BoU

- BoU Implementation
  - *k*-Means clustering
    - Similarity metric needs to be modified to incorporate **action sequences** AND **missing values**
    - Number of clusters used must balance cluster purity and coverage
  - Session repair
    - Genetic search for subsets of features contributing the most to session outcome
    - Query synthesis for **additional hands** in mixed regions

Results | Overview

- Validation
  - Basic agent vs. other basic (DONE)
  - EE agent vs. basic agents (DONE)
  - Deceptive agent vs. EE agent
- Investigation
  - AS agent vs. EE/deceptive agents
  - BoU agent vs. EE/deceptive agents
  - AS agent vs. BoU agent
    - **Ultimate showdown**

Results | Simple Agent Validation

- Simple Agent Hypotheses
  - SA-H1: None of these strategies will "dominate" all the others
  - SA-H2: Stochastic variance will allow an agent to win overall against another with the same strategy

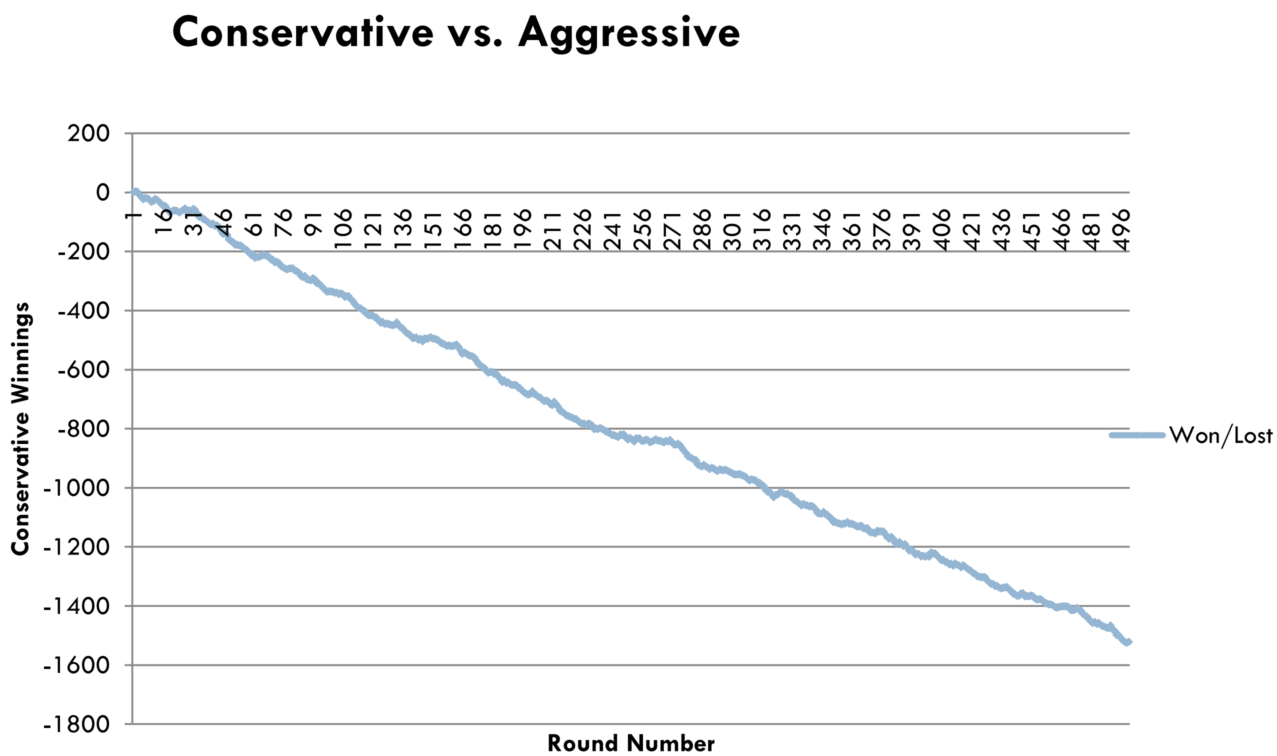
- Parameters
  - Hands = 500
  - Seeds = 30

Results | Simple Agent Validation

- Matchups
  - Conservative vs. Aggressive (DONE)
  - Aggressive vs. Optimistic (DONE)
  - Optimistic vs. Conservative (DONE)
  - Aggressive vs. Aggressive (DONE)
  - Optimistic vs. Optimistic (DONE)
  - Conservative vs. Conservative (DONE)

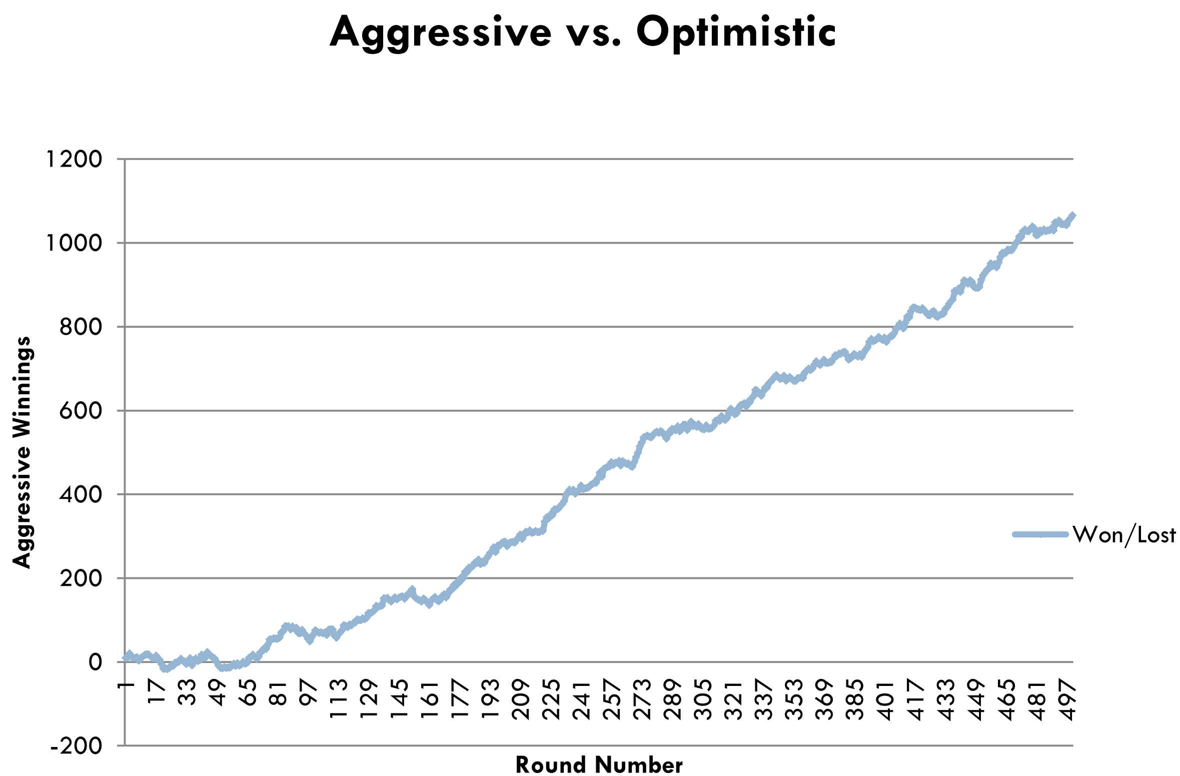
Results | Simple Agent Validation

- Matchup 1: Conservative vs. Aggressive



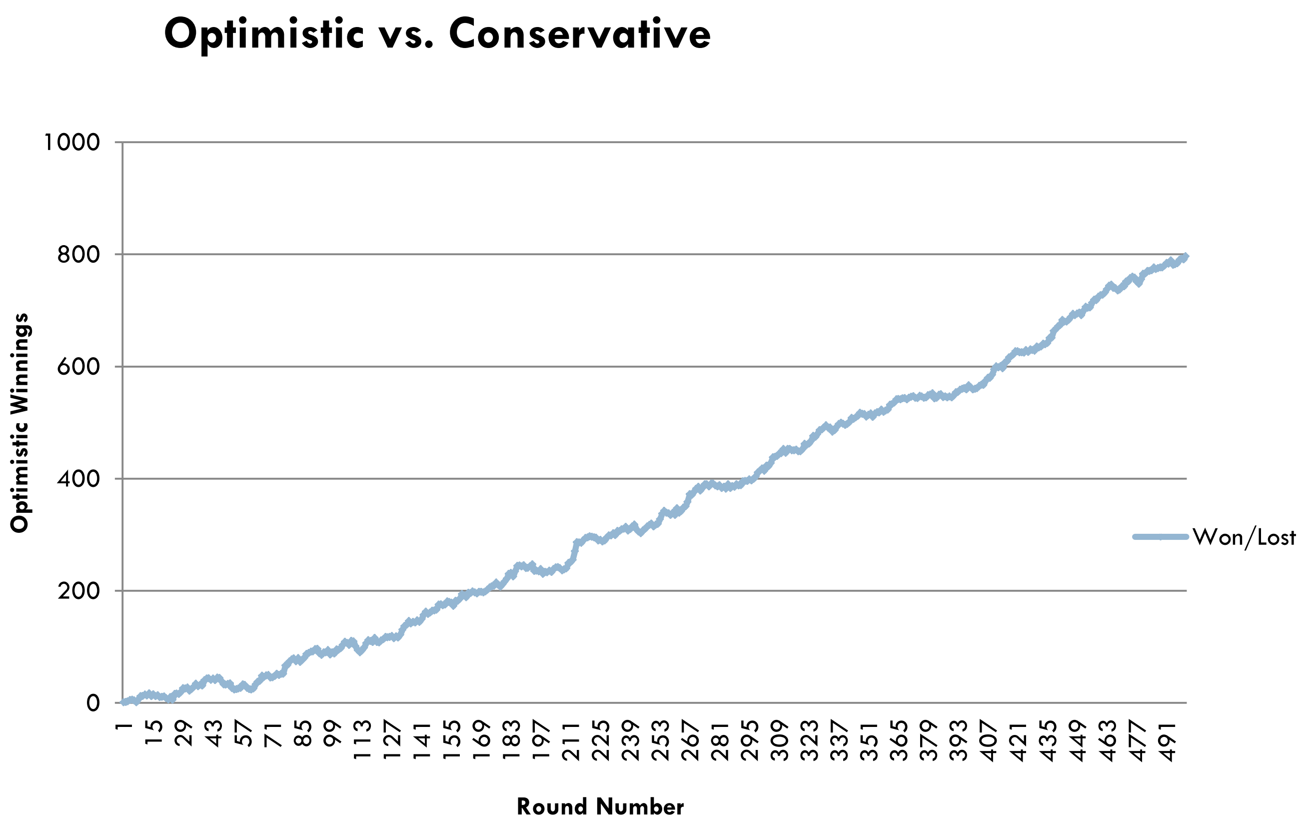
Results | Simple Agent Validation

- Matchup 2: Aggressive vs. Optimistic



Results | Simple Agent Validation

- Matchup 3: Optimistic vs. Conservative



Results | EE Validation

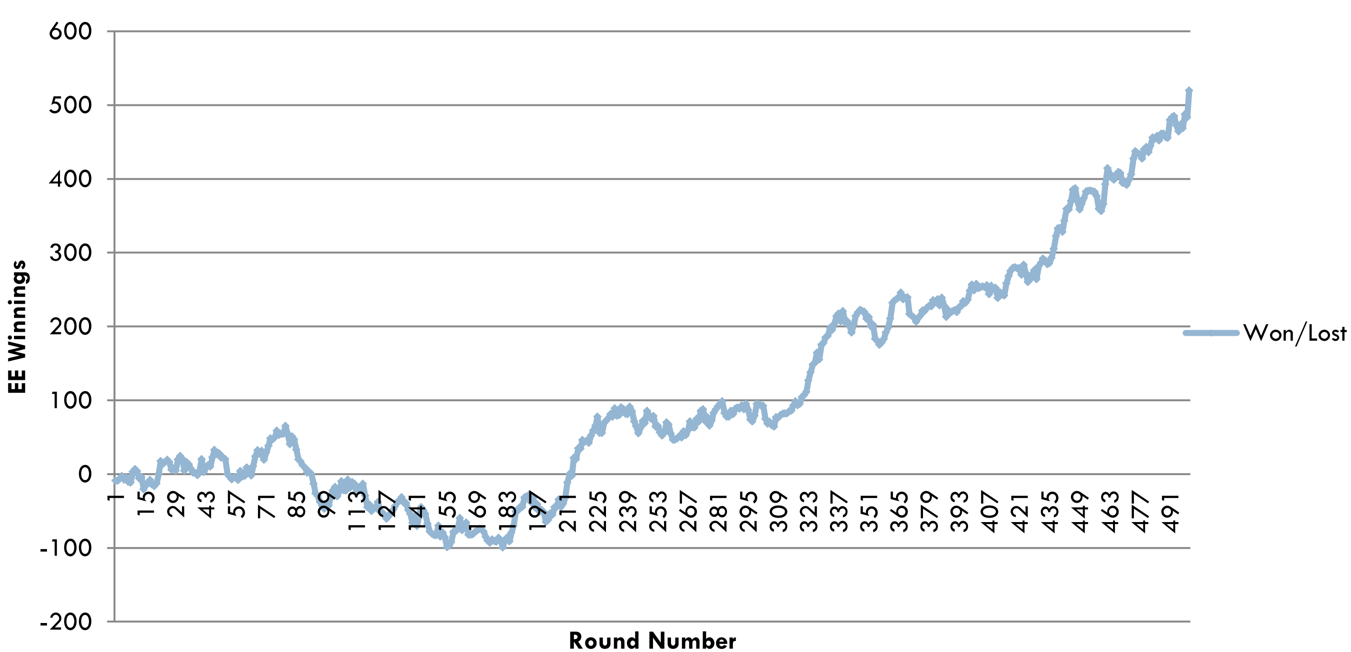
- EE Hypotheses
  - EE-H1: Explore/exploit will lose money early while it is exploring
  - EE-H2: Explore/exploit will eventually adapt and choose actions which exploit simple agents to improve its overall winnings

- Parameters
  - Hands = 500
  - Seeds = 30
  - Learning Rate = Discounted

Results | EE Validation

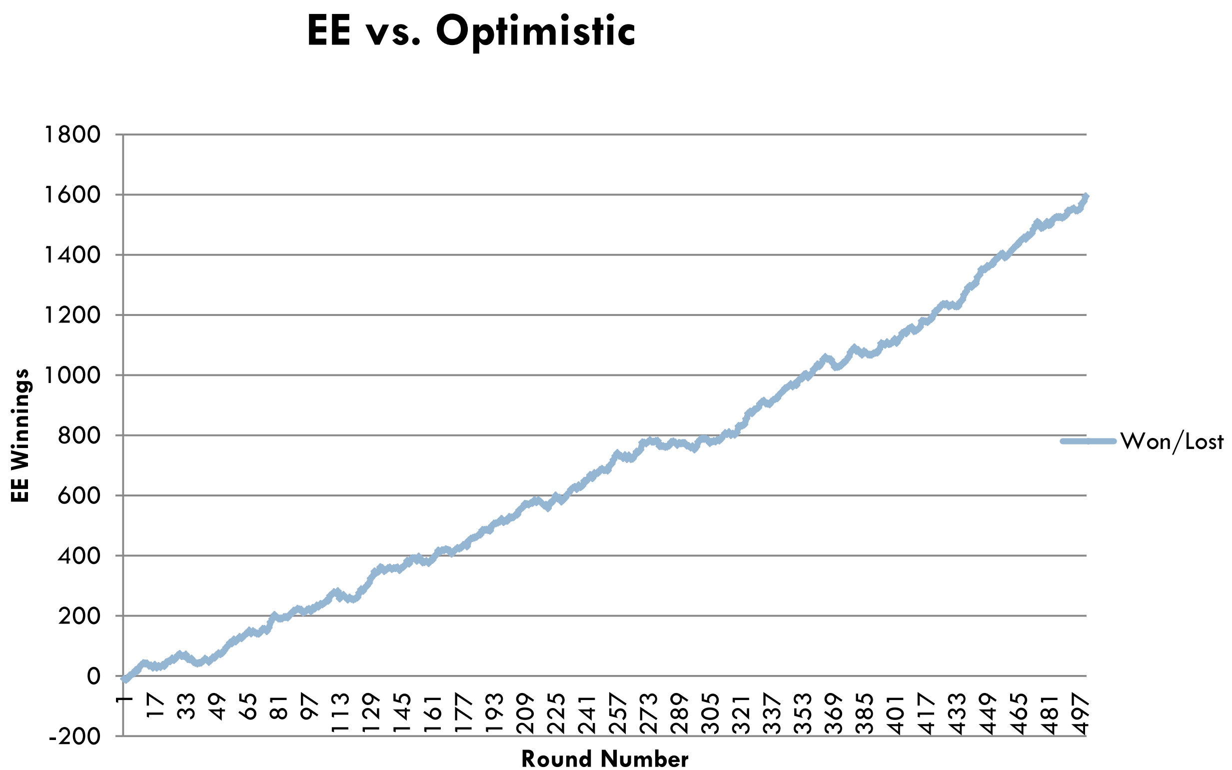
- Matchup 1: EE vs. Aggressive

EE vs. Aggressive



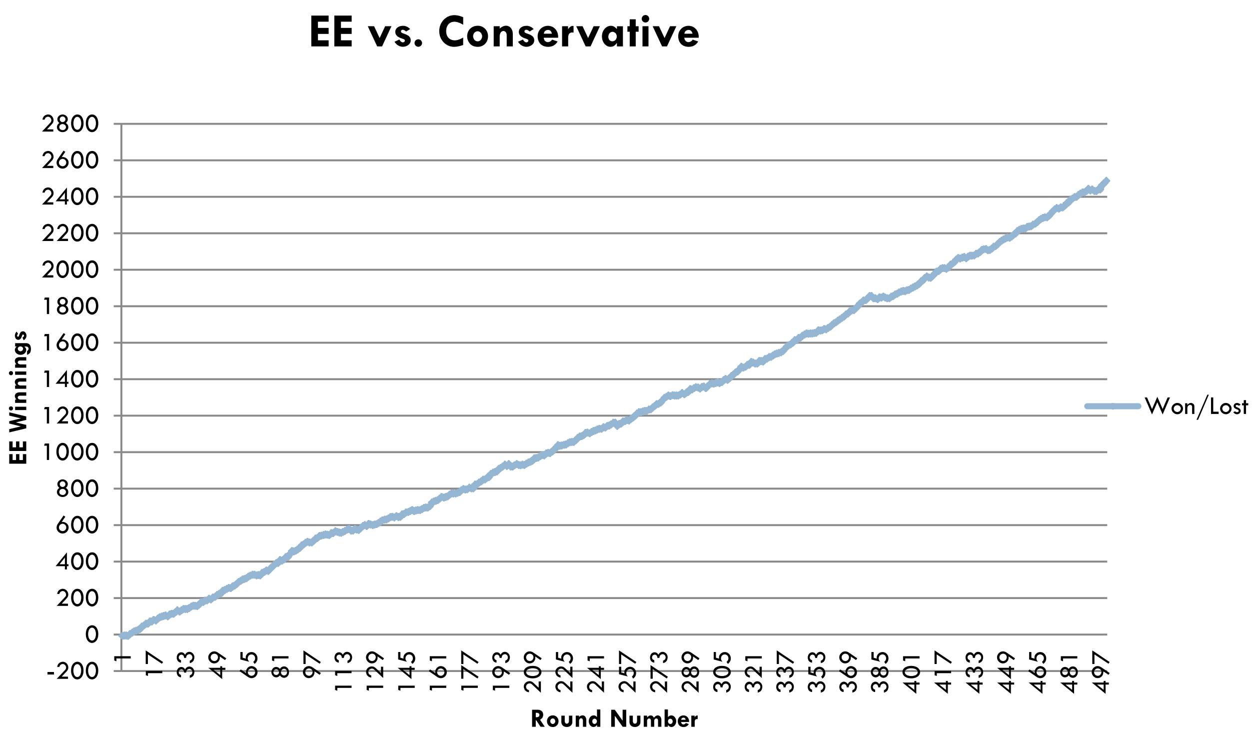
Results | EE Validation

- Matchup 2: EE vs. Optimistic



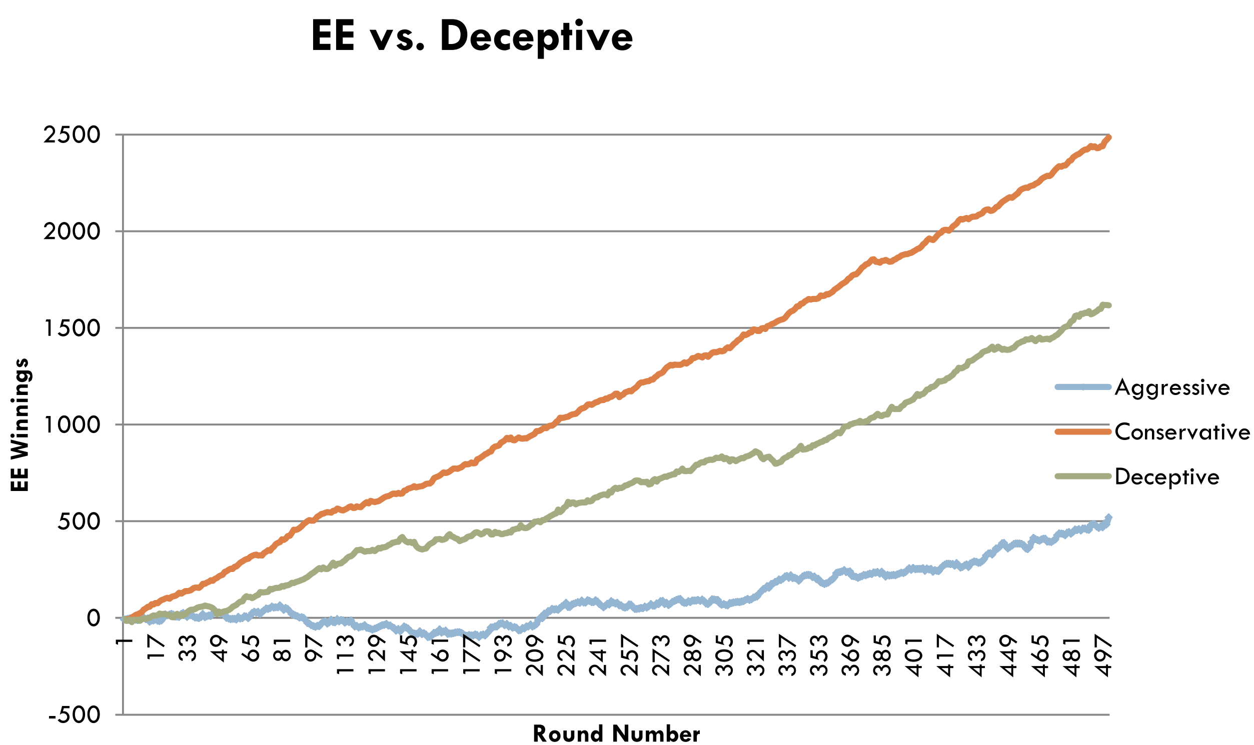
Results | EE Validation

- Matchup 3: EE vs. Conservative



Results | EE Validation

- Matchup 4: EE vs. Deceptive



Results | Active Sensing Setup

- Active Sensing Hypotheses
  - AS-H1: Including opponent modeling will improve agent winnings
  - AS-H2: Using AS to boost opponent modeling will improve agent winnings over non-AS opponent modeling

- Open questions:
  - How is agent performance affected by:
    - ε values?
    - Other opponent performs modeling?

Results | AS Setup

- Parameters
  - $\varepsilon = 0.0, 0.1, 0.2$
- Opponents
  - EE: implicit vs. explicit modeling, dynamic opponent
  - Deceptive: shifting opponent
  - Non-AS: effect of opponent's modeling
  - BOU: Offline learning/modeling

Results | BoU Setup

- BoU Hypotheses
  - BoU-H1: Including additional session information should improve agent reasoning
  - BoU-H2: Using the BoU to choose the correct strategy should improve winnings over agents which only use hand strength
- BoU Data Collection
  - Simple agent validation
  - Crowdsourcing agents vs. humans

Conclusion | Remaining Work

- Finish implementing AS
- Finish implementing BOU
- Run AS/BOU Experiments
- POJI results

Conclusion | Summary

- Introduced poker as an AI problem
- Described various agent strategies
  - Basic
  - Need for meta-strategies
  - AS/BOU
- Introduced experimental setup
  - Early validation results

Questions?

Demonstration

References

- (Daw et al., 2006) N.D. Daw et. al, 2006. Cortical substrates for exploratory decisions in humans, *Nature*, 441:876-879.
- (Economist, 2007) Poker: A big deal, *Economist*, Retrieved January 11, 2011, from http://www.economist.com/node/10281315?story_id=10281315, 2007.
- (Smith, 2009) Smith, G., Levere, M., and Kurtzman, R. Poker player behavior after big wins and big losses, *Management Science*, pp. 1547-1555, 2009.
- (WSOP, 2010) 2010 World series of poker shatters attendance records, Retrieved January 11, 2011, from http://www.wsop.com/news/2010/Jul/2962/2010-WORLD-SERIES-OF-POKER-SHATTERS-ATTENDANCE-RECORD.html

Acknowledgements

- Playing card images from David Bellot: http://www.eludication.org/playingcards.html#