AI for 2-Person Games: Adversarial Search

Keith L. Downing

January 16, 2023
AI is what humans are currently better at ... Jim Hendler
Games are a core part of AI’s history, and of AI present, even though bots now beat humans at many of them.
Classes of Games

- **Perfect Info** - state of playing arena and other players holdings known at all times.
- **Imperfect Info** - some info about arena or players not available.
- **Deterministic** - outcome of any action is certain.
- **Stochastic** - some actions have probabilistic outcomes, e.g. dealt cards, rolled dice, etc.

![Diagram of Classes of Games]

Keith L. Downing | AI for 2-Person Games: Adversarial Search
Top-Down -vs- Bottom-Up AI

Top-Down: Theory Driven

Human Expert

AI (E.g. Expert System)

Automated Problem Solver

Bottom-Up: Data Driven

AI (E.g. Neural Network)

World Experience (i.e. Data)
Early AI Poker (D. Waterman, 1968)

- 5-card draw using an expert system.
- Uses deep knowledge of the game and human strategy.

**Dynamic State Vector (DSV)**

(VDHAND, POT, LASTBET, BLUFFO, POTBET, ORP, OSTYLE)

**Game State**

**Action Rule**

- Action: Fold, Call, Bet

**Interpretation Rule**

- Updated DSV

Keith L. Downing
AI for 2-Person Games: Adversarial Search
1. Play poker, interview experts, read strategy books, etc.
2. Then produce a large set of situation-action rules:

**Example Rule 52**

IF your hole cards are a pair of face cards, AND the third such face card appears in the flop, THEN slow play the hand (i.e. do not raise, only call).

**Example Rule 1073**

IF it is the final (‘river’) round of betting, AND an opponent raises before you, AND the pot odds are favorable THEN call the raise.
Chance-Event Rollouts

- Offline: simulate millions of possible chance events (i.e. dealt cards) to assess winning probabilities of general game states.
- Online: simulates thousands of chance events to compute winning probabilities for specific game states.
- No deep knowledge of game; just brute force calc + simple statistics.

![Diagram of poker card rollouts](image-url)
Recent AI Poker: DeepStack (Moravcik et. al. 2017)

- Heads-up no-limit (HUNL) poker. Heads-up = just 2 players; no-limit = bets can be of any size (though the total amount bet in a game is still limited).
- DeepStack combines deep learning (DL) and self-play to achieve expert-level performance.
- The approach is very similar to deep reinforcement learning (DRL).
- It computes Nash-Equilibrium strategies for the states that arise in poker games via resolving. So although it does not solve Poker, it does solve for individual poker states by finding minimally-exploitable strategies.
- It has a strong mathematical foundation. It was the first theoretically-sound heuristic approach to imperfect-information games.

Chance-Event and Player-Action Rollouts

- Offline: Simulate millions of possible chance events (i.e. dealt cards) and **player actions** (fold, call, raise, etc.) to train neural networks that serve as heuristic functions: given probability distributions over hidden states, produce probability distributions for winning.

- Online: Use trained NNs plus thousands of simulated chance events and player actions to compute best-response moves for specific game states.

- Still no deep knowledge of game; just brute force calc + simple statistics. But now, the statistics encompass action strategies (a.k.a., policies) and beliefs about the opponent (in terms of probability distributions over possible hidden states).
DeepStack Rollouts

P1’s Range: 0.16 0.085 0.004 ...

Sampled Private Info:

P2 Hole Cards:

P1's Range (modified on the way down to the leaf), along with the sample private and public info.

Sampled Public Info:

Flop:

Turn:

River:

The expected winnings for Player 2 based on P1’s Range, along with the sample private and public info.
Modern AI poker players (e.g. DeepStack) find Game Theory Optimal (GTO) strategies.

Even though the AI pokerbot knows its own cards, it calculates with a range, not a single pair.

This reflects knowledge that the opponent could have about AI’s hole cards, based on public cards and the pokerbots recent actions.

So the AI’s strategies are based on what it thinks the opponent knows about the AI’s cards. **Beliefs are key!**
* When using a solver, this strategy is generated from scratch for each game state: the AI resolves at each decision point.
Resolving: Bootstrapped Strategy Learning

Start at Root

Traverse tree many times

Save root node's strategy after each traversal

Basis for whole search tree

Strategies

Elides through entire search tree

Average Strategy

Start at Root

Range Values

Regrets

Strategy Series

Current Game State: S

Ranges for S

Inputs

Next Action (a*)

Next Game State: S(a*)

Outputs

Can return as inputs during future calls to re-solve during the same hand of poker

Traverse tree many times

Can return as inputs during future calls to re-solve during the same hand of poker
1967 (IBM) Samuel's Checker's player learned own evaluation function by playing against itself.

1990 (University of Alberta) Chinook developed by Jonathan Schaeffer's group.

Blondie 24 (Fogel, 2001) - Evolved a Checker's player with very high ranking on zone.com. Beats Chinook once.

2004 - Chinook beats best human and becomes world champion.

2007 - (University of Alberta) - Schaeffer's group proves all Checkers games can be a tie if played perfectly by each side.

- 18 years of computer runtime.
- $5 \times 10^{20}$ positions were evaluated. Chess has $\approx 10^{40}$.
- "It's been 18 years!...obsessive-compulsive behavior...not normal ...Get a life, Jonathan." ... his wife.

*The Quest for Artificial Intelligence*, Nilsson (2010).
AI and Chess

- 1959-1962 (MIT) - *Most of the machine’s moves are neither brilliant nor stupid. It must be admitted that it occasionally blunders* - John McCarthy

- 1967 (Stanford + Moscow) - Russia -vs- USA; McCarthy + Kotok’s group -vs- Adelson-Velskiy’s group. USSR won.

- 1967 (MIT) - Greenblatt’s MAC HACK VI was first computer to play in a human tournament. Won 2; Tied 2 to achieve an amateur rating.

- 1967 (MIT) - MAC HACK VI beat Hubert Dreyfus, a renowned critic of AI.

- 1970 (Northwestern University) - CHESS 3.0 won first ACM computer chess tournament. Evaluated 100 states per second.

- 1974 (Stockholm) - Kaissa (Russian) became first world computer chess champion.

- Chess has been called the *Drosophila* of AI.

*The Quest for Artificial Intelligence*, Nilsson (2010).
A human uses prodigious amounts of knowledge in the pattern-recognition process and a small amount of calculation to verify the fact that the proposed solution is good in the present instance....However, the computer would make the same maneuver because it found at the end of a very large search that it was the most advantageous way to proceed out of the hundreds of thousands of possibilities it looked at. CHESS 4.6 has to date made several well known maneuvers without having the slightest knowledge of the maneuver, the conditions for its applications, and so on; but only knowing that the end result of the maneuver was good.... Hans Berliner (Nature, 1978) *The Quest for Artificial Intelligence*, Nilsson (2010).

Computers don’t really understand chess the ways humans do, but they search thoroughly and quickly. Does this matter?

McCarthy suggests tournaments where computers have limited computation time, like a millisecond.
Deep Blue

- Lost to Kasparov (reigning world champion) in 1996. IBM then improved it to handle some of its weaknesses. Beat Kasparov in 1997.
- I am a human being. When I see something that is well beyond my understanding, I’m afraid...Kasparov.
- State evaluations per second: Deep Blue - 200 million; Kasparov - 3
- Deep chess knowledge: Deep Blue - very little; Kasparov - volumes.
- But Deep Blue had *memorized* thousands of book moves and grandmaster games.
- 8000 features are analyzed by Deep Blue. Fast evaluation only checks for the simplest of them (like piece count), while a slow evaluation checks for complex patterns (like king safety, center control, etc.).
- Deep Blue uses no machine learning.
- Deep Blue was dismantled after 1997 match.
- Claimed to use no AI, just *brute force search* (with heuristics): minimax with alpha-beta pruning, which McCarthy deems essential.

*The Quest for Artificial Intelligence*, Nilsson (2010).
Adversarial Search Trees

Similar to an OR node in AND/OR Search. I can make ANY of these choices.

Similar to AND nodes in AND/OR. I must account for ALL of the opponent's possible moves.

Evaluate bottom states and propagate values upward.
Best-First Search (e.g. A*) - Review

- Generate **one** large search tree.
- Use heuristics to evaluate the promise of **all** states in the tree.
- When a goal state is found, return the **entire path** from the start state as the solution.
1. From the start state, generate a search tree in a depth-limited, depth-first manner, alternating between own and opponent’s possible moves.

2. Only use heuristics to evaluate the promise of the bottom-level nodes; then propagate values upwards, combining them using MAX and MIN operators.

3. Return to the root and choose the action, $A_{best}$, leading to the highest-rated child state, $S_{best}$.

4. Apply $A_{best}$ to the current game state, producing $S_{best}$.

5. Wait for the opponent to choose an action, which then produces the new game state, $S_{new}$.

6. Generate a **whole new** search tree, with root state = $S_{new}$.

7. GO TO step 2.
Pure Minimax: Lookahead

Apply Heuristic Function

All scores are from the perspective of the top (max) player.
Pure Minimax: Backup

These subtrees did not need to be generated.
Saving Evaluations in Min-Max Trees

Keith L. Downing

AI for 2-Person Games: Adversarial Search
Alpha-Beta Pruning

MAX

MIN

MAX

5 < a => Stop generating children. This node’s value cannot win at the root. Prune it!

There could be many large subtrees down here, but there is no need to generate them, since they cannot affect the choice made at the root.
Actor-Critic AI Systems

- Critic = a system that evaluates search states. Very similar to both heuristics and objective functions.
- Many AI applications involve standard AI search algorithms (A*, Minimax, etc.) plus problem-specific critics. Building the critic is the hard part.
- Actor = a system for mapping search states to actions.
- Actor-Critic systems use AI to both compute actions and evaluate states. In some versions, the best action is simply the one that leads to a state with the highest evaluation.
Actor-Critic Reinforcement Learning

Actor
- State(t)
- Action(t)
- reward(t+1)
- State(t+1)
- Action(t+1)

Critic
- TD error
- State(t), Action(t), Reward(t+1)
- State(t+1)
- Action(t+1)

Individual
- State(t), Action(t), Reward(t+1)
- State(t+1)
- Action(t+1)

Surroundings
- State(t+1)
- Reward(t+1)
- Action(t+1)
Neural Network as Critic

NN = Function Approximator ... instead of Lookup Table
ANN-based critic that improved via self-play.

Discovered new strategies that humans now use.

\[ V(S_t) = r_1 + V(S_{1t+1}) + r_2 + V(S_{2t+1}) + \ldots + r_k + V(S_{kt+1}) \]

\[ \delta_t = r^* + V(S^*_t) - V(S_t) \]

Do move $a^*$

Learn

How do you train these nets??
Monte Carlo Tree Search (MCTS)

- Similar to MiniMax: Generate a huge tree (but traverse it **many** times) as basis for selecting **one** move (from the root). Then retain the subtree rooted at the next state.
- Most AI GO players use MCTS.
- Originally designed for stochastic games: backgammon, poker and scrabble.
- Why use it for a deterministic game? The search space is so large, and most evaluation functions (i.e. heuristics) are so bad, that you need to try many alternate futures to assess the current state.
- AlphaGO combines MCTS with RL actors and critics produced by Deep Learning.

---

MCTS Dynamics

1. Rollout
   - Update Q(s,a) values
2. Rollout
   - Update
3. Leaf Expansion
4. More Rollouts
5. Eventually Choose Best Action from Root
6. Prune Tree

Eventually Choose Best Action from Root

Keith L. Downing
AI for 2-Person Games: Adversarial Search
The Monte Carlo Search Tree for GO

Expand
Rollout
Final State

Q + U: Q(s,a) + U(s,a)

Tree provides targets for training NN actor and critic.
AlphaGO = SL + RL + MCTS

8 million human cases

Actor-SL
Supervised Learning
Reinforcement Learning
Self-play

Actor-RL
Monte Carlo Tree Search (MCTS)

Critic-RL
More RL
Self-play

Supervised Learning
Actor-RO

Keith L. Downing
AI for 2-Person Games: Adversarial Search
October 2015: AlphaGo beats European champion Fan Hui, 5 - 0.

March 2016: AlphaGo beats a Grand Master (ranked 2nd in the world), Lee Sedol, 4-1.

- AlphaGo makes some moves that experts find very unusual, but prove to be brilliant.
- I guess I lost the game because I wasn’t able to find any weaknesses...Lee Sedol.
- Here in Korea, back in the U.S. and all over the world, there is much more sadness and introspection than I expected. Andrew Okun, President American Go Association.
- When you watch really great Go players play, it is like a thing of beauty. So I’m very excited that we’ve been able to instil that level of beauty inside a computer. Sergery Brin, Google.

May 2017: AlphaGo Master beats Ke Jie (World #1), 3 - 0.

- No expert knowledge: no supervised learning from expert games
- No human knowledge of proper features: minimal preprocessing of board states.
- A single neural network (actor + critic)
- No rollouts - uses critic to evaluate MCTS leaf nodes. Search complexity greatly reduced.
Neural Net Training in AlphaGO Zero

Actor + Critic Network

Action Probabilities

Evaluation

Targets for Supervised Learning

R = Reward at end of a completed game

Monte Carlo Search Tree

- No rollouts
  - Leaf nodes in MCTS are evaluated using the dual-purpose ANN to yield z values, which are then backed up.

So ANN training cases cannot be completed until each game ends.

Keith L. Downing

AI for 2-Person Games: Adversarial Search
3 days: Surpasses AlphaGO Lee (the one that beat Lee Sedol)

21 days: Surpasses AlphaGO Master (the one that beat Ke Jie)

40 days: Becomes best GO player in the world.
Mastering chess and shogi with self-play..., Silver et. al., arXiv, 2017.

- AlphaGO Zero **generalized** to play other games.
- Performance tests run on 5000+ TPUs (Tensor Processing Units)
- **2 hours**: Beat *Elmo* (world’s best shogi bot).
- **4 hours**: Beat *Stockfish* (world’s best chess bot).
- **8 hours**: Beat AlphaGO Lee.