EECS 391: Introduction to AI

Soumya Ray
Website: http://engr.case.edu/ray_soumya/eecs391_sp16/
Email: sray@case.edu
Office: Olin 516
Office hours: M 12:30-2pm
Announcements

• Quiz next Tuesday
  – 60 minutes
  – Up to and including search for optimization

• Read chapter 7 (propositional logic)
Today

• Adversarial Search (Chapter 5)
• Case study: Deep Blue
• (End of first part)
Adversarial search Algorithm

• Set \((\alpha, \beta) = (-\infty, +\infty)\)

• Perform standard minimax backup for each successor in turn

• If at a MIN node:
  – If backed-up value smaller than \(\alpha\), prune other successors
  – Else replace \(\beta\) with smaller of \(\beta\) and backed-up value

• If at a MAX node:
  – If backed-up value larger than \(\beta\), prune other successors
  – Else replace \(\alpha\) with larger of \(\alpha\) and backed-up value
Real-time Games

• So far, we have assumed both agents have enough time to exhaustively search the game tree

• In most games this is impossible, so they will be forced to stop early and make a move
  – How to best do this?
    • Iterative Deepening
    • Evaluation functions
Evaluation Functions

• If we are forced to stop the search at a non-terminal state, we can use heuristic evaluation functions to estimate what the backed up utility of that state could be
  – Similar idea as heuristics in goal-directed search
State Features

- To estimate the utility of a state, we compute *features* of a state
  - These are characteristics that we think might contribute to the overall utility
  - For example, in chess, we could have features that measured the advantage in pieces and control of the center
Weight of a feature

• Each feature has an associated weight, indicating how important it is
  – Note, weight can be negative
    • If negative and large, it is REALLY BAD to have a nonzero value for that feature
Linear Evaluation Functions

- A simple way to combine state features and express the heuristic utility of a state

\[ \hat{U}(s) = \sum_{i} w_i f_i(s) \]

- Assumes state features are additive
  - More complex functions are nonlinear e.g. neural networks (later)
Algorithm

• Set $(\alpha, \beta) = (-\infty, +\infty)$

• If (depth limit reached)
  – $V \leftarrow$ estimated value through heuristic; return $V$

• For each successor,
  – $V \leftarrow$ standard minimax backup
  – If at a MIN node:
    • If $V$ smaller than $\alpha$, prune other successors
    • Else replace $\beta$ with smaller of $\beta$ and $V$
  – If at a MAX node:
    • If $V$ larger than $\beta$, prune other successors
    • Else replace $\alpha$ with larger of $\alpha$ and $V$
Super famous game 1 of 1972
Bobby Fisher - Boris Spassky
Reykjavik cold war match

“Quiescent States”

Fisher

Spassky

bishop

pawn
Super famous game 1 of 1972
Bobby Fisher – Boris Spassky
Reykjavik cold war match

Black up 1 point
Super famous game 1 of 1972
Bobby Fisher – Boris Spassky
Reykjavik cold war match

Black bishop trapped
Still up material but bishop can’t attack, will eventually be captured
Issues with minimax (1)

• In some cases, there may be a “clear favorite” node, but minimax/α-β will still expand many other legal moves

• Along with a utility function, could have a function that evaluates the utility of expanding a node given what it has seen so far

• “Metareasoning”—thinking about the method of solution rather than the solution
Issues with minimax (2)

• Do humans play like this?
  – Maybe to some extent, but often not
  – Humans tend to be more goal-directed (long term strategic thinking)
  – This requires reasoning about the state---will see more of this in automated planning
Case Study: Deep Blue

• On May 11, 1997, in New York City.

2/2/2016 Soumya Ray, Case Western Reserve U. 16
How Does Tournament Chess Work?

• Chess time = 90m for 40 moves
  – And then 30m + 30s/move for rest
• Average time 2½ minutes
• Threefold repetition
  – Board in same state 3 times (not necessarily in a row)
• 50 moves with no capture or pawn moves
In the beginning...

• How well does minimax do in chess?
  – Chess: branching factor $b=35$
  – Suppose we could process 1,000,000 positions/sec
  – Chess time ≈ 2½ minutes
    • Can process 150m positions/move
  – What is our search depth?
  – Minimax time complexity $O(b^m)$
    • $35^4 = 1.5m$
    • $35^5 = 52m$ (5 plies = 2½ moves)
    • $35^6 = 1,838m$
  – One more ply – Importance?
    • Believed: one ply = 50-70 Elo points
Elo

- Way to rank chess players
  - Named after professor Arpad Elo, physicist
- Score based on opponents rating and a statistical model
- If win against opponent, some of their Elo points are transferred to you
  - How many depends on difference between your ratings
  - “Self correcting”

<table>
<thead>
<tr>
<th>Elo</th>
<th>Title</th>
<th>People</th>
</tr>
</thead>
<tbody>
<tr>
<td>2800</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2700</td>
<td></td>
<td>42</td>
</tr>
<tr>
<td>2600</td>
<td></td>
<td>178</td>
</tr>
<tr>
<td>2500</td>
<td>Intl. Grandmaster</td>
<td>587</td>
</tr>
<tr>
<td>2400</td>
<td>Intl. Master</td>
<td>1382</td>
</tr>
<tr>
<td>2300</td>
<td>FIDE Master</td>
<td>2998</td>
</tr>
<tr>
<td>2200</td>
<td>Candidate Master</td>
<td>5840</td>
</tr>
</tbody>
</table>
Alpha-Beta to the rescue

• Lets you ignore subtrees that are irrelevant
  – Can search deeper in same amount of time

• Best case: $O(b^{m/2})$
  – $b$ effectively becomes $b^{0.5}$
  – Chess $b = 35$ becomes $b = 6!$
  – $6^5 \approx 8k$, $6^6 \approx 48k$, $6^{10} \approx 60m$
  – 150m positions = 10 ply (5 moves)
    - Increase of 5 ply = 250-350 Elo
    - E.g. From Candidate Master to Intl Master/Grandmaster!

• Goal: Find accurate $\alpha$ and $\beta$ bounds early
  – Requires evaluating boundary (best) nodes first
Node ordering heuristics

1. Captures,
2. Threats,
3. Forward Moves,
4. Backwards Moves
Null Move Heuristic

• Generally every move made by opponent will improve their situation
  – Corollary: If I am able to skip my turn (give opponent free move) and I’m still really good, this position is probably unrealistic, so I will never realistically get here, so can skip examining further
Singular Extension Heuristic

• General idea – spend more time looking deeply at moves that are good
  – Not necessarily non-quiescent
• More specifically – look at “forced” moves
  – Includes “qualitatively forced”
• Forced 1 – you have no other choice
• Forced 2 – one move significantly better than all the rest
  – v=value, Δv=how much better, parameter we choose
• In general, SE turns out to be really good
  – Helped Deep Blue search up to 40 plies (!!) in some cases
State Evaluation Function

• Weighted linear function

\[ \hat{U}(s) = \sum_{i} w_i f_i(s) \]

• Features: pieces in center, pawn structure, king protection, etc.
  – Deep Blue had 8,000
    • Many respond to weird, rare situations
    • Written by chess experts (grandmasters)
How Else Can We Go Fast?

• Pondering
  – Search during *opponent’s* turn

• Representation
  – BitBoard

• Opening Book

• Endgame Database
Opening Book

• Start of game has a REALLY deep and wide search tree
• But, it’s the same every single time you play the game
• Every opponent normally plays the same moves
  – Well, one of a couple 100 patterns
• Why not just memorize the beginning?
• Works for roughly 10 moves
  – After that, end up in unexplored territory
Endgame Database

• Hard-coded moves
  – Experts know strategy for playing KRK

• Nalimov tables the most famous; on Web for free

• Heavily compressed to save room. Even so...
  – 5 piece – 7GB
  – 6 piece – 1.2 TB

• Issue: DB competes with search memory, transposition table
  – Less transpose = more searching / less search depth
Endgame Database (contd.)

• Pro: Can tell when you’re going to win
• Con: Can’t actually tell when you’ll lose or draw
  – Assumes opponent has perfect play
  – Humans screw up a lot, especially on long sequences
  – Computers often resign too early
• DB not say how easy/hard a pattern is
  – Some patterns very tricky, easy for human to screw up
  – Computer won’t pick those over easier ones (doesn’t model human opponent)
• Nalimov tables and others don’t know the 50-move rule
Legend of Deep Blue

- CMU had 2 chess enthusiasts
  - Professor Hans Berliner (chess/AI expert)
  - Student Feng-hsiung Hsu (hardware expert)
- 1989 IBM hires Hsu’s Deep Thought team
- 1996, Kasparov beats Deep Blue 4-2
  - Deep Blue does 100m positions/sec
  - Deep Blue says “I’ll be back” (metaphorically)
- 1997, Deep Blue upgraded for rematch
  - 200m positions/sec, #259 on TOP500
  - 30 P2SCs, 480 custom VLSI
  - P2SC single chip PowerPC CPU, 135MHz, 11.38 GFLOPS
    • Comparison: Single Intel i7-3770: 109 GFLOPS
Legend of Deep Blue (contd.)

- 200m positions/sec means Deep Blue looks at 30 billion positions per move (assuming 2½m per move)
- *Mostly hardware* but some software
  - Specifically Singular Extension and null move
- Depth limit =? (actual system unknown)
  - SE = 40
- Branching factor as low as 3
  - Down from 35 using *lots* of pruning
    - Null move heuristic, alpha-beta
- Position evaluation function – 8,000 features
- Opening book DB – 4,000 positions
- History DB – 700,000 grandmaster games
  - Selected by GMs Illescas, Fedorowicz and de Firmian
- Endgame DB – All 5 piece endings, many 6
Legend of Deep Blue (contd.)

• Round 1, Kasparov did not understand why Deep Blue resigned
  – Checking the last moves, he found there was a forced mate---in 20 moves!
    • “The conclusion was a little bit scary... Deep Blue had actually worked it all out, down to the very end and simply chosen the least obnoxious losing line. ... Garry [was] thankful that he had been on the right side of these awesome calculations.”—Frederic Friedel, attached to Kasparov’s team, chessbase.com

• Round 2: Kasparov loses
  • “Deep Blue was playing deep strategic chess, "just like Karpov", the experts on the stage agreed. They could hardly believe it was the same machine that lost yesterday's game. ...There were many mystifying moments in the game. [Kasparov] was especially puzzled by 36.Qb6 instead of 36.axb6, and 37.Be4, instead of 37.Qb6. No amount of coaxing would persuade Fritz4 or Hiarcs 6.0 to waver from Qb6 in both cases. Deep Blue must contain some highly sophisticated positional instructions that have never before been seen in a chess program. Or it calculated the variations to depths that defy imagination.”
Legend of Deep Blue (contd)

• Rounds 3 and 4, Kasparov played “anti-computer” chess (later)
  – Draws

• Round 5, Kasparov was in a winning position, but Deep Blue forced the match to a draw
  – “But just when everybody at last seemed to agree that [Kasparov] was actually winning Deep Blue pulled out a most amazing [defense], forcing Garry's king into a perpetual. The [maneuver] left the entire auditorium gasping.”
  – In the conference after the game, Kasparov admits he “is afraid of the machine”
Legend of Deep Blue (contd)

• Round 6

  – “Disaster. I could sense it already early that morning. In the two weeks we had all been together in New York I had never seen Garry so tense. He hardly spoke, and on the way to the site the mood was dark. When I saw the game start I really had a sinking feeling. In 19 moves it was all over, with the world champion falling into a well-known openings trap...Just over an hour later the world champion had resigned against the computer.”
Legacy of Deep Blue

- Houdini (3311 Elo CCRL)
  - alpha-beta search, bitboard, late move reduction, Nalimov end-game tablebases
  - Released 2011
- Rybka (3261 Elo)
- Deep Junior (3121 Elo)
- Deep Fritz (3095 Elo)
- Hydra (3000+, guesstimated)
- Deep Blue?
  - Didn’t play enough games. Maybe 2870?
- Kasparov 2851

All rankings by CCRL, one of many computer chess ranking organizations
Legacy of Deep Blue (contd.)

• 2002, Kramnik vs. Deep Fritz, tie
  – Five years after Deep Blue, computers are four times as fast
• 2003, Kasparov vs. Deep Junior, tie
• 2005, Adams (#7) vs. Hydra, 5½-½ Hydra
  – 64 CPU, custom FPGA chips, 150m positions/sec
  – D=18
• 2006, Kramnik vs. Deep Fritz, 4-2 Fritz
  – Dual Core 2 Duo, 1.3 GHz, 8m positions/sec
    • Two orders of magnitude less than Hydra
  – D=18
• 2009, Pocket Fritz wins Copa Mercosur in the National Master category
  – HTC Touch HD, 20,000 positions/second
    • It’s a freaking phone!
So chess is pretty much over

...or is it?

(Cue dramatic music)
BATTLESTAR GALACTICA
THE RESISTANCE
Anti-Computer Chess

• Move quick
  – Reduce pondering

• Use unusual (non-book) opening (Kasparov did this in games 3 and 4)
  – Force them to use their time early
    • Reduces # possible plies later

• Don’t swap pieces
  – Keep the branching factor high
  – Beware end games; 5 piece DBs are common

• Control the center from somewhere else
  – Most evaluation functions like center, rate it high
  – Fianchetto (Sicilian Dragon, Pirc Defense, Benko, Grob, etc.)
    • Fianchetto = stay out of center, put bishop on long diagonal
2008 Rybka vs. Nakamura

Move 111, White up 4

http://www.chessgames.com/perl/chessgame?gid=1497429
Summary

• We learned about:
  – Deep Blue was awesome
  – But humans aren’t done yet!
  – And neither are the computers...