Game playing

Why study games?

In, why try and write computer programs that can play games?

Why study games

Clear success criteria

Good motivator to push research

Important historically for AI

Fun 😊

Some real-world problems fit this model
- game theory (economics)
- multi-agent problems
We want to write a program to play Tic Tac Toe. How would you do it? What types of decisions does it have to make?

For any board configuration

Needs to make a move (ideally as good as possible)

What move should ‘X’ make?

For any board configuration

Needs to make a move (ideally as good as possible)

What move should ‘X’ make?
For any board configuration

What move should 'X' make?

Needs to make a move (ideally as good as possible)

How can we “learn” this (or figure it out)?

Tic Tac Toe as search

Key idea: search all possible moves/configurations

Explore all possible first moves by ‘X’
Then, explore all possible next moves by 'O'

Explore all possible next moves by 'O'

Then, explore all possible next moves by 'X'

Explore all possible next moves by 'X'
Tic Tac Toe as search

Eventually, we'll get to an end game

WIN
TIE
LOSE

How does this help us?
Try and make moves that move us towards a win, i.e. where there are leaves with a WIN.

Problem: we don't know what 'O' will do

I'm X, what will 'O' do?

O's turn

X's turn

O's turn

Problem: we don't know what 'O' will do
Minimizing risk

The program doesn’t know what move \( O \) (the opponent) will make. It can assume, though, that it will try and make the best move possible. Even if \( O \) actually makes a different move, we’re no worse off. Why?

![Diagram of a game board with X's and O's]

Defining a scoring function

Idea:
- define a function that gives us a “score” for how good each board state is for us
- higher scores mean better for us

<table>
<thead>
<tr>
<th>WIN</th>
<th>TIE</th>
<th>LOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>

Our (X) turn

What should be the score of this board state?

\(+1\): we can get to a win

Opponent’s (O) turn

What should be the score of this board state?

\(-1\): opponent can get to a win (we lose)
Defining a scoring function

Opponent’s (O) turn

Our (X) turn

What should be the score of this board state?
Defining a scoring function

Our (X) turn

What should be the score of this state?

0: If we play perfectly and so does O, the best we can do is a tie (could do better if O makes a mistake)

How should X play?

When it's "my" turn, pick the highest scoring state

When it's the opponent's turn, assume the lowest scoring state (from my perspective)

If we can reach the end games, we can percolate these answers all the way back up

How to calculate this tree

Start from the leaves and propagate the score up:
- If X's turn, pick the move that maximizes the utility
- If O's turn, pick the move that minimizes the utility

This results in an optimal strategy!
Another example: Baby Nim

Take 1 or 2 sticks at each turn
Goal: take the last stick

What move should I take?
Take 1 or 2 at each turn
Goal: take the last match
\[ \text{MAX wins} = 1.0 \]
\[ \text{MIN wins/ MAX loses} = -1.0 \]
Baby Nim

Take 1 or 2 at each turn
Goal: take the last match

MAX wins
\[= 1.0\]

MIN wins
\[= -1.0\]

MAX loses

MAX wins

MIN wins

MAX loses
Take 1 or 2 at each turn
Goal: take the last match

MAX wins

= 1.0

MIN wins/
MAX loses

= -1.0

Which move?

Baby Nim

One more minimax example

Which move should be made: A1, A2 or A3?
One more minimax example

Properties of minimax

Minimax is optimal!

What are the requirements to use minimax (i.e. what types of games can we solve)?

- Must have full view of board state (e.g. stratego wouldn’t work)
- No chance/probability
- Board state space can’t be too big!

Game state size

What impacts the size of the board state space?
Game state size

What impacts the size of the board state space?
- Number of possible moves from each board state
- Number of moves before the game finishes (depth of the tree)

Game state space size

How many of this can we solve optimally?

Branching Factor Estimates for different two-player games:
- Tic-tac-toe: 4
- Connect Four: 7
- Checkers: 10
- Othello: 30
- Chess: 35
- Go: 300

Game state space size

Can search entire space
- "solved" games: CHINOOK (2007)

Can't:
- computer-dominated

Mastermind revisited

Leverage some of these ideas for Mastermind!
3 colors, 3 pegs

3 Colors: Red, Green, Blue

3 pegs: [___, ___, ___]

How many different codes?

3 colors, 3 pegs

27! \((\text{colors}^{\text{pegs}} = 3^3)\)

Naive approach (assignment 3)

What would our naive approach guess first?

Codemaker chooses this code
### Naïve approach (assignment 3)

#### Guess 1: [Red, Red, Red]

<table>
<thead>
<tr>
<th>Exact</th>
<th>Inexact</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Red, Red, Red]</td>
<td>[Green, Red, Red]</td>
</tr>
<tr>
<td>[Red, Red, Green]</td>
<td>[Green, Red, Green]</td>
</tr>
<tr>
<td>[Red, Red, Blue]</td>
<td>[Green, Red, Blue]</td>
</tr>
<tr>
<td>[Red, Green, Red]</td>
<td>[Green, Green, Red]</td>
</tr>
<tr>
<td>[Red, Green, Green]</td>
<td>[Green, Green, Green]</td>
</tr>
<tr>
<td>[Red, Green, Blue]</td>
<td>[Green, Green, Blue]</td>
</tr>
<tr>
<td>[Red, Blue, Red]</td>
<td>[Green, Blue, Red]</td>
</tr>
<tr>
<td>[Red, Blue, Green]</td>
<td>[Green, Blue, Green]</td>
</tr>
<tr>
<td>[Red, Blue, Blue]</td>
<td>[Green, Blue, Blue]</td>
</tr>
</tbody>
</table>

Response?: [Blue, Red, Red]

#### Guess 1: [Red, Red, Red]

<table>
<thead>
<tr>
<th>Exact</th>
<th>Inexact</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Red, Red, Red]</td>
<td>[Green, Red, Red]</td>
</tr>
<tr>
<td>[Red, Red, Green]</td>
<td>[Green, Red, Green]</td>
</tr>
<tr>
<td>[Red, Red, Blue]</td>
<td>[Green, Red, Blue]</td>
</tr>
<tr>
<td>[Red, Green, Red]</td>
<td>[Green, Green, Red]</td>
</tr>
<tr>
<td>[Red, Green, Green]</td>
<td>[Green, Green, Green]</td>
</tr>
<tr>
<td>[Red, Green, Blue]</td>
<td>[Green, Green, Blue]</td>
</tr>
<tr>
<td>[Red, Blue, Red]</td>
<td>[Green, Blue, Red]</td>
</tr>
<tr>
<td>[Red, Blue, Green]</td>
<td>[Green, Blue, Green]</td>
</tr>
<tr>
<td>[Red, Blue, Blue]</td>
<td>[Green, Blue, Blue]</td>
</tr>
</tbody>
</table>

Response?: [Blue, Red, Red]

**Which ones can we eliminate?**

- [Red, Red, Red], [Red, Red, Green], [Red, Red, Blue], [Red, Green, Red], [Red, Green, Green], [Red, Green, Blue], [Red, Blue, Red], [Red, Blue, Green], [Red, Blue, Blue]

### Naïve approach (assignment 3)

#### Guess 1: [Red, Red, Red]

<table>
<thead>
<tr>
<th>Exact</th>
<th>Inexact</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Red, Red, Red]</td>
<td>[Green, Red, Red]</td>
</tr>
<tr>
<td>[Red, Red, Green]</td>
<td>[Green, Red, Green]</td>
</tr>
<tr>
<td>[Red, Red, Blue]</td>
<td>[Green, Red, Blue]</td>
</tr>
<tr>
<td>[Red, Green, Red]</td>
<td>[Green, Green, Red]</td>
</tr>
<tr>
<td>[Red, Green, Green]</td>
<td>[Green, Green, Green]</td>
</tr>
<tr>
<td>[Red, Green, Blue]</td>
<td>[Green, Green, Blue]</td>
</tr>
<tr>
<td>[Red, Blue, Red]</td>
<td>[Green, Blue, Red]</td>
</tr>
<tr>
<td>[Red, Blue, Green]</td>
<td>[Green, Blue, Green]</td>
</tr>
<tr>
<td>[Red, Blue, Blue]</td>
<td>[Green, Blue, Blue]</td>
</tr>
</tbody>
</table>

Response?: 0 0

### Naïve approach (assignment 3)

#### Guess 1: [Red, Red, Red]

<table>
<thead>
<tr>
<th>Exact</th>
<th>Inexact</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Red, Red, Red]</td>
<td>[Green, Red, Red]</td>
</tr>
<tr>
<td>[Red, Red, Green]</td>
<td>[Green, Red, Green]</td>
</tr>
<tr>
<td>[Red, Red, Blue]</td>
<td>[Green, Red, Blue]</td>
</tr>
<tr>
<td>[Red, Green, Red]</td>
<td>[Green, Green, Red]</td>
</tr>
<tr>
<td>[Red, Green, Green]</td>
<td>[Green, Green, Green]</td>
</tr>
<tr>
<td>[Red, Green, Blue]</td>
<td>[Green, Green, Blue]</td>
</tr>
<tr>
<td>[Red, Blue, Red]</td>
<td>[Green, Blue, Red]</td>
</tr>
<tr>
<td>[Red, Blue, Green]</td>
<td>[Green, Blue, Green]</td>
</tr>
<tr>
<td>[Red, Blue, Blue]</td>
<td>[Green, Blue, Blue]</td>
</tr>
</tbody>
</table>

Response?: 0 0

**What would our naïve approach guess next?**

- [Green, Green, Green], [Green, Green, Blue], [Green, Blue, Green], [Green, Blue, Blue], [Blue, Green, Green], [Blue, Green, Blue], [Blue, Blue, Green], [Blue, Blue, Blue]

Any with red in them: 19 removed
Naïve approach (assignment 3)

<table>
<thead>
<tr>
<th>Exact</th>
<th>Inexact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guess 2: [Green, Green, Green]</td>
<td>Response?</td>
</tr>
<tr>
<td>(codemaker)</td>
<td>1 0</td>
</tr>
</tbody>
</table>
Naïve approach (assignment 3)

Guess 3: [Green, Blue, Blue]  Response?

[Blue, Green, Blue]  [Blue, Blue, Green]  [Green, Blue, Blue]  

Which ones can we eliminate?

Naïve approach (assignment 3)

Guess 3: [Green, Blue, Blue]  Response?

[Blue, Green, Blue]  [Blue, Blue, Green]  [Green, Blue, Blue]  

Only 1!

Naïve approach (assignment 3)

Guess 3: [Green, Blue, Blue]  Response?

[Blue, Green, Blue]  [Blue, Blue, Green]  

What would our naïve approach guess next?
### Naïve approach (assignment 3)

<table>
<thead>
<tr>
<th>Exact</th>
<th>Inexact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guess 4: [Blue, Green, Blue]</td>
<td>Response? (codemaker)</td>
</tr>
</tbody>
</table>

| [Blue, Green, Blue] |
| [Blue, Blue, Green] |

---

### Naïve approach (assignment 3)

<table>
<thead>
<tr>
<th>Exact</th>
<th>Inexact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guess 4: [Blue, Green, Blue]</td>
<td>Response? 1</td>
</tr>
</tbody>
</table>

| [Blue, Green, Blue] |
| [Blue, Blue, Green] |

---

### Naïve approach (assignment 3)

<table>
<thead>
<tr>
<th>Exact</th>
<th>Inexact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guess 5: [Blue, Blue, Green]</td>
<td>Response? 3</td>
</tr>
</tbody>
</table>

| [Blue, Blue, Green] |

---

### Naïve approach (assignment 3)

It took us 5 guesses.

- Guess 1: [Red, Red, Red]
- Guess 2: [Green, Green, Green]
- Guess 3: [Green, Blue, Blue]
- Guess 4: [Blue, Green, Blue]
- Guess 5: [Blue, Blue, Green]

Can we do better (less guesses)?