Game playing

Thursday, November 21, 2024 4:35 PM

- Main topics:
  - Adversarial topics
  - Minmax algorithm
  - alpha-beta pruning
  - Evaluation functions
  - Isolation game player
  - multi-player, probabilistic games
- How do we tell the computer not to lose?
  - If computer makes poor first move, x can win no matter what
  - To avoid computer making a bad first move - we can keep a table of best first moves that is called an opening book
- Minimax algorithm
  - Top of tree is always going to be a max level
  - Work up from bottom of tree - until you get to the top node
  - The process of computing values for each node bottom to top = mini max algorithm
  - For each max node, pick the minimum value among its child nodes
  - If theres at least one plus one child, the computer can always pick that to win
- Book - adversarial search and games
  - Competitive environments
  - Two or more agents have conflicting goals
  - Game theory
    - multi-agent environment
    - economy- A very large number of agents
    - Mini max search - generalized and/or search
    - Pruning - ignore portions of the search tree that make no difference to the optimal move
    - For each state where we choose to stop searching, we ask who is winning
    - Evaluation function- to estimate who is winning based on features of the state or we can average the outcomes of many fast simulations of the game from that state all the way to the end
  - Two - player zero-sum games

- Two - player zero-sum games
    - Perfect information = "fully observable"
    - "zero sum" = what is good for one player is just-bad for the other
    - Transition model - defines the state resulting from taking an action a in state s
    - Utility function - objective function, pay-off function - defines the final numeric value to player p when to game ends in terminal states
    - State space graph - a graph where the vertices are states, the edges are moves and a state might be reached by multiple paths
    - Game tree- a search tree that follows every sequence of moves all the way to a terminal state
- Optimal decisions in games
    - Max wants to find a sequence of actions leading to a win
    - Max must be a conditional plan - a contingent strategy specifying in response to each of min's possible moves
    - Ply - used to unambiguously mean are move by one player to go are level deeper into game play
    - Optimal strategy can be determined by working out the minimum value of each state in the tree
- Mini max search algorithm
    - A recursive algorithm that proceeds all the way down to the leaves of the tree and then backs up the mini max values through the tree as the recursion unwinds
    - The functions max-value and min-value go through the whole game tree, all the way to the leaves, to determine the backed-up value of the state and the move to get there
    - Time complexity =
        - $O(b^m)$
    - Exponential complexity makes mini max impractical for complex games
    - Series as basis for mathematical analysis of games
    - Optimal decisions in multiplayer games
        - 3 player game -vector associated with each node - gives utility of the state from each player's viewpoint
        - Alliances are natural consequences - players will automatically cooperate to achieve a mutually desirable goal
    - alpha-beta pruning
        - No algorithm can completely eliminate the exponential depth of a tree

tree

        - Can prune large parts of the tree that make no difference to the outcome
        - By examining some of its descendants we find out enough about n, then we can prune it
        - Just ave to consider the modes along a single path in the tree
        - Describes the bounds on the backed-up values that appear anywhere along the path
            - Alpha = the value of the best (highest - value) choice we have found so far at any choice point along the path for max = "at least"
            - Beta = the value of the best (lowest value) choice we have found so far at any choice point along the path for min - "at most"
        - alpha-beta search updates the values of alpha and beta as it goes along and prunes remaining branches at a node (terminates recursive call) as soon as the value of the current node is known to be worse than tee current alpha and beta value for max or min respectively
    - Move ordering
        - Effectiveness of alpha-beta pruning is dependent on the order in which the states are examined
        - Improves performance to
        - $O(b^{m/2})$
        - In game tree search, repeated states can occur because of transpositions - different permutations of the more sequence that end up in the same position and the problem can be addressed with a transposition table that caches the heuristic value of states
        - Limited exploration time:
            - A strategies:
                - Type A strategy: considers all possible moves to a certain depth-in the search tree and then uses a heuristic evaluation function to estimate the utility of states at that depth - explores wide but shallow
                - Type B strategy: ignores moves that look bad and follows promising lines as far as possible - explores deep but narrow
    - How many nodes do you think mini max will need to visit? B= average

branching factor d= depth of the game tree

- Average branching factor - the number of non-root nodes (the size of the tree, minus are or the number of edges) divided by the number of non-leaf nodes (the number of nodes with children)
- Depth limited search
  - Depth limited search
    - $\log_a x = \dfrac{\log_b x}{\log_b a}$
    - need b
  - Used to find how many levels to go deep
- Evaluation function intro
  - Evaluate goodness of node a level to understand how much we should expect it to lead to a win for our computer player
  - You want an evaluation function that returns a higher number depending how good the board is for our computer player.
- Quiescent search
- Horizon effect
  - Iterative deepening
  - Computer player cannot search far enough into the future to see if the game will end the way a human can
- Evaluation function