Implementación del juego del gato vía algoritmo minimax

Rodolfo Emilio Escobar Uribe

4 de diciembre de 2019

Introducción

La inteligencia artificial símbolica es un conjunto de métodos que tienen el objetivo de resolver problemas complejos a través de la manipulación de símbolos utilizando lógica y procesos de búsqueda. Fue el paradigma dominante durante las primeras décadas de la inteligencia artifical. En este proyecto se utulizará este enfoque para crear un programa que pueda juegar al gato utilizando un algoritmo de busqueda inteligente que considera las decisiones de un rival racional.

Descripción del problema

El juego del gato pertenece a la categoría de juegos de información completa. Esto significa que toda la información requerida para tomar las mejores decisiones está disponible a la vista de todos los jugadores. En este caso, la información se despliega como las posiciones de las marcas de los jugadores sobre el tablero. El juego tiene reglas muy simples:

- A cada jugador le corresponde un símbolo: 'X', 'O'.
- Durante cada turno el jugador sólo puede escribir un símbolo.
- El primer jugador que forma una linea recta con sus simbolos en cualquier dirección gana.

el

Podemos formalizar este juego como un proceso de transición de estados. Bajo esta consideración, cada estado representa una configuración particular

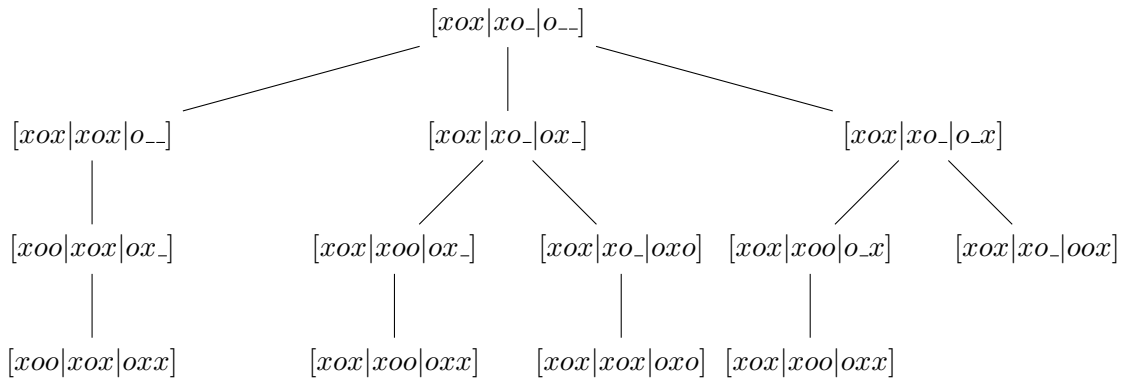

Figura 1: Arbol de transiciones de estados

del tablero. Los turnos están definidos como una restricción en la transiciones. Si el número de 'X's es igual a de 'O's entonces sólo pueden ocurrir transiciones que añadan una 'X' al tablero. En caso contrario, transiciones que añadan una 'O'. Podemos representar estas transiciones como se muestra en la Figura 1 partiendo del estado $[x\_x|xo\_|o\_\_]$.

Las transiciones se detienen cuando uno de los jugadores gana en el estado.

Algoritmo minimax

En teoría de juegos, el algoritmo minimax es un método de decisión que minimiza la perdida máxima para un agente que compite contra otro. Este algorimo considera que el adversario va actuar de la manera que mejor le convenga. Para aplicarlo al juego del gato, se debe asignar un valor numérico a cada una de las jugadas terminales de la siguiente manera:

- Si X gana $\rightarrow Estado.Eval = +10$.
- Si O gana $\rightarrow Estado.Eval = -10$.
- Si no hay ganador $\rightarrow Estado.Eval = 0$.

El objeto $Estado$ es una instancia de una clase definida de la siguiente manera: $Estado = (Tablero, Eval, tablleno(), turnoX())$. Dónde:

- Tablero: Matriz 3x3 donde 0,1,2 representan vacío,'x' y 'o' respectivamente.

- Eval: Campo de evaluación estática.
- tablleno(): Función que retorna verdadero si el tablero está lleno.
- turnoX(): Función que verifica si es el turno de "X".

El algorimto implementado funciona por fuerza bruta recorriendo todos los posibles juegos. La implementación del algoritmo fue hecha en lenguaje Python siendo la computadora el jugador "X". El pseudocódigo del programa se muestra en el Algoritmo 1.

```
1  Function minimax(Estado)
2  |  Estado.Eval = evaluacion(Estado);
3  |  if Estado.Eval ≠ 0 ∨ Estado.tablleno() then
4  |  |  return Estado;
5  |  end
6  |  Lista Ramas = EstadosSiguientes(Estado);
7  |  Lista RamasEvaluadas;
8  |  for EdSig in EstadosSiguientes do
9  |  |  RamasEvaluadas.agregar(minimax(EdSig);
10 |  end
11 |  if Estado.turnoX() then
12 |  |  Edminimax = max(RamasEvaluadas, clave = x.Eval);
13 |  else
14 |  |  Edminimax = min(RamasEvaluadas, clave = x.Eval);
15 |  end
16 |  minimaxIndex = index(RamasEvaluadas, Edminimax);
17 |  return Ramas[minimaxIndex];
18 end
```

Resultados

El tiempo de ejecución que toma la elección de la primera jugada de "X"toma cerca de 30 segundos. El primer turno requiere una búsqueda por todo el espacio de jugadas. Para resto de las jugadas el tiempo es despreciable. El desepeño del juego de la máquina mediante el algoritmo minimax resulta perfecto. Si el oponente elige siempre las jugadas más favorable, el juego siempre termina en empate.

1. Concluciones

Este método es efectivo sólo si el espacio de estados es pequeño en relación al poder de cómputo disponible o si es posible reducir el espacio de estados mediante podas heurísticas.