DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Machine Learning

Trabajo práctico 2

Qlearning

Resumen

Integrante	LU	Correo electrónico
Negri, Franco	893/13	franconegri2004@hotmail.com
Podavini Rey, Martín Gastón	483/12	marto.rey2006@gmail.com

Palabras claves:

TP, 4 en linea, qlearning

Índice

1. Introduccion

En este trabajo practico buscamos modelar el juego de 4 en linea, modelar jugadores que puedan jugar sobre ese modelo y por ultimo, implementar una clase jugador que utilice tecnicas de q-learning para observar su comportamiento al variar parametros o entrenandolo bajo ciertas circunstancias particulares que concideremos interesantes.

Algo interesante para notar aqui es que, en este juego se contará con la participación de dos ajentes que compiten entre ellos, por lo que deberá prestarse particular atención en donde se modelará la etapa de recompensa del algoritmo.

Otra idea que intentaremos explorar aqui es que politica de exploración utilizar en nuestro algoritmo. En particular en estre trabajo veremos que sucede al utilizar las estretegias:

- estrategia greedy: toma un camino random con probabilidad $\epsilon\,\%$ y en caso contrario utilice el mejor brazo conocido.
- estrategia ϵ-first: toma un camino random en las primeras ϵ iteraciones y luego toma el mejor camino conocido.
- estrategia softmax: basada en una formula probabilistica que desarrollaremos mas adelante.

2. Desarrollo

2.1. Modelado

El juego en si consistirá en una lista de listas donde cada lista representará una columna del tablero. En cada turno un jugador elejirá una columna numerada del 0 al n, siendo n el numero de columnas totales y el juego se encargará. Ambos jugadores contarán en el momento de la elección con el estado actual del tablero.

Luego de cada jugada, el juego chequará si el jugador ganó o si ya no hay mas movimientos posibles, terminando el juego e informando que jugador gano o si fue empate.

El pseudocodigo del juego será, entonces el siguiente:

```
1: While True
2:     columna = player.move(tablero)
3:     jugar_ficha(columna, tablero)
4:     if jugador_gano(tablero)
5:         return player
6:     if tablero_lleno(tablero)
7:         return empate
8:     player = otro_jugador(player)
```

Algorithm 1: jugar()

Como ya adelantamos en la introducción, es necesario definir en que etapa del algoritmo se le asignará la recompensa al algoritmo de q-learning. Una opción bastante sencilla e intuitiva es la de asignar recompensas en el momento en que algun jugador gana la partida. Por ejemplo al ganar la partida uno de los jugadores, se le asigna una recompensa de 1 a el y una recompensa de -1 al contrincante. Siguiendo con el lineamiento anterior, tambien sería posible asignarles recompensas en el momento en que se empata, por ejemplo, asignandoles a ambos jugadores una recompensa de 0,5

En caso de que no se haya llegado a un estado final, una posible propuesta es asignarle al otro jugador una recompenza por ejemplo de 0.

El pseudocodigo entonces, pasaría a verse de la siguiente manera:

```
While True
    columna = player.move(tablero)
    jugar_ficha(columna, tablero)
    if jugador_gano(tablero)
        player.recompensa(1)
        otro_jugador(player).recompensa(-1)
        return player
    if tablero_lleno(tablero)
        player.recompensa(0.5)
        otro_jugador(player).recompensa(0.5)
        return empate
    otro_jugador(player).recompensa(0)
    player = otro_jugador(player)
```

Algorithm 2: jugar()

3. Concluciones