TP2

Q-Learning
Inteligência Artificial

Giovanni Ferreira Martinelli

1 de Julho, de 2019

1 Introdução

O objetivo do trabalho consiste em implementar o algoritmo Q-Learning para o problema do paczmaze, que consiste no mapa do pacman e um agente, que deve percorrer esse mapa evitando os fantasmas e procurando a recompensa (a pilula).

2 Implementação

2.1 Abordagem

A abordagem básica utilizada para os algoritmos foi bastante similar, seguindo a seguinte lógica:

1. Inicia em um estado inicial aleatório possível no pacmaze
2. Para cada estado, se escolhe um estado próximo aleatório e também determina todos os estados possíveis após o próximo estado, os "estados do próximo próximo"
3. Examina-se os valores atuais de Q e encontra o maior valor do próximo estado para qualquer próximo próximo estado. Esse maior valor, "max Q", é usado na instrução que atualiza a matriz Q.
4. Para cada estado inicial aleatório, repete-se esse processo até atingir o estado contendo a pilula.

2.2 Modelagem

Para a modelagem do estado foi feita uma classe "Pacmaze", contendo os seguintes parâmetros:

2.2.1 State

1. O mapa(uma matriz) representando os estados do pacmaze
2. As recompensas de cada estado em uma matriz, de mesmo tamanho que o mapa
3. Um dicionário contendo os Q valores para cada estado (x,y)
4. Um array de tuplas contendo os estados possíveis de movimento (para facilitar a iteração)
5. Um array contendo tuplas (x,y) com a poisção as pilulas e outro para os fantasmas (os estados terminais)

3 Execução

O projeto foi feito em Python 3, utilizando a biblioteca numpy. Sendo obrigatória a presença da mesma para execução do código. Ela pode ser instalada utilizando o comando: *pip*3 *install numpy*

4 Resultados

Foram feitos os testes para comparar as performances dos algoritmos. Utilizando o mapa disponibilizado no enunciado como base para os testes foi possível notar a influência do γ e seu desconto no algoritmo. Quando colocado com outro valor a não ser 0.9 havia uma solução sub-ótima. Para esse problema a alteração do E-greedy não mudou o resultado

Gamma	E-Greedy	Solução
0.1	0.1	Subótima
0.5	0.1	Subótima
0.9	0.1	Ótima
0.1	0.5	Não alterou o resultado
0.1	0.9	Não alterou o resultado

Para os problemas maiores, o programa não conseguiu a solução ótima, as paredes não tinham um desconto o suficiente, mesmo utilizando a sugestão do enunciado do agente receber a mesma recompensa do estado atual.

5 Conclusão

O trabalho foi muito educativo sobre o algoritmos de inteligência artificial e na modelagem de problemas para serem resolvidos utilizando técnicas e metodologias aprendidas em sala.