上海交通大学安泰经济与管理学院

BUSS3620 人工智能导论

代码实践 #1. 搜索

刘佳璐 助理教授

Q1. DFS

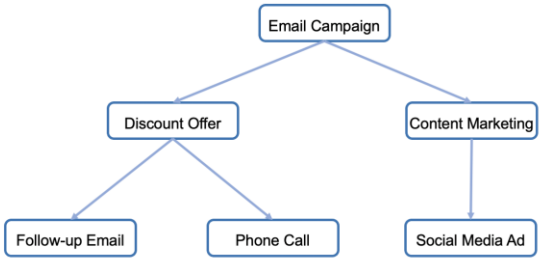
假设你是一名为公司工作的顾问，公司需要探索各种潜在的营销策略。公司有一个可能的营销策略的集合，每个策略都会根据客户的反应导致不同的结果。你的任务是帮助公司确定可能最大化客户参与度的营销策略序列。然而，公司希望在做出决定之前先探索所有可能的路径。

任务:

假设营销策略可以构成一张搜索图，每个节点代表一个营销策略，每条边代表可能的客户反应，实现一个深度优先搜索（DFS）算法来探索所有可能的从根节点到子节点的营销策略序列。假设这棵树是二叉树（每个节点最多有两个子节点）。你的任务是编写一个程序，返回所有可能的从根到子节点的营销策略序列。

输入:

- 营销策略图。



输出:

- 一个列表，列表中的每个内部列表代表一个从根节点到叶节点的营销策略序列。

提示:

- 你可以参考课上的 `maze.py`。
- 你可以参考以下的代码框架

```
class Node():
    """记录营销策略"""

class Frontier():
    """构建 Frontier"""

class Graph():
    """搜索"""
    def __init__(self, start):
        """储存必要信息"""

    def addNode(self,strategy, parent):
```

```
        """搜索过程中创建节点"""

    def solve(self):
        """搜索所有的解"""

##储存营销策略图的信息
advertising={…}

g=Graph('Email Campaign')
g.solve()
g.solution
```

Q2. MiniMax

两家公司（A 和 B）进行两轮竞标。每家公司在每轮中只能选择三个价格之一：高（H）、中（M）、低（L）。每轮两家公司依次出价（A 先 B 后），每次出价必须比自己上一轮的出价相等或更高。根据最后一轮出价，竞标收益如下：

- 如果两者都出价 H，A 获得 80,000 元，B 得 20,000 元。
- 如果 A 出价 H，B 出价 M，A 获得 70,000 元，B 得 30,000 元。
- 如果 A 出价 H，B 出价 L，A 获得 60,000 元，B 得 40,000 元。
- 如果 A 出价 M，B 出价 H，A 获得 30,000 元，B 得 70,000 元。
- 如果 A 出价 M，B 出价 M，A 和 B 各获得 50,000 元。
- 如果 A 出价 M，B 出价 L，A 获得 40,000 元，B 得 60,000 元。
- 如果 A 出价 L，B 出价 H，A 获得 20,000 元，B 得 80,000 元。
- 如果 A 出价 L，B 出价 M，A 获得 30,000 元，B 得 70,000 元。
- 如果两者都出价 L，A 和 B 各获得 50,000 元。

任务：

实现一个程序，使用极小极大算法模拟这个决策过程，模拟 A，B 公司竞标过程中每次的最佳出价。

输入:

- 竞标收益

输出:

- 公司 A，B 两轮的出价策略。

提示:

- 你可以参考课上的 `tictactoe.py`。
- 你可以参考以下的代码框架

```
class Bid():
    def __init__(self, payoff):
        """记录需要的信息"""

    def actions(self,previous_action=None):
        """每次出价可以竞标的价格"""

    def A_value(self,round,A_action,B_action):
        """让 A 收益最大化策略时 A，B 的收益"""

    def B_value(self,round,A_action,B_action):
        """让 B 收益最大化策略时 A，B 的收益"""

    def minimax(self,round,isA,A_action,B_action):
        """返回每轮出价价格"""
```

```
#竞价收益
gain={}

#每次最优出价策略
b=Bid(gain)
b.minimax(1,True,None,None) #A:L
b.minimax(2,False,'L',None)#B:L
b.minimax(3,True,'L','L') #A:H
b.minimax(4,False,'H','L')#B:L
```