Evolving Decision Making based on Bayesian Network

**Abstract**

Agents may need to handle uncertainty, whether due to partial observability, nondeterminism, or a combination of the two. An agent may never know for certain what state it’s in or where it will end up after a sequence of actions.

**Introduction**

Bayes’ rule: P(B|A) =P(A|B)P(B)/P(A), P(A|B)是在B发生的情况下A发生的可能性，称为先验概率.

我们先考虑对方是否携带武器，通过玩家训练会得到大量数据。假设对于每次战斗A=敌方携带武器，B=我方胜利。AI通过数据可以得到P(A)=0.6, P(B)=0.3, P(A|B)=0.2 (P(B)=P(B|A)P(A)+P(B|~A)P(~A)) (每场战斗如果胜利的话，对手携带武器的概率)(这里存在一个问题，既然数据都是一样的，那么我们也自然可以得到对方携带武器时我方的胜率)，那么我们可以得到P(B|A)=0.2\*0.3/0.6=0.1,即当对方带有武器时我方胜利可能性为0.1。

这个继续往下扩展，如果刚才给出的概率是认为规定的，假设比重是100场战斗数据，那么在游戏进行的过程中，玩家又进行了100场战斗，AI得到数据：P(A)=0.8, P(B)=0.8, P(A|B)=0.4。现在因为我们将玩家数据放入到真个个数据库中，现在数据变成了P(A)=0.7, P(B)=0.55, P(A|B)=0.3，那么继续刚才的计算可以得到P(B|A)=0.3\*0.55/0.7=0.236，发现因为玩家变弱（可以从数据中看出），电脑评估胜率提高了。从宏观上来看是一个进化的过程（随环境调整而调整）。

**Evaluation**

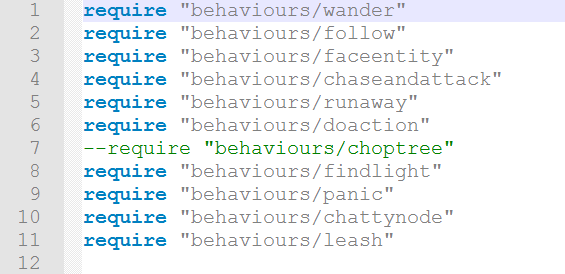
The cost of Bayesian Network：基础评估

刚才我们只是考虑了某一单一变量，假如说现在考虑对方的攻击力、是否携带武器以及生命值，先用模糊逻辑将攻击力和生命值分别分成三个档（低(L)中(M)高(H)，具体模糊函数待定），这三个变量是互相独立的。明显可以看出我们需要储存数据和概率，概率空间为O(N1\*N2\*N3…Nm),计算所消耗的时间为O(m)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IsArmed | HP | Attack | Win | Lost |
| T | L | L | P(W|T^L^L) | P(~W|T^L^L) |
| T | L | M | … | … |
| T | L | H |  |  |
| T | M | L |  |  |
| T | M | M |  |  |
| T | M | H |  |  |
| T | H | L |  |  |
| T | H | M |  |  |
| T | H | H |  |  |
| F | L | L |  |  |
| F | L | M |  |  |
| F | L | H |  |  |
| F | M | L |  |  |
| F | M | M |  |  |
| F | M | H |  |  |
| F | H | L |  |  |
| F | H | M |  |  |
| F | H | H |  |  |

**Games**

我理想的游戏是Don’t Starve，首先这是个沙盒游戏，里面生物共享一个脚本，因为里面有大量写好的脚本，LUA语言也比较好读，其中很多行动代码都已经写好了，我们可以加以利用。如果能够实现的话，展示效果也比较好



**Tasks**

找出影响决定的变量（自身和环境）

↓

建立模糊函数，将连续变化的变量模糊化（方便建立贝叶斯网络）

↓

建立相应的贝叶斯网络

↓

与其他AI techniques合并（FSM）

↓

评估结果（最好是再设计出来一些相对简单的算法，比较结果）