

人工智能实验

原创

rocketeerLi 于 2018-11-19 22:40:04 发布 3089 收藏 21

分类专栏: [人工智能](#) 文章标签: [人工智能](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/rocketeerLi/article/details/84231437>

版权



[人工智能](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

介绍

这学期的人工智能实验, 一共有三个。个人感觉, 除了第二个实验稍微复杂一些之外, 剩下两个还是比较简单的。我们是6个人组成了一组, 我负责做第二个实验, 虽然复杂了一些, 但网上的参考还是很多的。人工智能实验已经验收完了, 考试也已经完事了, 刚好轻松一下, 在这里简单介绍一下这几个实验。

所有的实验代码在这里给出: [人工智能实验](#) 这里就不再粘贴代码了, 只给出思想和解决方案。

实验介绍

实验一: 知识表示

实验简介

实验一主要是设计一个解决方案, 一个猴子摘香蕉的解决方案。刚开始看到这个问题的时候, 还是比较懵逼的, 一度觉得, 这问题也需要写代码? 后来队友把代码搞出来了, 才知道, 原来可以这样写。

实验的要求很简单:

参照课程第二部分讲授的知识表示方法完成, 包括产生式系统、框架系统、语义网络等 (还可以选择其他方法) 解决以下问题 (不限于此), 必要时上网查找有关参考文献。最好事先编好源代码, 然后再调试运行。

一个房间里, 天花板上挂有一串香蕉, 有一只猴子可在房间里任意活动 (到处走动, 推移箱子, 攀登箱子等)。设房间里还有一只可被猴子移动的箱子, 且猴子登上箱子时才能摘到香蕉, 问猴子在某一状态下 (设猴子位置为A, 香蕉位置在B, 箱子位置为C), 如何行动可摘取到香蕉。

该问题的本质就是利用一种知识表示的方法来解决猴子摘香蕉的问题。如果把猴子、香蕉和箱子的位置用 $\{-1,0,1\}$ 里的数字表示 (对应为A,B,C), 那么就可以用一个三元组来表示问题的状态, 任意输入一个合法的状态, 即可输出一个从初始状态到目标状态的过程。实际实现需要使用四元组, 还需要记录猴子与箱子的相对位置。

解决方案:

按照题目要求, 选择产生式表示法。

1. 首先定义一个综合数据库，即包含所有状态的状态空间（monkey, box, banana, monbox），分别表示猴子、香蕉和箱子的位置，用{-1,0,1}表示。
2. 定义猴子的行动规则，即规则数据库：走到X位置处；跳到箱子上；从箱子上跳下来；将箱子移到另一个位置；如果站在箱子上且有香蕉，则可以摘香蕉。
3. 根据问题，确定终止状态是猴子摘到香蕉{ (x,x,x,0) } (x属于{0, -1, 1})。
4. 输入一个初始状态 (a, b, c, d)
5. 有了上述条件，即可利用产生式的方法输出一个从初始状态到目标状态的解决方案。

实验二：搜索策略

这是一个吃豆人吃豆的游戏，项目整体其实挺庞大的，但大部分代码已经给出，我们只需要在这个框架之上完成相应的功能即可。这是一个国外大学的实验，地址在这里：[UC Berkeley CS188 Intro to AI – Course Materials](#)

先来看实验要求：

采用且不限于课程第四章内各种搜索算法此编写一系列吃豆人程序解决以下列出的问题1-8，包括到达指定位置以及有效的吃豆等。在这些已经设计好的框架中需要做的就是实现不同的图的搜索算法。并通过策略选择以最少的步数，在搜索最少节点的情况下达成问题目标。

问题1：应用深度优先算法找到一个特定的位置的豆

问题2：宽度优先算法

问题3：代价一致算法

问题4：A*算法

问题5：找到所有的角落

问题6：角落问题（启发式）

问题7：吃掉所有的豆子

问题8：次最优搜索

简单地说，这个实验就是分成了两大部分，前四个是写四种不同的搜索算法；后四个是在前四个的基础上再去解决相对应的问题。

实验要求中有这样一句话：

注意：因为不同的搜索方法的不同之处仅仅在于open表的排序不同，因此请定义一个通用的搜索算法解决问题1-4。提示：问题1-4的不同之处在于用不同的数据结构对open表进行排序。

前四个问题的策略其实也确实差不多，但还是有点区别的，所以，我都写在了一个函数中，根据不同的参数来调用不同的算法。

后四个小问题的可发挥性就很高了，可以自由发挥。我这里借鉴了同学的代码，感觉比网上大部分的参考代码都要简洁许多。由于后四个问题基本上都不是我自己写的，这里就不解释了，我在代码中添加了一些注释，想看解释就去看注释吧。

实验三：不确定性推理

这个实验就是利用贝叶斯网络进行计算，思路还是比较简单的。

还是先来看一下实验内容：

参照课程第五部分讲授的贝叶斯网络完成，给定事件和事件之间的关系，并且给出每个事件的CPT图，根据贝叶斯公式根据上述条件求出目标概率，编写程序实现基于贝叶斯网络的推理。在这里用到的贝叶斯算法是建立在有向无环图和CPT表的技术上实现的。

大概的思想就是，根据输入文件给的概率矩阵，构建贝叶斯网络。然后读取问题文件，利用构建的贝叶斯网络，计算出一个CPT表，然后读取解决问题文件中的概率问题，给出概率值。

具体一些，就是说：

1. 首先，利用读取文件的函数读取文件，然后构建一个贝叶斯网络图（使用`geteverycpt`函数，信息存储在存储在`count,strings,q,p`中），计算出一个CPT表。
2. 利用数据结构，计算并存储联合概率。
3. 然后读取问题文件（存储在`arrayList2`中），根据贝叶斯网络的公式，计算相应的联合概率，然后进行计算。将所有的全概率算出，利用穷举法，将各种情况相加，相除得到结果。

其实可以不需要存储所有的联合概率的，太复杂了。后来复习的时候我们才知道有简单一些的方法，但已经写完代码了，并不想改了。

改进的方法也只是相对简单很多，但本质都是先求联合概率，然后相加。不同的是，我们利用的是存储所有的联合概率，然后用的时候去查找，这样存储，感觉方法比较笨。根据课件上的指导，可以根据条件概率与联合概率成正比来计算，有一个正比系数（其实就是计算概率的真假概率值之和的倒数），利用联合概率乘该正比系数即可求出条件概率。

总结

本次人工智能课程的三个实验，其实都比较简单。由于课程较多，时间紧迫，有些东西就没有深究。其实，相比于其他同学选的机器学习课程，人工智能课已经很简单了。但是自己总感觉没有学到太多实用的东西，感觉没有机器学习有意思，以后有时间一定把机器学习的四个实验写一遍。