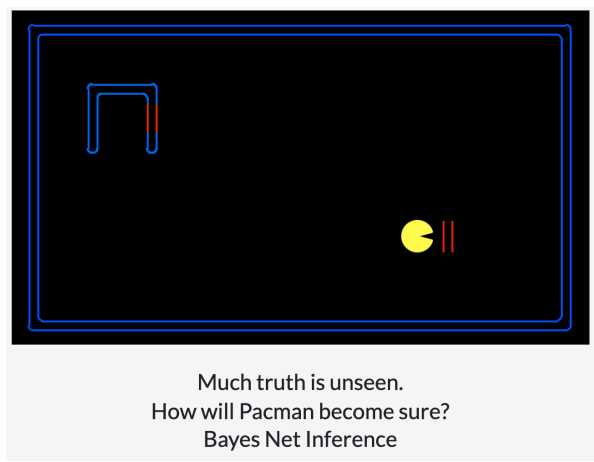


14 Inference in Bayes Net II



本节实验目标：完成贝叶斯网络精确推理中的变量消元法。

报告说明（重要）：本部分“贝叶斯网络”包含两周内容，本节课为第二周内容，请留意该文档目录中标有“报告应实现任务[分值]”的部分，请在实现后及时保存文件和运行结果截图，与上周完成的问题一起在对分易提交第六次课程报告。

Introduction

- [回顾 Treasure-Hunting Pacman 游戏说明](#)
- [回顾 Bayes Nets and Factors](#)
- [Question 1-3 请查看上节课的实验文档并先完成](#)
- [Question 4: Eliminate 报告应实现任务\[1分\]](#)
- [Question 5: Normalize 报告应实现任务\[1分\]](#)
- [Question 6: Variable Elimination 报告应实现任务\[1分\]](#)
- [报告6 “Bayesnet”提交说明](#)

Introduction

在本堂课中，你将在上节课熟悉贝叶斯网络的一种构造方法和上节课实现的逐点相乘算法的基础上，进一步完成贝叶斯网络中的精确推理变量消元法。

你可以通过以下形式的命令为特定测试运行自动评分器：

```
python autograder.py -t test_cases/q4/1-simple-eliminate
```

本堂课你将要编辑的文件	
factorOperations.py	Operations on Factors (join, eliminate, normalize).
inference.py	Inference algorithms (enumeration, variable elimination, likelihood weighting).
bayesAgents.py	Pacman agents that reason under uncertainty.
你应该阅读但不要编辑的文件：	
bayesNet.py	The BayesNet and Factor classes.
你可以忽略的文件	
graphicsDisplay.py	Graphics for Pacman
graphicsUtils.py	Support for Pacman graphics
textDisplay.py	ASCII graphics for Pacman
ghostAgents.py	Agents to control ghosts
keyboardAgents.py	Keyboard interfaces to control Pacman

本堂课你将要编辑的文件	
layout.py	Code for reading layout files and storing their contents
autograder.py	Project autograder
testParser.py	Parses autograder test and solution files
testClasses.py	General autograding test classes
test_cases/	Directory containing the test cases for each question
bayesNets2TestClasses.py	Project's specific autograding test classes

回顾 Treasure-Hunting Pacman 游戏说明

吃豆人进入了一个神秘的世界。最初，整个地图是不可见的。当他探索地图时，他会了解有关相邻格子的信息。地图上有两个房子：一个可能大部分墙壁是红色的鬼屋和一个可能大部分墙壁是蓝色的食物屋。Pacman 的目标是在避开鬼屋的同时进入食物屋。

Pacman 将根据他的观察来推断哪个房子是鬼屋哪个是食物屋，并在冒险进屋或收集更多证据之间的权衡作出决定。为此，你将使用贝叶斯网络实现概率推理。

Again, 你可以运行以下代码自己试玩几轮：

```
python hunters.py -p KeyboardAgent -r
```

回顾 Bayes Nets and Factors

首先，我们把这个问题用贝叶斯网络建模。查看 `bayesNet.py` 了解你将使用的类 - `BayesNet` 和 `Factor`。你还可以运行此文件 `python bayesNet.py` 以查看下面这个示例的 `BayesNet` 和相关的 `Factors`。

你应该查看 `bayesNet.py` 中的 `printStarterBayesNet` 函数来对比观察 `python bayesNet.py` 的输出 - 有一些有用的注释可以让你在本堂课中显得更轻松。

在 `printStarterBayesNet` 这个函数中创建的贝叶斯网络将如下图所示：

$(Raining \rightarrow Traffic \leftarrow Ballgame)$

一些术语如下：

- Bayes Net**：这是一个有向无环图和一组条件概率表的概率模型，上面的交通贝叶斯网络就是一个例子。
- Factor**：这存储了一个概率表，尽管表中条目的总和不一定是 1。一个因子的一般形式是 $P(X_1, \dots, X_m, y_1, \dots, y_n | Z_1, \dots, Z_p, w_1, \dots, w_q)$ 。其中小写字母形式的变量已经获得赋值。对于 X_i 和 Z_j 变量的每个可能的赋值，这个因子存储一个数值。 Z_j, w_k 是这个因子的条件变量， X_i, y_l 是非条件变量。
- 条件概率表 Conditional Probability Table (CPT): 这是一个满足以下两个性质的因子。
 - 对于条件变量的每个赋值，其条目的总和必须为 1
 - 非条件变量只有一个（回忆一下，贝叶斯网络中的 CPT 的形式，每个变量结点有一个 CPT）

例如：交通贝叶斯网络存储以下条件概率表: $P(Raining), P(Ballgame), P(Traffic|Ballgame, Raining)$ 。

Question 1-3 请查看上节课的实验文档并先完成

Question 4: Eliminate 报告应实现任务[1分]

- 说明**：该小题截图只需截 `python autograder.py -q q4` 的运行结果，获得评分器给的4分中，这道题才算得它占的1分。

在 `factorOperations.py` 中实现 `eliminate` 函数。它输入一个 `Factor` 和一个要被消除的变量并返回一个不包含该变量的新 `Factor`。这对应于将 `Factor` 中的所有条目相加，这些条目仅在要被消除的变量的值上有所不同。

要测试和调试你的代码，请运行

```
python autograder.py -q q4
```

在调试期间运行特定测试可能会很有用，以便只看到一组因子打印出来。例如，要仅运行第一个测试，请运行：

```
python autograder.py -t test_cases/q4/1-simple-eliminate
```

Hints and Observations:

- 你的 `eliminate` 应该返回一个新的 `Factor`。
- `eliminate` 可用于边缘化概率表中的变量。例如：
 - $\text{eliminate}(P(X, Y|Z), Y) = P(X|Z)$
 - $\text{eliminate}(P(X, Y|Z), X) = P(Y|Z)$
- 对于一般消元操作，哪些变量在返回的因子中是非条件（非证据）变量的？哪些变量是条件变量？
- 请记住，`Factor` 不仅存储它们使用的无条件变量和条件变量，还存储原始 `BayesNet` 的 `variableDomainsDict`。因此，返回的 `Factor` 应该具有与输入 `Factor` 相同的 `variableDomainsDict`。

Question 5: Normalize 报告应实现任务[1分]

- 说明：该小题截图只需截 `python autograder.py -q q5` 的运行结果，获得评分器给的4分中，这道题才算得它占的1分。

在 `factorOperations.py` 中实现 `normalize` 函数。它输入 `Factor` 并对其进行归一化，也就是说，它缩放 `Factor` 中的所有条目，使得 `Factor` 中的条目之和为 1。如果输入因子中的概率之和为 0，你应该返回 `None`。

要测试和调试你的代码，请运行

```
python autograder.py -q q5
```

在调试期间写一些print函数打印一些过程变量，运行特定测试查看可能会很有用，以便只看到一组因子打印出来。例如，要仅运行第一个测试，请运行：

```
python autograder.py -t test_cases/q5/1-preNormalized
```

Hints and Observations:

- 你的 `normalize` 应该返回一个新的 `Factor`。
- `normalize` 不会影响概率分布（因为概率分布的总和必须已经为 1）。
- 对于一般的 `normalize` 操作，返回的 `Factor` 中哪些变量是无条件的？哪些变量是有条件的？请务必阅读 `normalize` 的文档注释以获取更多说明。
- 请记住，`Factors` 存储原始 `BayesNet` 的 `variableDomainsDict`，不仅存储它们使用的无条件变量和条件变量。因此，返回的 `Factor` 应该具有与输入 `Factor` 相同的 `variableDomainsDict`。

Question 6: Variable Elimination 报告应实现任务[1分]

- 说明：该小题截图只需截 `python autograder.py -q q6` 的运行结果，获得评分器给的4分中，这道题才算得它占的1分。

在 `inference.py` 中实现 `inferenceByVariableElimination` 函数。它能回答一个概率查询，该查询使用 `BayesNet`、查询变量列表和证据表示。

要测试和调试你的代码，请运行

```
python autograder.py -q q6
```

同样，在调试期间写一些print函数打印一些过程变量，运行特定测试查看可能会很有用，以便只看到一组因子打印出来。例如，要仅运行第一个测试，请运行：

```
python autograder.py -t test_cases/q6/1-disconnected-eliminate
```

Hints and Observations:

- 算法应该以参数中的 `eliminationOrder` 遍历隐藏变量，合并并消除该变量，直到只剩下查询和证据变量。
- 输出的因子的概率总和应为 1（因此它是一个真实的条件概率，以证据为条件）。
- 查看 `inference.py` 中的 `inferenceByEnumeration` 函数下的注释，了解如何使用所需函数的示例。（显然，你应该需要用上你在 `factorOperations.py` 写的几个函数）
- 你需要注意你加入的因子只有一个无条件变量的特殊情况（函数下面的文档说明更详细地指定了要做什么）。

报告6 “Bayesnet”提交说明

- 按照第13周和本周第14周的Question 1, 3-6 的要求实现代码和获得运行结果截图
- 提交压缩包命名为“姓名_学号_报告序号.zip” (如“彭振辉_2106666_报告6.zip”)
- 压缩包应包含内容：
 - 已实现的完整项目文件夹“**Project_6_Bayesnet_Full**”
 - 其中应包含 Question 1, 3-6 的要求实现代码
 - 一个doc或pdf说明文档，上面需要有：
 - 开头一段说明 “整体实现参考 + 2-3句简要体会（如教训、思路、拓展应用等）”，如： -
 - “自行实现。挑战最大的是xxx内容，初始时报了什么错，通过什么方式解决，该部分的实现思路为xxx”
 - “xxx内容参考xxx同学/xxx网址。思考不出算法思路，探究后学习到了什么方法。”
 - 要求实现的5个任务的成功运行截图，说明截图对应任务。