

设计实验室11 6.01 – 2011年秋季

走廊里的机器人

目标： 设计实验室11为机器人定位估算奠定了基础
它从一个不确定的位置开始，沿着走廊向下移动。你将：

- 探索观测模型与转换模型
- 机器人在走廊中移动的贝叶斯状态估计仿真
- 构建机器人位置定位真实系统

资源： 本实验应单独完成。

Do `athrun 6.01 getFiles`。相关文件（在`~/ Desktop/6.01/designLab11`）是 `•designLab11 Work.py`：用于模拟彩色走廊状态估计的代码

1 介绍

当我们有一个内部状态无法直接观测的系统时，就可以考虑 *状态估计* 问题，即试图根据与系统状态相关的观测来了解系统内部隐藏状态。此类系统的例子包括：

- 复印机的隐状态是其内部机械的运行状态，我们可采取的操作是进行复印，而观察结果则是复印质量。
- 机器人在走廊中移动，其隐藏状态即为移动位置。可执行的动作包括向东或向西移动，而观察结果为颜色（这些颜色可能无法准确反映墙壁的真实底色）。
- 在视频游戏中，玩家的隐藏状态是其试图击杀的怪物，游戏中不存在显性动作，观察指标为玩家的移动行为。

状态估计的核心在于：通过接收系统输入（我们通常称之为“动作”）和系统观测数据，计算出系统隐藏状态的概率分布。在接下来的实验课中，我们将运用基础 **状态估计技术**，基于含噪的声呐和里程计读数，构建一个机器人姿态估计系统。

进行概率状态估计时，我们需要一个包含三个组成部分的模型：

- 初始分布：指系统状态的概率分布，用于描述任意状态作为初始状态的概率。
- 观测模型：一种条件概率分布，用于描述在给定状态下观察到各可能观测结果的概率。
- 转移模型：一个条件概率分布，告诉我们 $t+1$ 时刻处于每个状态的概率，给定 t 时刻的状态和 t 时刻的动作

部分软件与设计实验室配置了`athrun 6.01 getFiles`命令。请忽略该指令。

相同文件以.zip文件形式在6.01 OCW Scholar网站上提供，标记为[设计或软件实验室编号]代码。

我们将首先系统学习观测模型与状态转移模型，通过简单模拟场景培养您对这些理论应用的直觉，随后深入建模真实机器人系统，为第13次设计实验做好充分准备。

2 走廊世界

我们将采用一种抽象模拟方法，研究机器人在由固定数量彩色房间构成的一维走廊中上下移动的行为。

- 机器人初始时知晓房间总数，但无法确定自身所在房间。
- 该机器人知晓其在每个房间内将进行哪些观察（颜色）的概率分布。
- 该机器人可在走廊各方向尝试移动，但其移动并不总是完全可靠。若机器人紧贴走廊左端并向左移动，则会停留在最左侧房间；同理，当处于最右侧房间时向右移动也会停留在该房间。

机器人的目标是确定它所在的房间。我们将使用[状态估计](#)来解决这个问题。

步骤1.以-n参数启动空闲状态，并加载fiLedesignLab11Work.py并运行它。（我们将不使用Soar，这是一个独立的软件）。现在，输入

```
p = makePerfect()
p.运行(10)
```

此时，p是一个小应用程序的实例，它既模拟了机器人在世界中的移动，又显示了估计的置信状态。

每个方格对应世界中的一个房间。每个房间具有一个“真实”颜色，该颜色决定了机器人在该房间内进行观察时的分布情况。房间的真实颜色显示在每个方格的外缘。

机器人[信念状态](#)通过概率分布来表示其所在房间。在窗口界面中，当前信念状态通过各区块内方框的颜色呈现——颜色越接近零值时红色越鲜艳，接近1值时蓝色越明亮。黑色则对应均匀分布的概率值（此处为0.2）。每个房间对应的概率值也会以文字形式显示在对应的方框内。

事实上，p是一个[随机状态机](#)的组合，它模拟了机器人世界系统的行为和一个[状态机](#)通过机器人动作和观测数据进行[状态估计](#)，在每个时间步计算新的置信状态。

该复杂机器的工作原理如下：

- 初始化p时，每次调用p.run的开始时：
 1. 状态估计器将[信念状态初始化](#)为起始信念状态，在此情况下，起始信念状态是房间上的均匀分布。
 2. 模拟器从起始分布中随机选择机器人的[初始起始位置](#)。请注意，[状态估计器并不知道这个真实位置](#)；它只是在模拟中使用。它也不会窗口或打印输出中以任何方式显示。
- 在p:

1. 模拟器生成一个观察值，从与机器人当前实际所在房间相关的观察值分布中抽取。该观察值是一个颜色名称，如“白色”或“绿色”。
 2. 模拟器会提示用户执行一个操作。该操作必须是介于 -4 和 4 之间的整数。
 3. 模拟机器人移动的房间数量取决于指定的动作；但若机器人的运动模型存在噪声（我们将在后文详细讨论其具体含义），则其实际移动的房间数未必与指令完全一致。此外，机器人不会移动超过走廊的任一端点。当动作值为0时，机器人将尝试保持在当前位置。
 4. 状态估计器根据其旧的置信状态和观测值进行观测值更新。更新取决于观测值模型，该模型为每个状态规定了观测值的概率分布。此置信状态被打印出来。
 5. 状态估计器根据观测更新和指定动作产生的置信状态，进行状态估计的转移更新。该更新取决于转移模型，该模型规定了在给定先前状态和动作的情况下，下一状态的概率分布。
 6. 此时，方块被重新绘制，其颜色和数字反映了新信念状态中的概率。
- 如果将quit输入为操作，整个计算机将终止。除非您调用带较大数字参数的run，否则它将在10步后停止。

如果要再次运行机器，最好创建一个新实例；如果在旧实例上再次调用run也可以，但在执行一次更新之前，初始信念显示将是不正确的。

检查自己 1.在完美的模拟器中移动机器人。确保你理解代表信念状态的颜色含义，以及Python shell中打印出的数字是否有意义。如果需要，可以随时向工作人员寻求帮助。

步骤2.你刚刚创建的世界具有完美的运动和完美的感知。您可按以下步骤创建并运行一个包含噪声运动与感知的系统：

```
n = makeNoisy()
n.运行(20)
```

检查自己 2.在嘈杂的模拟器中移动机器人。确保你理解颜色的含义，并对可能发生的情况有一个基本的概念。如果需要，可以随时向工作人员询问澄清。

2.1 观测模型

读数的第7.6节可能有助于理解接下来的章节。

观察模型是一个条件概率分布，它规定了机器人在特定房间内看到的颜色： $P(O_t = o_t | S_t = s_t)$ 。在我们的案例中， o_t 的取值范围是

黑色、白色、红色、绿色、蓝色、紫色、橙色、深绿色、金色、巧克力色、百香果纹、午夜蓝、热粉、黄绿色

并且遍历机器人可能所在的所有房间。

当存在 m 种可能的观测结果和 n 种可能的位置时，通常需要 $m \cdot n$ 个参数来定义观测模型。为简化模型，我们在此问题中引入一个关键假设：机器人的观测结果仅取决于其所处房间的实际颜色。具体而言，所有实际为白色的房间具有相同的观测结果分布，而所有实际为绿色的房间也具有相同的观测结果分布（这种分布通常与实际白色房间的观测分布存在差异）。基于这个假设，我们只需要指定 $P(\text{observedColor} | \text{actualColor})$ ，然后我们就可以找到在任何房间中观察到每种颜色的概率，只要我们知道那个房间的实际颜色：

$$P(O_t = \text{观测颜色} | S_t = s_t) = P(O_t = \text{观测颜色} | \text{实际颜色} = \text{actualColor}(s_t))$$

条件概率分布

$$P(O_t = \text{观测颜色} | \text{实际颜色})$$

观测噪声分布是指根据机器人所在房间的实际颜色来确定观测颜色分布的函数。在Python中，我们可以将观测噪声分布定义为一个过程，该过程以实际颜色作为输入，并返回观测颜色的分布。这里有一个非常简单的示例，它始终能观测到真实颜色。

```
def perfectObsNoiseModel(actualColor): 返回
    距离。DDist({actualColor: 1.0})
```

现在，给定一个观测噪声分布`obsNoise`，例如在完美观测噪声模型下，我们可以构建完整的观测模型（即给定机器人位置时对观测颜色的条件概率分布），如下所示：

```
def makeObservationModel(hallwayColors, obsNoise):
    返回局部变量 loc: obsNoise(hallwayColors[loc])
```

其中，`hallwayColors`是一个列表，用于指定走廊中每个位置的真实颜色；`loc`是一个整数，表示机器人的位置；`obsNoise`则是给定真实颜色时观察到颜色的条件分布。该过程会返回一个条件概率分布，其作用是将位置作为输入参数，输出对应颜色的观测概率分布。

我们一直使用的示例世界是通过以下方式指定的

标准走廊颜色配置：白色、白色、绿色、白色、白色

根据这些过程，我们可以用以下方式指定完美观测的观测模型：`perfectObsModel=makeObservationModel`

`(standardHallway, perfectObsNoiseModel)`

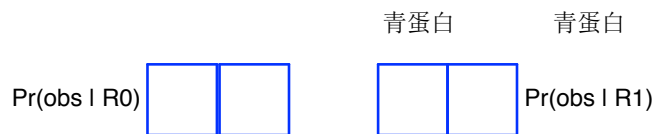
步骤3。

W周11.1.1

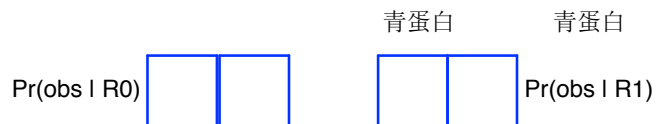
完成这个关于定义观察模型的辅导题。

检查自己3.为了确保你理解观察模型，考虑一个有两个房间的世界：房间R0实际上是绿色的，R1实际上是白色的。

- 对于一个完美的传感器，每个房间的观测值概率分布如何？



- 使用whiteEqGreenObsDist，每个房间的观测值概率分布是什么？



- 对于whiteVsGreenObsDist，每个房间的观测值的概率分布是什么？



步骤 4.现在考虑一个更大的测试世界，称为testHallway。两端的房间具有真彩色，chocolate。其余的房间为绿色或白色。

尝试whiteEqGreenObsDist和whiteVsGreenObsDist，看看信念状态会发生什么变化。将教程中这些过程的定义粘贴到design □ Lab11Work.py文件中。使用下面所示的完美运动模型。**请确保使用类似 whiteEqGreenObsDist 的变量作为下面的第三个参数。**（actions、standardDynamics和perfectTransNoiseModel变量已在designLab11Work.py中为您定义。）

```
w = makeSim (testHallway, actions, <your observedColor given actualColor model>,
            standardDynamics, perfectTransNoiseModel)w.run
```

(50)

2.2 状态转移模型

状态转移模型描述的是在给定时间 t 的状态和所选动作 a 时，系统在时间 $t+1$ 的状态 $t+1$ 的条件概率分布。具体来说，就是 $P(S_{t+1}|S_t = s_t, A_t = a_t)$ 。系统的下一个状态既取决于当前状态，也与所采取的动作相关。如果将这个模型用矩阵表示，其规模会非常庞大：假设 n 为系统状态数， m 为动作数，那么矩阵的维度就是 mn^2 。

通常，转移模型可以更简洁或系统化的方式进行描述。在此特定场景中，机器人可尝试将若干房间向左或向右移动，或保持当前位置不变。我们假设机器人在尝试沿特定方向移动时产生的错误类型不依赖于其实际位置（除非位于世界边缘），并进一步假设大多数状态的转移概率为零（例如机器人不可能瞬间移动到走廊的另一端）。这将使我们能够更简洁地描述转移模型。

我们将首先通过定义动态过程（例如下文的`standardDynamics`）来构建状态转移模型。该动态过程用于计算机器人在通道长度个位置中的当前位置采取某个动作后的名义新位置。这种机制在其他问题中同样适用。机器人可执行的动作包括： $-4, \dots, -1, 0, 1, \dots, 4$ 。因此，我们只需将机器人当前位置与其动作值相加即可得到名义新位置，但需确保其移动不会超出世界边界，为此我们使用`util`函数进行边界校验。`clip`以防止数值低于0或高于`hallwayLength-1`。

默认标准动态（位置，动作，走廊长度）：返回`utilclip`（位置 + 动作，0，走廊长度-1）

接下来，我们定义一个噪声模型，它与位置无关。该模型返回一个分布，该分布描述了在给定动力学条件下，从某个动作得出的名义位置可能产生的结果位置。最简单的噪声模型假设状态转换是完美的，因此结果位置将与名义位置相同。

调用`def perfectTransNoiseModel (nominalLoc, hallwayLength)`：返回距离。`DDist ({nominalLoc : 1.0})`

最终，我们需要建立一个完整的转移模型，即形式为 $Pr(S_{t+1}|S_t, A_t)$ 的条件概率分布。由于它以两个变量为条件，我们将使用嵌套过程（表示条件分布）来表示它。所以，我们可以把它看作是 $Pr(S_{t+1}|S_t|A_t)$ 的某种形式，或者，我们可以把它看作是一个过程，它接受一个 a_t 并返回一个过程，该过程接受一个 s_t 并返回一个 S_{t+1} 的分布。

该过渡模型的基本形式包含两个过程：动力学过程和噪声模型，以及一个表示走廊长度的整数。

```
def def transitionModel (动力学参数、噪声分布、走廊长度):
    return lambda act: lambda loc: noiseDist(dynamics(loc, act, hallwayLength),
                                              hallwayLength)
```

在标准动力学条件下，我们构建了标准走廊的完美过渡模型，具体如下：

```
perfectTransModel = makeTransitionModel(standardDynamics, perfectTransNoiseModel, 5)
```

步骤5。

W周11.1.2 完成导师布置的关于定义转移模型的习题。

步骤6.现在设想一个仅包含白色房间的测试场景，其状态转移与观测数据均存在噪声干扰。我们将使用初始信念状态对状态估计器进行初始化，该初始状态将所有位置的概率值统一设为1。

7（当然，所有其他位置的概率均为0）。变量sterile用于定义由16个白色房间组成的走廊。您可以通过以下方式创建该世界模型和状态估计器：

```
w = makeNoisyKnownInitLoc (7, sterile)
```

w运行(50)

连续0几次选择动作，会发生什么？当你驾驶机器人时，会发生什么？

Checkoff 1. whiteEqGreenObsDist和whiteVsGreenObsDist如何观测模型（来自《Check Yourself 3》）与以下内容进行比较：

- 一个完美的传感器模型
- 无论处于何种房间，传感器始终显示‘黑色’。向工作人员演示步骤6中具有噪声动态特征的场景，并解释其工作原理。

3 走廊世界中的状态估计

务必仔细阅读7.5-7.7章节的阅读材料。

我们将通过若干状态估计的数值示例，帮助您建立状态估计的直觉。这类问题正是我们期望您在测验和考试中能够解决的类型。

步骤7。

W周11.1.4 在走廊世界中进行状态估计问题的求解。

步骤8。

W周11.1.5 在走廊世界中进行状态估计问题的求解。

4 准备本地化

在设计实验室13中，我们将构建一个系统，使机器人能够实现‘自我定位’：即根据世界障碍物地图及局部环境感知能力，估算其在世界中的位置。

声呐读数。这些问题构成了定位系统的重要概念和组成部分。

步骤9。

W周11.1.6 声纳几何结构解析

步骤10。

W周11.1.7 计算机器人姿态下的理想声呐读数。

MIT开放课程

<http://ocw.mit.edu>

6.01SC 电气工程与计算机科学导论

2011年春季

关于引用这些材料或我们的使用条款的信息，请访问：<http://ocw.mit.edu/terms>。