

第20章 用贝叶斯网学习和动作

20.1 学习贝叶斯网

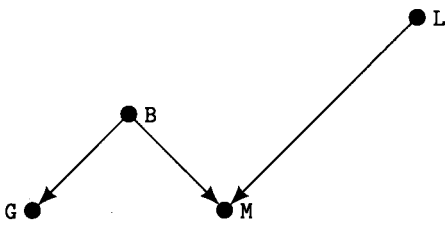
学习一个贝叶斯网的问题是寻找一个网络，它能最好地匹配（按照某个记分度量）一个数据训练集（*training set*），是所有（至少有一些）变量值的实例集合。说“寻找一个网”，我们的意思是既要找到DAG结构，也要找到与DAG中每个节点相关的条件概率表（CPT）。

20.1.1 已知网络结构

如果知道网络的结构，那么只需找到CPT，我们首先描述这种情况。通常，人类专家能对一个问题领域提出适当的结构但不能做出CPT。在我们必须学习网络结构的情况下，学习CPT仍是必需的。学习CPT有一个比较容易的和一個比较难的两种情况。在容易的情况下，没有缺失的数据，即训练集合的每个成员对网中表达的每个变量有一个值。然而在更实际的设置中，情况常常是一些训练记录的变量值缺失了；缺失的数据导致更难以学习CPT。

1. 无缺失数据

首先假定没有缺失任何数据。这里，如果有充足的训练样本，我们只要计算每个节点和它的双亲的采样统计信息。假如给定双亲 $\mathcal{P}(V_i)$ ，我们想得到某个节点 V_i 的CPT。遵守前面的约定，我们用 v_i 指称 V_i 的值。 V_i 的表和它具有的不同值（小于1）一样多。在布尔表达式中，再次假定，对每个节点仅有一个CPT。设 V_i 有 k_i 个父节点。因为每个双亲有两个可能值，那么在表中有 2^{k_i} 项（行）。我们用向量变量 P_i 指称与 V_i 的双亲有关的变量，用向量值 p_i 指称这些变量的值。采样统计结果 $\hat{p}(V_i = v_i | P_i = p_i)$ ，由中有 $V_i = v_i$ 和 $P_i = p_i$ 的采样数除以有 $P_i = p_i$ 的采样数得到。为了学习CPT，我们仅仅将实际的这些采样统计结果用于网中的所有节点。



G	M	B	L	实例数
True	True	True	True	54
True	True	True	False	1
True	False	True	True	7
True	False	True	False	27
False	True	True	True	3
False	False	True	False	2
False	False	False	True	4
False	False	False	False	2
				100

图20-1 一个网络和一些示例值

用一个例子可以使这个计算更清楚。考虑一个同图 19-2有相同结构的贝叶斯网，在图 20-1中重复它，但没有CPT。假如我们观察了图中G、M、B和L的100组值（注意到有些组合没有出现，有些比其他出现得更频繁）。为了计算采样概率 $\hat{p}(B = \text{True})$ ，我们只要计算在所有的采样中B为True出现的次数。得到 $\hat{p}(B = \text{True}) = 0.94$ 。同样， $\hat{p}(L = \text{True}) = 0.68$ 。对节点B和L，这些概率正是它们的CPT所需要的。

我们用下面解释的典型计算方式计算节点M的CPT行：为了计算 $\hat{p}(M = \text{True} | B = \text{True}, L = \text{False})$ （简称为 $\hat{p}(M | B, \neg L)$ ），计算M为True、B为True，L为False的次数，并除以B为True、L为

False的次数。我们得到 $\hat{p}(M|B, \neg L) = 0.03$ 。对节点G我们进行相似的计算。可以计算整个采样统计结果集，把它们与图19-2中给出的CPT比较。

注意，在例子中，有些采样统计是基于很小的采样的——导致相应的基础概率的可能不精确评估。一般地讲，一个CPT的指数级数量的大量参数可能无法使训练集对这些参数产生良好评估的能力。如果很多参数有相同（或接近相同）的值，可能会减轻这个问题。[Friedman & Goldszmidt 1996a]已经探索过如何使用这个冗余，减少在一个CPT中必须评估的参数数量的技术。

同样，在观察采样前，我们可以有CPT中项的先验概率。给定一个训练集，给先验概率合适的权值。Bayesian更新了CPT，当然这个过程有点复杂（见[Heckerman, Geiger, & Chickering 1995]）。当有一个非常大的训练集时，先验的作用被极大地减小了。

2. 缺失数据

在收集被一个学习过程使用的训练数据中，常常发生数据缺失。有时，要被捕获的数据不经意地缺失了，有时数据缺失本身是重要的。这里处理第一种情况。一个简单、收敛的迭代计算采样统计过程已被证明是对之有效的[Lauritzen 1991]。用刚刚描述的例子来介绍该方法的主要思想。假如，不用图20-1中的数据，而用如下数据：

G	M	B	L	实例数
True	True	True	True	54
True	True	True	False	1
*	*	True	True	7
True	False	True	False	27
False	True	*	True	3
False	False	True	False	2
False	False	False	True	4
False	False	False	False	2

星号*表示变量值组中与那个位置相关的变量值缺失了。问题是当试图评估这个网络的CPT时，我们如何处理这些缺失值？先考虑三次采样，其中 $G = \text{False}$ ， $M = \text{True}$ ， $L = \text{True}$ ，B的值缺失的情况。这三次采样中的每一次可能有 $B = \text{True}$ 或 $B = \text{False}$ ，我们不知道是哪一个。但对这些采样，我们知道G、M和L的值。因此，虽然不知道B的值，但给定了G、M和L的值，我们能计算B的概率 $p(B|\neg G, M, L)$ 。这个概率能用前面讲述的概率推理方法计算——工作在网络结构（图20-1）和网络的CPT上，条件是我们有这些CPT（当然，我们还没有它们，但是将简要讨论这个问题）。因此，为了计算采样统计以估计该网络的CPT，三次采样中的每一次能用两个加权采样代替——一个是 $B = \text{True}$ ，用 $p(B|\neg G, M, L)$ 加权，另一个是 $B = \text{False}$ ，权值为 $p(\neg B|\neg G, M, L) = 1 - p(B|\neg G, M, L)$ （顺便提一下，从图20-1有 $p(B|\neg G, M, L) = p(B|\neg G, M)$ ，因为给定G和M，B条件独立于L。）

我们把相同的过程应用到7次采样 $B = \text{True}$ ， $L = \text{True}$ ，G和M的值缺失的情形。这些采样中的每一个可由对应组合 (G, M) ， $(G, \neg M)$ ， $(\neg G, M)$ 和 $(\neg G, \neg M)$ 的4个加权采样代替，权值分别是概率 $p(G, M|B, L)$ ， $p(G, \neg M|B, L)$ ， $p(\neg G, M|B, L)$ 和 $p(\neg G, \neg M|B, L)$ 。我们可以再次用网络结构和CPT计算这些概率（当在任何一个采样中的缺失值数量很大时，存在指数爆炸的采样危险）。

现在，我们能用加权采样（其中，缺失值已被填充——用所有可能的方法）和其他采样（它们中没有缺失值）一起进行频率统计以计算 CPT 的估计。这个过程与在没有缺失值中描述的过程相同，除了一些计数现在不是整个数量（因为加权）外。但是正如前面提到的，为了通过概率推理计算权值，我们需要 CPT，然而我们没有它。一个叫期望最大化（EM）[Dempster, Laird, & Rubin 1977]的方法可以用来对一个 CPT 集合调零。首先，我们为整个网络的 CPT 中的参数选择随机值，用这些随机值计算需要的权值（给定观察要数据的值时缺失数据值的条件概率），然后用这些权值反过来估计新的 CPT。我们迭代这个过程，直到 CPT 收敛，保证收敛能达到。在大部应用中，收敛是快速的。

即使这个有缺失值的小例子也需要冗长的计算，应用 EM 方法时最好让计算机程序去执行概率推理和频率统计。

20.1.2 学习网络结构

如果不知道网络结构，那么我们必须设法去找出这个结构以及相关的 CPT，以使它能最好地符合训练数据。为此，需要一个尺度对候选网络打分，我们需要指定一个过程以在可能的结构中搜索。在本小节将讨论这两个问题。

1. 记分制

有几个度量方法可用来对网络打分。一种方法是基于描述长度的。它的思想是：假如我们想发送训练集 给某个人，为此，把变量值编码成一个位串，然后发送位。我们需要多少位呢？即，消息的长度是多少？有效的编码利用要发送数据的统计属性，这些统计属性正是我们企图在贝叶斯网中建模的。如果找到一个适当的贝叶斯网，我们能基于它使用哈夫曼编码对要传输的数据编码。根据信息理论 [Cover & Thomas 1991]，一组数据 \mathcal{D} ，按照一个给定贝叶斯网 \mathcal{B} 的组合概率分布，其最好编码需要 $L(\mathcal{D}, \mathcal{B})$ 比特：

$$L(\mathcal{D}, \mathcal{B}) = -\log p[\mathcal{D}]$$

其中 $p[\mathcal{D}]$ 是被发送的特殊数据的概率（按照 \mathcal{B} 规定的联合概率）。给定一些特殊数据 \mathcal{D} ，我们企图找到网络 \mathcal{B} ，它使 $L(\mathcal{D}, \mathcal{B})$ 最小化。在了解这个方法之前，我们先计算 $\log p[\mathcal{D}]$ 。假定数据 \mathcal{D} 包含 m 个采样值： v_1, \dots, v_m ，每个 v_i 是 n 个变量值的一个 n 维向量， $p(\mathcal{D})$ 是联合概率 $p[v_1, \dots, v_m]$ 。假定每个数据按照 \mathcal{B} 指定的概率分布被独立提供，我们有

$$p(\mathcal{D}) = \prod_{i=1}^m p(v_i)$$

和

$$-\log p(\mathcal{D}) = -\sum_{i=1}^m \log p(v_i)$$

其中每个 $p(v_i)$ （由 v_i 指称变量值的联合概率）从贝叶斯网 \mathcal{B} 计算。这些计算虽然是冗长的，但确实被用来对各种试验贝叶斯网计分。当然，每个网络不但包括网络图结构，而且有 CPT。可以证明：给定一个网络结构和一个训练集 \mathcal{D} ，使 $L(\mathcal{D}, \mathcal{B})$ 最小的 CPT 是从 \mathcal{D} 计算的采样统计结果获得的 [Friedman & Goldszmidt 1996a]。

但是单独的 $L(\mathcal{D}, \mathcal{B})$ 不是一个非常好的度量，因为它的使用促成了有很多弧的大网络。这样一个网络对 \mathcal{D} 过于特殊化，即它过于适合数据。对记分度量的适当调整能这样实现；为了

用一个基于 \mathcal{B} 的有效编码把 发送给某个人，我们也必须传递 \mathcal{B} 的描述以便接收者能够对消息解码。这样，我们必须加一项到 $L(\mathcal{E}, \mathcal{B})$ ，这个项是传输 \mathcal{B} 需要的消息长度。粗略地讲，传送 \mathcal{B} 要求的比特数是 $\frac{|\mathcal{B}| \log m}{2}$ ，其中 $|\mathcal{B}|$ 是 \mathcal{B} 中的参数个数， $\frac{\log m}{2}$ 一般被认为是适合表示每个数字参数的比特数。因此调整后的记分制 $L'(\mathcal{E}, \mathcal{B})$ 为：

$$L'(\mathcal{E}, \mathcal{B}) = - \sum_{i=1}^m \log p(v_i) + \frac{|\mathcal{B}| \log m}{2}$$

现在，我们搜索一个给出数据和网络编码的最小描述长度的网络。使用两个因子允许我们对发送数据和网络做出更好的权衡。

作为一个例子，对图 19-2 中显示的网络和图 20-1 中所示的发送数据，我们计算 $L'(\mathcal{E}, \mathcal{B})$ 。首先，我们计算 $L(\mathcal{E}, \mathcal{B})$ ，即发送图 20-1 中的数据要求的比特数——假定数据由图 19-2 中的贝叶斯网给定的一个概率分布提供。图 20-1 中表的第 1 项的概率是

$$\begin{aligned} p(\text{第1项}) &= p(G|B)p(M|B, L)p(B)p(L) \\ &= 0.95 \times 0.9 \times 0.95 \times 0.7 = 0.569 \end{aligned}$$

采用负对数（对以 2 为底的），产生

$$-\log p(\text{第1项}) = 0.814$$

在数据中有 54 个这种“第 1 项”，因此 54 个第 1 项对 $L(\mathcal{E}, \mathcal{B})$ 的作用结果是 $54 \times 0.814 = 43.9$ 。表中其他项的总结果分别是 6.16、27.9、52.92、16.33、12.32、24.83 和 12.32。把这些结果加起来得到：

$$L(\mathcal{E}, \mathcal{B}) = 196.68 \text{ bit}$$

下面，我们计算 $\frac{|\mathcal{B}| \log m}{2}$ ，发送图 19-2 的网络需要的位数。在这个网中有 8 个参数。因此

$$\frac{|\mathcal{B}| \log 100}{2} = 4 \times 6.64 = 26.58 \text{ bit}$$

因此这个网络的记分度量是

$$L'(\mathcal{E}, \mathcal{B}) = 196.68 + 26.58 = 223.26 \text{ bit}$$

其他的网络可用同样的方式评估，而图 19-2 中的网络大概是最理想的。

2. 搜索网络空间

当然，所有可能的贝叶斯网集合是如此之大，以致我们不可能企求一种详尽的搜索以发现一个使 $L'(\mathcal{E}, \mathcal{B})$ 最小的网络。一种可能是利用下山搜索或“贪婪”搜索。即我们从一个给定的网络开始（比如一个没有弧的网络，假定它独立于所有的变量），估计该网络的 $L'(\mathcal{E}, \mathcal{B})$ ，然后对它做一些小改变，来看这些改变产生的网络是否减少了 $L'(\mathcal{E}, \mathcal{B})$ 。这些小改变可以是加一条弧，或减一条弧，或者掉转一条弧的方向。每发生一个改变，我们用从 \mathcal{E} 中导出的采样统计计算改变网络的 CPT。然后这些 CPT 被用来计算 $L'(\mathcal{E}, \mathcal{B})$ 的部分 $-\sum_{i=1}^m \log p(V_i)$ 。新网络中的参数数量用来计算部分 $(|\mathcal{B}| \log m)/2$ 。这些计算可以简化，由于描述长度计算可分解成网络中每个 CPT 上的计算。当记分度量可分解时，总度量是局部度量的和 [Friedman & Goldszmidt 1996a]。因此，当一个弧被加、被减或被反向时，我们只需计算在采样统计中的变

化和改变涉及到的节点 V_i 的 $p(V_i | \mathcal{P}(V_i))$, 其他的 $p(V_i | \mathcal{P}(V_i))$ 保持不变。除了描述长度, 可分解度量也被用来评价一个网络适合数据的程度 (参见 [Heckerman, Geiger & Chickering 1995])。

对一些非试验问题, 贝叶斯网学习方法已用来学习网络结构和PT。作为一个例子, 考虑图20-2中的网络, 给出三个网络, 第一个是对一个问题中37个变量的关系编码, 这个问题是一个用于医院特护单元通风设备管理的警报系统。这个已知网络被用来产生7个节点随机值的、大小为10 000的训练集。用这个随机采样, 从第二个网络开始 (相互之间无任何依赖性), 第三个网络用类似于刚刚描述的方法来学习 (详见 [Spirtes & Meek 1995])。注意在结构上非常相似——仅去掉了一个弧。

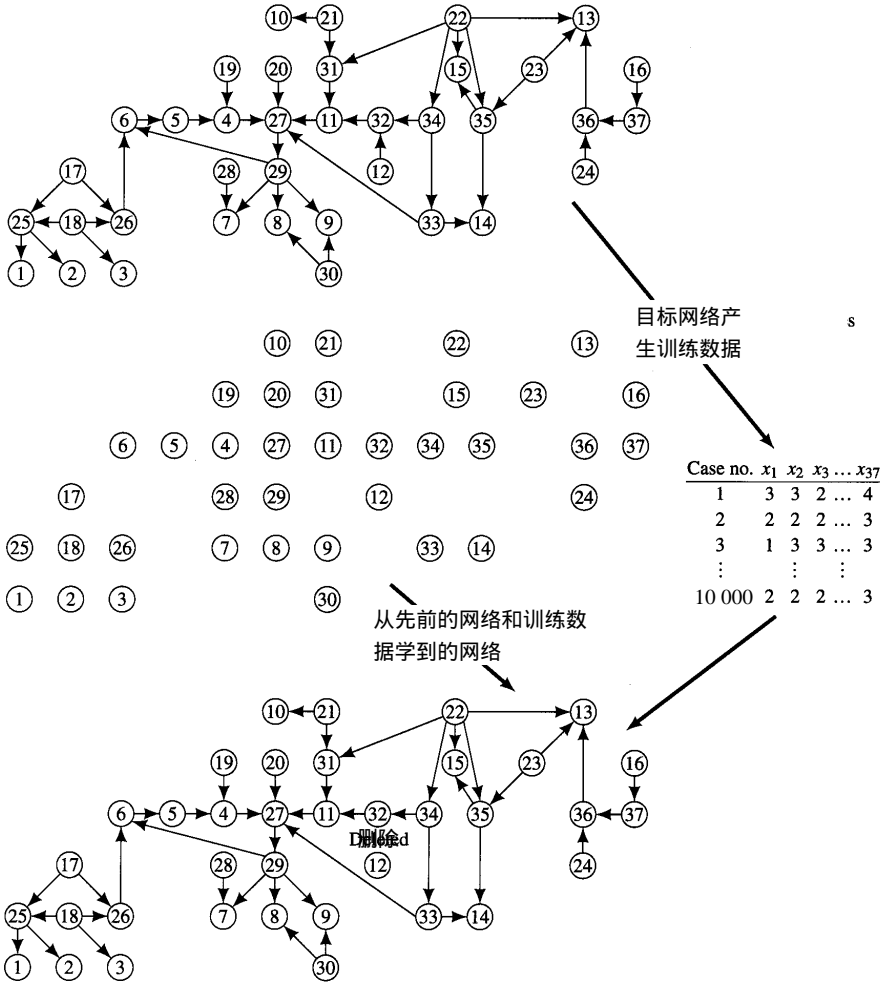


图20-2 在网中学习的一个实验

有时, 网络结构可以通过增加节点而大大地简化。这些节点所代表的变量值没有在训练集中给定。这些节点叫做隐藏节点。作为一个简单的例子, 考虑图 20-3中所示的两个贝叶斯网。左边的网络比右边有隐藏节点 H 的网络有更多的参数; 如果右边的网和它一样好或更好 (如果它是变量因果关系的更好表示), 那么它的描述长度分数将会更糟。由于我们不能度量隐藏变量, 必须用搜索过程虚构它的存在。因此, 贪婪搜索必须向它的可能改变的列表中添加一个新节点 (见 [Heckerman 1996])。相应的变量值当然是“缺失数据”, 和这个变量相关的概率必须

通过前面描述的EM方法引证[⊖]。

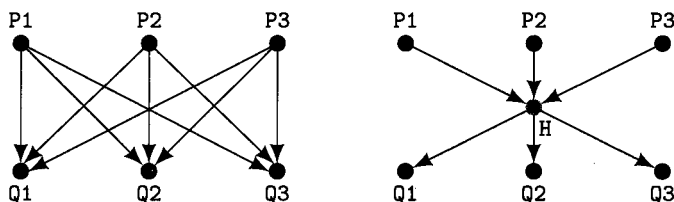


图20-3 两个网络——其中一个有一个隐藏变量节点

20.2 概率推理与动作

20.2.1 一般设置

有很多概率推理的应用。一个著名的应用是在专家系统中根据应用到特殊数据的一般知识做出最好的猜测。诊断给定症状的疾病就是一个例子。下面描述使用贝叶斯网的概率推理如何能被一个agent利用，这个agent必须根据给定感知信息和关于环境状态的记分尺度决定下一个最好动作（我的处理基于[Russell & Norvig 1995, 第17章]，它利用了[Dean & Wellman 1991, 第17章]描述的一个方法）。

待研究的问题是第10章所处理问题的一个一般化。回忆一下，在那里描述了一个agent使用一个感知/计划/动作循环，计划阶段通过朝着目标方向预测在当前环境状态下计算要被采用的下一个最好动作（有时仅仅到达一个有限的范围），动作阶段执行由计划者推荐的第一个动作，感知阶段设法辨别下一次循环的结果环境状态。在第10章，也已将一个“目标”符号一般化为在一定的环境状态下给出（或得到）的一个“奖赏”目录。这个奖赏反过来按照总计的打折未来奖赏为每个状态导出一个“值”，其中的未来奖赏被一个执行动作以使它的奖赏最大化的agent实现。这里保留这个一般化的目标符号，把一个应用归因于每个环境状态。

前面，对感知/计划/动作的结构采用了关于环境和感知动作可靠性的有些不一致的假设集合。假定一个agent通过感知能准确决定它的当前状态，能精确预测它的所有动作结果。然而，倘若这些假设是不合理的（它们常常是不合理的），我们的agent采用一个简单动作，并立即用它的传感器去发现实现产生的环境状态。用下面的说法使这个方法合理化：即使采用的动作并不总会有它的预期效果，传感器有时会出错，但传感器（平均地）将使agent通过状态空间知道它的进展，并且重复计划将再次引导agent朝着需要的目标（或奖赏）前进。

现在，有工具允许我们更适当地处理不确定性，就能明确地采用更现实的假设。新的agent不知道它在哪个状态，它仅仅知道它在各种状态的概率。agent的传感器不能给出环境状态的确切知识，它最多只能加深这些状态的概率。动作结果仅仅被大概地了解：在一个给定状态下采取的动作可能导致一组新状态中的任何一个——每一个都有相应的概率。通过计划和感知，我们想让agent选择使它的预期应用最大化的那个动作。一个agent能在它的完全一般化中处理这个问题需要的计算量非常大，并需要再一次近似和进一步的限制假设。在下面的一个特殊例子中讨论这些问题。

[⊖] 在学习贝叶斯网中虚构隐藏节点类似于在学习命题规则中虚构新原子或在学习逻辑程序中虚构新的关系量（谓词）。

20.2.2 一个扩展的例子

由于各种假设极大地增加了计算难度，下面的例子使用一个简单 agent 和环境，以免模糊了我们的主要思想。考虑一个机器人，它处于图 20-4 所示的一个有 5 个单元的一维网格中。环境状态只涉及到机器人的位置，我们用一个状态变量 E 表示它， E 有 5 个可能值 $\{-2, -1, 0, 1, 2\}$ ，每个位置对机器人有一个应用 U 。中间单元有一个应用 0，它和其他的应用值显示在图中。为了开始过程，我们假定机器人（精确地）知道当 $t = 0$ 时，它在标志为 0 的单元中，即 $E_0 = 0$ 。

机器人在第 i 个时间步骤采取的动作用符号 A_i 指称。它试图向左移一个单元 ($A_i = L$) 或向右移一个单元 ($A_i = R$)。无论哪种情况，一个移动有预期结果的概率为 0.5；每个动作没有任何结果的概率是 0.25，一个动作使机器人以与预期相反的方向移动到相邻单元的概率为 0.25。因此，在几次移动后，机器人只有它实际位置的概率知识。

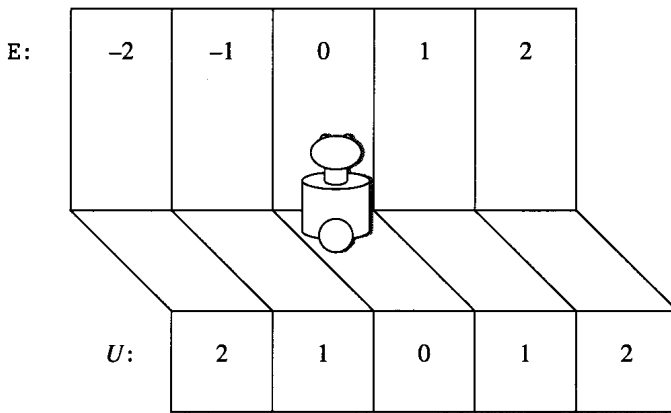


图20-4 限制在5个单元的机器人

机器人在第 i 个时间步骤通过一个感知信号 s_i 感知它的位置。但是我们假定那个传感器有点不可靠。假如 E_i 有确定值， s_i 有同样值的概率是 0.9。 s_i 有每个其他值的概率是 0.025。

面对各种障碍，机器人的问题是做出使它的应用的期望值提前几步最大化的移动。如果我们使预期的应用仅提前一步最大化，用来选择机器人的下一步动作的决策技术最容易解释。假定当 $t = 0$ 时机器人企图向右移动，即 $A_0 = R$ 。所得到的环境状态由 E_1 给出。为了继续感知/计划/动作循环，机器人通过观察 $s_1 = 1$ 感知它的位置。它下一步采取什么移动呢？确实，在做了那个移动后，它如何基于感知数据、对当前状态所做推理和动作和效果来？

使用一个叫做动态决策网的特殊信念网的概率推理，来选择最大化应用动作。在图 20-5 中显示了适合这个问题的网络。这个网络允许机器人在采用动作时和新感知的信息变得可用时迭代地进行推理，给定 $E_0 = 0$ 、 $A_0 = R$ 和 $s_0 = 1$ 后，我们能用普通的概率推理计算期望的应用值 U_2 ，它先由 $A_1 = R$ 产生，再由 $A_1 = L$ 产生。然后机器人选择给出更大值的动作（这种情况

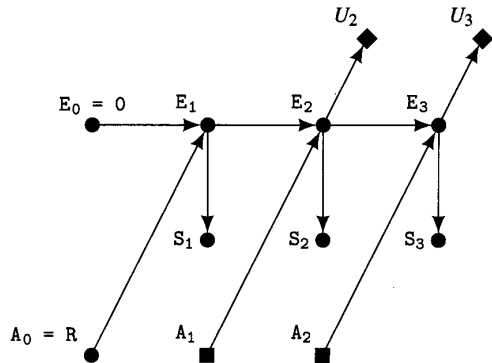


图20-5 一个动态决策网

下的动作选择基于机器人仅向前看一步。进一步的向前看将涉及到对可选动作序列计算更远的应用)。在动态决策网中使用不同形状节点表示这些节点的不同假设。方形节点表示变量的值仍在agent的完全控制之下,它们被称为决策点。在一个动态决策网中,在agent已经做了决策之后,它们成为(有已知值)普通信念网节点。菱形节点表示应用值的变量。应用变量的期望值是它们双亲的各种值的概率函数。

由网络结构注意到该环境是马尔可夫模型,也就是说,在 $t+1$ 时刻,环境状态所能知道的东西(没有感知的)完全取决于 t 时刻的环境状态和动作(概率地)。在这种意义下,大多数agent环境能被假设是马尔可夫模型(或者能引入附加的变量以使它们成为那样)。也要注意网络结构包含了假设——给定时刻 t 的环境,在时刻 t 感知的东西条件独立于以前的每个事情。

在 $t=1$ 时,通过采取一个计划的动作把时间前沿向前移动一步(通过使预期应用 U_2 最大化的过程进行计划),并在感知了下一个状态 E_2 时能被感知的东西 S_2 后,整个处理用另一个感知/计划/动作循环重复。执行一些计算将有助于理解。

首先,给定时刻 $t=0$ 时已知的东西,和假定 A_1 的两个不同概率,我们需要计算两个预期应用:

$$\text{Ex}[U_2 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R]$$

和

$$\text{Ex}[U_2 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = L]$$

为了计算这两个预期值,对 A_1 的两个值中的每一个的 E_2 的不同值,我们要计算 $P(E_2 | E_0 = 0, A_0 = R, S_1 = 1, A_1)$ 。这些计算形式在后面的步骤中将被重复,但为了具体,在此问题的步骤中都写出它。这个形式对 A_1 的每个值也是相同的,因此仅写出 $A_1=R$ 的情形。我们用上一章解释的polytree算法计算 $P(E_2 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R)$ 。

首先,我们“包含没有给定的双亲”,得到:

$$\begin{aligned} p(E_2 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R) \\ = \sum_{E_1} p(E_2 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R, E_1) p(E_1 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R) \end{aligned}$$

考虑条件独立性,这个等式能被简化:

$$\begin{aligned} p(E_2 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R) \\ = \sum_{E_1} p(E_2 | A_1 = R, E_1) p(E_1 | E_0 = 0, A_0 = R, S_1 = 1) \end{aligned}$$

在用于机器人动作选择的决策网中,上述求和的第一项叫做马尔可夫过程的动作模型。对一个给定的前一个状态和动作,它给出各种可能的随后状态的概率。根据polytree算法,求和中的第二项用贝叶斯法则重写为:

$$p(E_1 | E_0 = 0, A_0 = R, S_1 = 1) = k p(S_1 = 1 | E_0 = 0, A_0 = R, E_1) p(E_1 | E_0 = 0, A_0 = R)$$

其中 k 是一个标准化因子,它被选择(后面)以使概率和为1。再使用条件独立,我们有

$$p(E_1 | E_0 = 0, A_0 = R, S_1 = 1) = k p(S_1 = 1 | E_1) p(E_1 | E_0 = 0, A_0 = R)$$

上述结果中的第1项叫感知模型。对很多环境状态,它给出了传感器将有各种值的概率(对每个环境状态,一个完全可靠和提供最多信息的传感器,将把所有的概率集中到一个传感器值)。

第2项又是一个动作模型。

汇集我们的结果，产生：

$$p(E_2 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R) \\ = k \sum_{E_1} p(E_2 | A_1 = R, E_1) p(S_1 = 1 | E_1) p(E_1 | E_0 = 0, A_0 = R)$$

为了评估这个表达式（为 E_2 的各种值），使用我们已经做出的关于动作结果和传感器可信度的概率假设。为了说明，我们仅计算 $p(E_2 = 1 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R)$ 。计算涉及到下面的概率，它们是图20-5中网络的CPT条目（我们必须对所有的 E_1 值求和）。

$$p(E_2 = 1 | A_1 = R, E_1 = 0) = 0.5$$

$$p(E_2 = 1 | A_1 = R, E_1 = 1) = 0.25$$

$$p(E_2 = 1 | A_1 = R, E_1 = 2) = 0.25$$

$$p(E_2 = 1 | A_1 = R, E_1 = -1) \text{ 和 } p(E_2 = 1 | A_1 = R, E_1 = -2) \text{ 都等于 } 0。$$

$$p(S_1 = 1 | E_1 = -2) = 0.025$$

$$p(S_1 = 1 | E_1 = -1) = 0.025$$

$$p(S_1 = 1 | E_1 = 0) = 0.025$$

$$p(S_1 = 1 | E_1 = 1) = 0.9$$

$$p(S_1 = 1 | E_1 = 2) = 0.025$$

$$p(E_1 = -1 | E_0 = 0, A_0 = R) = 0.25$$

$$p(E_1 = 0 | E_0 = 0, A_0 = R) = 0.25$$

$$p(E_1 = 1 | E_0 = 0, A_0 = R) = 0.5$$

$$p(E_1 = -2 | E_0 = 0, A_0 = R) \text{ 和 } p(E_1 = 2 | E_0 = 0, A_0 = R) \text{ 都等于 } 0。$$

执行求和产生

$$p(E_2 = 1 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R) \\ = k \times [(0.5 \times 0.025 \times 0.25) + (0.25 \times 0.9 \times 0.5)] \\ = k \times 0.14375$$

我们对 E_2 的其他值执行类似的计算以得到 $p(E_2 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R)$ 。由于所有这些的和为1，我们能求出 k 。用 E_2 的这些概率，给定 $A_1 = R$ ，我们计算 U_2 的预期值。重复这个过程来计算给定 $A_1 = L$ 的 U_2 的预期值，并选择产生更大值的动作（从问题的结构我们知道它将是 $A_1 = R$ 。但是，用这个例子仅仅说明这个方法——没有什么惊奇的）。

20.2.3 一般化举例

分析方程 $p(E_2 | E_0 = 0, A_0 = R, S_1 = 1, A_1 = R)$ ，允许我们把它扩展到随后的时间步。这个方程在 $t = 1$ 时被估计，就像 $S_1 = 1$ 已被感知和动作 $A_1 = R$ 被采取前一样。其他“给定”的值成为过去。因此，我们能改写为 $p(E_2 | < \text{values before } t = 1 >, S_1 = 1, A_1 = R)$ 。同样，在求和中的表达式 $p(E_1 | E_0 = 0, A_0 = R)$ 能被写为 $p(E_1 | < \text{values before } t = 1 >)$ 。用这些改变，我们有：

$$p(E_2 | < \text{values before } t = 1 >, S_1 = 1, A_1) \\ = k \sum_{E_1} p(E_2 | A_1, E_1) p(S_1 = 1 | E_1) p(E_1 | < \text{values before } t = 1 >)$$

将等式一般化为：

$$p(E_{i+1} | < \text{values before } t = i >, S_i = s_i, A_i) \\ = k \sum_{E_i} p(E_{i+1} | A_i, E_i) p(S_i = s_i | E_i) p(E_i | < \text{values before } t = i >)$$

为了决定动作，当我们处理时，用下面的方式使用这个等式。为了计算在 $t = i$ 时要采取的动作 A_i ：

- 1) 从上一个时间步 ($i - 1$) (以及感知到 $S_{i-1} = s_{i-1}$ 后)，我们已经对 E_i 的所有值计算了 $p(E_i | < \text{values before } t = i >)$ 。
- 2) 在 $t = i$ 时，感知 $S_i = s_i$ ，并且用感知模型计算 E_i 所有值的 $p(S_i = s_i | E_i)$ 。
- 3) 根据动作模型，计算 E_i 和 A_i 所有值的 $p(E_{i+1} | A_i, E_i)$ 。
- 4) 对 A_i 的每个值和 E_{i+1} 的一个特定值，我们对 E_i 所有值上的 $p(E_{i+1} | A_i, E_i) p(S_i = s_i | E_i) p(E_i | < \text{values before } t = i >)$ 求和，并乘以一个常量 k ，产生和 $p(E_{i+1} | < \text{values before } t = i >, S_i = s_i, A_i)$ 成比例的值。
- 5) 对 E_{i+1} 的所有其他值，重复上述过程，计算常量 k 得到对每个 E_{i+1} 和 A_i 值的 $p(E_{i+1} | < \text{values before } t = i >, S_i = s_i, A_i)$ 的实际值。
- 6) 用这些概率值，计算对 A_i 的每个值 U_{i+1} 的预期值，选择使预期值最大的那个 A_i 。
- 7) 采取在上一步选择的动作， i 加 1，循环。

这就是使用一个动态决策网络选择动作的要素。该方法可扩展到其他应用。动作影响环境的方式和环境影响传感刺激的方式一般通过类似图 20-5 中的网络来很好地建模。代替有一个单一环境变量 E_i ，在每个时间步，我们可以有一个值向量 $E_i = (E_{i1}, \dots, E_{in})$ 。同样，代替有一个单一的传感变量 S_i ，我们可以有一个值向量 $S_i = (S_{i1}, \dots, S_{im})$ 。当然动态决策网在每个时间步必须扩展到包括所有的这些变量节点和它们的依赖，但是计算的一般形式和这个简单例子中的一样。虽然网络对一个给定的时间步本质上会复杂一些，但马尔可夫假设简化了时间步之间的依赖。

向前看超过一步涉及到向前传播概率到要计算一个预期应用的点。显然，这样一个扩展迅速地变得不实际。同时也变得不是非常有用，因为 E_i 的概率分布变得相当扩散。因此，在表达这个概率扩展之前所做的有点不一致的假设保留了一个合理的选择。

20.3 补充读物和讨论

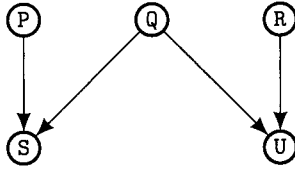
贝叶斯网学习是一个每年有很多重要新论文出现的活跃的研究领域。[Neal 1991]描述了用神经网络学习贝叶斯网的方法。这里描述的用于评估建议的贝叶斯网的技术使用了最小描述长度的概念[Rissanen 1984]。[Friedman 1997]描述了当不知道一个网络结构且有缺失数据时学习贝叶斯网的技术。[Cooper & Herskovitz 1992]做了关于学习贝叶斯网的早期工作。

[Forbes, et al. 1995]提出了一个使用动态决策网驱动一个自动设备的系统。

在随机状态下的应用评估构成了决策理论的主旨。在 AI 中使用决策理论的一个事例见 [Horvitz, Breese, & Henrion 1988]。Markov 决策问题 (MDP) [Puterman 1994] 和部分可观察 Markov 决策问题 (POMDP) [Cassandra, Kaelbling, & Littman 1994] 理论提供了在随机状态下动作结果的基本理论模型。

习题

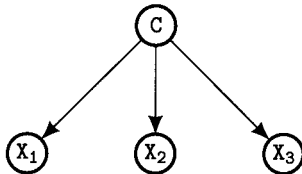
20.1 考虑有下面结构的贝叶斯网：



假如我们能控制“原因变量”P、Q和R，即，我们能安排实验以让这些变量可以取值（True或False）的任意组合，然后观察S和U的结果。为了学习S和U的CPT，将需要这些变量的什么组合？从这个实验，我们能学到P、Q和R的先验概率吗？

20.2 计算图20-3中每个网络的参数数量。

20.3 一个贝叶斯网有下图所示的结构（注意，这个结构暗示 x_1 条件独立于给定的C）。给定 x_1, x_2, x_3 值的一个训练集，每一对有一个C值，给出如何应用采样统计以计算 $p(C | x_1, x_2, x_3)$ 的一个估计。



20.4 用上面习题的假设和结果，假定C代表两个动作之一的名字，这两个动作是一个机器人在它的二值传感器输入有由 x_1, x_2, x_3 给出的值时采取的。假定选择一个动作的策略是：如果 $p(C = 1 | x_1, x_2, x_3) > p(C = 2 | x_1, x_2, x_3)$ ，那么选择 $C = 1$ 。证明这样一个策略能被一个输入为 x_1, x_2, x_3 的TLU实现，把你的结果解释为一个训练该TLU的方法（在这个问题中，信念网没有一个因果解释；即应该采取的动作不是传感器输入的一个“原因”）。

20.5 一个机器人生活在一个如下图所示的 5×5 网格中。网格中的数字代表相对温度值。假设机器人开始在一个温度 = 2的单元中。它不知道从哪一个温度 = 2的单元开始，但它有一个网格地图，显示了每个单元的温度。

4	5	6	7	8
3	4	5	6	7
2	3	4	5	6
1	2	3	4	5
0	1	2	3	4

假定机器人能精确感知它所占据单元的温度值。它有4个动作，用north、east、south和west表示，为了使它的预期温度最大化，它必须选择其中之一。一股强风正从西南方向吹来。通常，每个动作将按指示的方向移动机器人一个单元，但由于有风，无论何时当机器人在一个温度 = 2的单元中时，动作有下述结果：

south 没影响

west 没影响
 north 向北移动一个单元的概率为0.5
 north 向北移动两个单元的概率为0.25
 north 向东移动一个单元的概率为0.25
 east 向东移动一个单元的概率为0.5
 east 向东移动二个单元的概率为0.5

- 1) 画出对应一个动作步的一个动态决策网。给出在 CPT 中的相关项。
- 2) 对4个动作中的每一个，计算期望的温度 $Ex[T_1 | A_0]$ 。
- 3) 现在假定在时刻 $t = t_0$ 机器人不能确信它的温度。相反，它感知到一个信号 $S_0 = 2$ ，它告诉机器人关于它的温度。 S_0 的传感模型包括

$$p(S_0 = 2 | T_0 = 2) = 0.9$$

$$p(S_0 = 2 | T_0 = 3) = 0.3$$

$$p(S_0 = 2 | T_0 = i) = 0 \quad \text{对所有的 } i \text{ 不等于 } 2 \text{ 或 } 3 \text{ 的值。}$$

为以上情况画一个一步动态决策网。

- 4) 假定在一个温度 = 3 的单元中采取的动作和温度 = 2 的单元中有相同的结果。在这种情况下，来自4个动作的每一个的期望温度是什么？