第十六章　深度学习中的结构化概率模型

深度学习为研究者们提供了许多建模方式，用以设计以及描述算法。其中一种形式是**结构化概率模型**（structured probabilistic model）的思想。我们曾经在第3.14节中简要讨论过结构化概率模型。此前简要的介绍已经足够使我们充分了解如何使用结构化概率模型作为描述第二部分中某些算法的语言。现在在第三部分，我们可以看到结构化概率模型是许多深度学习重要研究方向的关键组成部分。作为讨论这些研究方向的预备知识，本章将更加详细地描述结构化概率模型。本章内容是自洽的，所以在阅读本章之前读者不需要回顾之前的介绍。

结构化概率模型使用图来描述概率分布中随机变量之间的直接相互作用，从而描述一个概率分布。在这里我们使用了图论（一系列结点通过一系列边来连接）中“图”的概念，由于模型结构是由图定义的，所以这些模型也通常被称为**图模型**（graphical model）。

图模型的研究社群是巨大的，并提出过大量的模型、训练算法和推断算法。在本章中，我们将介绍图模型中几个核心方法的基本背景，并且重点描述已被证明对深度学习社群最有用的观点。如果你已经熟知图模型，那么你可以跳过本章的绝大部分。然而，我们相信即使是资深的图模型方向的研究者也会从本章的最后一节中获益匪浅，详见第16.7节，其中我们强调了在深度学习算法中使用图模型的独特方式。相比于其他图模型研究领域的是，深度学习的研究者们通常会使用完全不同的模型结构、学习算法和推断过程。在本章中，我们将指明这种区别并解释其中的原因。

我们首先介绍了构建大规模概率模型时面临的挑战。之后，我们介绍如何使用一个图来描述概率分布的结构。尽管这个方法能够帮助我们解决许多挑战和问题，它本身仍有很多缺陷。图模型中的一个主要难点就是判断哪些变量之间存在直接的相

互作用关系，也就是对于给定的问题哪一种图结构是最适合的。在第 16.5 节中，我们通过了解 **依赖**（dependency），简要概括了解决这个难点的两种方法。最后，作为本章的收尾，我们在第 16.7 节中讨论深度学习研究者使用图模型特定方式的独特之处。

16.1 非结构化建模的挑战

深度学习的目标是使得机器学习能够解决许多人工智能中亟需解决的挑战。这也意味着它们能够理解具有丰富结构的高维数据。举个例子，我们希望AI的算法能够理解自然图片[1]，表示语音的声音信号和包含许多词和标点的文档。

分类问题可以把这样一个来自高维分布的数据作为输入，然后使用一个类别的标签来概括它——这个标签可以是照片中是什么物品，一段语音中说的是哪个单词，也可以是一段文档描述的是哪个话题。这个分类过程丢弃了输入数据中的大部分信息，然后产生单个值的输出（或者是关于单个输出值的概率分布）。这个分类器通常可以忽略输入数据的很多部分。例如，当我们识别一张照片中的一个物体时，我们通常可以忽略图片的背景。

我们也可以使用概率模型完成许多其他的任务。这些任务通常相比于分类成本更高。其中的一些任务需要产生多个输出。大部分任务需要对输入数据整个结构的完整理解，所以并不能舍弃数据的一部分。这些任务包括以下几个：

- **估计密度函数**：给定一个输入 $\boldsymbol{x}$，机器学习系统返回一个对数据生成分布的真实密度函数 $p(\boldsymbol{x})$ 的估计。这只需要一个输出，但它需要完全理解整个输入。即使向量中只有一个元素不太正常，系统也会给它赋予很低的概率。

- **去噪**：给定一个受损的或者观察有误的输入数据 $\tilde{\boldsymbol{x}}$，机器学习系统返回一个对原始的真实 $\boldsymbol{x}$ 的估计。举个例子，有时候机器学习系统需要从一张老相片中去除灰尘或者抓痕。这个系统会产生多个输出值（对应着估计的干净样本 $\boldsymbol{x}$ 的每一个元素），并且需要我们有一个对输入的整体理解（因为即使只有一个损坏的区域，仍然会显示最终估计被损坏）。

- **缺失值的填补**：给定 $\boldsymbol{x}$ 的某些元素作为观察值，模型被要求返回一个 $\boldsymbol{x}$ 一些或者全部未观察值的估计或者概率分布。这个模型返回的也是多个输出。由于这个模型需要恢复 $\boldsymbol{x}$ 的每一个元素，所以它必须理解整个输入。

[1] 自然图片指的是能够在正常的环境下被照相机拍摄的图片，不同于合成的图片，或者一个网页的截图等等。

- **采样**：模型从分布 $p(\boldsymbol{x})$ 中抽取新的样本。其应用包括语音合成，即产生一个听起来很像人说话的声音。这个模型也需要多个输出以及对输入整体的良好建模。即使样本只有一个从错误分布中产生的元素，那么采样的过程也是错误的。

图 16.1 中描述了一个使用较小的自然图片的采样任务。

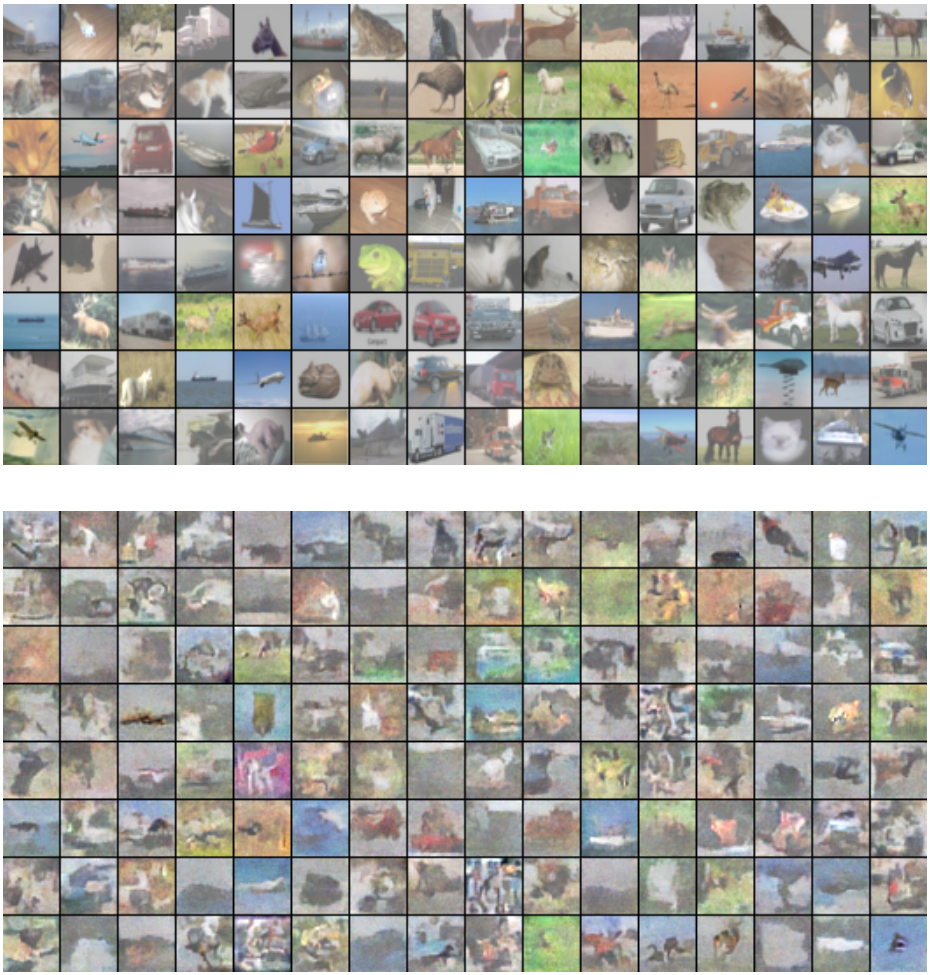

图 16.1: 自然图片的概率建模。*(上)* CIFAR-10 数据集 (Krizhevsky and Hinton, 2009) 中的 32×32 像素的样例图片。*(下)* 从这个数据集上训练的结构化概率模型中抽出的样本。每一个样本都出现在与其欧氏距离最近的训练样本的格点中。这种比较使得我们发现这个模型确实能够生成新的图片，而不是记住训练样本。为了方便展示，两个集合的图片都经过了微调。图片经 Courville *et al.* (2011a) 许可转载。

对上千甚至是上百万随机变量的分布建模，无论从计算上还是从统计意义上说，都是一个极具挑战性的任务。假设我们只想对二值的随机变量建模。这是一个最简单的例子，但是我们仍然无能为力。对一个只有 32×32 像素的彩色（RGB）图片来说，存在 2^{3072} 种可能的二值图片。这个数量已经超过了 10^{800}，比宇宙中的原子总数还要多。

通常意义上讲，如果我们希望对一个包含 n 个离散变量并且每个变量都能取 k 个值的 $\boldsymbol{x}$ 的分布建模，那么最简单的表示 $P(\boldsymbol{x})$ 的方法需要存储一个可以查询的表格。这个表格记录了每一种可能值的概率，则需要 k^n 个参数。

基于下述几个原因，这种方式是不可行的：

- *内存：存储参数的开销。* 除了极小的 n 和 k 的值，用表格的形式来表示这样一个分布需要太多的存储空间。

- *统计的高效性：*当模型中的参数个数增加时，使用统计估计器估计这些参数所需要的训练数据数量也需要相应地增加。因为基于查表的模型拥有天文数字级别的参数，为了准确地拟合，相应的训练集的大小也是相同级别的。任何这样的模型都会导致严重的过拟合，除非我们添加一些额外的假设来联系表格中的不同元素（正如第 12.4.1 节中所举的回退或者平滑 n-gram 模型）。

- *运行时间：推断的开销。* 假设我们需要完成这样一个推断的任务，其中我们需要使用联合分布 $P(\mathbf{x})$ 来计算某些其他的分布，比如说边缘分布 $P(\mathrm{x}_1)$ 或者是条件分布 $P(\mathrm{x}_2 \mid \mathrm{x}_1)$。计算这样的分布需要对整个表格的某些项进行求和操作，因此这样的操作的运行时间和上述高昂的内存开销是一个级别的。

- *运行时间：采样的开销。* 类似的，假设我们想要从这样的模型中采样。最简单的方法就是从均匀分布中采样，$u \sim \mathrm{U}(0, 1)$，然后把表格中的元素累加起来，直到和大于 u，然后返回最后一个加上的元素。最差情况下，这个操作需要读取整个表格，所以和其他操作一样，它也需要指数级别的时间。

基于表格操作的方法的主要问题是我们显式地对每一种可能的变量子集所产生的每一种可能类型的相互作用建模。在实际问题中我们遇到的概率分布远比这个简单。通常，许多变量只是间接地相互作用。

例如，我们想要对接力跑步比赛中一个队伍完成比赛的时间进行建模。假设这个队伍有三名成员：Alice，Bob 和 Carol。在比赛开始时，Alice 拿着接力棒，开始

跑第一段距离。在跑完她的路程以后，她把棒递给了 Bob。然后 Bob 开始跑，再把棒给 Carol，Carol 跑最后一棒。我们可以用连续变量来建模他们每个人完成的时间。因为 Alice 第一个跑，所以她的完成时间并不依赖于其他的人。Bob 的完成时间依赖于 Alice 的完成时间，因为 Bob 只能在 Alice 跑完以后才能开始跑。如果 Alice 跑得更快，那么 Bob 也会完成得更快。所有其他关系都可以被类似地推出。最后，Carol 的完成时间依赖于她的两个队友。如果 Alice 跑得很慢，那么 Bob 也会完成得更慢。结果，Carol 将会更晚开始跑步，因此她的完成时间也更有可能要晚。然而，在给定 Bob 完成时间的情况下，Carol 的完成时间只是*间接地*依赖于 Alice 的完成时间。如果我们已经知道了 Bob 的完成时间，知道 Alice 的完成时间对估计 Carol 的完成时间并无任何帮助。这意味着我们可以通过仅仅两个相互作用来建模这个接力赛。这两个相互作用分别是 Alice 的完成时间对 Bob 的完成时间的影响和 Bob 的完成时间对 Carol 的完成时间的影响。在这个模型中，我们可以忽略第三种间接的相互作用，即 Alice 的完成时间对 Carol 的完成时间的影响。

结构化概率模型为随机变量之间的直接作用提供了一个正式的建模框架。这种方式大大减少了模型的参数个数以致于模型只需要更少的数据来进行有效的估计。这些更小的模型大大减小了在模型存储、模型推断以及从模型中采样时的计算开销。

16.2 使用图描述模型结构

结构化概率模型使用图（在图论中“结点”是通过“边”来连接的）来表示随机变量之间的相互作用。每一个结点代表一个随机变量。每一条边代表一个直接相互作用。这些直接相互作用隐含着其他的间接相互作用，但是只有直接的相互作用会被显式地建模。

使用图来描述概率分布中相互作用的方法不止一种。在下文中我们会介绍几种最为流行和有用的方法。图模型可以被大致分为两类：基于有向无环图的模型和基于无向图的模型。

16.2.1 有向模型

有向图模型（directed graphical model）是一种结构化概率模型，也被称为**信念网络**（belief network）或者**贝叶斯网络**（Bayesian network）[2](Pearl, 1985)。

之所以命名为有向图模型是因为所有的边都是有方向的，即从一个结点指向另一个结点。这个方向可以通过画一个箭头来表示。箭头所指的方向表示了这个随机变量的概率分布是由其他变量的概率分布所定义的。画一个从结点 a 到结点 b 的箭头表示了我们用一个条件分布来定义 b，而 a 是作为这个条件分布符号右边的一个变量。换句话说，b 的概率分布依赖于 a 的取值。

我们继续第 16.1 节所讲的接力赛的例子，我们假设 Alice 的完成时间为 t_0，Bob 的完成时间为 t_1，Carol 的完成时间为 t_2。就像我们之前看到的一样，t_1 的估计是依赖于 t_0 的，t_2 的估计是直接依赖于 t_1 的，但是仅仅间接地依赖于 t_0。我们用一个有向图模型来建模这种关系，如图 16.2 所示。

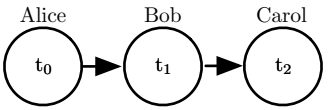

图 16.2: 描述接力赛例子的有向图模型。Alice 的完成时间 t_0 影响了 Bob 的完成时间 t_1，因为 Bob 只能在 Alice 完成比赛后才开始。类似的，Carol 也只会在 Bob 完成之后才开始，所以 Bob 的完成时间 t_1 直接影响了 Carol 的完成时间 t_2。

正式地说，变量 $\mathbf{x}$ 的有向概率模型是通过有向无环图 $\mathcal{G}$（每个结点都是模型中的随机变量）和一系列**局部条件概率分布**（local conditional probability distribution）$p(\mathrm{x}_i \mid Pa_{\mathcal{G}}(\mathrm{x}_i))$ 来定义的，其中 $Pa_{\mathcal{G}}(\mathrm{x}_i)$ 表示结点 x_i 的所有父结点。$\mathbf{x}$ 的概率分布可以表示为

$$p(\mathbf{x}) = \prod_i p(\mathrm{x}_i \mid Pa_{\mathcal{G}}(\mathrm{x}_i)). \tag{16.1}$$

在之前所述的接力赛的例子中，参考图 16.2，这意味着概率分布可以被表示为

$$p(\mathrm{t}_0, \mathrm{t}_1, \mathrm{t}_2) = p(\mathrm{t}_0)p(\mathrm{t}_1 \mid \mathrm{t}_0)p(\mathrm{t}_2 \mid \mathrm{t}_1). \tag{16.2}$$

[2]当我们希望 ‘‘强调’’ 从网络中计算出的值的 ‘‘推断’’ 本质，即强调这些值代表的是置信程度大小而不是事件的频率时，Judea Pearl 建议使用 “贝叶斯网络’’ 这个术语。

这是我们看到的第一个结构化概率模型的实际例子。我们能够检查这样建模的计算开销，为了验证相比于非结构化建模，结构化建模为什么有那么多的优势。

假设我们采用从第 0 分钟到第 10 分钟每 6 秒一块的方式离散化地表示时间。这使得 t_0，t_1 和 t_2 都是一个有 100 个取值可能的离散变量。如果我们尝试着用一个表来表示 $p(t_0, t_1, t_2)$，那么我们需要存储 $999,999$ 个值（100 个 t_0 的可能取值 $\times$ 100 个 t_1 的可能取值 $\times$ 100 个 t_2 的可能取值减去 1，由于存在所有的概率之和为 1 的限制，所以其中有 1 个值的存储是多余的）。反之，如果我们用一个表来记录每一种条件概率分布，那么表中记录 t_0 的分布需要存储 99 个值，给定 t_0 情况下 t_1 的分布需要存储 9900 个值，给定 t_1 情况下 t_2 的分布也需要存储 9900 个值。加起来总共需要存储 $19,899$ 个值。这意味着使用有向图模型将参数的个数减少了超过 50 倍！

通常意义上说，对每个变量都能取 k 个值的 n 个变量建模，基于建表的方法需要的复杂度是 $O(k^n)$，就像我们之前观察到的一样。现在假设我们用一个有向图模型来对这些变量建模。如果 m 代表图模型的单个条件概率分布中最大的变量数目（在条件符号的左右皆可），那么对这个有向模型建表的复杂度大致为 $O(k^m)$。只要我们在设计模型时使其满足 $m \ll n$，那么复杂度就会被大大地减小。

换一句话说，只要图中的每个变量都只有少量的父结点，那么这个分布就可以用较少的参数来表示。图结构上的一些限制条件，比如说要求这个图为一棵树，也可以保证一些操作（例如求一小部分变量的边缘或者条件分布）更加地高效。

决定哪些信息需要被包含在图中而哪些不需要是很重要的。如果变量之间可以被假设为是条件独立的，那么这个图可以包含这种简化假设。当然也存在其他类型的简化图模型的假设。例如，我们可以假设无论 Alice 的表现如何，Bob 总是跑得一样快（实际上，Alice 的表现很大概率会影响 Bob 的表现，这取决于 Bob 的性格，如果在之前的比赛中 Alice 跑得特别快，这有可能鼓励 Bob 更加努力并取得更好的成绩，当然这也有可能使得 Bob 过分自信或者变得懒惰）。那么 Alice 对 Bob 的唯一影响就是在计算 Bob 的完成时间时需要加上 Alice 的时间。这个假设使得我们所需要的参数量从 $O(k^2)$ 降到了 $O(k)$。然而，值得注意的是在这个假设下 t_0 和 t_1 仍然是直接相关的，因为 t_1 表示的是 Bob 完成时的时间，并不是他跑的总时间。这也意味着图中会有一个从 t_0 指向 t_1 的箭头。“Bob 的个人跑步时间相对于其他因素是独立的’’ 这个假设无法在 t_0，t_1，t_2 的图中被表示出来。反之，我们只能将这个关系表示在条件分布的定义中。这个条件分布不再是一个大小为 $k \times k - 1$ 的分别对应着 t_0，t_1 的表格，而是一个包含了 $k - 1$ 个参数的略微复杂的公式。有向图模型的语法并不能对我们如何定义条件分布作出任何限制。它只定义了哪些变量可以作为其中

的参数。

16.2.2 无向模型

有向图模型为我们提供了一种描述结构化概率模型的语言。而另一种常见的语言则是**无向模型**（undirected Model），也被称为**马尔可夫随机场**（Markov random field, MRF）或者是**马尔可夫网络**（Markov network）(Kindermann, 1980)。就像它们的名字所说的那样，无向模型中所有的边都是没有方向的。

当存在很明显的理由画出每一个指向特定方向的箭头时，有向模型显然最适用。有向模型中，经常存在我们理解的具有因果关系以及因果关系有明确方向的情况。接力赛的例子就是一个这样的情况。之前运动员的表现会影响后面运动员的完成时间，而后面运动员却不会影响前面运动员的完成时间。

然而并不是所有情况的相互作用都有一个明确的方向关系。当相互的作用并没有本质性的指向，或者是明确的双向相互作用时，使用无向模型更加合适。

作为一个这种情况的例子，假设我们希望对三个二值随机变量建模：你是否生病，你的同事是否生病以及你的室友是否生病。就像在接力赛的例子中所作的简化假设一样，我们可以在这里做一些关于相互作用的简化假设。假设你的室友和同事并不认识，所以他们不太可能直接相互传染一些疾病，比如说感冒。这个事件太过罕见，所以我们不对此事件建模。然而，很有可能其中之一将感冒传染给你，然后通过你再传染给了另一个人。我们通过对你的同事传染给你以及你传染给你的室友建模来对这种间接的从你的同事到你的室友的感冒传染建模。

在这种情况下，你传染给你的室友和你的室友传染给你都是非常容易的，所以模型不存在一个明确的单向箭头。这启发我们使用无向模型。其中随机变量对应着图中的相互作用的结点。与有向模型相同的是，如果在无向模型中的两个结点通过一条边相连接，那么对应这些结点的随机变量相互之间是直接作用的。不同于有向模型，在无向模型中的边是没有方向的，并不与一个条件分布相关联。

我们把对应你健康状况的随机变量记作 h_y，对应你的室友健康状况的随机变量记作 h_r，你的同事健康的变量记作 h_c。图 16.3 表示这种关系。

正式地说，一个无向模型是一个定义在无向模型 $\mathcal{G}$ 上的结构化概率模型。对于图中的每一个团[3] $\mathcal{C}$，一个**因子**（factor）$\phi(\mathcal{C})$(也称为**团势能**（clique potential）)，

[3]图的一个团是图中结点的一个子集，并且其中的点是全连接的

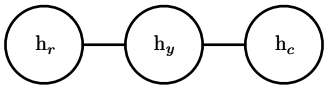

图 16.3: 表示你室友健康状况的 h_r、你健康状况的 h_y 和你同事健康状况的 h_c 之间如何相互影响的一个无向图。你和你的室友可能会相互传染感冒，你和你的同事之间也是如此，但是假设你室友和同事之间相互不认识，他们只能通过你来间接传染。

	$h_y = 0$	$h_y = 1$
$h_c = 0$	2	1
$h_c = 1$	1	10

衡量了团中变量每一种可能的联合状态所对应的密切程度。这些因子都被限制为是非负的。它们一起定义了 **未归一化概率函数**（unnormalized probability function）：

$$\tilde{p}(\mathbf{x}) = \prod_{\mathcal{C} \in \mathcal{G}} \phi(\mathcal{C}). \tag{16.3}$$

只要所有团中的结点数都不大，那么我们就能够高效地处理这些未归一化概率函数。它包含了这样的思想，密切度越高的状态有越大的概率。然而，不像贝叶斯网络，几乎不存在团定义的结构，所以不能保证把它们乘在一起能够得到一个有效的概率分布。图 16.4 展示了一个从无向模型中读取分解信息的例子。

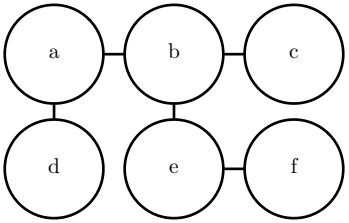

图 16.4: 这个图说明通过选择适当的 ϕ，函数 $p(\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e}, \mathrm{f})$ 可以写作 $\frac{1}{Z}\phi_{\mathrm{a,b}}(\mathrm{a}, \mathrm{b})\phi_{\mathrm{b,c}}(\mathrm{b}, \mathrm{c})\phi_{\mathrm{a,d}}(\mathrm{a}, \mathrm{d})\phi_{\mathrm{b,e}}(\mathrm{b}, \mathrm{e})\phi_{\mathrm{e,f}}(\mathrm{e}, \mathrm{f})$。

在你、你的室友和同事之间感冒传染的例子中包含了两个团。一个团包含了 h_y 和 h_c。这个团的因子可以通过一个表来定义，可能取到下面的值：

状态为 1 代表了健康的状态，相对的状态为 0 则表示不好的健康状态（即感染了感冒）。你们两个通常都是健康的，所以对应的状态拥有最高的密切程度。两个人中只有一个人是生病的密切程度是最低的，因为这是一个很罕见的状态。两个人都

生病的状态（通过一个人来传染给了另一个人）有一个稍高的密切程度，尽管仍然不及两个人都健康的密切程度。

为了完整地定义这个模型，我们需要对包含 h_y 和 h_r 的团定义类似的因子。

16.2.3 配分函数

尽管这个未归一化概率函数处处不为零，我们仍然无法保证它的概率之和或者积分为 1。为了得到一个有效的概率分布，我们需要使用对应的归一化的概率分布 [4]：

$$p(\mathbf{x}) = \frac{1}{Z}\tilde{p}(\mathbf{x}), \tag{16.4}$$

其中，Z 是使得所有的概率之和或者积分为 1 的常数，并且满足：

$$Z = \int \tilde{p}(\mathbf{x})d\mathbf{x}. \tag{16.5}$$

当函数 ϕ 固定时，我们可以把 Z 当成是一个常数。值得注意的是如果函数 ϕ 带有参数时，那么 Z 是这些参数的一个函数。在相关文献中为了节省空间忽略控制 Z 的变量而直接写 Z 是一个常用的方式。归一化常数 Z 被称作是配分函数，这是一个从统计物理学中借鉴的术语。

由于 Z 通常是由对所有可能的 $\mathbf{x}$ 状态的联合分布空间求和或者求积分得到的，它通常是很难计算的。为了获得一个无向模型的归一化概率分布，模型的结构和函数 ϕ 的定义通常需要设计为有助于高效地计算 Z。在深度学习中，Z 通常是难以处理的。由于 Z 难以精确地计算出，我们只能使用一些近似的方法。这样的近似方法是第十八章的主要内容。

在设计无向模型时，我们必须牢记在心的一个要点是设定一些使得 Z 不存在的因子也是有可能的。当模型中的一些变量是连续的，且 $\tilde{p}$ 在其定义域上的积分发散时这种情况就会发生。例如，当我们需要对一个单独的标量变量 $\mathrm{x} \in \mathbb{R}$ 建模，并且单个团势能定义为 $\phi(x) = x^2$ 时。在这种情况下，

$$Z = \int x^2 dx. \tag{16.6}$$

由于这个积分是发散的，所以不存在一个对应着这个势能函数 $\phi(x)$ 的概率分布。有时候 ϕ 函数某些参数的选择可以决定相应的概率分布是否能够被定义。例如，对 ϕ

[4] 一个通过归一化团势能乘积定义的分布也被称作是 **吉布斯分布**（Gibbs distribution）

函数 $\phi(x;\beta) = \exp(-\beta x^2)$ 来说，参数 β 决定了归一化常数 Z 是否存在。正的 β 使得 ϕ 函数是一个关于 x 的高斯分布，但是非正的参数 β 则使得 ϕ 不可能被归一化。

有向建模和无向建模之间一个重要的区别就是有向模型是通过从起始点的概率分布直接定义的，反之无向模型的定义显得更加宽松，通过 ϕ 函数转化为概率分布而定义。这改变了我们处理这些建模问题的直觉。当我们处理无向模型时需要牢记一点，每一个变量的定义域对于一系列给定的 ϕ 函数所对应的概率分布有着重要的影响。举个例子，我们考虑一个 n 维向量的随机变量 $\mathbf{x}$ 以及一个由偏置向量 $\boldsymbol{b}$ 参数化的无向模型。假设 $\mathbf{x}$ 的每一个元素对应着一个团，并且满足 $\phi^{(i)}(\mathrm{x}_i) = \exp(b_i \mathrm{x}_i)$。在这种情况下概率分布是怎样的呢？答案是我们无法确定，因为我们并没有指定 $\mathbf{x}$ 的定义域。如果 $\mathbf{x}$ 满足 $\mathbf{x} \in \mathbb{R}^n$，那么有关归一化常数 Z 的积分是发散的，这导致了对应的概率分布是不存在的。如果 $\mathbf{x} \in \{0,1\}^n$，那么 $p(\mathbf{x})$ 可以被分解成 n 个独立的分布，并且满足 $p(\mathrm{x}_i = 1) = \mathrm{sigmoid}(b_i)$。如果 $\mathbf{x}$ 的定义域是基本单位向量 $(\{[1,0,\dots,0],[0,1,\dots,0],\dots,[0,0,\dots,1]\})$ 的集合，那么 $p(\mathrm{x}) = \mathrm{softmax}(\boldsymbol{b})$，因此对于 $j \neq i$，一个较大的 b_i 的值会降低所有 $p(\mathrm{x}_j = 1)$ 的概率。通常情况下，通过仔细选择变量的定义域，能够从一个相对简单的 ϕ 函数的集合可以获得一个相对复杂的表达。我们会在第 20.6 节中讨论这个想法的实际应用。

16.2.4 基于能量的模型

无向模型中许多有趣的理论结果都依赖于 $\forall \boldsymbol{x},\ \tilde{p}(\boldsymbol{x}) > 0$ 这个假设。使这个条件满足的一种简单方式是使用**基于能量的模型**（Energy-based model, EBM），其中

$$\tilde{p}(\mathbf{x}) = \exp(-E(\mathbf{x})), \tag{16.7}$$

$E(\mathbf{x})$ 被称作是**能量函数**（energy function）。对所有的 z，exp(z) 都是正的，这保证了没有一个能量函数会使得某一个状态 $\mathbf{x}$ 的概率为 0。我们可以完全自由地选择那些能够简化学习过程的能量函数。如果我们直接学习各个团势能，我们需要利用约束优化方法来任意地指定一些特定的最小概率值。学习能量函数的过程中，我们可以采用无约束的优化方法[5]。基于能量的模型中的概率可以无限趋近于 0 但是永远达不到 0。

服从式 (16.7) 形式的任意分布都是**玻尔兹曼分布**（Boltzmann distribution）的一个实例。正是基于这个原因，我们把许多基于能量的模型称为**玻尔兹曼机**

[5]对于某些模型，我们可以仍然使用约束优化方法来确保 Z 存在。

（Boltzmann Machine） (Fahlman *et al.*, 1983; Ackley *et al.*, 1985; Hinton *et al.*, 1984a; Hinton and Sejnowski, 1986)。关于什么时候称之为基于能量的模型，什么时候称之为玻尔兹曼机不存在一个公认的判别标准。一开始玻尔兹曼机这个术语是用来描述一个只有二值变量的模型，但是如今许多模型，比如均值-协方差 RBM，也涉及到了实值变量。虽然玻尔兹曼机最初的定义既可以包含潜变量也可以不包含潜变量，但是时至今日玻尔兹曼机这个术语通常用于指拥有潜变量的模型，而没有潜变量的玻尔兹曼机则经常被称为马尔可夫随机场或对数线性模型。

无向模型中的团对应于未归一化概率函数中的因子。通过 $\exp(a+b) = \exp(a)\exp(b)$，我们发现无向模型中的不同团对应于能量函数的不同项。换句话说，基于能量的模型只是一种特殊的马尔可夫网络：求幂使能量函数中的每个项对应于不同团的一个因子。关于如何从无向模型结构中获得能量函数形式的示例可以参考图 16.5。人们可以将能量函数中带有多个项的基于能量的模型视作是**专家之积**（product of expert） (Hinton, 1999)。能量函数中的每一项对应的是概率分布中的一个因子。能量函数中的每一项都可以看作决定一个特定的软约束是否能够满足的“专家”。每个专家只执行一个约束，而这个约束仅仅涉及随机变量的一个低维投影，但是当其结合概率的乘法时，专家们一同构造了复杂的高维约束。

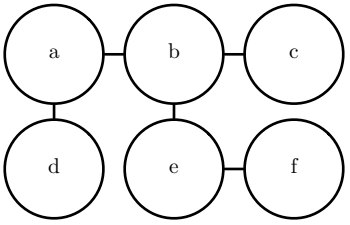

图 16.5: 这个图说明通过为每个团选择适当的能量函数 $E(\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{d},\mathrm{e},\mathrm{f})$ 可以写作 $E_{\mathrm{a,b}}(\mathrm{a},\mathrm{b}) + E_{\mathrm{b,c}}(\mathrm{b},\mathrm{c}) + E_{\mathrm{a,d}}(\mathrm{a},\mathrm{d}) + E_{\mathrm{b,e}}(\mathrm{b},\mathrm{e}) + E_{\mathrm{e,f}}(\mathrm{e},\mathrm{f})$。值得注意的是，我们令 ϕ 等于对应负能量的指数，可以获得图16.4中的 ϕ 函数，比如，$\phi_{\mathrm{a,b}}(\mathrm{a},\mathrm{b}) = \exp(-E(\mathrm{a},\mathrm{b}))$。

基于能量的模型定义的一部分无法用机器学习观点来解释：即式 (16.7) 中的 “-” 符号。这个 “-” 符号可以被包含在 E 的定义之中。对于很多 E 函数的选择来说，学习算法可以自由地决定能量的符号。这个负号的存在主要是为了保持机器学习文献和物理学文献之间的兼容性。概率建模的许多研究最初都是由统计物理学家做出的，其中 E 是指实际的、物理概念的能量，没有任何符号。诸如 “能量” 和 “配分函数” 这类术语仍然与这些技术相关联，尽管它们的数学适用性比在物理中更宽。一些机器学习研究者（例如，Smolensky (1986) 将负能量称为 **harmony**）发出了不同的声

音，但这些都不是标准惯例。

许多对概率模型进行操作的算法不需要计算 $p_{\text{model}}(\boldsymbol{x})$，而只需要计算 $\log \tilde{p}_{\text{model}}(\boldsymbol{x})$。对于具有潜变量 $\boldsymbol{h}$ 的基于能量的模型，这些算法有时会将该量的负数称为 **自由能**（free energy）：

$$\mathcal{F}(\boldsymbol{x}) = -\log \sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{x}, \boldsymbol{h})). \tag{16.8}$$

在本书中，我们更倾向于更为通用的基于 $\log \tilde{p}_{\text{model}}(\boldsymbol{x})$ 的定义。

16.2.5 分离和 d-分离

图模型中的边告诉我们哪些变量直接相互作用。我们经常需要知道哪些变量间接相互作用。某些间接相互作用可以通过观察其他变量来启用或禁用。更正式地，我们想知道在给定其他变量子集的值时，哪些变量子集彼此条件独立。

在无向模型中，识别图中的条件独立性是非常简单的。在这种情况下，图中隐含的条件独立性称为 **分离**（separation）。如果图结构显示给定变量集 $\mathbb{S}$ 的情况下变量集 $\mathbb{A}$ 与变量集 $\mathbb{B}$ 无关，那么我们声称给定变量集 $\mathbb{S}$ 时，变量集 $\mathbb{A}$ 与另一组变量集 $\mathbb{B}$ 是分离的。如果连接两个变量 a 和 b 的连接路径仅涉及未观察变量，那么这些变量不是分离的。如果它们之间没有路径，或者所有路径都包含可观测的变量，那么它们是分离的。我们认为仅涉及未观察到的变量的路径是 “活跃” 的，而包括可观察变量的路径称为 “非活跃” 的。

当我们画图时，我们可以通过加阴影来表示观察到的变量。图 16.6 用于描述当以这种方式绘图时无向模型中的活跃和非活跃路径的样子。图 16.7 描述了一个从无向模型中读取分离信息的例子。

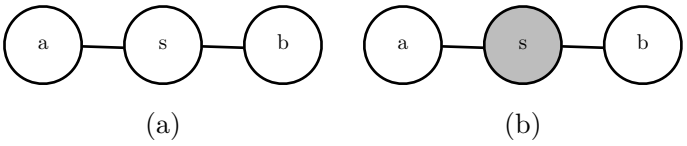

图 16.6: *(a)* 随机变量 a 和随机变量 b 之间穿过 s 的路径是活跃的，因为 s 是观察不到的。这意味着 a，b 之间不是分离的。*(b)* 图中 s 用阴影填充，表示它是可观察的。因为 a 和 b 之间的唯一路径通过 s，并且这条路径是不活跃的，我们可以得出结论，在给定 s 的条件下 a 和 b 是分离的。

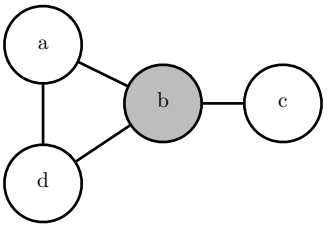

图 16.7: 从一个无向图中读取分离性质的一个例子。这里 b 用阴影填充，表示它是可观察的。由于 b 挡住了从 a 到 c 的唯一路径，我们说在给定 b 的情况下 a 和 c 是相互分离的。观察值 b 同样挡住了从 a 到 d 的一条路径，但是它们之间有另一条活跃路径。因此给定 b 的情况下 a 和 d 不是分离的。

类似的概念适用于有向模型，只是在有向模型中，这些概念被称为 **d-分离**（d-separation）。“d” 代表 “依赖” 的意思。有向图中 d-分离的定义与无向模型中分离的定义相同：如果图结构显示给定变量集 $\mathbb{S}$ 时，变量集 $\mathbb{A}$ 与变量集 $\mathbb{B}$ 无关，那么我们认为给定变量集 $\mathbb{S}$ 时，变量集 $\mathbb{A}$ d-分离于变量集 $\mathbb{B}$。

与无向模型一样，我们可以通过查看图中存在的活跃路径来检查图中隐含的独立性。如前所述，如果两个变量之间存在活跃路径，则两个变量是依赖的，如果没有活跃路径，则为d-分离。在有向网络中，确定路径是否活跃有点复杂。关于在有向模型中识别活跃路径的方法可以参考图 16.8 。图 16.9 是从一个图中读取一些属性的例子。

尤其重要的是要记住分离和d-分离只能告诉我们图中隐含的条件独立性。图并不需要表示所有存在的独立性。进一步的，使用完全图（具有所有可能的边的图）来表示任何分布总是合法的。事实上，一些分布包含不可能用现有图形符号表示的独立性。**特定环境下的独立**（context-specific independences）指的是取决于网络中一些变量值的独立性。例如，考虑三个二值变量的模型：a，b 和 c。假设当 a 是 0 时，b 和 c 是独立的，但是当 a 是 1 时，b 确定地等于 c。当 $a = 1$ 时图模型需要连接 b 和 c 的边。但是图不能说明当 $a = 0$ 时 b 和 c 不是独立的。

一般来说，当独立性不存在时，图不会显示独立性。然而，图可能无法编码独立性。

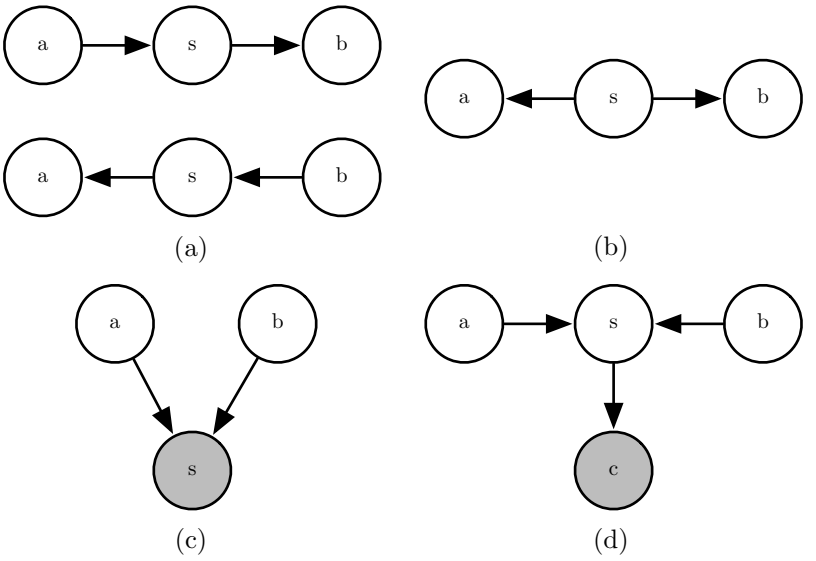

图 16.8: 两个随机变量 a，b 之间存在的长度为 2 的所有种类的活跃路径。*(a)* 箭头方向从 a 指向 b 的任何路径，反过来也一样。如果 s 可以被观察到，这种路径就是阻塞的。在接力赛的例子中，我们已经看到过这种类型的路径。*(b)* 变量 a 和 b 通过共因 s 相连。举个例子，假设 s 是一个表示是否存在飓风的变量，a 和 b 表示两个相邻气象监控区域的风速。如果我们在 a 处观察到很高的风速，我们可以期望在 *b* 处也观察到高速的风。如果观察到 s，那么这条路径就被阻塞了。如果我们已经知道存在飓风，那么无论 a 处观察到什么，我们都能期望 b 处有较高的风速。在 a 处观察到一个低于预期的风速（对飓风而言）并不会改变我们对 b 处风速的期望（已知有飓风的情况下）。然而，如果 s 不被观测到，那么 a 和 b 是依赖的，即路径是活跃的。*(c)* 变量 a 和 b 都是 s 的父节点。这称为 **V-结构**（V-structure）或者 **碰撞情况**（the collider case）。根据 **相消解释作用**（explaining away effect），V-结构导致 a 和 b 是相关的。在这种情况下，当 s 被观测到时路径是活跃的。举个例子，假设 s 是一个表示你的同事不在工作的变量。变量 a 表示她生病了，而变量 b 表示她在休假。如果你观察到了她不在工作，你可以假设她很有可能是生病了或者是在度假，但是这两件事同时发生是不太可能的。如果你发现她在休假，那么这个事实足够解释她的缺席了。你可以推断她很可能没有生病。*(d)* 即使 s 的任意后代都被观察到，相消解释作用也会起作用。举个例子，假设 c 是一个表示你是否收到你同事的报告的一个变量。如果你注意到你还没有收到这个报告，这会增加你估计的她今天不在工作的概率，这反过来又会增加她今天生病或者度假的概率。阻塞 V-结构中路径的唯一方法就是共享子节点的后代一个都观察不到。

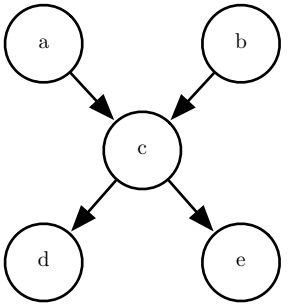

图 16.9: 从这张图中，我们可以发现一些 d-分离的性质。这包括了：

- 给定空集的情况下，a 和 b 是 d-分离的。
- 给定 c 的情况下，a 和 e 是 d-分离的。
- 给定 c 的情况下，d 和 e 是 d-分离的。

我们还可以发现当我们观察到一些变量时，一些变量不再是 d-分离的：

- 给定 c 的情况下，a 和 b 不是 d-分离的。
- 给定 d 的情况下，a 和 b 不是 d-分离的。

16.2.6 在有向模型和无向模型中转换

我们经常将特定的机器学习模型称为无向模型或有向模型。例如，我们通常将受限玻尔兹曼机称为无向模型，而稀疏编码则被称为有向模型。这种措辞的选择可能有点误导，因为没有概率模型本质上是有向或无向的。但是，一些模型很适合使用有向图描述，而另一些模型很适合使用无向模型描述。

有向模型和无向模型都有其优点和缺点。这两种方法都不是明显优越和普遍优选的。相反，我们根据具体的每个任务来决定使用哪一种模型。这个选择部分取决于我们希望描述的概率分布。根据哪种方法可以最大程度地捕捉到概率分布中的独立性，或者哪种方法使用最少的边来描述分布，我们可以决定使用有向建模还是无向建模。还有其他因素可以影响我们决定使用哪种建模方式。即使在使用单个概率分布时，我们有时也可以在不同的建模方式之间切换。有时，如果我们观察到变量的某个子集，或者如果我们希望执行不同的计算任务，换一种建模方式可能更合适。例如，有向模型通常提供了一种高效地从模型中抽取样本（在第 16.3 节中描述）的直接方法。而无向模型形式通常对于推导近似推断过程（我们将在第十九章中看到，式 (19.56) 强调了无向模型的作用）是很有用的。

每个概率分布可以由有向模型或由无向模型表示。在最坏的情况下，我们可以使用 “完全图’’ 来表示任何分布。在有向模型的情况下，完全图是任意有向无环图，其中我们对随机变量排序，并且每个变量在排序中位于其之前的所有其他变量作为其图中的祖先。对于无向模型，完全图只是包含所有变量的单个团。图 16.10 给出了一个实例。

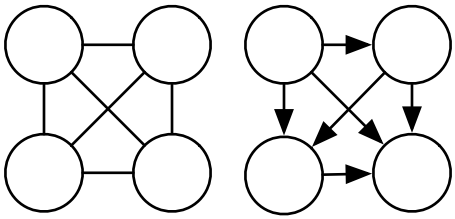

图 16.10: 完全图的例子，完全图能够描述任何的概率分布。这里我们展示了一个带有四个随机变量的例子。*(左)* 完全无向图。在无向图中，完全图是唯一的。*(右)* 一个完全有向图。在有向图中，并不存在唯一的完全图。我们选择一种变量的排序，然后对每一个变量，从它本身开始，向每一个指向顺序在其后面的变量画一条弧。因此存在着关于变量数阶乘数量级的不同种完全图。在这个例子中，我们从左到右从上到下地排序变量。

当然，图模型的优势在于图能够包含一些变量不直接相互作用的信息。完全图并不是很有用，因为它并不隐含任何独立性。

当我们用图表示概率分布时，我们想要选择一个包含尽可能多独立性的图，但是并不会假设任何实际上不存在的独立性。

从这个角度来看，一些分布可以使用有向模型更高效地表示，而其他分布可以使用无向模型更高效地表示。换句话说，有向模型可以编码一些无向模型所不能编码的独立性，反之亦然。

有向模型能够使用一种无向模型无法完美表示的特定类型的子结构。这个子结构被称为 **不道德**（immorality）。这种结构出现在当两个随机变量 a 和 b 都是第三个随机变量 c 的父结点，并且不存在任一方向上直接连接 a 和 b 的边时。（“不道德’’ 的名字可能看起来很奇怪; 它在图模型文献中使用源于一个关于未婚父母的笑话。）为了将有向模型图 $\mathcal{D}$ 转换为无向模型，我们需要创建一个新图 $\mathcal{U}$。对于每对变量 x 和 y，如果存在连接 $\mathcal{D}$ 中的 x 和 y 的有向边（在任一方向上），或者如果 x 和 y 都是图 $\mathcal{D}$ 中另一个变量 z 的父节点，则在 $\mathcal{U}$ 中添加连接 x 和 y 的无向边。得到的图 $\mathcal{U}$ 被称为是 **道德图**（moralized graph）。关于一个通过道德化将有向图模型转化为无向模型的例子可以参考图 16.11 。

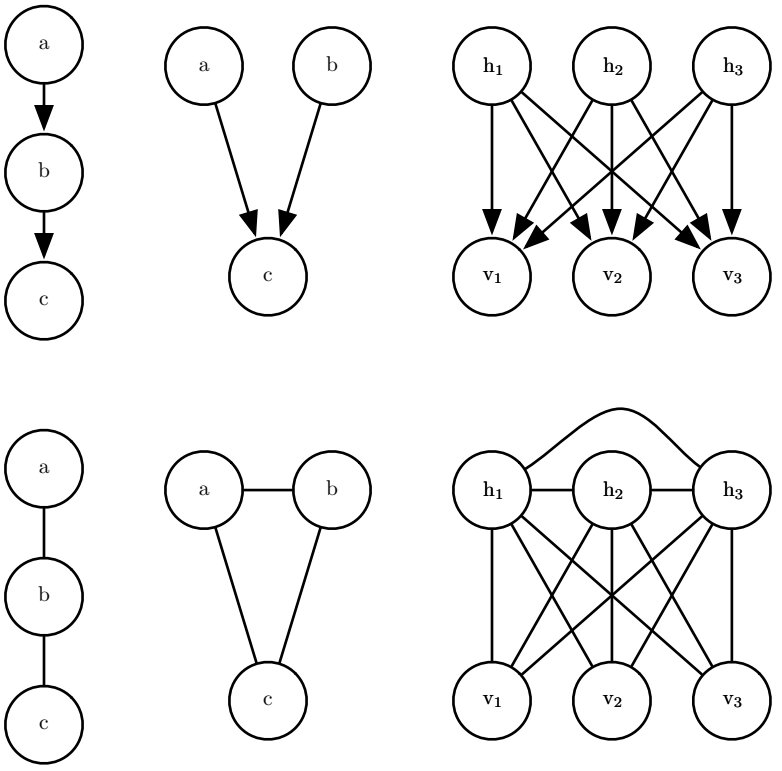

图 16.11: 通过构造道德图将有向模型（上一行）转化为无向模型（下一行）的例子。*(左)* 只需要把有向边替换成无向边就可以把这个简单的链转化为一个道德图。得到的无向模型包含了完全相同的独立关系和条件独立关系。*(中)* 这个图是在不丢失独立性的情况下是无法转化为无向模型的最简单的有向模型。这个图包含了单个完整的不道德结构。因为 a 和 b 都是 c 的父节点，当 c 被观察到时，它们之间通过活跃路径相连。为了捕捉这个依赖，无向模型必须包含一个含有所有三个变量的团。这个团无法编码 $a \perp b$ 这个信息。*(右)* 一般来说，道德化的过程会给图添加许多边，因此丢失了一些隐含的独立性。举个例子，这个稀疏编码图需要在每一对隐藏单元之间添加道德化的边，因此也引入了二次数量级的新的直接依赖。

同样的，无向模型可以包括有向模型不能完美表示的子结构。具体来说，如果 $\mathcal{U}$ 包含长度大于 3 的 **环**（loop），则有向图 $\mathcal{D}$ 不能捕获无向模型 $\mathcal{U}$ 所包含的所有条件独立性，除非该环还包含 **弦**（chord）。环指的是由无向边连接的变量序列，并且满足序列中的最后一个变量连接回序列中的第一个变量。弦是定义环序列中任意两个非连续变量之间的连接。如果 $\mathcal{U}$ 具有长度为 4 或更大的环，并且这些环没有弦，我们必须在将它们转换为有向模型之前添加弦。添加这些弦会丢弃在 $\mathcal{U}$ 中编码的一些

独立信息。通过将弦添加到 $\mathcal{U}$ 形成的图被称为 **弦图**（chordal graph）或者 **三角形化图**（triangulated graph），因为我们现在可以用更小的、三角的环来描述所有的环。要从弦图构建有向图 $\mathcal{D}$，我们还需要为边指定方向。当这样做时，我们不能在 $\mathcal{D}$ 中创建有向循环，否则将无法定义有效的有向概率模型。为 $\mathcal{D}$ 中的边分配方向的一种方法是对随机变量排序，然后将每个边从排序较早的节点指向排序稍后的节点。一个简单的实例可以参考图 16.12。

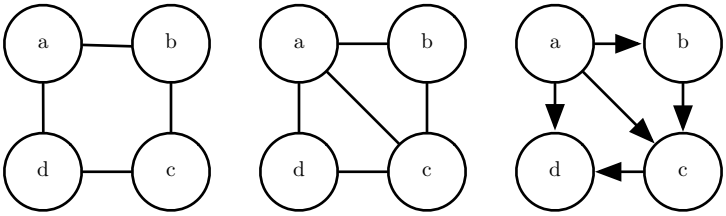

图 16.12: 将一个无向模型转化为一个有向模型。*(左)* 这个无向模型无法转化为有向模型，因为它有一个长度为 4 且不带有弦的环。具体说来，这个无向模型包含了两种不同的独立性，并且不存在一个有向模型可以同时描述这两种性质：$\mathrm{a} \perp \mathrm{c} \mid \{\mathrm{b}, \mathrm{d}\}$ 和 $\mathrm{b} \perp \mathrm{d} \mid \{\mathrm{a}, \mathrm{c}\}$。*(中)* 为了将无向图转化为有向图，我们必须通过保证所有长度大于 3 的环都有弦来三角形化图。为了实现这个目标，我们可以加一条连接 a 和 c 或者连接 b 和 d 的边。在这个例子中，我们选择添加一条连接 a 和 c 的边。*(右)* 为了完成转化的过程，我们必须给每条边分配一个方向。执行这个任务时，我们必须保证不产生任何有向环。避免出现有向环的一种方法是赋予节点一定的顺序，然后将每个边从排序较早的节点指向排序稍后的节点。在这个例子中，我们根据变量名的字母进行排序。

16.2.7 因子图

因子图（factor graph）是从无向模型中抽样的另一种方法，它可以解决标准无向模型语法中图表达的模糊性。在无向模型中，每个 ϕ 函数的范围必须是图中某个团的子集。我们无法确定每一个团是否含有一个作用域包含整个团的因子——比如说一个包含三个结点的团可能对应的是一个有三个结点的因子，也可能对应的是三个因子并且每个因子包含了一对结点，这通常会导致模糊性。通过显式地表示每一个 ϕ 函数的作用域，因子图解决了这种模糊性。具体来说，因子图是一个包含无向二分图的无向模型的图形化表示。一些节点被绘制为圆形。就像在标准无向模型中一样，这些节点对应于随机变量。其余节点绘制为方块。这些节点对应于未归一化概率函数的因子 ϕ。变量和因子可以通过无向边连接。当且仅当变量包含在未归一化概率函数的因子中时，变量和因子在图中存在连接。没有因子可以连接到图中的

另一个因子，也不能将变量连接到变量。图 16.2.7 给出了一个例子来说明因子图如何解决无向网络中的模糊性。

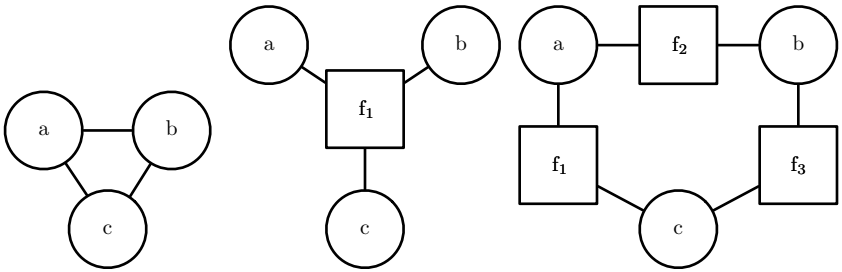

图 16.13: 因子图如何解决无向网络中的模糊性的一个例子。*(左)* 一个包含三个变量（a、b 和 c）的团组成的无向网络。*(中)* 对应这个无向模型的因子图。这个因子图有一个包含三个变量的因子。*(右)* 对应这个无向模型的另一种有效的因子图。这个因子图包含了三个因子，每个因子只对应两个变量。即使它们表示的是同一个无向模型，这个因子图上进行的表示、推断和学习相比于中图描述的因子图都要渐近地廉价。

16.3 从图模型中采样

图模型同样简化了从模型中采样的过程。

有向图模型的一个优点是，可以通过一个简单高效的过程从模型所表示的联合分布中产生样本，这个过程被称为 **原始采样**（Ancestral Sampling）。

原始采样的基本思想是将图中的变量 x_i 使用拓扑排序，使得对于所有 i 和 j，如果 x_i 是 x_j 的一个父亲结点，则 j 大于 i。然后可以按此顺序对变量进行采样。换句话说，我们可以首先采 $\mathrm{x}_1 \sim P(\mathrm{x}_1)$，然后采 $\mathrm{x}_2 \sim P(\mathrm{x}_2 \mid Pa_{\mathcal{G}}(\mathrm{x}_2))$，以此类推，直到最后我们从 $P(\mathrm{x}_n \mid Pa_{\mathcal{G}}(\mathrm{x}_n))$ 中采样。只要不难从每个条件分布 $\mathrm{x}_i \sim P(\mathrm{x}_i \mid Pa_{\mathcal{G}}(\mathrm{x}_i))$ 中采样，那么从整个模型中采样也是容易的。拓扑排序操作保证我们可以按照式 (16.1) 中条件分布的顺序依次采样。如果没有拓扑排序，我们可能会在其父节点可用之前试图对该变量进行抽样。

有些图可能存在多个拓扑排序。原始采样可以使用这些拓扑排序中的任何一个。

原始采样通常非常快（假设从每个条件分布中采样都是很容易的）并且非常简便。

原始采样的一个缺点是其仅适用于有向图模型。另一个缺点是它并不是每次采

样都是条件采样操作。当我们希望从有向图模型中变量的子集中采样时，给定一些其他变量，我们经常要求所有给定的条件变量在顺序图中比要采样的变量的顺序要早。在这种情况下，我们可以从模型分布指定的局部条件概率分布中采样。否则，我们需要采样的条件分布是给定观测变量的后验分布。这些后验分布在模型中通常没有明确指定和参数化。推断这些后验分布的代价可能是很高的。在这种情况下的模型中，原始采样不再有效。

不幸的是，原始采样仅适用于有向模型。我们可以通过将无向模型转换为有向模型来实现从无向模型中抽样，但是这通常需要解决棘手的推断问题（要确定新有向图的根节点上的边缘分布），或者需要引入许多边从而会使得到的有向模型变得难以处理。从无向模型采样，而不首先将其转换为有向模型的做法似乎需要解决循环依赖的问题。每个变量与每个其他变量相互作用，因此对于采样过程没有明确的起点。不幸的是，从无向模型中抽取样本是一个成本很高的多次迭代的过程。理论上最简单的方法是**Gibbs 采样**（Gibbs Sampling）。假设我们在一个 n 维向量的随机变量 $\mathbf{x}$ 上有一个图模型。我们迭代地访问每个变量 x_i，在给定其他变量的条件下从 $p(\mathrm{x}_i \mid \mathrm{x}_{-i})$ 中抽样。由于图模型的分离性质，抽取 x_i 时我们可以等价地仅对 x_i 的邻居条件化。不幸的是，在我们遍历图模型一次并采样所有 n 个变量之后，我们仍然无法得到一个来自 $p(\mathbf{x})$ 的客观样本。相反，我们必须重复该过程并使用它们邻居的更新值对所有 n 个变量重新取样。在多次重复之后，该过程渐近地收敛到正确的目标分布。我们很难确定样本何时达到所期望分布的足够精确的近似。无向模型的采样技术是一个高级的研究方向，第十七章将对此进行更详细的讨论。

16.4 结构化建模的优势

使用结构化概率模型的主要优点是它们能够显著降低表示概率分布、学习和推断的成本。有向模型中采样还可以被加速，但是对于无向模型情况则较为复杂。选择不对某些变量的相互作用进行建模是允许所有这些操作使用较少的运行时间和内存的主要机制。图模型通过省略某些边来传达信息。在没有边的情况下，模型假设不对变量间直接的相互作用建模。

结构化概率模型允许我们明确地将给定的现有知识与知识的学习或者推断分开，这是一个不容易量化的益处。这使我们的模型更容易开发和调试。我们可以设计、分析和评估适用于更广范围的图的学习算法和推断算法。同时，我们可以设计能够捕捉到我们认为数据中存在的重要关系的模型。然后，我们可以组合这些不同的算

法和结构，并获得不同可能性的笛卡尔乘积。然而，为每种可能的情况设计端到端的算法会更加困难。

16.5 学习依赖关系

良好的生成模型需要准确地捕获所观察到的或“可见”变量 $\mathbf{v}$ 上的分布。通常 $\mathbf{v}$ 的不同元素彼此高度依赖。在深度学习中，最常用于建模这些依赖关系的方法是引入几个潜在或“隐藏”变量 $\mathbf{h}$。然后，该模型可以捕获任何对（变量 v_i 和 v_j 间接依赖可以通过 v_i 和 $\mathbf{h}$ 之间直接依赖和 $\mathbf{h}$ 和 v_j 直接依赖捕获）之间的依赖关系。

如果一个良好的关于 $\mathbf{v}$ 的模型不包含任何潜变量，那么它在贝叶斯网络中的每个节点需要具有大量父节点或在马尔可夫网络中具有非常大的团。仅仅表示这些高阶相互作用的成本就很高了，首先从计算角度上考虑，存储在存储器中的参数数量是团中成员数量的指数级别，接着在统计学意义上，因为这些指数数量的参数需要大量的数据来准确估计。

当模型旨在描述直接连接的可见变量之间的依赖关系时，通常不可能连接所有变量，因此设计图模型时需要连接那些紧密相关的变量，并忽略其他变量之间的作用。机器学习中有一个称为**结构学习**（structure learning）的领域专门讨论这个问题。Koller and Friedman (2009) 是一个不错的结构学习参考资料。大多数结构学习技术基于一种贪婪搜索的形式。它们提出了一种结构，对具有该结构的模型进行训练，然后给出分数。该分数奖励训练集上的高精度并对模型的复杂度进行惩罚。然后提出添加或移除少量边的候选结构作为搜索的下一步。搜索向一个预计会增加分数的新结构发展。

使用潜变量而不是自适应结构避免了离散搜索和多轮训练的需要。可见变量和潜变量之间的固定结构可以使用可见单元和隐藏单元之间的直接作用，从而建模可见单元之间的间接作用。使用简单的参数学习技术，我们可以学习到一个具有固定结构的模型，这个模型在边缘分布 $p(\boldsymbol{v})$ 上拥有正确的结构。

潜变量除了发挥本来的作用，即能够高效地描述 $p(\mathbf{v})$ 以外，还具有另外的优势。新变量 $\mathbf{h}$ 还提供了 $\mathbf{v}$ 的替代表示。例如，如第 3.9.6 节所示，高斯混合模型学习了一个潜变量，这个潜变量对应于输入样本是从哪一个混合体中抽出。这意味着高斯混合模型中的潜变量可以用于做分类。我们可以看到第十四章中简单的概率模型如稀疏编码，是如何学习可以用作分类器输入特征或者作为流形上坐标的潜变量的。

其他模型也可以使用相同的方式，但是更深的模型和具有多种相互作用方式的模型可以获得更丰富的输入描述。许多方法通过学习潜变量来完成特征学习。通常，给定 $\mathbf{v}$ 和 $\mathbf{h}$，实验观察显示 $\mathbb{E}[\mathbf{h} \mid \mathbf{v}]$ 或 $\arg\max_{\boldsymbol{h}} p(\boldsymbol{h}, \boldsymbol{v})$ 都是 $\boldsymbol{v}$ 的良好特征映射。

16.6 推断和近似推断

解决变量之间如何相互关联的问题是我们使用概率模型的一个主要方式。给定一组医学测试，我们可以询问患者可能患有什么疾病。在一个潜变量模型中，我们可能需要提取能够描述可观察变量 $\mathbf{v}$ 的特征 $\mathbb{E}[\mathbf{h} \mid \mathbf{v}]$。有时我们需要解决这些问题来执行其他任务。我们经常使用最大似然的准则来训练我们的模型。由于

$$\log p(\boldsymbol{v}) = \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h} \mid \boldsymbol{v})}[\log p(\boldsymbol{h}, \boldsymbol{v}) - \log p(\boldsymbol{h} \mid \boldsymbol{v})], \tag{16.9}$$

学习过程中，我们经常需要计算 $p(\mathbf{h} \mid \boldsymbol{v})$。所有这些都是 **推断**（inference）问题的例子，其中我们必须预测给定其他变量的情况下一些变量的值，或者在给定其他变量值的情况下预测一些变量的概率分布。

不幸的是，对于大多数有趣的深度模型来说，即使我们使用结构化图模型来简化这些推断问题，它们仍然是难以处理的。图结构允许我们用合理数量的参数来表示复杂的高维分布，但是用于深度学习的图并不满足这样的条件，从而难以实现高效地推断。

我们可以直接看出，计算一般图模型的边缘概率是 #P-hard 的。复杂性类别 #P 是复杂性类别 NP 的泛化。NP 中的问题只需确定其中一个问题是否有解决方案，并找到一个解决方案（如果存在）就可以解决。#P 中的问题需要计算解决方案的数量。为了构建最坏情况的图模型，我们可以设想一下我们在 3-SAT 问题中定义二值变量的图模型。我们可以对这些变量施加均匀分布。然后我们可以为每个子句添加一个二值潜变量，来表示每个子句是否成立。然后，我们可以添加另一个潜变量，来表示所有子句是否成立。这可以通过构造一个潜变量的缩减树来完成，树中的每个结点表示其他两个变量是否成立，从而不需要构造一个大的团。该树的叶是每个子句的变量。树的根表示整个问题是否成立。由于子句的均匀分布，缩减树根结点的边缘分布表示子句有多少比例是成立的。虽然这是一个设计的最坏情况的例子，NP-hard 图确实会频繁地出现在现实世界的场景中。

这促使我们使用近似推断。在深度学习中，这通常涉及变分推断，其中通过寻求尽可能接近真实分布的近似分布 $q(\mathbf{h} \mid \mathbf{v})$ 来逼近真实分布 $p(\mathbf{h} \mid \boldsymbol{v})$。这个技术将在

第十九章中深入讨论。

16.7 结构化概率模型的深度学习方法

深度学习从业者通常与其他从事结构化概率模型研究的机器学习研究者使用相同的基本计算工具。然而，在深度学习中，我们通常对如何组合这些工具作出不同的设计决定，导致总体算法、模型与更传统的图模型具有非常不同的风格。

深度学习并不总是涉及特别深的图模型。在图模型中，我们可以根据图模型的图而不是计算图来定义模型的深度。如果从潜变量 h_i 到可观察变量的最短路径是 j 步，我们可以认为潜变量 h_j 处于深度 j。我们通常将模型的深度描述为任何这样的 h_j 的最大深度。这种深度不同于由计算图定义的深度。用于深度学习的许多生成模型没有潜变量或只有一层潜变量，但使用深度计算图来定义模型中的条件分布。

深度学习基本上总是利用分布式表示的思想。即使是用于深度学习目的的浅层模型（例如预训练浅层模型，稍后将形成深层模型），也几乎总是具有单个大的潜变量层。深度学习模型通常具有比可观察变量更多的潜变量。变量之间复杂的非线性相互作用通过多个潜变量的间接连接来实现。

相比之下，传统的图模型通常包含至少是偶尔观察到的变量，即使一些训练样本中的许多变量随机地丢失。传统模型大多使用高阶项和结构学习来捕获变量之间复杂的非线性相互作用。如果有潜变量，它们的数量通常很少。

潜变量的设计方式在深度学习中也有所不同。深度学习从业者通常不希望潜变量提前包含了任何特定的含义——训练算法可以自由地开发对特定数据集建模所需要的概念。在事后解释潜变量通常是很困难的，但是可视化技术可以得到它们表示的一些粗略表征。当潜变量在传统图模型中使用时，它们通常被赋予一些特定含义——比如文档的主题、学生的智力、导致患者症状的疾病等。这些模型通常由研究者解释，并且通常具有更多的理论保证，但是不能扩展到复杂的问题，并且不能像深度模型一样在许多不同背景中重复使用。

另一个明显的区别是深度学习方法中经常使用的连接类型。深度图模型通常具有大的与其他单元组全连接的单元组，使得两个组之间的相互作用可以由单个矩阵描述。传统的图模型具有非常少的连接，并且每个变量的连接选择可以单独设计。模型结构的设计与推断算法的选择紧密相关。图模型的传统方法通常旨在保持精确推断的可解性。当这个约束太强时，我们可以采用一种流行的被称为 **环状信念传播**

（loopy belief propagation）的近似推断算法。这两种方法通常在稀疏连接图上都有很好的效果。相比之下，在深度学习中使用的模型倾向于将每个可见单元 v_i 连接到非常多的隐藏单元 h_j 上，从而使得 $\mathbf{h}$ 可以获得一个 v_i 的分布式表示（也可能是其他几个可观察变量）。分布式表示具有许多优点，但是从图模型和计算复杂性的观点来看，分布式表示有一个缺点就是很难产生对于精确推断和环状信念传播等传统技术来说足够稀疏的图。结果，大规模图模型和深度图模型最大的区别之一就是深度学习中几乎从来不会使用环状信念传播。相反的，许多深度学习模型可以设计来加速 Gibbs 采样或者变分推断。此外，深度学习模型包含了大量的潜变量，使得高效的数值计算代码显得格外重要。除了选择高级推断算法之外，这提供了另外的动机，用于将结点分组成层，相邻两层之间用一个矩阵来描述相互作用。这要求实现算法的单个步骤可以实现高效的矩阵乘积运算，或者专门适用于稀疏连接的操作，例如块对角矩阵乘积或卷积。

最后，图模型的深度学习方法的一个主要特征在于对未知量的较高容忍度。与简化模型直到它的每一个量都可以被精确计算不同的是，我们仅仅直接使用数据运行或者是训练，以增强模型的能力。我们一般使用边缘分布不能计算的模型，但可以从中简单地采近似样本。我们经常训练具有难以处理的目标函数的模型，我们甚至不能在合理的时间内近似，但是如果我们能够高效地获得这样一个函数的梯度估计，我们仍然能够近似训练模型。深度学习方法通常是找出我们绝对需要的最小量信息，然后找出如何尽快得到该信息的合理近似。

16.7.1 实例：受限玻尔兹曼机

受限玻尔兹曼机（Restricted Boltzmann Machine, RBM）(Smolensky, 1986) 或者 **簧风琴**（harmonium）是图模型如何用于深度学习的典型例子。RBM 本身不是一个深层模型。相反，它有一层潜变量，可用于学习输入的表示。在第二十章中，我们将看到 RBM 如何被用来构建许多的深层模型。在这里，我们举例展示了 RBM 在许多深度图模型中使用的实践：它的单元被分成很大的组，这种组称作层，层之间的连接由矩阵描述，连通性相对密集。该模型被设计为能够进行高效的 Gibbs 采样，并且模型设计的重点在于以很高的自由度来学习潜变量，而潜变量的含义并不是设计者指定的。之后在第 20.2 节，我们将更详细地再次讨论 RBM。

标准的 RBM 是具有二值的可见和隐藏单元的基于能量的模型。其能量函数为

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h} - \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h}, \tag{16.10}$$

其中 $\boldsymbol{b}, \boldsymbol{c}$ 和 $\boldsymbol{W}$ 都是无约束、实值的可学习参数。我们可以看到，模型被分成两组单元：$\boldsymbol{v}$ 和 $\boldsymbol{h}$，它们之间的相互作用由矩阵 $\boldsymbol{W}$ 来描述。该模型在图 16.14 中以图的形式描绘。该图能够使我们更清楚地发现，该模型的一个重要方面是在任何两个可见单元之间或任何两个隐藏单元之间没有直接的相互作用（因此称为“受限”，一般的玻尔兹曼机可以具有任意连接）。

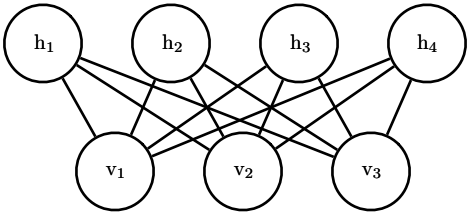

图 16.14: 一个画成马尔可夫网络形式的RBM。

对 RBM 结构的限制产生了良好的属性

$$p(\mathbf{h} \mid \mathbf{v}) = \prod_i p(\mathrm{h}_i \mid \mathbf{v}) \tag{16.11}$$

以及

$$p(\mathbf{v} \mid \mathbf{h}) = \prod_i p(\mathrm{v}_i \mid \mathbf{h}). \tag{16.12}$$

独立的条件分布很容易计算。对于二元的受限玻尔兹曼机，我们可以得到：

$$p(\mathrm{h}_i = 1 \mid \mathbf{v}) = \sigma\big(\mathbf{v}^\top \boldsymbol{W}_{:,i} + b_i\big), \tag{16.13}$$

$$p(\mathrm{h}_i = 0 \mid \mathbf{v}) = 1 - \sigma\big(\mathbf{v}^\top \boldsymbol{W}_{:,i} + b_i\big). \tag{16.14}$$

结合这些属性可以得到高效的 **块吉布斯采样**（block Gibbs Sampling），它在同时采样所有 $\boldsymbol{h}$ 和同时采样所有 $\boldsymbol{v}$ 之间交替。RBM 模型通过 Gibbs 采样产生的样本展示在图 16.15 中。

由于能量函数本身只是参数的线性函数，很容易获取能量函数的导数。例如，

$$\frac{\partial}{\partial W_{i,j}} E(\mathbf{v}, \mathbf{h}) = -\mathrm{v}_i \mathrm{h}_j. \tag{16.15}$$

这两个属性，高效的 Gibbs 采样和导数计算，使训练过程变得非常方便。在第十八章中，我们将看到，可以通过计算应用于这种来自模型样本的导数来训练无向模型。