

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

10 Directed graphical models (Bayes nets)

10.1 Introduction

I basically know of two principles for treating complicated systems in simple ways: the first is the principle of modularity and the second is the principle of abstraction. I am an apologist for computational probability in machine learning because I believe that probability theory implements these two principles in deep and intriguing ways — namely through factorization and through averaging. Exploiting these two mechanisms as fully as possible seems to me to be the way forward in machine learning. — Michael Jordan, 1997 (quoted in (Frey 1998)). Suppose we observe multiple correlated variables, such as words in a document, pixels in an image, or genes in a microarray. How can we compactly represent the joint distribution $p(x|\theta)$? How can we use this distribution to infer one set of variables given another in a reasonable amount of computation time? And how can we learn the parameters of this distribution with a reasonable amount of data? These questions are at the core of probabilistic modeling, inference and learning, and form the topic of this chapter.

10.1 Introduction

I basically know of two principles for treating complicated systems in simple ways: the first is the principle of modularity and the second is the principle of abstraction. I am an apologist for computational probability in machine learning because I believe that probability theory implements these two principles in deep and intriguing ways — namely through factorization and through averaging. Exploiting these two mechanisms as fully as possible seems to me to be the way forward in machine learning.

— Michael Jordan, 1997 (quoted in (Frey 1998)). Suppose we observe multiple correlated variables, such as words in a document, pixels in an image, or genes in a microarray. How can we compactly represent the joint distribution $p(x|\theta)$? How can we use this distribution to infer one set of variables given another in a reasonable amount of computation time? And how can we learn the parameters of this distribution with a reasonable amount of data?

These questions are at the core of probabilistic modeling, inference and learning, and form the topic of this chapter.

10.1.1 Chain rule

By the chain rule of probability, we can always represent a joint distribution as follows, using any ordering of the variables:

$$p(x_1:V) = p(x_1)p(x_2|x_1)p(x_3|x_2,x_1)p(x_4|x_1,x_2,x_3) \dots p(x_V|x_1:V-1) \quad (10.1)$$

where V is the number of variables, the Matlab-like notation $1:V$ denotes the set $\{1, 2, \dots, V\}$, and where we have dropped the conditioning on the fixed parameters θ for brevity. The problem with this expression is that it becomes more and more complicated to represent the conditional distributions $p(x_t|x_1:t-1)$ as t gets large. For example, suppose all the variables have K states. We can represent $p(x_1)$ as a table of $O(K)$ numbers, representing a discrete distribution (there are actually only $K - 1$ free parameters, due to the sum-to-one constraint, but we write $O(K)$ for simplicity). Similarly, we can represent $p(x_2|x_1)$ as a table of $O(K^2)$ numbers by writing $p(x_2 = j|x_1 = i) = T_{ij}$; we say that T is a stochastic matrix, since it satisfies the constraint $\sum_j T_{ij} = 1$ for all rows i , and $0 \leq T_{ij} \leq 1$ for all entries. Similarly, we can represent $p(x_3|x_1,x_2)$ as a 3d table with $O(K^3)$ numbers. These are called conditional probability tables or CPTs. We see that there are $O(KV)$ parameters in the model. We would need an awful lot of data to learn so many parameters. One solution is to replace each CPT with a more parsimonious conditional probability distribution or CPD, such as multinomial logistic regression, i.e., $p(x_t = k|x_1:t-1) = S(Wx_1:t-1)k$. The total number of parameters is now only $O(K^2V^2)$, making this a compact density model (Neal 1992; Frey 1998). This is adequate if all we want to do is evaluate the probability of a fully observed vector $x_1:T$. For example, we can use this model to define a class-conditional density, $p(x|y = c)$, thus making a generative classifier (Bengio and Bengio 2000). However, this model is not useful for other kinds of prediction tasks, since each variable depends on all the previous variables. So we need another approach.

10.1.2 Conditional independence

The key to efficiently representing large joint distributions is to make some assumptions about conditional independence (CI). Recall from Section 2.2.4 that X and Y

are conditionally independent given Z , denoted $X \perp Y|Z$, if and only if (iff) the conditional joint can be written as a product of conditional marginals, i.e.,

$$X \perp Y|Z \iff p(X, Y|Z) = p(X|Z)p(Y|Z) \quad (10.2)$$

Let us see why this might help. Suppose we assume that $x_{t+1} \perp x_{1:t-1}|x_t$, or in words, "the future is independent of the past given the present". This is called the (first order) Markov assumption. Using this assumption, plus the chain rule, we can write the joint distribution as follows:

$$p(x_1:V) = p(x_1) \vee t=1 p(x_t|x_{t-1}) \quad (10.3)$$

This is called a (first-order) Markov chain. They can be characterized by an initial distribution over states, $p(x_1 = i)$, plus a state transition matrix $p(x_t = j|x_{t-1} = i)$. See Section 17.2 for more information.

已思考 30s

好的！我先把你给的英文原文逐句、无遗漏地翻译成中文（保留式样与公式编号），然后用通俗的方式详细解释，再给一个小例子帮你直观理解，最后用要点总结。注意：你给的“10.1 Introduction”段落在原文中出现了两次，我会分别翻译两遍以确保没有遗漏。

翻译

10 有向图模型（贝叶斯网）

10.1 引言

我基本上知道两条以简单方式处理复杂系统的原则：第一是**模块化**原则，第二是**抽象**原则。我为机器学习中的计算概率方法辩护，是因为我相信概率论以深刻而耐人寻味的方式实现了这两条原则——即通过**因子分解** (factorization) 和**平均** (averaging)。在我看来，尽可能充分地利用这两种机制，是机器学习向前发展的道路。——Michael Jordan, 1997 (引自 (Frey 1998))。

假设我们观测到多个相关的变量，比如文档中的单词、图像中的像素或芯片数据中的基因。我们如何紧凑地表示联合分布 $p(\mathbf{x} | \theta)$ ？我们如何在**合理的计算时间**内，用这个分布去推断在给定一部分变量的情况下另一部分变量的值？以及我们如何用**合理数量的数据**来学习这个分布的参数？这些问题**是概率建模、推断与学习**的核心，也是本章要讨论的主题。

(以下段落为原文重复的“10.1 引言”，按要求再次翻译)

10.1 引言

我基本上知道两条以简单方式处理复杂系统的原则：第一是模块化原则，第二是抽象原则。我为机器学习中的计算概率方法辩护，是因为我相信概率论以深刻而耐人寻味的方式实现了这两条原则——即通过因子分解和平均。在我看来，尽可能充分地利用这两种机制，是机器学习向前发展的道路。——Michael Jordan, 1997 (引自 (Frey 1998))。

假设我们观测到多个相关的变量，比如文档中的单词、图像中的像素或芯片数据中的基因。我们如何紧凑地表示联合分布 $p(\mathbf{x} \mid \theta)$ ？我们如何在合理的计算时间内，用这个分布去推断在给定一部分变量的情况下另一部分变量的值？以及我们如何用合理数量的数据来学习这个分布的参数？这些问题时概率建模、推断与学习的核心，也是本章要讨论的主题。

10.1.1 链式法则

根据概率的链式法则，我们总是可以用**任意的变量次序**把一个联合分布写成下面的形式：

$$p(x_{1:V}) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2, x_1)p(x_4 \mid x_1, x_2, x_3) \dots p(x_V \mid x_{1:V-1}) \quad (10.1)$$

其中 V 是变量的个数，Matlab 式记号 $1:V$ 表示集合 $\{1, 2, \dots, V\}$ ，并且为简洁起见，我们省略了对固定参数 θ 的条件。

这个表达式的问题在于：当 t 变大时，表示条件分布 $p(x_t \mid x_{1:t-1})$ 会变得越来越复杂。举例来说，假设所有变量都有 K 个取值。我们可以把 $p(x_1)$ 表示成一个包含 $O(K)$ 个数的表，代表一个离散分布（实际上只有 $K - 1$ 个自由参数，因为要满足“求和为 1”的约束，但为简单起见我们记为 $O(K)$ ）。类似地，我们可以把 $p(x_2 \mid x_1)$ 表示成一个包含 $O(K^2)$ 个数的表，写作 $p(x_2 = j \mid x_1 = i) = T_{ij}$ ；我们称 T 为一个**随机矩阵 (stochastic matrix)**，因为它满足对所有行 i 都有 $\sum_j T_{ij} = 1$ ，且所有元素满足 $0 \leq T_{ij} \leq 1$ 。同理，我们可以把 $p(x_3 \mid x_1, x_2)$ 表示成一个包含 $O(K^3)$ 个数的三维表。这些表被称为**条件概率表 (CPTs)**。我们看到，该模型里有 $O(K^V)$ 个参数。要学习这么多参数，我们需要海量数据。

一种解决办法是用更节省参数的**条件概率分布 (CPD)** 来替代每个 CPT，比如使用多项逻辑回归 (multinomial logistic regression)，即

$$p(x_t = k \mid x_{1:t-1}) = S(W_t x_{1:t-1})_k.$$

这时总参数数目只有 $O(K^2 V^2)$ ，使其成为一个**紧凑的密度模型** (Neal 1992; Frey 1998)。如果我们只想对一个**完全观测**的向量 $x_{1:T}$ 计算概率，这已经足够了。例如，我们可用这个模型定义一个**类条件密度** $p(x \mid y = c)$ ，从而得到一个**生成式分类器** (Bengio

and Bengio 2000)。然而，对于其他类型的预测任务，这个模型并不实用，因为**每个变量都依赖于所有先前的变量**。因此我们需要另一种方法。

10.1.2 条件独立

高效表示大规模联合分布的关键在于做出一些**条件独立 (CI)** 的假设。回忆第 2.2.4 节：若给定 Z 时 X 与 Y 条件独立，记作 $X \perp Y \mid Z$ ，当且仅当 (iff) 条件联合分布可以写成条件边缘分布的乘积，即

$$X \perp Y \mid Z \Leftrightarrow p(X, Y \mid Z) = p(X \mid Z)p(Y \mid Z). \quad (10.2)$$

看看这如何有帮助。假设我们断言 $x_{t+1} \perp x_{1:t-1} \mid x_t$ ，或者用语言说，“**在给定现在的情况下，未来与过去独立**”。这叫做 (一阶) **马尔可夫假设**。利用这个假设加上链式法则，我们可以把联合分布写为

$$p(x_{1:V}) = p(x_1) \prod_{t=2}^V p(x_t \mid x_{t-1}). \quad (10.3)$$

这称为 (一阶) **马尔可夫链**。它可以由对状态的初始分布 $p(x_1 = i)$ 以及**状态转移矩阵** $p(x_t = j \mid x_{t-1} = i)$ 来刻画。更多内容见第 17.2 节。

通俗详解

1) 这一章在讲什么？

- 我们面对很多**相关联**的变量（如句子里的词、图像里的像素）。要做三件事：
 - ① 用尽量少的参数表示它们的**联合分布**；
 - ② 能够**高效推断**（已知一部分变量，算另一部分的概率/取值）；
 - ③ 能够用**不太的数据**把模型参数学出来。
- 概率图模型（尤其是**有向图模型 / 贝叶斯网**）的思想：用**因子分解**（把大概率拆成很多小概率的乘积）和**条件独立**（恰当“断开”不必要的依赖）来做到“又准又省”。

2) 链式法则：任何联合分布都能“拆乘”

- 链式法则说：无论变量怎么排序，都能把 $p(x_1, \dots, x_V)$ 写成一串条件概率的乘积。
直观比喻：做菜流程——先决定第一步（油多少），再决定第二步（放不放蒜，取决于第一步），再第三步（番茄放多少，取决于前两步）……按顺序一点点“条件化”。
- **问题：**如果“第 t 步”要依赖“前 $t - 1$ 步的所有细节”，那要存的表会越来越大，参数爆炸。

3) 为什么参数会爆炸?

- 如果每个变量只有 K 种取值:
 - $p(x_1)$ 是长度为 K 的表;
 - $p(x_2 | x_1)$ 是 $K \times K$ 的表;
 - $p(x_3 | x_1, x_2)$ 是 $K \times K \times K$ 的三维表;
 - ...一直到 $p(x_V | x_{1:V-1})$ 需要 K^V 个数。
- 总的参数量级被最后一项“主导”，是指数级 $O(K^V)$ 。这意味着：变量一多、取值一多，所需数据和计算都会爆炸。

4) 省参数的一种招：用“函数”代替“大表”

- 思路：别把每个条件概率都存成巨大表 (CPT)，而是让它由一个带参数的函数 (CPD) 算出来，比如多项逻辑回归 (softmax 回归)：

$$p(x_t = k | x_{1:t-1}) = \text{softmax}(W_t \phi(x_{1:t-1}))_k,$$

其中 ϕ 是把“历史”变成特征向量的编码。

- 这样参数量可以变成多项式级别 (文中给的数量级是 $O(K^2 V^2)$)，比指数级小太多了。
- 局限：若每个 x_t 仍然“看见所有过去变量”，很多预测任务（要灵活地只用一小部分信息）仍不够方便。

5) 真正好使的招：条件独立 (CI)

- 条件独立：给定 Z 时， X 与 Y 互不影响。数学上等价于

$$p(X, Y | Z) = p(X | Z)p(Y | Z).$$

- 这能把“大概率”拆成几个“小概率”，大幅减少参数与计算。

6) 最经典的 CI 假设：马尔可夫性

- 马尔可夫假设：在知道现在的情况下，未来与过去无关。
写作： $x_{t+1} \perp x_{1:t-1} | x_t$ 。
- 套上链式法则，联合分布就变成马尔可夫链：

$$p(x_{1:V}) = p(x_1) \prod_{t=2}^V p(x_t | x_{t-1}).$$

- 参数量骤减：

- 一个初始分布 $p(x_1)$ 需要 K 个数；
- 一个转移矩阵 $K \times K$ (每一行加起来是 1)。

相比完全依赖所有历史的情形，简直是“从天文数字降到可管理”。

小例子 (超直观)

设 $V = 3$ 个变量、每个变量只有 $K = 2$ 个取值 (比如 0/1) :

1) 不加任何独立性 (全历史依赖)

- $p(x_1)$: 需要 2 个数 (实际上 1 个自由度)。
- $p(x_2 | x_1)$: 需要 $2 \times 2 = 4$ 个数 (两行, 每行和为 1)。
- $p(x_3 | x_1, x_2)$: 需要 $2 \times 2 \times 2 = 8$ 个数。
- 总计要存 $2 + 4 + 8 = 14$ 个数 (量级被 8 主导, 对一般 V 来说是 $O(2^V)$)。

2) 加一阶马尔可夫 (只看紧邻的过去)

- $p(x_1)$: 2 个数。
- 转移矩阵 $p(x_t | x_{t-1})$: $2 \times 2 = 4$ 个数 (同一个矩阵复用给 $t = 2, 3$)。
- 总计只需存 $2 + 4 = 6$ 个数。
- 对更大 V , 代价几乎不再随 V 爆炸增长——这就是“条件独立”的威力。

常见易错点 / 勘误提示

1. 参数量的量级

文中先展示了 $p(x_t | x_{1:t-1})$ 的表随 t 呈指数增长, 结论应为**总参数量级** $O(K^V)$ (由最后一项主导)。如果你在别的版本里见到 $O(KV)$, 那多半是印刷或排版错误。

2. 马尔可夫链的乘积下标

正确写法应是 $\prod_{t=2}^V p(x_t | x_{t-1})$ (从第 2 个变量开始条件化前一个), 有时材料里会把下标打错成从 1 开始。

要点总结

- **核心目标**: 用少参数、可学习、可高效推断的方式表示多变量的**联合分布**。
- **链式法则**: 任何联合分布都能拆成条件概率的乘积, 但**直接存表**会导致参数呈**指数级增长** ($O(K^V)$)。
- **CPT vs. CPD**: 把条件概率表 (CPT) 换成带参数的函数形式 (CPD, 如 softmax 回归) 可显著**降维**到多项式级别。
- **条件独立 (CI) 是关键**: 合适的 CI 假设能把“大问题”分解成“小块”, 既体现模块化, 又体现**抽象/平均**。
- **马尔可夫假设** (未来与过去在“给定现在”下独立) → **马尔可夫链**:

V

$$p(x_{1:V}) = p(x_1) \prod_{t=2}^V p(x_t | x_{t-1}),$$

只需初始分布 + 转移矩阵，参数量大幅下降。

- **应用直觉：**当“远处的历史细节”对“当前的下一步”影响不大时，用“一阶依赖”近似通常就很有效，这正是图模型（尤其是贝叶斯网与马尔可夫模型）在工程上好用的原因。

如果你愿意，我可以把这里的公式变成更直观的图（小图示：节点和箭头）并配上几道练习题，帮助进一步消化。

在马尔科夫假设下，公式10.1怎么推导到公式10.3的

已思考 16s

下面把“在一阶马尔可夫假设下，如何由公式(10.1)推到公式(10.3)”分三步讲清楚，并给一个小例子核对一下下标。

公式回顾

- **链式法则 (10.1)** (对任意变量次序都成立)

$$p(x_{1:V}) = p(x_1)p(x_2 | x_1)p(x_3 | x_{1:2}) \cdots p(x_V | x_{1:V-1}) = p(x_1) \prod_{t=2}^V p(x_t | x_{1:t-1}).$$

- **一阶马尔可夫假设**

$$x_t \perp x_{1:t-2} \mid x_{t-1} \iff p(x_t | x_{1:t-1}) = p(x_t | x_{t-1}) \quad (t \geq 2).$$

- **目标 (10.3)**

$$p(x_{1:V}) = p(x_1) \prod_{t=2}^V p(x_t | x_{t-1}).$$

推导三步走

第1步：写出链式法则的乘积形式

$$p(x_{1:V}) = p(x_1) \prod_{t=2}^V p(x_t | x_{1:t-1}).$$

第 2 步：用马尔可夫假设替换每个条件因子

对每个 $t \geq 2$, 有

$$p(x_t | x_{1:t-1}) = p(x_t | x_{t-1}).$$

第 3 步：逐项替换得到紧凑形式

把上式代回乘积：

$$p(x_{1:V}) = p(x_1) \prod_{t=2}^V p(x_t | x_{t-1}),$$

这就是 (10.3)。

一眼看懂的小例子 ($V = 4$)

链式法则：

$$p(x_1, x_2, x_3, x_4) = p(x_1) p(x_2 | x_1) p(x_3 | x_1, x_2) p(x_4 | x_1, x_2, x_3).$$

马尔可夫假设让“只看紧邻的过去”成立：

$$p(x_3 | x_1, x_2) = p(x_3 | x_2), \quad p(x_4 | x_1, x_2, x_3) = p(x_4 | x_3).$$

代入得：

$$p(x_1, x_2, x_3, x_4) = p(x_1) p(x_2 | x_1) p(x_3 | x_2) p(x_4 | x_3),$$

正是 (10.3) 的特例。

下标小提示（避免常见困惑）

- 最标准的写法是 $p(x_{1:V}) = p(x_1) \prod_{t=2}^V p(x_t | x_{t-1})$ 。
 - 如果你看到有人写成 $\prod_{t=1}^V p(x_t | x_{t-1})$, 那相当于引入了一个虚拟的“起始状态” x_0 , 并把 $p(x_1)$ 写成 $p(x_1 | x_0)$ 。两者本质一致, 但教科书里更常见的是从 $t = 2$ 开始的形式。
-

要点小结

- 链式法则把联合分布分解成一串条件分布的乘积。
- 一阶马尔可夫假设把每个 $p(x_t | x_{1:t-1})$ 简化为 $p(x_t | x_{t-1})$ 。

- 逐项替换后立刻得到马尔可夫链形式：

$$p(x_{1:V}) = p(x_1) \prod_{t=2}^V p(x_t | x_{t-1}).$$

- 直观理解：知道“现在”就足以预测“下一步”，无需再看更久远的过去。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 10.1 (a) A simple DAG on 5 nodes, numbered in topological order. Node 1 is the root, nodes 4 and 5 are the leaves. (b) A simple undirected graph, with the following maximal cliques: {1, 2, 3}, {2, 3, 4}, {3, 5}.

10.1.3 Graphical models

Although the first-order Markov assumption is useful for defining distributions on 1d sequences, how can we define distributions on 2d images, or 3d videos, or, in general, arbitrary collections of variables (such as genes belonging to some biological pathway)? This is where graphical models come in.

A graphical model (GM) is a way to represent a joint distribution by making CI assumptions. In particular, the nodes in the graph represent random variables, and the (lack of) edges represent CI assumptions. (A better name for these models would in fact be “independence diagrams”, but the term “graphical models” is now entrenched.) There are several kinds of graphical model, depending on whether the graph is directed, undirected,

or some combination of directed and undirected. In this chapter, we just study directed graphs. We consider undirected graphs in Chapter 19.

10.1.4 Graph terminology

Before we continue, we must define a few basic terms, most of which are very intuitive. A graph $G = (V, E)$ consists of a set of nodes or vertices, $V = \{1, \dots, V\}$, and a set of edges, $E = \{(s, t) : s, t \in V\}$. We can represent the graph by its adjacency matrix, in which we write $G(s, t) = 1$ to denote $(s, t) \in E$, that is, if $s \rightarrow t$ is an edge in the graph. If $G(s, t) = 1$ iff $G(t, s) = 1$, we say the graph is undirected, otherwise it is directed. We usually assume $G(s, s) = 0$, which means there are no self loops. Here are some other terms we will commonly use:

- Parent For a directed graph, the parents of a node is the set of all nodes that feed into it: $pa(s) = \{t : G(t, s) = 1\}$.
- Child For a directed graph, the children of a node is the set of all nodes that feed out of it: $ch(s) = \{t : G(s, t) = 1\}$.
- Family For a directed graph, the family of a node is the node and its parents, $fam(s) = \{s\} \cup pa(s)$.
- Root For a directed graph, a root is a node with no parents.
- Leaf For a directed graph, a leaf is a node with no children.
- Ancestors For a directed graph, the ancestors are the parents, grand-parents, etc of a node. That is, the ancestors of t is the set of nodes that connect to t via a trail: $anc(t) = \{s : s \rightarrow t\}$.
- Descendants For a directed graph, the descendants are the children, grand-children, etc of a node. That is, the descendants of s is the set of nodes that can be reached via trails from s : $desc(s) = \{t : s \rightarrow t\}$.
- Neighbors For any graph, we define the neighbors of a node as the set of all immediately connected nodes, $nbr(s) = \{t : G(s, t) = 1 \vee G(t, s) = 1\}$. For an undirected graph, we write $s \sim t$ to indicate that s and t are neighbors (so $(s, t) \in E$ is an edge in the graph).
- Degree The degree of a node is the number of neighbors. For directed graphs, we speak of the in-degree and out-degree, which count the number of parents and children.
- Cycle or loop For any graph, we define a cycle or loop to be a series of nodes such that we can get back to where

we started by following edges, $s_1 - s_2 \dots - s_n - s_1$, $n \geq 2$.

If the graph is directed, we may speak of a directed cycle.

For example, in Figure 10.1(a), there are no directed cycles, but $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$ is an undirected cycle.

- DAG A directed acyclic graph or DAG is a directed graph with no directed cycles. See Figure 10.1(a) for an example.

- Topological ordering For a DAG, a topological ordering or total ordering is a numbering of the nodes such that parents have lower numbers than their children. For example, in Figure 10.1(a), we can use (1, 2, 3, 4, 5), or (1, 3, 2, 5, 4), etc.

- Path or trail A path or trail $s ; t$ is a series of directed edges leading from s to t .

- Tree An undirected tree is an undirected graph with no cycles. A directed tree is a DAG in which there are no directed cycles. If we allow a node to have multiple parents, we call it a polytree, otherwise we call it a moral directed tree.

- Forest A forest is a set of trees.

- Subgraph A (node-induced) subgraph G_A is the graph created by using the nodes in A and their corresponding edges, $G_A = (V_A, E_A)$.

- Clique For an undirected graph, a clique is a set of nodes that are all neighbors of each other. A maximal clique is a clique which cannot be made any larger without losing the clique property. For example, in Figure 10.1(b), {1, 2} is a clique but it is not maximal, since we can add 3 and still maintain the clique property. In fact, the maximal cliques are as follows: {1, 2, 3}, {2, 3, 4}, {3, 5}.

10.1.5 Directed graphical models

A directed graphical model or DGM is a GM whose graph is a DAG. These are more commonly known as Bayesian networks. However, there is nothing inherently “Bayesian” about Bayesian networks: they are just a way of defining probability distributions. These models are also called belief networks. The term “belief” here refers to subjective probability. Once again, there is nothing inherently subjective about the kinds of probability distributions represented by DGMs. Finally, these models are sometimes called causal networks, because the directed arrows are sometimes interpreted as representing causal relations. However, there is nothing inherently causal about DGMs. (See Section 26.6.1 for a discussion of causal

DGMs.) For these reasons, we use the more neutral (but less glamorous) term DGM. The key property of DAGs is that the nodes can be ordered such that parents come before children. This is called a topological ordering, and it can be constructed from any DAG. Given such an order, we define the ordered Markov property to be the assumption that a node only depends on its immediate parents, not on all predecessors in the ordering, i.e.,
 $x_s \perp \text{xpred}(s) \setminus \text{pa}(s) \mid \text{xpa}(s)$ (10.4)

where $\text{pa}(s)$ are the parents of node s , and $\text{pred}(s)$ are the predecessors of node s in the ordering. This is a natural generalization of the first-order Markov property to from chains to general DAGs.

For example, the DAG in Figure 10.1(a) encodes the following joint distribution:

$$p(x_1:5) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_2, x_3, x_4) \quad (10.5)$$

$$= p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3)p(x_5|x_3) \quad (10.6)$$

In general, we have

$$p(x_1:V|G) = \bigvee_{t=1}^V p(x_t|x_{\text{pa}(t)}) \quad (10.7)$$

where each term $p(x_t|x_{\text{pa}(t)})$ is a CPD. We have written the distribution as $p(x|G)$ to emphasize that this equation only holds if the CI assumptions encoded in DAG G are correct. However, we will usual drop this explicit conditioning for brevity. If each node has $O(F)$ parents and K states, the number of parameters in the model is $O(VKF)$, which is much less than the $O(KV)$ needed by a model which makes no CI assumptions.

已思考 20s

好的！我先把你给的英文原文逐句、无遗漏地翻译成中文（保留图注、公式和要点），再用通俗方式详细解释，最后给出要点总结。

翻译

图 10.1

- (a) 一个包含 5 个结点的简单有向无环图 (DAG)，结点按拓扑顺序编号。结点 1 是根，结点 4 和 5 是叶子。
- (b) 一个简单的无向图，其极大团 (maximal cliques) 为：{1, 2, 3}, {2, 3, 4}, {3, 5}。

10.1.3 图模型 (Graphical models)

尽管一阶马尔可夫假设对定义一维序列上的分布很有用，那我们怎样在二维图像、三维视频，或者更一般地，在任意的变量集合（例如属于某个生物通路的一组基因）上定义分布呢？这正是图模型发挥作用的地方。

图模型 (GM) 是一种通过做条件独立 (CI) 假设来表示联合分布的方法。具体地，图中的结点表示随机变量，而边的存在或缺失表示相应的条件独立假设。（其实把这些模型称作“独立性示意图”更贴切，但“图模型”这个术语已经深入人心。）

根据图是有向的、无向的，还是有向与无向的混合，存在多种图模型。在本章我们只研究**有向图**；无向图留到第 19 章讨论。

10.1.4 图论术语

在继续之前，我们需要定义一些基本术语，其中大多数都很直观。一个图 $G = (V, E)$ 由一组**结点或顶点** $V = \{1, \dots, V\}$ 和一组**边** $E = \{(s, t) : s, t \in V\}$ 组成。我们可以用**邻接矩阵**来表示该图：若 $(s, t) \in E$ （也就是图中存在从 s 指向 t 的一条边 $s \rightarrow t$ ），则记 $G(s, t) = 1$ 。若且唯若 $G(s, t) = 1$ 与 $G(t, s) = 1$ 同时成立，我们称该图是**无向的**，否则称为**有向的**。通常假设 $G(s, s) = 0$ ，表示没有自环。下面是我们将频繁使用的一些术语：

- **父结点 (Parent)**：对有向图而言，某结点的父结点集合是所有**指向它的**结点：
 $\text{pa}(s) = \{t : G(t, s) = 1\}$ 。
- **子结点 (Child)**：对有向图而言，某结点的子结点集合是所有**从它出发**的结点：
 $\text{ch}(s) = \{t : G(s, t) = 1\}$ 。
- **家族 (Family)**：对有向图而言，某结点的家族是该结点及其父结点：
 $\text{fam}(s) = \{s\} \cup \text{pa}(s)$ 。
- **根 (Root)**：对有向图而言，根是**没有父结点**的结点。
- **叶 (Leaf)**：对有向图而言，叶是**没有子结点**的结点。
- **祖先 (Ancestors)**：对有向图而言，某结点的祖先是指它的父、祖父等。也就是说，结点 t 的祖先是可以通过一条轨迹 (trail) 连接到 t 的那些结点：
 $\text{anc}(t) = \{s : s \rightsquigarrow t\}$ 。
- **后代 (Descendants)**：对有向图而言，某结点的后代是它的子、孙等。也就是说，结点 s 的后代是从 s 出发沿轨迹可以到达的那些结点：
 $\text{desc}(s) = \{t : s \rightsquigarrow t\}$ 。
- **邻居 (Neighbors)**：对任意图而言，某结点的邻居是与其**直接相连**的结点集合：
 $\text{nbr}(s) = \{t : G(s, t) = 1 \vee G(t, s) = 1\}$ 。
 对无向图，我们写作 $s \sim t$ 表示 s 与 t 是邻居（即 $(s, t) \in E$ 是图中的一条边）。

- **度 (Degree)** : 结点的度是其邻居的数量。对有向图，还可以区分**入度与出度**，分别统计父结点数与子结点数。
 - **环 / 回路 (Cycle or loop)** : 对任意图，环/回路指这样一串结点：沿着边可以回到起点， $s_1 - s_2 - \dots - s_n - s_1, n \geq 2$ 。如果图是有向的，我们可以说**有向环**。例如，在图 10.1(a) 中，没有**有向环**；但 $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$ 组成了一个**无向环**。
 - **DAG: 有向无环图** (Directed Acyclic Graph, DAG) 是没有有向环的有向图。示例见图 10.1(a)。
 - **拓扑排序 (Topological ordering)** : 对 DAG，拓扑排序或全序是给结点编号，使得**父结点的编号小于其子结点**。例如，在图 10.1(a) 中，可以使用 $(1, 2, 3, 4, 5)$ 或 $(1, 3, 2, 5, 4)$ 等。
 - **路径 / 轨迹 (Path or trail)** : 一条从 s 到 t 的路径/轨迹 $s \rightsquigarrow t$ 是从 s 到 t 的一系列有向边。
 - **树 (Tree)** : **无向树**是没有环的无向图。**有向树**是没有有向环的 DAG。若允许一个结点有多个父结点，称为**多树 (polytree)**；否则称为**道德有向树 (moral directed tree)**。
 - **森林 (Forest)** : 一组树的并集称为森林。
 - **子图 (Subgraph)** : 一个 (由结点诱导的) 子图 G_A 是只使用 A 中的结点及其对应边所得到的图， $G_A = (V_A, E_A)$ 。
 - **团 / 极大团 (Clique / Maximal clique)** : 对无向图，团是一个结点集合，其中任意两个结点互为邻居。**极大团**是无法再加入任何新结点而仍保持团性质的团。例如，在图 10.1(b) 中， $\{1, 2\}$ 是一个团，但不是极大团，因为我们还能加入 3 仍保持团性质。事实上，极大团为： $\{1, 2, 3\}, \{2, 3, 4\}, \{3, 5\}$ 。
-

10.1.5 有向图模型 (Directed graphical models)

有向图模型 (DGM) 是其图为 DAG 的图模型。它们更常被称为**贝叶斯网络**。然而，贝叶斯网络并没有什么本质上的“贝叶斯性”：它们只是定义概率分布的一种方式。这些模型也被称为**信念网络 (belief networks)**。“信念”一词指的是主观概率；再强调一次，DGM 所表示的概率分布并不必然是主观的。最后，这些模型有时被称为**因果网络**，因为有向箭头有时会被解释为因果关系。然而，DGM 也没有内在的因果含义。（关于因果 DGM 的讨论见第 26.6.1 节。）基于这些原因，我们使用更中性（但不那么响亮）的术语 DGM。

DAG 的关键性质是：可以为结点排序，使得**父结点排在子结点之前**。这称为**拓扑排序**，可由任意 DAG 构造。给定这样一个顺序，我们定义**有序马尔可夫性质 (ordered Markov**

property)：一个结点仅依赖其**直接父结点**，而不依赖于序中所有的前驱，形式化为

$$x_s \perp x_{\text{pred}(s) \setminus \text{pa}(s)} \mid x_{\text{pa}(s)} \quad (10.4)$$

其中 $\text{pa}(s)$ 为结点 s 的父结点， $\text{pred}(s)$ 为在该排序中 s 的所有前驱结点。这是把一阶马尔可夫性质从**链**推广到**一般 DAG**的自然方式。

例如，图 10.1(a) 的 DAG 编码了如下联合分布：

```
\begin{aligned} p(x_{1:5}) &= p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \cdot p(x_4 \mid x_1, x_2, x_3) \cdot p(x_5 \mid x_1, x_2, x_3, x_4) \\ &\quad \cdot p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1) \cdot p(x_4 \mid x_2, x_3) \cdot p(x_5 \mid x_3) \end{aligned}
```

一般地，有

$$p(x_{1:V} \mid G) = \prod_{t=1}^V p(x_t \mid x_{\text{pa}(t)}) \quad (10.7)$$

其中每一项 $p(x_t \mid x_{\text{pa}(t)})$ 都是一个 CPD (条件概率分布)。我们把该分布写为 $p(x \mid G)$ 是为了强调：只有当 DAG G 中编码的条件独立假设成立时，这个等式才成立。不过为了简洁，我们通常会省略这种显式条件。

若每个结点有 $O(F)$ 个父结点、且每个变量有 K 个可能取值，则模型中的参数个数为 $O(VKF)$ ，这远小于**不做任何条件独立假设**时所需的 $O(K^V)$ 。

通俗解释

1) 图模型想解决什么？

当变量很多且彼此相关时，直接存整个联合分布需要指数级参数，学不动、算不动。图模型用“**画图**”的方式把谁依赖谁、谁与谁条件独立表达清楚：

- **结点** = 一个随机变量；
- **边** = 直接依赖关系；
- **没有边** = 在合适的条件下彼此独立。

这样就能把一个巨大概率分开成许多“**小块相乘**”，每块只管一个结点“看它的父母”。

2) 图注直观理解

- 图 10.1(a) (有向图)：1 指向 2 和 3；2、3 又指向 4；3 指向 5。1 是根 (无父母)，4、5 是叶 (无孩子)。
- 图 10.1(b) (无向图)：没有箭头，只有连线。极大团是“互相都连着”的最大结点组： $\{1,2,3\}, \{2,3,4\}, \{3,5\}$ 。

3) 关键图论词汇，用大白话记

- 父/子：箭头的起点是父，终点是子。
- 祖先/后代：多步的父系/子系。
- 根/叶：没有父/没有子。
- 邻居/度：直接连接的结点/邻居数量（有向图里可分入度与出度）。
- 环：兜一圈回到自己；DAG 就是“有向边兜不回去”。
- 拓扑排序：把父节点排在前面、孩子排在后面的一种编号。
- 团/极大团（无向图）：团=内部全互连；极大团=再也加不进结点了。

4) 为什么 DGM 有用？（有序马尔可夫性质）

- 先给 DAG 做个**拓扑排序**（父在前、子在后）。
- **有序马尔可夫性质**说：每个变量只依赖**它的父结点**，不需要依赖所有在它前面的结点。
- 因此联合分布就能写成

$$p(x_{1:V}) = \prod_{t=1}^V p(x_t | x_{\text{pa}(t)}),$$

这就是 (10.7)。

- 对图 10.1(a) 的例子：先按链式法则写出 (10.5)，再把“只依赖父母”这个事实逐项代入，就得到 (10.6)。其中
 - x_3 只有父亲 1，所以 $p(x_3 | x_1, x_2) = p(x_3 | x_1)$;
 - x_4 的父母是 2、3，所以条件里只留 (x_2, x_3) ;
 - x_5 的父亲是 3，所以条件只留 x_3 。

5) 参数节省到底有多大？

- 不做独立性假设：如果每个变量有 K 个取值，联合表需要大约 $O(K^V)$ 个参数，指数爆炸。
- 做 DGM、每个结点只有 F 个父结点：每个 CPD 是一个“ $K \times K^F$ ”的表（对子结点的每种父配置给出一个分布），所以严格来说每个结点大约 $O(K^{F+1})$ 个参数，总共 $O(VK^{F+1})$ 。

说明：原文给了 $O(VKF)$ 这一更粗糙的数量级写法（常见教材里也会简写/有排版差异）。实际精确计数通常按 K^{F+1} 量级理解更直观：固定 F 时是多项式而不是指数随 V 增长。

小示例 (把 10.5 → 10.6 走一遍)

图 10.1(a) 的父集：

- $\text{pa}(1) = \emptyset, \text{pa}(2) = \{1\}, \text{pa}(3) = \{1\}, \text{pa}(4) = \{2, 3\}, \text{pa}(5) = \{3\}$ 。

链式法则 (随便用一种拓扑顺序 1,2,3,4,5) :

$$p(x_{1:5}) = p(x_1)p(x_2 | x_1)p(x_3 | x_1, x_2)p(x_4 | x_1, x_2, x_3)p(x_5 | x_1, x_2, x_3, x_4).$$

按“只依赖父母”逐项删掉多余条件：

$$p(x_{1:5}) = p(x_1)p(x_2 | x_1)p(x_3 | x_1)p(x_4 | x_2, x_3)p(x_5 | x_3),$$

与 (10.6) 一致。

要点总结

- 图模型用图表达谁依赖谁、谁与谁在何种条件下独立，从而把大联合分布因子分解成许多小的条件分布。
- DAG (贝叶斯网/DGM) : 没有有向环；存在拓扑排序；满足有序马尔可夫性质：

$$p(x) = \prod_t p(x_t | x_{\text{pa}(t)}).$$

- 图论术语要熟：父/子、祖先/后代、根/叶、邻居/度、环、拓扑排序、树/森林、子图、(无向图的) 团与极大团。
- 参数量：不作 CI 假设是 $O(K^V)$ (指数爆炸)；DGM 在每结点最多 F 个父时，参数量约为 $O(VK^{F+1})$ (常见教材也会写成更粗略的量级)，随 V 线性、随 F 多项式增长。
- 理解 10.5→10.6 的关键：先用链式法则写全条件，再用“只依赖父结点”把多余条件剔除。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 10.2 (a) A naive Bayes classifier represented as a DGM. We assume there are $D = 4$ features, for simplicity. Shaded nodes are observed, unshaded nodes are hidden.
 (b) Tree-augmented naive Bayes classifier for $D = 4$ features. In general, the tree topology can change depending on the value of y .

Figure 10.3 A first and second order Markov chain.

Figure 10.4 A first-order HMM.

10.2 Examples

In this section, we show a wide variety of commonly used probabilistic models can be conveniently represented as DGMs.

10.2.1 Naive Bayes classifiers

In Section 3.5, we introduced the naive Bayes classifier. This assumes the features are conditionally independent given the class label. This assumption is illustrated in Figure 10.2(a). This allows us to write the joint distribution as follows:

$$p(y, \mathbf{x}) = p(y) \prod_{j=1}^D p(x_j|y) \quad (10.8)$$

The naive Bayes assumption is rather naive, since it assumes the features are conditionally independent. One way to capture correlation between the features is to use a graphical model. In particular, if the model is a tree, the method is known as a tree-augmented naive Bayes classifier or TAN model (Friedman et al. 1997). This is illustrated in Figure 10.2(b). The reason to use a tree, as opposed to a generic graph, is two-fold. First, it is easy to find the optimal tree structure using the Chow-Liu algorithm, as explained in Section 26.3. Second, it is easy to handle missing features in a tree-structured model, as we explain in Section 20.2.

10.2.2 Markov and hidden Markov models

Figure 10.3(a) illustrates a first-order Markov chain as a DAG. Of course, the assumption that the immediate past, x_{t-1} , captures everything we need to know about the entire history, $x_1:t-2$, is a bit strong. We can relax it a little by adding a dependence from x_{t-2} to x_t as well; this is called a second order Markov chain, and is illustrated in Figure 10.3(b). The corresponding joint has the following form:

$$p(x_1:T) = p(x_1, x_2)p(x_3|x_1, x_2)p(x_4|x_2, x_3) \dots = p(x_1, x_2) T \text{ t=3}$$

$p(x_t|x_{t-1}, x_{t-2}) \quad (10.9)$

We can create higher-order Markov models in a similar way. See Section 17.2 for a more detailed discussion of Markov models.

Unfortunately, even the second-order Markov assumption may be inadequate if there are longrange correlations amongst the observations. We can't keep building ever higher order models, since the number of parameters will blow up. An alternative approach is to assume that there is an underlying hidden process, that can be modeled by a first-order Markov chain, but that the data is a noisy observation of this process. The result is known as a hidden Markov model or HMM, and is illustrated in Figure 10.4. Here z_t is known as a hidden variable at "time" t , and x_t is the observed variable. (We put "time" in quotation marks, since these models can be applied to any kind of sequence data, such as genomics or language, where t represents location rather than time.) The CPD $p(z_t|z_{t-1})$ is the transition model, and the CPD $p(x_t|z_t)$ is the observation model.

The hidden variables often represent quantities of interest, such as the identity of the word that someone is currently speaking. The observed variables are what we measure, such as the acoustic waveform. What we would like to do is estimate the hidden state given the data, i.e., to compute $p(z_t|x_1:t, \theta)$. This is called state estimation, and is just another form of probabilistic inference. See Chapter 17 for further details on HMMs.

已思考 12s

好的！我先把你给的英文原文逐句、无遗漏地译成中文（保留图注、公式与术语），然后用通俗方式详细解释，并在最后做要点总结。

翻译

图 10.2

(a) 以 DGM 表示的**朴素贝叶斯分类器**。为简单起见，我们假设有 $D = 4$ 个特征。**实心（灰色）结点**为已观测变量，**空心结点**为隐变量。

(b) $D = 4$ 个特征的**树增强朴素贝叶斯分类器** (tree-augmented naive Bayes, TAN)。一般来说，树的拓扑结构可以随 γ 的取值而变化。

图 10.3 一阶与二阶马尔可夫链。

图 10.4 一阶隐马尔可夫模型 (HMM)。

10.2 例子

在本节中，我们展示：许多常用的概率模型都可以方便地表示为 DGM。

10.2.1 朴素贝叶斯分类器

在第 3.5 节，我们介绍了朴素贝叶斯分类器。它假设在给定类别标签的条件下，各特征相互条件独立。这一假设如图 10.2(a) 所示。由此我们可以把联合分布写成

$$p(y, \mathbf{x}) = p(y) \prod_{j=1}^D p(x_j | y) \quad (10.8)$$

朴素贝叶斯假设之所以“朴素”，是因为它假定特征之间条件独立。捕捉特征间相关性的一种方法是使用图模型。特别地，若模型是一棵树，该方法称为树增强朴素贝叶斯 (TAN) (Friedman 等, 1997)，如图 10.2(b) 所示。之所以用“树”而不是一般的图，有两个原因。第一，用 Chow-Liu 算法 (第 26.3 节) 可以容易地找到最优的树结构。第二，如第 20.2 节所述，在树结构模型中处理缺失特征更容易。

10.2.2 马尔可夫模型与隐马尔可夫模型

图 10.3(a) 以一个 DAG 形式展示了一阶马尔可夫链。当然，认为最近的过去 x_{t-1} 就完全概括了整个历史 $x_{1:t-2}$ 的信息，这个假设有点强。我们可以稍微放宽它：再让 x_{t-2} 也影响 x_t ；这称为二阶马尔可夫链，如图 10.3(b) 所示。相应的联合分布为

$$p(x_{1:T}) = p(x_1, x_2) p(x_3 | x_1, x_2) p(x_4 | x_2, x_3) \cdots = p(x_1, x_2) \prod_{t=3}^T p(x_t | x_{t-1}, x_{t-2}) \quad (10.9)$$

以类似方式，我们可以构造更高阶的马尔可夫模型。关于马尔可夫模型的更详细讨论见第 17.2 节。

不幸的是，如果观测之间存在远距离（长程）相关，即使二阶马尔可夫假设也可能不够好。我们不能无限制地提升阶数，否则参数数量会爆炸。一种替代做法是假设存在一个潜在的隐过程，它本身可由一阶马尔可夫链建模，而数据是该过程的有噪声观测。这就得到隐马尔可夫模型 (HMM)，如图 10.4 所示。此处 z_t 称为“时间” t 的隐变量， x_t 是观测变量。（我们给“时间”加引号，是因为这些模型可用于任何序列数据，比如基因组或语言，此时 t 表示位置而非时间。）CPD $p(z_t | z_{t-1})$ 称为转移模型，CPD $p(x_t | z_t)$ 称为观测模型。

隐变量通常代表我们感兴趣的量，比如某人当前说出的**词的身份**。观测变量则是我们能测量到的，比如**声学波形**。我们希望做的是：在给定数据时估计隐状态，即计算 $p(z_t | x_{1:t}, \theta)$ 。这称为**状态估计**，本质上就是一种概率推断。关于 HMM 的更多细节参见第 17 章。

通俗解释

1) 图 10.2: 朴素贝叶斯 vs. 树增强朴素贝叶斯 (TAN)

- **朴素贝叶斯 (NB)**

结构: 一个类别结点 Y 指向所有特征 X_1, \dots, X_D 。

含义: 知道类别 Y 之后，各特征彼此独立。

公式 (10.8): 整件事拆成“先选类别 $p(y)$ ”，再“对每个特征独立地产生 $p(x_j | y)$ ”并相乘。

优点: 参数少、训练快、鲁棒。

缺点: 忽略了特征之间的相关性（比如“free”和“win”这两个词常一起出现）。

- **TAN (树增强)**

在 NB 的基础上，**允许特征之间再加一些有向边**，但这些边构成一棵树（不出现环）。

为什么用树？

1. Chow-Liu 能在多项式时间内找到“最优的特征树”（最大化似然/互信息）。
2. 树上推断简单、**缺失值**容易处理（把缺失特征当未知变量，用边结构把信息传递过来）。

注意: 结构通常是**针对每个类别 y 学的**——因此“树的拓扑可以随 y 变化”。

2) 图 10.3: 一阶/二阶马尔可夫链

- **一阶**: 只看前一个状态 x_{t-1} 。联合分布就是“初始分布 \times 一串转移概率”。
- **二阶**: 看前两个状态 x_{t-1}, x_{t-2} 。联合分布见式 (10.9): 先给定 (x_1, x_2) ，然后从 $t = 3$ 起每一步用 $p(x_t | x_{t-1}, x_{t-2})$ 。
- **为什么不总用高阶?** 阶数每升 1，参数量成倍增加（每个条件概率表要对更多的父状态组合给出分布），很快不可控。

3) 图 10.4: 隐马尔可夫模型 (HMM)

- **思路**: 让“有记忆的部分”放到**隐状态** z_t 里，隐状态之间只是一阶马尔可夫；观测 x_t 由当前隐状态产生。
- **两块 CPD**:
 - **转移** $p(z_t | z_{t-1})$: 系统如何从上一隐状态走到当前隐状态；

- 观测 $p(x_t | z_t)$: 当前隐状态会“发射/产生”什么观测。
 - 优势: 即使 x_t 之间存在长程相关, 也能通过“穿过隐状态链”的方式间接表达, 而不用提升马尔可夫阶数。
 - 要做的事: 给定到目前为止的观测 $x_{1:t}$, 求当前隐状态的分布 $p(z_t | x_{1:t})$ (**过滤/状态估计**)。常用算法: 前向-后向、Viterbi 等 (第 17 章详述)。
-

小例子帮你直觉化

- NB 邮件分类: $Y \in \{\text{spam}, \text{ham}\}$, 特征是若干词出现与否。NB 假设“在给定 Y 时, 每个词是否出现互不影响”。因此

$$p(y, \mathbf{x}) = p(y) \prod_j p(x_j | y).$$

- TAN: 如果你发现“free”和“win”强相关, 就在 X_{free} 和 X_{win} 之间加一条边 (同时仍由 Y 指向它们), 更贴近真实。
 - 一阶马尔可夫 (天气): 今天的天气只和昨天有关。
 - HMM (语音识别): 隐状态 z_t =当前发的音素, 观测 x_t =声学特征。音素序列是一阶马尔可夫, 但声学观测是对音素的“嘈杂”测量。
-

要点总结

- NB 结构: $Y \rightarrow X_1, \dots, X_D$ 。假设“给定 Y 时特征独立”; 联合分布为式 (10.8)。简单高效但忽略特征相关。
- TAN: 在 NB 上给特征之间加树结构来建模相关; Chow-Liu 易求最优; 对**缺失特征**友好; 树结构可随类 y 变化。
- 马尔可夫链: 一阶只看 x_{t-1} , 二阶看 x_{t-1}, x_{t-2} ; 二阶联合分布见式 (10.9)。阶数高→参数爆炸。
- HMM: 用隐状态链 z_t (一阶马尔可夫) + 观测模型 $p(x_t | z_t)$ 来表达长程依赖; 核心任务是**状态估计** $p(z_t | x_{1:t}, \theta)$ 。
- DGM 表达的好处: 把“谁依赖谁”画清楚, 使联合分布因子分解, 推断与学习都更可控。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 10.5 (a) The alarm network. Figure generated by visualizeAlarmNetwork. (b) The QMR network.

Table 10.1 Noisy-OR CPD for 2 parents augmented with leak node. We have omitted the t subscript for brevity.

10.2.3 Medical diagnosis

Consider modeling the relationship between various variables that are measured in an intensive care unit (ICU), such as the breathing rate of a patient, their blood pressure, etc. The alarm network in Figure 10.5(a) is one way to represent these (in)dependencies (Beinlich et al. 1989). This model has 37 variables and 504 parameters. Since this model was created by hand, by a process called knowledge engineering, it is known as a probabilistic expert system. In Section 10.4, we discuss how to learn the parameters of DGMs from data, assuming the graph structure is known, and in Chapter 26, we discuss how to learn the graph structure itself. A different kind of medical diagnosis network, known as the quick medical reference or QMR network (Shwe et al. 1991), is shown in Figure 10.5(b). This was designed to model infectious diseases.

The QMR model is a bipartite graph structure, with diseases (causes) at the top and symptoms or findings at the bottom. All nodes are binary. We can write the distribution as follows:

$$p(v, h) = s p(hs) t p(vt|hpa(t)) \quad (10.10)$$

where hs represent the hidden nodes (diseases), and vt represent the visible nodes (symptoms). The CPD for the root nodes are just Bernoulli distributions, representing the prior probability of that disease. Representing the CPDs for the leaves (symptoms) using CPTs would require too many parameters, because the fan-in (number of

parents) of many leaf nodes is very high. A natural alternative is to use logistic regression to model the CPD, $p(v_t = 1 | hpa(t)) = \text{sigm}(w^T t hpa(t))$. (A DGM in which the CPDs are logistic regression distributions is known as a sigmoid belief net (Neal 1992).) However, since the parameters of this model were created by hand, an alternative CPD, known as the noisy-OR model, was used.

The noisy-OR model assumes that if a parent is on, then the child will usually also be on (since it is an or-gate), but occasionally the “links” from parents to child may fail, independently at random. In this case, even if the parent is on, the child may be off. To model this more precisely, let $\theta_{st} = 1 - q_{st}$ be the probability that the $s \rightarrow t$ link fails, so $q_{st} = 1 - \theta_{st} = p(v_t = 1 | hs = 1, h_s = 0)$ is the probability that s can activate t on its own (its “causal power”). The only way for the child to be off is if all the links from all parents that are on fail independently at random. Thus

$$p(v_t = 0 | h) = \prod_{s \in pa(t)} \theta_{st} \quad (10.11)$$

Obviously, $p(v_t = 1 | h) = 1 - p(v_t = 0 | h)$.

If we observe that $v_t = 1$ but all its parents are off, then this contradicts the model. Such a data case would get probability zero under the model, which is problematic, because it is possible that someone exhibits a symptom but does not have any of the specified diseases. To handle this, we add a dummy leak node h_0 , which is always on; this represents “all other causes”. The parameter q_{0t} represents the probability that the background leak can cause the effect on its own. The modified CPD becomes

$$p(v_t = 0 | h) = \theta_{0t} + \prod_{s \in pa(t)} \theta_{hs} \quad (10.11)$$

See Table 10.1 for a numerical example. If we define $w_{st} = \log(\theta_{st})$, we can rewrite the CPD as

$$p(v_t = 1 | h) = 1 - \exp(-\sum_{s \in pa(t)} w_{st} + \theta_{0t}) \quad (10.12)$$

We see that this is similar to a logistic regression model. Bipartite models with noisy-OR CPDs are called BN2O models. It is relatively easy to set the θ_{st} parameters by hand, based on domain expertise. However, it is also possible to learn them from data (see e.g., (Neal 1992; Meek and Heckerman 1997)). Noisy-OR CPDs have also proved useful in modeling human causal learning (Griffiths and Tenenbaum 2005), as well as general binary classification settings (Yuille and Zheng 2009).

好的！我先把你给的英文原文逐句、无遗漏地译成中文（保留图注、公式与术语），再用通俗方式详细解释，最后做要点总结。

翻译

图 10.5

- (a) 报警网络 (alarm network)。该图由 *visualizeAlarmNetwork* 生成。
- (b) QMR 网络。

表 10.1 对**两个父结点并加入泄漏结点 (leak node) **的 Noisy-OR 条件概率分布 (CPD)。为简洁起见，我们省略了下标 t 。

10.2.3 医学诊断 (Medical diagnosis)

考虑对重症监护室 (ICU) 中测量到的各种变量之间的关系进行建模，例如病人的呼吸频率、血压等。图 10.5(a) 中的报警网络就是表示这些 (不) 独立性的一个方法 (Beinlich 等, 1989)。该模型包含 37 个变量和 504 个参数。由于此模型是通过一种称为**知识工程** (knowledge engineering) 的手工过程创建的，因此它被称为**概率专家系统**。在第 10.4 节，我们讨论在**图结构已知**的情况下如何从数据中学习 DGM 的参数；在第 26 章，我们讨论如何学习**图结构本身**。

另一种医学诊断网络称为**快速医学参考** (quick medical reference, QMR) 网络 (Shwe 等, 1991)，如图 10.5(b) 所示。它被设计用来建模传染病。QMR 模型是一个**二部图**结构，上层是疾病 (原因)，下层是症状或体征 (结果)。**所有结点都是二元变量**。我们可以把分布写为

$$p(\mathbf{v}, \mathbf{h}) = \prod_s p(h_s) \prod_t p(v_t | \mathbf{h}_{\text{pa}(t)}) \quad (10.10)$$

其中 $\{h_s\}$ 表示**隐结点 (疾病)**， $\{v_t\}$ 表示**可见结点 (症状)**。根结点 (疾病) 的 CPD 只是伯努利分布，表示该疾病的**先验概率**。

若用**条件概率表** (CPT) 来表示叶结点 (症状) 的 CPD，则需要太多参数，因为许多叶结点的**扇入** (fan-in, 父结点个数) 非常大。一个自然的替代方法是用**逻辑回归**来建模 CPD，即

$$p(v_t = 1 | \mathbf{h}_{\text{pa}(t)}) = \text{sigm}(\mathbf{w}_t^\top \mathbf{h}_{\text{pa}(t)}).$$

(CPD 都是逻辑回归分布的 DGM 被称为 Sigmoid belief net (Neal, 1992)。) 然而，由于该模型的参数是**手工指定**的，作者采用了另一种 CPD，称为 Noisy-OR 模型。Noisy-OR 的假设是：**如果某个父结点为“开 (1)”，子结点通常也会“开”** (因为它像一个

OR 门)；但从父到子的“连接”可能以相互独立的方式随机失效。在这种情况下，即使父是“开”，子也可能是“关 (0)”。更精确地建模时，令

$\theta_{st} = 1 - q_{st}$ 为链路 $s \rightarrow t$ 失效的概率，因此

$$q_{st} = 1 - \theta_{st} = p(v_t = 1 \mid h_s = 1, \mathbf{h}_{-s} = 0)$$

是结点 s 单独激活 t 的概率 (称为它的“因果能力 (causal power)”)。

子结点为 0 的唯一方式是：所有处于“开”状态的父结点到子的链路都独立地失效。于是

$$p(v_t = 0 \mid \mathbf{h}) = \prod_{s \in \text{pa}(t)} \theta_{st}^{I(h_s=1)} \quad (10.11)$$

显然， $p(v_t = 1 \mid \mathbf{h}) = 1 - p(v_t = 0 \mid \mathbf{h})$ 。

如果我们观测到 $v_t = 1$ ，而它的所有父结点都为 0，那么这与模型矛盾。这种数据在该模型下会得到概率 0，这有问题——因为某人可能出现某个症状，但并没有任何列出的疾病。为了解决这一点，我们加入一个虚拟的泄漏结点 h_0 ，它始终为 1；它代表“所有其他原因”。参数 q_{0t} 表示背景泄漏单独导致该效应的概率。修改后的 CPD 为

$$p(v_t = 0 \mid \mathbf{h}) = \theta_{0t} \prod_{s \in \text{pa}(t)} \theta_{st}^{h_s}.$$

参见表 10.1 的数值示例。若定义 $w_{st} = \log \theta_{st}$ ，则可把 CPD 改写为

$$p(v_t = 1 \mid \mathbf{h}) = 1 - \exp\left(w_{0t} + \sum_s w_{st}\right). \quad (10.12)$$

可见它与逻辑回归模型相似。带有 Noisy-OR CPD 的二部图模型称为 BN2O 模型。根据领域知识，手工设定 θ_{st} 相对容易；当然，也可以从数据中学习这些参数 (例如 Neal, 1992; Meek 与 Heckerman, 1997)。Noisy-OR CPD 也被证明在人类因果学习建模 (Griffiths 与 Tenenbaum, 2005) 以及一般的二分类场景 (Yuille 与 Zheng, 2009) 中很有用。

通俗解释

1) 两个经典医学网络

- 报警网络 (Alarm)：把 ICU 里诸如血压、心率、呼吸机状态等变量都画成一个 DGM。手工 (专家) 建立：37 个结点、504 个参数。它告诉我们：谁直接影响谁，以及在这些假设下如何计算联合概率、做诊断推断。
- QMR：为感染病设计。二部图：上层是疾病 (隐变量 h_s)，下层是症状 (可见变量 v_t)，边只从疾病指向症状。所有变量二元 (有/无)。联合分布的结构就是“各疾病的先验 \times 每个症状在给定它的父疾病时的条件概率”。

2) 为什么不用大表 (CPT) 来写症状的 CPD?

- 很多症状由**很多疾病共同影响** (扇入高)。如果用 CPT, 就要为“父疾病的每一种组合”都存一个概率——**参数爆炸**。
- 两个常见替代:

- 逻辑回归 CPD:**

$$p(v_t = 1 \mid \mathbf{h}_{\text{pa}(t)}) = \sigma(\mathbf{w}_t^\top \mathbf{h}). \text{ 参数随父数量线性增长。}$$

- Noisy-OR CPD** (本节主角) : 抓住“**有一个原因能生效就会触发症状**, 但每条因果链可能**独立失效**”的直觉。

3) Noisy-OR 的直觉与公式

- 把每条“疾病→症状”的连接看成有**失效率** θ_{st} (**失效**=该疾病即便存在也**没有触发症状**)。
- 若某时刻“打开”的父疾病集合为 A , 则**症状不出现的概率**等于“这些已打开的链路都失效”的概率乘积:

$$p(v_t = 0 \mid \mathbf{h}) = \prod_{s \in A} \theta_{st}.$$

于是

$$p(v_t = 1 \mid \mathbf{h}) = 1 - \prod_{s \in A} \theta_{st}.$$

- 泄漏结点** h_0 : 解决“父全关但症状仍出现”的情况。它始终为 1, 代表“其他未建模原因”, 有自己的失效概率 θ_{0t} 。加入后:

$$p(v_t = 0 \mid \mathbf{h}) = \theta_{0t} \prod_{s \in A} \theta_{st}, \quad p(v_t = 1 \mid \mathbf{h}) = 1 - \theta_{0t} \prod_{s \in A} \theta_{st}.$$

- 令 $w_{st} = \log \theta_{st}$, 乘积变和, 得到式 (10.12):

$$p(v_t = 1 \mid \mathbf{h}) = 1 - \exp(w_{0t} + \sum_s h_s w_{st}),$$

形式上类似逻辑回归 (但注意是 $1 - \exp(\cdot)$, 不是标准 sigmoid)。

4) 表 10.1 (两个父 + 泄漏) 的读法

- h_0 是泄漏结点, 恒为 1; h_1, h_2 是两个疾病父结点; v 是症状。
- 记 $\theta_0, \theta_1, \theta_2$ 为三条链路的**失效概率**。则
 - 若 $h_1 = 0, h_2 = 0$: 只有泄漏链路可能触发,

$$p(v = 0) = \theta_0, \quad p(v = 1) = 1 - \theta_0.$$

- 若仅 $h_1 = 1$:
 $p(v = 0) = \theta_0\theta_1, \quad p(v = 1) = 1 - \theta_0\theta_1.$
- 若仅 $h_2 = 1$:
 $p(v = 0) = \theta_0\theta_2, \quad p(v = 1) = 1 - \theta_0\theta_2.$
- 若两者都 1:
 $p(v = 0) = \theta_0\theta_1\theta_2, \quad p(v = 1) = 1 - \theta_0\theta_1\theta_2.$
- 这正是表中的四行数值。

5) 实操层面

- BN2O (bipartite noisy-OR) : 二部图 + Noisy-OR CPD。
 - 好处:
 - 易于由专家给出 θ_{st} ("这个病单独导致这个症状的大致概率是多少?")。
 - 也能从数据学习这些参数。
 - 适合大规模疾病×症状 (父多、叶多) 的建模; 在心理学、二分类等领域也常用。
-

要点总结

- 报警网络: 手工构建的 DGM (专家系统), 37 个变量、504 个参数, 表达 ICU 变量间的因果/依赖关系。
- QMR: 用于传染病的二部图: 上层疾病 (隐、二元), 下层症状 (显、二元); 联合分布为式 (10.10)。
- 为什么不用 CPT: 症状父亲多→CPT 参数爆炸; 改用逻辑回归 CPD 或 Noisy-OR CPD。
- Noisy-OR 核心: 父为 1 会“激活”子, 但每条父→子链路有独立失效概率 θ_{st} 。

$$p(v_t = 0 \mid \mathbf{h}) = \prod_{s \in \text{pa}(t)} \theta_{st}^{h_s}, \quad p(v_t = 1 \mid \mathbf{h}) = 1 - \dots$$

泄漏结点 h_0 解决“父全关而症状仍出现”的情形。

- 与逻辑回归的联系: 取对数把乘积变加和, 得到形式类似的式 (10.12)。
- BN2O: Noisy-OR 的二部图模型; 参数可手工设定也可学习; 在因果学习与二分类中有广泛应用。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Table 10.2 CPT which encodes a mapping from genotype to phenotype (bloodtype). This is a deterministic, but many-to-one, mapping.

Figure 10.6 Left: family tree, circles are females, squares are males. Individuals with the disease of interest are highlighted. Right: DGM for two loci. Blue nodes X_{ij} is the observed phenotype for individual i at locus j . All other nodes are hidden. Orange nodes $G_{p/m\ ij}$ is the paternal/ maternal allele. Small red nodes $z_{p/m\ ijl}$ are the paternal/ maternal selection switching variables. These are linked across loci, $z_{m\ ij} \rightarrow z_{m\ i,j+1}$ and $z_{p\ ij} \rightarrow z_{p\ i,j+1}$. The founder (root) nodes do not have any parents, and hence do no need switching variables. Based on Figure 3 from (Friedman et al. 2000).

10.2.4 Genetic linkage analysis

Another important (and historically very early) application of DGMs is to the problem of genetic linkage analysis. We start with a pedigree graph, which is a DAG that representing the relationship between parents and children, as shown in Figure 10.6(a). We then convert this to a DGM, as we explain below. Finally we perform probabilistic inference in the resulting model.

In more detail, for each person (or animal) i and location or locus j along the genome, we create three nodes: the observed marker X_{ij} (which can be a property such as blood type, or just a fragment of DNA that can be measured), and two hidden alleles, $G_{m\ ij}$ and $G_{p\ ij}$, one inherited from i 's mother (maternal allele) and the other from i 's father (paternal allele). Together, the ordered pair $G_{ij} = (G_{m\ ij}, G_{p\ ij})$ constitutes i 's hidden genotype at locus j .

Obviously we must add $Gm_{ij} \rightarrow X_{ij}$ and $Gp_{ij} \rightarrow X_{ij}$ arcs representing the fact that genotypes cause phenotypes (observed manifestations of genotypes). The CPD $p(X_{ij}|Gm_{ij}, Gp_{ij})$ is called the penetrance model. As a very simple example, suppose $X_{ij} \in \{A, B, O, AB\}$ represents person i's observed bloodtype, and $Gm_{ij}, Gp_{ij} \in \{A, B, O\}$ is their genotype. We can represent the penetrance model using the deterministic CPD shown in Table 10.2. For example, A dominates O, so if a person has genotype AO or OA, their phenotype will be A.

In addition, we add arcs from i's mother and father into G_{ij} , reflecting the Mendelian inheritance of genetic material from one's parents. More precisely, let $m_i = k$ be i's mother. Then Gm_{ij} could either be equal to Gm_{kj} or Gp_{kj} , that is, i's maternal allele is a copy of one of its mother's two alleles. Let Zm_{ij} be a hidden variable than specifies the choice. We can model this using the following CPD, known as the inheritance model:

$$p(Gm_{ij} | Gm_{kj}, Gp_{kj}, Zm_{ij}) = I(Gm_{ij} = Gm_{kj}) \text{ if } Zm_{ij} = m \\ I(Gm_{ij} = Gp_{kj}) \text{ if } Zm_{ij} = p \quad (10.13)$$

We can define $p(Gp_{ij}|Gm_{kj}, Gp_{kj}, Zp_{ij})$ similarly, where $k = p_i$ is i's father. The values of the Z_{ij} are said to specify the phase of the genotype. The values of $Gp_{ij}, Gm_{ij}, Zp_{ij}$ and Zm_{ij} constitute the haplotype of person i at locus j.1

Next, we need to specify the prior for the root nodes, $p(Gm_{ij})$ and $p(Gp_{ij})$. This is called the founder model, and represents the overall prevalence of different kinds of alleles in the population. We usually assume independence between the loci for these founder alleles. Finally, we need to specify priors for the switch variables that control the inheritance process. These variables are spatially correlated, since adjacent sites on the genome are typically inherited together (recombination events are rare). We can model this by imposing a two-state Markov chain on the Z 's, where the probability of switching state at locus j is given by $\theta_j = 1/2(1 - e^{-2d_j})$, where d_j is the distance between loci j and j + 1. This is called the recombination model.

The resulting DGM is shown in Figure 10.6(b): it is a series of replicated pedigree DAGs, augmented with switching Z variables, which are linked using Markov chains. (There is a related model known as phylogenetic HMM (Siepel and

Haussler 2003), which is used to model evolution amongst phylogenies.)

As a simplified example of how this model can be used, suppose we only have one locus, corresponding to blood type. For brevity, we will drop the j index. Suppose we observe $x_i = A$. Then there are 3 possible genotypes: G_i is (A,A) , (A,O) or (O,A) . There is ambiguity because the genotype to phenotype mapping is many-to-one. We want to reverse this mapping. This is known as an inverse problem. Fortunately, we can use the blood types of relatives to help disambiguate the evidence. Information will “flow” from the other x_i ’s up to their G_i ’s, then across to i ’s G_i via the pedigree DAG. Thus we can combine our local evidence $p(x_i|G_i)$ with an informative prior, $p(G_i|x_{-i})$, conditioned on the other data, to get a less entropic local posterior, $p(G_i|x) \propto p(x_i|G_i)p(G_i|x_{-i})$. In practice, the model is used to try to determine where along the genome a given disease-causing gene is assumed to lie — this is the genetic linkage analysis task. The method works as follows. First, suppose all the parameters of the model, including the distance between all the marker loci, are known. The only unknown is the location of the disease-causing gene. If there are L marker loci, we construct $L + 1$ models: in model i , we postulate that the disease gene comes after marker i , for $0 < i < L + 1$. We can estimate the Markov switching parameter θ , and hence the distance d between the disease gene and its nearest known locus. We measure the quality of that model using its likelihood, $p(D|\theta)$. We then can then pick the model with highest likelihood (which is equivalent to the MAP model under a uniform prior). Note, however, that computing the likelihood requires marginalizing out all the hidden Z and G variables. See (Fishelson and Geiger 2002) and the references therein for some exact methods for this task; these are based on the variable elimination algorithm, which we discuss in Section 20.3.

Unfortunately, for reasons we explain in Section 20.5, exact methods can be computationally intractable if the number of individuals and/or loci is large. See (Albers et al. 2006) for an approximate method for computing the likelihood; this is based on a form of variational inference, which we will discuss in Section 22.4.1.

已思考 18s

好的！我先把你给的文字与图注逐句完整翻译（包含表格内容与公式），再做通俗详解，最后要点总结。

翻译

表 10.2

把“基因型 (genotype) → 表型 (phenotype, 血型)”的映射编码成的 CPT (条件概率表)。这是一个**确定性的 (deterministic)**、但**多对一 (many-to-one) **的映射。

下表的两列 G^p, G^m 分别表示**父方等位基因与母方等位基因**；右侧四列是血型取值的条件概率：

G^p	G^m	$p(X = a)$	$p(X = b)$	$p(X = o)$	$p(X = ab)$
a	a	1	0	0	0
a	b	0	0	0	1
a	o	1	0	0	0
b	a	0	0	0	1
b	b	0	1	0	0
b	o	0	1	0	0
o	a	1	0	0	0
o	b	0	1	0	0
o	o	0	0	1	0

图 10.6

左：家系谱（家族树），圆圈表示女性，方形表示男性。被高亮的个体患有关注的疾病。

右：**两个位点 (loci) 的 DGM**。蓝色结点 X_{ij} 是个体 i 在位点 j 的**观测表型**；其他结点均为隐变量。橙色结点 $G_{ij}^{p/m}$ 分别是父方 / 母方**等位基因**。小的红色结点 $z_{ijl}^{p/m}$ 是父方 / 母方的**选择开关变量**。这些开关变量跨位点相连： $z_{ij}^m \rightarrow z_{i,j+1}^m$ 与 $z_{ij}^p \rightarrow z_{i,j+1}^p$ 。**创始者**

(founder, 根) 结点没有父结点, 因此不需要开关变量。基于 (Friedman 等, 2000) 图 3。

10.2.4 遗传连锁分析 (Genetic linkage analysis)

DGMs 的另一个重要 (且历史上很早) 的应用就是**遗传连锁分析**。我们先从**谱系图 (pedigree graph) **开始, 它是一个表示父母与子女关系的 DAG, 如图 10.6(a)。然后按下述方法把它转换成一个 DGM, 最后在得到的模型上做概率推断。

更具体地说: 对每个个体 (或动物) i 以及基因组上的每个位置 (位点) j , 我们创建三个结点:

- 一个**观测标记** X_{ij} (可以是血型这样的性质, 或某段可测的 DNA 片段) ;
- 两个**隐等位基因** G_{ij}^m 与 G_{ij}^p , 分别从 i 的母亲与父亲遗传而来。
有序对 $G_{ij} = (G_{ij}^m, G_{ij}^p)$ 就是个体 i 在位点 j 的**隐基因型**。

显然, 我们必须添加 $G_{ij}^m \rightarrow X_{ij}$ 与 $G_{ij}^p \rightarrow X_{ij}$ 的弧, 以表达“基因型**导致**表型 (基因型的可观测表现)”。CPD $p(X_{ij} | G_{ij}^m, G_{ij}^p)$ 称为**外显率模型** (penetrance model)。

一个非常简单的例子是: 令 $X_{ij} \in \{A, B, O, AB\}$ 表示血型, 且 $G_{ij}^m, G_{ij}^p \in \{A, B, O\}$ 表示基因型。我们可以用表 10.2 给出的**确定性** CPD 来表示外显率模型。例如, A 对 O 为显性, 因此若基因型是 AO 或 OA, 则表型为 A。

此外, 我们还要从个体 i 的母亲和父亲分别向 G_{ij} 加弧, 以反映孟德尔遗传: 子代从父母那儿拷贝等位基因。更精确地, 设 $m_i = k$ 是 i 的母亲, 则 G_{ij}^m 可能等于 G_{kj}^m 或 G_{kj}^p , 即 i 的母方等位基因是其母亲两等位基因之一的拷贝。令 Z_{ij}^m 为一个隐变量来指示这个选择。我们用下式 (称为**遗传模型**) 来建模:

$$p(G_{ij}^m | G_{kj}^m, G_{kj}^p, Z_{ij}^m) = \begin{cases} \mathbf{I}(G_{ij}^m = G_{kj}^m), & Z_{ij}^m = m \\ \mathbf{I}(G_{ij}^m = G_{kj}^p), & Z_{ij}^m = p \end{cases} \quad (10.13)$$

同理可定义 $p(G_{ij}^p | G_{kj}^m, G_{kj}^p, Z_{ij}^p)$, 其中 $k = p_i$ 是 i 的父亲。变量 Z_{ij} 的取值被称为给定位点的**相位** (phase)。四个变量 $G_{ij}^m, G_{ij}^p, Z_{ij}^m, Z_{ij}^p$ 一起构成个体 i 在位点 j 的**单倍型** (haplotype)。

接下来, 需要给根结点指定先验 $p(G_{ij}^m)$ 与 $p(G_{ij}^p)$ 。这叫作**创始者模型** (founder model), 表示群体中不同等位基因的总体流行率。通常假定这些创始等位基因在不同位点之间**相互独立**。

最后, 要为控制遗传过程的**开关变量**指定先验。这些变量在“空间” (沿基因组的位置) 上是**相关的**, 因为相邻位点往往一起遗传 (**重组**事件很少)。我们可以对 Z 施加一个**两状态**

马尔可夫链：在位点 j 处切换状态的概率为

$$\theta_j = \frac{1}{2}(1 - e^{-2d_j}),$$

其中 d_j 是位点 j 与 $j + 1$ 之间的距离。这称为**重组模型** (recombination model)。

得到的 DGM 如图 10.6(b)：它是一系列**重复的谱系 DAG**，并用马尔可夫链连接的开关变量 Z 进行增强。（一个相关模型叫**系统发育 HMM** (phylogenetic HMM) (Siepel & Haussler, 2003)，用于建模系统发育间的进化。）

作为一个简化示例，假设我们只有一个位点（血型）。为简洁起见省略下标 j 。若观测到 $x_i = A$ ，则可能的基因型有三种：(A, A), (A, O), (O, A)。这是因为“基因型→表型”的映射是**多对一的**。我们想要**反推**（求逆映射）。幸运的是，可以利用亲属的血型来消除这种歧义：信息会从其他人的 $x_{i'}$ 沿边“向上”传到他们的 $G_{i'}$ ，再通过谱系 DAG 传到个体 i 的 G_i 。因此我们可以把本人的局部证据 $p(x_i | G_i)$ 与“由其他数据提供的信息性先验” $p(G_i | x_{-i})$ 结合，得到一个**熵更小的后验**：

$$p(G_i | x) \propto p(x_i | G_i) p(G_i | x_{-i}).$$

在实际中，该模型用于推断**致病基因位于基因组的哪个位置**——这就是**遗传连锁分析**任务。方法如下：

首先，假设模型的所有参数（包括所有标记位点之间的距离）已知，唯一未知的是**致病基因的位置**。如果有 L 个标记位点，我们构造 $L + 1$ 个模型：在第 ℓ 个模型中，假设致病基因位于标记 ℓ 之后 ($0 < \ell < L + 1$)。我们可以估计该模型的马尔可夫切换参数 $\hat{\theta}_\ell$ ，从而得到致病基因与最近已知位点之间的距离 d 。用该模型的似然 $p(D | \hat{\theta}_\ell)$ 来度量其质量，然后选择似然最大的模型（在均匀先验下这等价于 MAP）。

但要计算似然，必须把所有隐变量 Z, G **积分消去**（求边缘化）。参见 Fishelson & Geiger (2002) 及其中引用的文献，有一些**精确方法**（基于我们在第 20.3 节将讨论的**变量消去算法**）。然而，正如我们在第 20.5 节解释的那样，当个体数和/或位点数很大时，精确方法在计算上可能不可行。参见 Albers 等 (2006) 提出的一种**近似似然计算方法**；它基于一种**变分推断**，我们将在第 22.4.1 节讨论。

通俗详解

1) 这在做什么？

- 有家族树 → 变成一个**概率图模型**：节点代表“这个人这个位点的等位基因/表型/是否从母还是父拷来的选择”，边代表“谁影响谁”。
- 目标：根据家族的观测（如血型或 DNA 标记），**推断隐藏的遗传状态，并进一步定位致病基因可能在什么位置**（连锁）。

2) 三类关键部件

1. 外显率模型 (基因型→表型)

- 例：血型由 $(G^m, G^p) \in \{A, B, O\}^2$ 决定；表 10.2 是确定性规则（如 A 显性于 O，AB 对应 A 与 B 同时存在）。

2. 遗传模型 (父母→子代等位基因)

- 子代的母方等位基因 G_{ij}^m 复制自母亲的两条之一；哪个被选用由开关变量 Z_{ij}^m 指示（父方类似）。

3. 重组模型 (跨位点相关)

- 相邻位点大概率不切换来源（一起遗传），偶尔切换（重组）。用两状态马尔可夫链建模，切换概率 $\theta_j = \frac{1}{2}(1 - e^{-2d_j})$ 与位点距离相关。

3) “创始者”与“单倍型”

- 家系最上层（没有父母）的个体称 **founder**，他们的等位基因从群体先验采样（**创始者模型**）。
- 单个位点的四个量 G^p, G^m, Z^p, Z^m 共同描述了**单倍型**；跨位点由 Z 的马尔可夫链把“来源相位”连起来。

4) 反推的直觉

- 表型往往不足以唯一确定基因型（多对一），但亲属的观测能提供强有力的先验：

$$\text{后验} \propto \underbrace{p(x_i | G_i)}_{\text{自己证据}} \times \underbrace{p(G_i | x_{-i})}_{\text{家族提供的先验}}.$$

- 因为图结构把“谁遗传谁”的路径连通起来，信息能在家族中传播、互相约束。

5) 连锁定位怎么做？

- 把“致病位点在标记 ℓ 之后”当作不同模型假设；对每个假设估计对应的切换率 $\hat{\theta}_\ell$ ，进而得到距离 d 。
- 计算每个假设的似然并选择最大者。
- 但这需要对大量隐变量求和/积分：小规模可用**变量消去**精确算；大规模需用**变分**等近似法。

要点总结

- 节点设置**：每人每位点有 (G^m, G^p) 两个隐等位基因、一个观测标记 X ，以及“复制自父/母哪条染色体”的开关 Z （根个体无需 Z ）。

- **三大 CPD**: 外显率 $p(X \mid G^m, G^p)$ (如表 10.2 的确定性规则)、遗传复制 $p(G_{\text{child}}^{m/p} \mid G_{\text{parent}}^m, G_{\text{parent}}^p, Z)$ 、重组的马尔可夫先验 $p(Z_j \mid Z_{j-1})$ 。
- **创始者模型**: 给根个体的等位基因设群体先验 (跨位点常假设独立)。
- **信息融合**: 利用家族其他成员的观测构造信息性先验, 提升对个体基因型的确定性。
- **连锁分析流程**: 为每个候选区间建模 → 估计切换率/距离 → 计算似然并选最大; 计算量大时用变分近似。
- **直觉**: DGM 把“谁把哪条染色体拷给了谁”与“基因型如何表现成表型”统一在一个可推断的概率框架里, 从而能**反推**隐含的遗传结构与致病位点位置。

请简要概括

下面是对“遗传连锁分析”这段内容的**精简概括**:

- **问题设定**: 用概率图模型把家系 (父母—子女关系) 转成可推断的模型, 以观测到的标记/表型来**反推隐含的基因型并定位致病基因**。
- **节点与边** (对每人 i 、每位点 j)
 - 观测: X_{ij} (如血型或 DNA 标记)。
 - 隐变量: 母/父等位基因 G_{ij}^m, G_{ij}^p ; 拷贝选择开关 Z_{ij}^m, Z_{ij}^p (指示来自父母哪条染色体)。
 - 结构: $G_{ij}^m, G_{ij}^p \rightarrow X_{ij}$ (**外显率模型**) ; 父母的等位基因通过 Z 复制到子代 (**遗传模型**)。创始者 (家系最上层) 用群体先验 $p(G)$ 。
- **外显率示例** (表 10.2) : 血型由 $(G^m, G^p) \in \{A, B, O\}^2$ **确定性映射**到 $\{A, B, O, AB\}$ (如 AO/ OA → A; AB → AB), 属于**多对一映射**。
- **跨位点相关 (重组模型)** : 相邻位点的 Z 用两状态马尔可夫链相连; 位点 j 的切换概率

$$\theta_j = \frac{1}{2}(1 - e^{-2d_j}) \quad (d_j \text{ 为位点间距离})$$
- **推断直觉**: 单人表型常**不能唯一确定**基因型; 利用亲属信息形成先验并结合自身证据:

$$p(G_i \mid x) \propto p(x_i \mid G_i) p(G_i \mid x_{-i})$$
- **连锁定位流程**: 对每个候选区间构建模型 (共 $L + 1$ 个), 估计切换率 $\hat{\theta}_\ell$ → 推得距离 d , 计算似然 $p(D \mid \hat{\theta}_\ell)$, 取最大者 (等价于 MAP)。需边缘化所有隐变量; 规模大时用**近似推断** (如变分法), 小规模可用**变量消去**精确计算。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

10.2.5 Directed Gaussian graphical models

Consider a DGM where all the variables are real-valued, and all the CPDs have the following form:

$$p(x_t|x_{pa}(t)) = N(x_t|\mu_t + w^T t x_{pa}(t), \sigma^2_t) \quad (10.14)$$

This is called a linear Gaussian CPD. As we show below, multiplying all these CPDs together results in a large joint Gaussian distribution of the form $p(x) = N(x|\mu, \Sigma)$. This is called a directed GGM, or a Gaussian Bayes net. We now explain how to derive μ and Σ from the CPD parameters, following (Shachter and Kenley 1989, App. B). For convenience, we will rewrite the CPDs in the following form:

$$x_t = \mu_t + s \in pa(t) wts(x_s - \mu_s) + \sigma_t z_t \quad (10.15)$$

where $z_t \sim N(0, 1)$, σ_t is the conditional standard deviation of x_t given its parents, wts is the strength of the $s \rightarrow t$ edge, and μ_t is the local mean.² It is easy to see that the global mean is just the concatenation of the local means, $\mu = (\mu_1, \dots, \mu_D)$. We now derive the global covariance, Σ . Let $S = \text{diag}(\sigma)$ be a diagonal matrix containing the standard deviations. We can rewrite Equation 10.15 in matrix-vector form as follows:

$$(x - \mu) = W(x - \mu) + Sz \quad (10.16)$$

Now let e be a vector of noise terms:

$$e = Sz \quad (10.17)$$

We can rearrange this to get

$$e = (I - W)(x - \mu) \quad (10.18)$$

Since W is lower triangular (because $wts = 0$ if $t > s$ in the topological ordering), we have that $I - W$ is lower triangular with 1s on the diagonal. Hence

$$\begin{aligned} (| | | | | e_1 e_2 \dots e_d | | | |) &= (| | | | | 1 -w_{11} 1 -w_{21} -w_{31} \\ &1 \dots \dots -w_{d1} -w_{d2} \dots -w_{d,d-1} | | | | |) (| | | | | x_1 - \mu_1 \\ &x_2 - \mu_2 \dots x_d - \mu_d | | | |) \end{aligned} \quad (10.19)$$

Since $I - W$ is always invertible, we can write

$$x - \mu = (I - W)^{-1} e = U e = USz \quad (10.20)$$

where we defined $U = (I - W)^{-1}$. Thus the regression weights correspond to a Cholesky decomposition of Σ , as we now show:

$$\Sigma = \text{cov} [x] = \text{cov} [x - \mu] \quad (10.21)$$

$$= \text{cov} [\mathbf{U}\mathbf{S}\mathbf{z}] = \mathbf{U}\mathbf{S} \text{cov} [\mathbf{z}] \mathbf{S}\mathbf{U}^T = \mathbf{U}\mathbf{S}^2\mathbf{U}^T \quad (10.22)$$

已思考 14s

下面先把原文逐句完整翻译（保留公式编号），再用通俗方式详细解释，最后给出要点总结。

翻译

10.2.5 有向高斯图模型 (Directed Gaussian graphical models)

考虑这样一个 DGM：所有变量都为实数，并且所有 CPD 都具有如下形式：

$$p(x_t \mid x_{\text{pa}(t)}) = N(x_t \mid \mu_t + \mathbf{w}_t^\top x_{\text{pa}(t)}, \sigma_t^2) \quad (10.14)$$

这被称为**线性高斯** CPD。如下所示，把这些 CPD 全部相乘，会得到一个大的联合高斯分布，形式为 $p(x) = N(x \mid \mu, \Sigma)$ 。这称为**有向 GGM**，或**高斯贝叶斯网**。下面我们按照 (Shachter and Kenley 1989, 附录 B) 解释如何由 CPD 的参数推出 μ 与 Σ 。

为方便起见，我们把 CPD 改写为：

$$x_t = \mu_t + \sum_{s \in \text{pa}(t)} w_{ts}(x_s - \mu_s) + \sigma_t z_t \quad (10.15)$$

其中 $z_t \sim N(0, 1)$ ， σ_t 是在给定父结点时 x_t 的条件标准差， w_{ts} 是边 $s \rightarrow t$ 的强度， μ_t 是**局部均值**。很容易看出，全局均值就是各局部均值的拼接： $\mu = (\mu_1, \dots, \mu_D)$ 。

现在推导全局协方差 Σ 。令 $S = \text{diag}(\sigma)$ 为装入这些标准差的对角矩阵。我们可把式 (10.15) 写成矩阵-向量形式：

$$(x - \mu) = W(x - \mu) + Sz \quad (10.16)$$

再令噪声向量

$$e = Sz \quad (10.17)$$

即可改写为

$$e = (I - W)(x - \mu). \quad (10.18)$$

由于在拓扑顺序下若 $t > s$ 则 $w_{ts} = 0$ ，矩阵 W 是**下三角**，因此 $I - W$ 也是下三角且对角线上全是 1。于是

$$\begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_d \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ -w_{21} & 1 & & & \\ -w_{31} & -w_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -w_{d1} & -w_{d2} & \cdots & -w_{d,d-1} & 1 \end{pmatrix} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \\ \vdots \\ x_d - \mu_d \end{pmatrix}. \quad (10.19)$$

因为 $I - W$ 总是可逆，我们有

$$x - \mu = (I - W)^{-1}e = Ue = USz \quad (10.20)$$

其中定义 $U = (I - W)^{-1}$ 。因此这些回归权重与 Σ 的 Cholesky 分解相对应，如下所示：

$$\Sigma = \text{cov}[x] = \text{cov}[x - \mu] \quad (10.21)$$

$$= \text{cov}[USz] = US \text{cov}[z] SU^\top = US^2 U^\top. \quad (10.22)$$

通俗解释

在每个结点做线性回归 + 高斯噪声。

式 (10.15) 说：每个变量 x_t 等于“自己的均值 μ_t ”+“父结点（先减去各自均值）按权重 w_{ts} 的线性组合”+“独立标准正态 z_t 乘以噪声尺度 σ_t ”。

- 写成“父变量的去均值形式 $(x_s - \mu_s)$ ”的好处是：直接可见 $E[x_t] = \mu_t$ ，因为 $E[z_t] = 0$ 、 $E[x_s - \mu_s] = 0$ 。所以整网的均值就是把这些局部均值拼起来： $\mu = (\mu_1, \dots, \mu_D)$ 。

把所有结点堆起来得到矩阵式。

- 令 W 是由边权 w_{ts} 组成的矩阵。因为是 DAG、按拓扑顺序编号，只有从“早”到“晚”的边存在，所以 W 是**严格下三角**。
- 把所有等式堆叠后得到 $(x - \mu) = W(x - \mu) + Sz$ 。把噪声写成 $e = Sz$ 后得到 $(I - W)(x - \mu) = e$ 。

为什么一定可解？

- $I - W$ 是下三角且对角为 1 \Rightarrow 一定可逆。用 $U = (I - W)^{-1}$ 可得 $x - \mu = Ue = USz$ 。

协方差怎么来？

- z 的各分量独立且方差 1 ($\text{cov}[z] = I$)，而 $e = Sz$ 使每一维噪声的方差变成 σ_t^2 。
- 把噪声通过“网络传递矩阵” U 传播到各结点，就得到整体的协方差：

$$\Sigma = \text{cov}[x] = US^2 U^\top.$$

设 $L = US$ ，则 $\Sigma = LL^\top$ 。这正是 Cholesky 分解（下三角因子 L 乘其转置）。

直观理解

- S 控制每个结点的**本地噪声大小**;
- $U = (I - W)^{-1}$ 把这些噪声沿着图的有向边“**传播/放大/混合**”到后续结点;
- 结果是全网联合为一个**多元正态**, 其相关性由“边权 w ”与“噪声尺度 σ ”共同决定。

一个 2 结点小例子 (帮助直觉)

拓扑: $1 \rightarrow 2$.

$$x_1 = \mu_1 + \sigma_1 z_1, \quad x_2 = \mu_2 + w_{21}(x_1 - \mu_1) + \sigma_2 z_2.$$

去均值: $\delta x_1 = \sigma_1 z_1$, $\delta x_2 = w_{21}\sigma_1 z_1 + \sigma_2 z_2$.

因此

$$\begin{aligned} \text{var}(x_1) &= \sigma_1^2, \\ \text{cov}(x_1, x_2) &= w_{21}\sigma_1^2, \\ \text{var}(x_2) &= w_{21}^2\sigma_1^2 + \sigma_2^2, \end{aligned}$$

这与 $\Sigma = US^2U^\top$ 完全一致 (此时 $U=\begin{pmatrix} 1 & 0 \\ w_{21} & 1 \end{pmatrix}$, $S = \text{diag}(\sigma_1, \sigma_2)$) 。

要点总结

- **线性高斯 CPD:**
 $x_t = \mu_t + \sum_{s \in \text{pa}(t)} w_{ts}(x_s - \mu_s) + \sigma_t z_t$, $z_t \sim N(0, 1)$.
- **联合分布:** 把所有 CPD 相乘 \Rightarrow 全局仍是**多元高斯** $N(\mu, \Sigma)$ 。
 全局均值: $\mu = (\mu_1, \dots, \mu_D)$.
- **矩阵形式与可逆性:** $(x - \mu) = W(x - \mu) + Sz$, W 严格下三角 $\Rightarrow I - W$ 可逆。
 $x - \mu = USz$, 其中 $U = (I - W)^{-1}$.
- **协方差:** $\Sigma = US^2U^\top$ 。令 $L = US$ 得 $\Sigma = LL^\top$ (**Cholesky 分解**) 。
 解释: σ_t 控制本地噪声, w_{ts} 控制噪声如何沿图传播并形成相关性。
- **直觉:** 这是把“每个结点线性回归其父 + 独立高斯噪声”的网络堆叠起来; DAG 确保可以从前到后求解, 最终得到一个整体的高斯模型。

我不太理解CPD和CPT

已思考 21s

这两个词很容易混:

- **CPD** (Conditional Probability Distribution / Density) = **条件概率分布/密度**。
就是“在给定父节点后，这个节点的概率规律”。记作 $p(X \mid \text{pa}(X))$ 。它可以是**任何形式**：表格、公式、回归模型、Noisy-OR、线性高斯……都算 CPD。
 - **CPT** (Conditional Probability Table) = **条件概率表**。
这是 CPD 的一种具体表示方法，只适用于离散变量：把“父节点的每种取值组合”逐行列出来，每行给一个对 X 的**离散分布**（各列相加为 1）。本质就是“查表”。
换句话说： $\text{CPT} \subset \text{CPD}$ 。所有 CPT 都是 CPD，但 CPD 不一定是 CPT（可能是公式/回归/高斯等）。
-

小例子：一个 CPT 长什么样？

设父节点 $Y \in \{\text{晴, 雨}\}$ ，子节点 $X \in \{\text{带伞, 不带}\}$ 。CPT 可以写成：

Y	$p(X = \text{带伞} \mid Y)$	$p(X = \text{不带} \mid Y)$
晴	0.1	0.9
雨	0.9	0.1

每行都加和为 1。这里的整个表就是 $p(X \mid Y)$ 的 CPD；因为是表格，所以它同时也是 CPT。

一般地：若 X 有 r 个取值，父节点的所有组合有 q 种，那么一张 CPT 需要 $q \times (r - 1)$ 个**自由参数**（每行少 1 个，因为要和为 1）。

不是表格的 CPD (几个常见形式)

- **逻辑回归 (离散子节点)**：
$$p(X = 1 \mid \text{pa}(X)) = \sigma(w^\top \text{pa}(X)).$$

好处：参数量随父节点数**线性**增长，适合父很多时；坏处：假设了特定函数形状。
- **Noisy-OR (离散子节点)**：
$$p(X = 1 \mid \mathbf{h}) = 1 - \prod_{s \in \text{pa}(X)} \theta_s^{h_s}$$
 (可加泄漏项)。
常用于“多原因触发一个症状”的场景。
- **线性高斯 (连续子节点)**：
$$p(X \mid \text{pa}(X)) = N(\mu + w^\top \text{pa}(X), \sigma^2).$$

适合连续变量，整网合起来仍是多元高斯。
- **确定性映射**：
如“基因型 → 血型”的表 (AO 或 OA 都映成 A)。这是 CPD 的一种极端：每行概率要么 0 要么 1；也可用 CPT 表示。

什么时候用 CPT，什么时候用“公式型” CPD？

- **用 CPT (查表)**
 - 变量都离散；
 - 父节点个数少、每个取值也少（表不会太大）；
 - 需要**完全灵活、最好解释**（每个条件下的概率一目了然）。
 - 缺点：父节点多时表大小随组合数**指数级膨胀**，数据需求大。
- **用公式/参数化 CPD (回归、Noisy-OR、线性高斯等)**
 - 父节点多或连续；
 - 想**节省参数、更好地泛化**；
 - 缺点：有模型假设，可能与真实分布不匹配。

和整张图的关系

在贝叶斯网 (DGM) 里，联合分布按节点相乘：

$$p(x_1, \dots, x_V) = \prod_{t=1}^V p(x_t \mid \text{pa}(x_t)).$$

这里每一项 $p(x_t \mid \text{pa}(x_t))$ 就是一个 CPD；

如果它是张表，那就是CPT；如果是回归/高斯/Noisy-OR，就是**非表格的** CPD。

一句话记忆

- CPD：条件概率的“**内容**”（分布本身，表也好、公式也好）。
- CPT：CPD 的一种“**表现形式**”（Table），只对离散变量，用查表法列出所有条件下的概率。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

10.3 Inference

We have seen that graphical models provide a compact way to define joint probability distributions. Given such a

joint distribution, what can we do with it? The main use for such a joint distribution is to perform probabilistic inference. This refers to the task of estimating unknown quantities from known quantities. For example, in Section 10.2.2, we introduced HMMs, and said that one of the goals is to estimate the hidden states (e.g., words) from the observations (e.g., speech signal). And in Section 10.2.4, we discussed genetic linkage analysis, and said that one of the goals is to estimate the likelihood of the data under various DAGs, corresponding to different hypotheses about the location of the disease-causing gene. In general, we can pose the inference problem as follows. Suppose we have a set of correlated random variables with joint distribution $p(x_1:V|\theta)$. (In this section, we are assuming the parameters θ of the model are known. We discuss how to learn the parameters in Section 10.4.) Let us partition this vector into the visible variables x_v , which are observed, and the hidden variables, x_h , which are unobserved. Inference refers to computing the posterior distribution of the unknowns given the knowns:

$$p(x_h|x_v, \theta) = p(x_h, x_v|\theta) / p(x_v|\theta) = p(x_h, x_v|\theta) / \sum_h p(x_h, x_v|\theta) \quad (10.23)$$

Essentially we are conditioning on the data by clamping the visible variables to their observed values, x_v , and then normalizing, to go from $p(x_h, x_v)$ to $p(x_h|x_v)$. The normalization constant $p(x_v|\theta)$ is the likelihood of the data, also called the probability of the evidence.

Sometimes only some of the hidden variables are of interest to us. So let us partition the hidden variables into query variables, x_q , whose value we wish to know, and the remaining nuisance variables, x_n , which we are not interested in. We can compute what we are interested in by marginalizing out the nuisance variables:

$$p(x_q|x_v, \theta) = \sum_{x_n} p(x_q, x_n|x_v, \theta) \quad (10.24)$$

In Section 4.3.1, we saw how to perform all these operations for a multivariate Gaussian in $O(V^3)$ time, where V is the number of variables. What if we have discrete random variables, with say K states each? If the joint distribution is represented as a multi-dimensional table, we can always perform these operations exactly, but this will take $O(KV)$ time. In Chapter 20, we explain how to exploit the factorization encoded by the GM to perform these operations in $O(VKw+1)$ time, where w is a quantity

known as the treewidth of the graph. This measures how “tree-like” the graph is. If the graph is a tree (or a chain), we have $w = 1$, so for these models, inference takes time linear in the number of nodes. Unfortunately, for more general graphs, exact inference can take time exponential in the number of nodes, as we explain in Section 20.5. We will therefore examine various approximate inference schemes later in the book.

好的！先把原文逐句完整翻译（保留公式与编号），再给出通俗解释，最后要点总结。

翻译

10.3 推断 (Inference)

我们已经看到，图模型为定义联合概率分布提供了一种紧凑的方法。给定这样一个联合分布，我们能做什么？这种联合分布的主要用途就是进行**概率推断**。这指的是：从已知量中估计未知量的任务。

例如，在第 10.2.2 节，我们介绍了 HMM，并说其目标之一是**从观测（如语音信号）估计隐状态（如单词）**。又如在第 10.2.4 节，我们讨论了遗传连锁分析，并说其目标之一是在**不同 DAG（对应于关于致病基因位置的不同假设）下估计数据的似然**。

一般而言，我们可以如下表述推断问题。假设我们有一组相关随机变量，其联合分布为 $p(x_{1:V} | \theta)$ 。（在本节中，我们假设模型的参数 θ 已知。如何学习参数见第 10.4 节。）把该向量划分为**可见变量 x_v （已观测）和**隐变量 x_h （未观测）****。推断指的是**在给定已知量的条件下计算未知量的后验分布**：

$$p(x_h | x_v, \theta) = \frac{p(x_h, x_v | \theta)}{p(x_v | \theta)} = \frac{p(x_h, x_v | \theta)}{\sum_{x_h} p(x_h, x_v | \theta)} \quad (10.23)$$

本质上，我们通过把可见变量“夹定”（clamping）为其观测值 x_v ，然后归一化，把 $p(x_h, x_v)$ 变为 $p(x_h | x_v)$ 。归一化常数 $p(x_v | \theta)$ 就是数据的**似然**，也称为**证据的概率**（probability of the evidence）。

有时我们只关心部分隐变量。于是把隐变量再划分为**查询变量 x_q （我们想知道的）与其余的杂散变量 x_n （不感兴趣）**。我们可以通过**边缘化**掉杂散变量来得到所需结果：

$$p(x_q | x_v, \theta) = \sum_{x_n} p(x_q, x_n | x_v, \theta). \quad (10.24)$$

在第 4.3.1 节，我们看到对一个**多元高斯**执行上述操作的时间是 $O(V^3)$ ，其中 V 为变量个数。

如果是**离散**随机变量（例如每个都有 K 个状态）怎么办？若把联合分布表示为一个多维表，我们总能精确地执行这些操作，但代价是 $O(K^V)$ 时间。

在第 20 章，我们解释如何**利用图模型编码的分解**在 $O(VK^{w+1})$ 时间内完成这些操作，其中 w 称为图的**树宽** (treewidth)，它度量图形“像树”的程度。若图是一棵树（或一条链），则 $w = 1$ ，因此对这些模型，推断的时间对结点数是**线性的**。

不幸的是，对更一般的图，如第 20.5 节所述，**精确推断**的时间可能对结点数呈**指数增长**。因此我们将在本书后面考察各种**近似推断方法**。

通俗解释

推断 = 用模型回答问题。

我们先用图模型把“所有变量一起的分布” $p(x_{1:V})$ 描述好。推断的典型问题有两类：

1. **给定观测，求未知的后验** $p(x_h \mid x_v)$ ：

- 把已观测的变量 x_v 固定（夹定）为事实；
- 只剩 x_h 不确定；
- 用“联合 / 证据”公式 (10.23) 做**归一化**得到后验。
- 证据 $p(x_v)$ 既是“这份数据在模型下有多可能”，又是很多学习算法里的关键量。

2. **只关心其中一部分未知** x_q ：

- 把不关心的 x_n **求和（边缘化）掉** (10.24)，就得到我们想要的分布。

计算代价为什么会差这么多？

- 如果什么结构信息都不用，离散情形需要遍历“所有变量的所有取值组合”——**指教级** $O(K^V)$ 。
- 多元高斯可用线性代数（求逆/分解）做推断，代价 $O(V^3)$ （矩阵操作的典型复杂度）。
- 图模型的威力：把联合分布**因子分解**到每个结点的 CPD 上。如果图像“树”（树宽 $w = 1$ ），可以用**消息传递**在线性时间里完成推断；一般图的代价是 $O(VK^{w+1})$ ——只与**局部团大小**（由树宽决定）**指数相关**，而不是对整个图指数。
- 但当图很“稠密”、树宽大时，**精确推断**仍可能不可承受，这时才需要**近似推断**（如采样、变分等）。

小比喻

- 想象拼图：联合分布是完整拼图；观测到的碎片 x_v 被固定到桌面；我们要推断剩下缺口 x_h 如何填。
- 如果拼图是“树状”分块，消息沿树传一次就行；如果块与块之间错综复杂（树宽大），就得付出更多计算，甚至只能“估着拼”（近似）。

要点总结

- **推断的定义**: 给定联合分布和观测, 求未知的**后验** $p(x_h \mid x_v)$ (式 10.23) ; 若只关心部分未知 x_q , 则对其余 x_n **边缘化** (式 10.24) 。
- **证据/似然**: 归一化常数 $p(x_v \mid \theta)$ 是“数据在模型下的概率”, 也是模型比较与学习的重要量。
- **复杂度对比**:
 - 多元高斯: $O(V^3)$;
 - 朴素的离散联合表: $O(K^V)$ (指数) ;
 - 利用图分解: $O(VK^{w+1})$, 其中 **树宽** w 衡量图“像树”的程度; 树/链有 $w = 1 \Rightarrow$ 线性时间。
- **现实结论**: 复杂图上精确推断常指数困难 \Rightarrow 需要**近似推断** (后续章节介绍) 。
- **大图思维**: 结构 (因子分解、树宽) 决定了推断是否高效; 把“谁依赖谁”画清楚, 就是图模型的价值所在。

DAG是什么

已思考 12s

DAG 是什么?

DAG 是 *Directed Acyclic Graph* 的缩写, 中文叫**有向无环图**。

一句话定义

- **有向** (Directed) : 每条边都有方向, 用箭头表示 ($s \rightarrow t$) 。
- **无环** (Acyclic) : **不存在**沿箭头走一圈又回到起点的**有向环** (如 $A \rightarrow B \rightarrow C \rightarrow A$ 这种不允许) 。

等价理解 (很实用)

- 只有当图里**能做拓扑排序**时, 它才是 DAG: 也就是能把所有结点排成一个顺序, 使**每条边都从“前面的结点”指向“后面的结点”**。
- 树 (把边定个方向) 一定是 DAG; DAG 可以比树更一般 (一个结点可有多个父) 。

在概率图模型里的意义

- 在贝叶斯网 (有向图模型, DGM) 中:
 - 结点 = 随机变量; 箭头 = 直接依赖关系; **没有有向环** \rightarrow 不会出现“自我循环定义”。

- 关键分解式（因子分解）：

$$p(x_1, \dots, x_V) = \prod_{t=1}^V p(x_t | x_{\text{pa}(t)}),$$

其中 $\text{pa}(t)$ 是结点 t 的父结点集合。

这使得建模、推断、学习更高效。

小图感受

- 链: $A \rightarrow B \rightarrow C$ (DAG ✓)
- 叉: $A \rightarrow C, B \rightarrow C$ (DAG ✓, 又叫“V 结构”)
- 环: $A \rightarrow B \rightarrow C \rightarrow A$ (不是 DAG ✗)

注意：无向环（把箭头忽略后形成的圈）可以存在；DAG 只禁止有向环。

常用术语（快速对照）

- 父母 (parents)：指向该结点的前驱。
- 子 (children)：由该结点指向的后继。
- 根 (root)：没有父的结点。
- 叶 (leaf)：没有子的结点。
- 祖先/后代：多步父系/子系结点。
- 拓扑序 (topological order)：父在前、子在后的编号顺序。

其他领域也常见

- 任务/课程依赖调度（先修课 → 后修课）。
- 版本控制（Git 的提交历史是一个 DAG）。
- 计算图/数据流（深度学习的前向/反向图）。

要点小结

- DAG = 有向 + 无有向环；可做拓扑排序。
- 在贝叶斯网里，DAG 保证联合分布可写成父子因子分解，避免“循环依赖”。
- 树是 DAG 的特例；DAG 更灵活，能表达更复杂的依赖结构。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 10.7 Left: data points x_i are conditionally independent given θ . Right: Plate notation. This represents the same model as the one on the left, except the repeated x_i nodes are inside a box, known as a plate; the number in the lower right hand corner, N , specifies the number of repetitions of the X_i node.

Figure 10.8 Naive Bayes classifier as a DGM. (a) With single plates. (b) With nested plates.

10.4 Learning

In the graphical models literature, it is common to distinguish between inference and learning. Inference means computing (functions of) $p(x_h|x_v, \theta)$, where v are the visible nodes, h are the hidden nodes, and θ are the parameters of the model, assumed to be known. Learning usually means computing a MAP estimate of the parameters given data:

$$\hat{\theta} = \operatorname{argmax}_{\theta} \sum_{i=1}^N \log p(x_i, v_i | \theta) + \log p(\theta) \quad (10.25)$$

where x_i, v_i are the visible variables in case i . If we have a uniform prior, $p(\theta) \propto 1$, this reduces to the MLE, as usual. If we adopt a Bayesian view, the parameters are unknown variables and should also be inferred. Thus to a Bayesian, there is no distinction between inference and learning. In fact, we can just add the parameters as nodes to the graph, condition on D , and then infer the values of all the nodes. (We discuss this in more detail below.) In this view, the main difference between hidden variables and parameters is that the number of hidden variables grows with the amount of training data (since there is usually a set of hidden variables for each observed data case), whereas the number of parameters is usually fixed (at least in a parametric model). This means that we must

integrate out the hidden variables to avoid overfitting, but we may be able to get away with point estimation techniques for parameters, which are fewer in number.

10.4.1 Plate notation

When inferring parameters from data, we often assume the data is iid. We can represent this assumption explicitly using a graphical model, as shown in Figure 10.7(a). This illustrates the assumption that each data case was generated independently but from the same distribution. Notice that the data cases are only independent conditional on the parameters θ ; marginally, the data cases are dependent. Nevertheless, we can see that, in this example, the order in which the data cases arrive makes no difference to our beliefs about θ , since all orderings will have the same sufficient statistics. Hence we say the data is exchangeable.

To avoid visual clutter, it is common to use a form of syntactic sugar called plates: we simply draw a little box around the repeated variables, with the convention that nodes within the box will get repeated when the model is unrolled. We often write the number of copies or repetitions in the bottom right corner of the box. See Figure 10.7(b) for a simple example. The corresponding joint distribution has the form

$$p(\theta, D) = p(\theta) \prod_{i=1}^N p(x_i | \theta) \quad (10.26)$$

This DGM represents the CI assumptions behind the models we considered in Chapter 5.

A slightly more complex example is shown in Figure 10.8. On the left we show a naive Bayes classifier that has been "unrolled" for D features, but uses a plate to represent repetition over cases $i = 1: N$. The version on the right shows the same model using nested plate notation. When a variable is inside two plates, it will have two sub-indices. For example, we write θ_{jc} to represent the parameter for feature j in class-conditional density c . Note that plates can be nested or crossing. Notational devices for modeling more complex parameter tying patterns can be devised (e.g., (Heckerman et al. 2004)), but these are not widely used. What is not clear from the figure is that θ_{jc} is used to generate x_{ij} iff $y_i = c$, otherwise it is ignored. This is an example of context specific independence, since the CI relationship $x_{ij} \perp \theta_{jc}$ only holds if $y_i = c$.

下面先把原文逐句完整翻译（含公式与图注，保留编号），再给出通俗解释，最后要点总结。

翻译

图 10.7 左：在给定 θ 的条件下，数据点 x_i 相互条件独立。右：**板式记号 (plate notation)**。它表示与左图相同的模型，只是把重复出现的 x_i 结点放在一个称为“板 (plate)”的方框里；方框右下角的数字 N 指出 X_i 这个结点重复的次数。

图 10.8 作为 DGM 的朴素贝叶斯分类器。**(a)** 使用单层 plate。**(b)** 使用嵌套的 plates。

10.4 学习 (Learning)

在图模型文献中，通常把**推断 (inference)** 和**学习 (learning)** 区分开来。

推断指计算 $p(x_h \mid x_v, \theta)$ (**或其函数**)，其中 v 是**可见结点**、 h 是**隐结点**， θ 是已知的模型参数。

学习通常是指：在给定数据的情况下计算参数的 MAP 估计：

$$\hat{\theta} = \arg \max_{\theta} \sum_{i=1}^N \log p(x_{i,v} \mid \theta) + \log p(\theta) \quad (10.25)$$

其中 $x_{i,v}$ 是第 i 个样本的可见变量。如果先验是均匀的 $p(\theta) \propto 1$ ，这就退化为**极大似然估计 (MLE)**。

如果采取**贝叶斯视角**，参数本身也是未知变量，也应该被推断。因此，对贝叶斯派而言，推断与学习并无本质差别：我们只需把参数当作图中的结点，给定数据 D 后，对所有结点一起做推断（下面会更详细讨论）。在这种观点下，**隐变量与参数**的主要区别在于：隐变量的数量**随训练数据量增长**（通常每条观测会带一组隐变量），而参数的数量通常是**固定的**（至少在参数模型中）。这意味着：为了避免过拟合，我们必须**积分消去 (边缘化)** 隐变量；而参数的数量较少，往往可以用**点估计**（如 MAP/MLE）来应对。

10.4.1 板式记号 (Plate notation)

从数据中推断参数时，我们经常假设数据是 iid。可以像图 10.7(a) 那样用图模型**显式**表示这个假设：每条数据是**独立生成的**，但来自**同一个分布**。注意：数据样本**仅在给定参数 θ** 的条件下彼此独立；**边缘意义上**（把 θ 积分掉）它们是相关的。尽管如此，在该例中，数据到达的顺序**不会改变**我们对 θ 的看法，因为任何顺序下形成的**充分统计量相同**——因此我们说数据是**可交换的 (exchangeable)**。

为避免图上画太多重复结点，通常使用一种“语法糖”——plate：把要重复的结点用一个小方框圈起来，表示模型**展开（unroll）**时，方框里的结点会被多次拷贝。我们常在板的右下角写上重复的次数。见图 10.7(b) 的简单示例。其对应的联合分布为

$$p(\theta, D) = p(\theta) \prod_{i=1}^N p(x_i | \theta). \quad (10.26)$$

这个 DGM 表示了我们在第 5 章所讨论模型背后的条件独立假设。

稍微复杂一点的例子见图 10.8。左图是把朴素贝叶斯分类器对特征数 D “展开”，并用一个 plate 表示样本索引 $i = 1:N$ 的重复。右图是同一模型的嵌套 plate 写法。某个变量同时位于两个 plate 内时，它会有两个下标；例如 θ_{jc} 表示类条件密度 c 下第 j 个特征的参数。注意 plate 可以嵌套也可以相交。为了表达更复杂的“参数共享（parameter tying）”模式，还可以设计其他记号（如 Heckerman 等，2004），但使用并不广泛。图中不明显的一点是： θ_{jc} 只有在 $y_i = c$ 时才用来生成 x_{ij} ，否则被忽略。这是**情境特异独立（context specific independence）**的例子，因为 CI 关系 $x_{ij} \perp \theta_{jc}$ 只有在 $y_i = c$ 时才成立。

通俗解释

1) 推断 vs. 学习 (直白版)

- **推断**：参数 θ 当已知；给我观测 x_v ，我要算未知 x_h 的分布（比如 HMM 里估计隐状态）。
- **学习**：参数也未知；给我一堆数据 D ，我要估计 θ (MAP 或 MLE)。
- **贝叶斯想法**：别把 θ 当常数，把它当“图上的变量”。那“学习”就是“推断”在更大的图上做一次而已。

2) 为什么要有 plate?

- 同一个结构**重复很多次**（每个样本、每个特征、每个时间步……）时，如果每次都画一个结点，图会乱成一团。
- Plate 把“重复发生”的部分圈起来，并在角上标明重复次数。展开后就是左图那样的多个拷贝。
- 联合分布就会写成“先验 \times 每次重复的条件概率的乘积”（式 10.26）。

3) iid 与可交换

- iid：独立、同分布；在图里体现为每个 x_i 都从同一个 θ 出发生成，且 x_i 之间没有边。
- 但要注意：只在**条件下** θ 时才独立；把 θ 积掉后，样本之间通常会有关联。

- **可交换**: 换个数据顺序也不影响对 θ 的看法 (充分统计量相同)。

4) 朴素贝叶斯的两种 plate 画法 (图 10.8)

- **单层 plate (左)** : 外层 plate 循环样本 i ; 在每个样本里, 类别 Y_i 生成 D 个特征 X_{i1}, \dots, X_{iD} 。
- **嵌套 plate (右)** : 再加一层遍历特征 j 的 plate, 于是参数写成 θ_{jc} ("第 j 个特征、类别 c ")。
- **情境特异独立**: 只有当 $y_i = c$ 时, θ_{jc} 与 x_{ij} 相关; 否则 x_{ij} 与 θ_{jc} 无关。

5) 实操提示

- 学习 θ 时:
 - **频率学派**: 求 $\arg \max_{\theta} \sum_i \log p(x_i | \theta)$ (或加上 $\log p(\theta)$ 做 MAP)。
 - **贝叶斯**: 把 θ 放到图里, 指定先验 $p(\theta)$, 和数据一起做**后验推断**。
 - 隐变量多、每个样本都有一套 \Rightarrow 常用EM、**变分或采样**来同时处理“积分隐变量”和“估计参数”。
-

要点总结

- **推断**: 已知 θ , 算 $p(x_h | x_v, \theta)$ 。
- **学习**: 未知 θ , 根据数据估计 (MAP/MLE); 贝叶斯把学习视作“把 θ 也纳入推断”。
- **Plate 记号**: 用方框表示**重复的子结构**, 右下角写重复次数; 联合分布形如 $p(\theta) \prod_i p(x_i | \theta)$ 。
- **iid 与可交换**: 样本在给定 θ 时独立同分布; 顺序不影响对 θ 的信念。
- **朴素贝叶斯的 plate**: 可画成单层或嵌套; θ_{jc} 只在 $y_i = c$ 时用于生成 x_{ij} ——体现**情境特异独立**。
- **隐变量 vs. 参数**: 隐变量数量随数据增长, 要**边缘化**; 参数通常固定, 常用**点估计**即可 (或做完整贝叶斯)。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

10.4.2 Learning from complete data

If all the variables are fully observed in each case, so there is no missing data and there are no hidden variables, we say the data is complete. For a DGM with complete data, the likelihood is given by

$$p(D|\theta) = \prod_{i=1}^N p(x_i|\theta) = \prod_{i=1}^N \prod_{t=1}^T p(x_{it}|x_i, \theta_t) = \prod_{t=1}^T p(D_t|\theta_t) \quad (10.27)$$

where D_t is the data associated with node t and its parents, i.e., the t 'th family. This is a product of terms, one per CPD. We say that the likelihood decomposes according to the graph structure. Now suppose that the prior factorizes as well:

$$p(\theta) = \prod_{t=1}^T p(\theta_t) \quad (10.28)$$

Then clearly the posterior also factorizes:

$$p(\theta|D) \propto p(D|\theta)p(\theta) = \prod_{t=1}^T p(D_t|\theta_t)p(\theta_t) \quad (10.29)$$

This means we can compute the posterior of each CPD independently. In other words, factored prior plus factored likelihood implies factored posterior (10.30)

Let us consider an example, where all CPDs are tabular, thus extending the earlier results of Section 3.5.1.2, where discussed Bayesian naive Bayes. We have a separate row (i.e., a separate multinoulli distribution) for each conditioning case, i.e., for each combination of parent values, as in Table 10.2. Formally, we can write the t 'th CPT as $x_t|x_{pa(t)} = c \sim \text{Cat}(\theta_{tc})$, where $\theta_{tc} = p(x_t = k|x_{pa(t)} = c)$, for $k = 1: K_t$, $c = 1: C_t$ and $t = 1: T$. Here K_t is the number of states for node t , $C_t = |\{s \in pa(t)\}|$, K_s is the number of parent combinations, and T is the number of nodes.

Obviously $\sum_k \theta_{tc} = 1$ for each row of each CPT. Let us put a separate Dirichlet prior on each row of each CPT, i.e., $\theta_{tc} \sim \text{Dir}(\alpha_{tc})$. Then we can compute the posterior by simply adding the pseudo counts to the empirical counts to get $\theta_{tc|D} \sim \text{Dir}(N_{tc} + \alpha_{tc})$, where N_{tc} is the number of times that node t is in state k while its parents are in state c :

$$N_{tc} = \sum_i I(x_{it} = k, x_{i,pa(t)} = c) \quad (10.31)$$

From Equation 2.77, the mean of this distribution is given by the following:

$$\theta_{tc} = N_{tc} + \alpha_{tc} \frac{k}{N_{tc} + \alpha_{tc}} \quad (10.32)$$

For example, consider the DGM in Figure 10.1(a). Suppose

the training data consists of the following 5 cases:

Below we list all the sufficient statistics N_{tck} , and the posterior mean parameters θ_{ick} under a Dirichlet prior with $\alpha_{ick} = 1$ (corresponding to add-one smoothing) for the $t = 4$ node:

It is easy to show that the MLE has the same form as Equation 10.32, except without the α_{tck} terms, i.e.,

$$\hat{\theta}_{tck} = N_{tck} / \sum N_{tck} \quad (10.33)$$

Of course, the MLE suffers from the zero-count problem discussed in Section 3.3.4.1, so it is important to use a prior to regularize the estimation problem.

10.4.3 Learning with missing and/or latent variables

If we have missing data and/or hidden variables, the likelihood no longer factorizes, and indeed it is no longer convex, as we explain in detail in Section 11.3. This means we will usually can only compute a locally optimal ML or MAP estimate. Bayesian inference of the parameters is even harder. We discuss suitable approximate inference techniques in later chapters.

已思考 21s

下面先把原文逐句完整翻译（含公式与编号），再给出通俗解释与表格示例算一遍，最后总结要点。

翻译

10.4.2 用完整数据学习 (Learning from complete data)

如果每个样本中的所有变量都被完全观测（既没有缺失值也没有隐变量），我们称数据是完整的（complete）。对一个具有完整数据的 DGM，其似然为

$$p(D \mid \theta) = \prod_{i=1}^N p(x_i \mid \theta) = \prod_{i=1}^N \prod_{t=1}^{V_i} p(x_{it} \mid x_{i,\text{pa}(t)}, \theta_t) = \prod_{t=1}^V p(D_t \mid \theta_t) \quad (10.27)$$

其中 D_t 是与结点 t 及其父结点相关的那部分数据（即第 t 个“家族”）。这是每个 CPD 各出一项的乘积；我们说似然按图结构分解。

现在再假设先验也能分解：

$$p(\theta) = \prod_{t=1}^V p(\theta_t) \quad (10.28)$$

则显然后验也分解：

$$p(\theta | D) \propto p(D | \theta) p(\theta) = \prod_{t=1}^V p(D_t | \theta_t) p(\theta_t) \quad (10.29)$$

这意味着我们可以**分别**计算每个 CPD 的后验。换句话说：

$$\text{“先验可分解” + “似然可分解”} \Rightarrow \text{“后验可分解”。} \quad (10.30)$$

看一个例子：所有 CPD 用**表格 (CPT) **表示（扩展第 3.5.1.2 节的贝叶斯朴素贝叶斯结果）。对每个“父取值组合”各有一行（即一个多项分布）。形式化地，第 t 个 CPT 可写为

$$x_t | x_{\text{pa}(t)} = c \sim \text{Cat}(\theta_{tc}), \quad \theta_{tck} = p(x_t = k | x_{\text{pa}(t)} = c),$$

其中 $k = 1:K_t$, $c = 1:C_t$, $t = 1:T$ 。这里 K_t 是结点 t 的状态数, $C_t = \prod_{s \in \text{pa}(t)} K_s$ 是父组合数, T 是结点数。显然每行满足 $\sum_k \theta_{tck} = 1$ 。

对每个 CPT 的每一行放一个**狄利克雷先验**: $\theta_{tc} \sim \text{Dir}(\alpha_{tc})$ 。则把**伪计数加到经验计数上**即可得到后验：

$$\theta_{tc} | D \sim \text{Dir}(N_{tc} + \alpha_{tc}),$$

其中 N_{tck} 是在“父为 c ”时结点 t 取值为 k 的次数：

$$N_{tck} = \sum_{i=1}^N \mathbf{1}(x_{i,t} = k, x_{i,\text{pa}(t)} = c). \quad (10.31)$$

由式 (2.77)，其**后验均值**为

$$\bar{\theta}_{tck} = \frac{N_{tck} + \alpha_{tck}}{\sum_k (N_{tck} + \alpha_{tck})}. \quad (10.32)$$

例如，考虑图 10.1(a) 的 DGM。设训练集为下表 5 个样本 (0/1)：

x_1	x_2	x_3	x_4	x_5
0	0	1	0	0
0	1	1	1	1
1	1	0	1	0
0	1	1	0	0
0	1	1	1	0

下面列出结点 $t = 4$ (其父为 x_2, x_3) 的所有充分统计量 N_{tck} , 以及在 $\alpha_{ick} = 1$ (加一平滑) 时的后验均值 $\bar{\theta}_{ick}$ (二元: $k = 1$ 表示 $x_4 = 1$, $k = 0$ 表示 $x_4 = 0$) :

x_2	x_3	$N_{tck=1}$	$N_{tck=0}$	$\bar{\theta}_{tck=1}$	$\bar{\theta}_{tck=0}$
0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
1	0	1	0	$\frac{2}{3}$	$\frac{1}{3}$
0	1	0	1	$\frac{1}{3}$	$\frac{2}{3}$
1	1	2	1	$\frac{3}{5}$	$\frac{2}{5}$

很容易证明, 极大似然估计 (MLE) 与式 (10.32) 同形, 只是去掉 α :

$$\hat{\theta}_{tck} = \frac{N_{tck}}{\sum_k N_{tck}}. \quad (10.33)$$

当然, MLE 会遇到第 3.3.4.1 节讨论的零计数问题, 因此使用先验来正则化很重要。

10.4.3 有缺失和/或潜在变量时的学习

若存在缺失数据和/或隐变量, 似然就不再能分解, 而且 (如 11.3 节详述) 通常也非凸。这意味着我们往往只能得到局部最优的 ML/MAP 估计; 做完全的贝叶斯参数推断更难。本书后续章节将讨论相应的近似推断技术。

通俗解释

- 完整数据时: 每个结点的参数只跟它的“家族数据”有关 \rightarrow 整体似然是所有 CPD 的乘积; 如果先验也按结点独立, 后验就能逐结点独立计算 (式 10.27–10.30)。
- 离散表格 (CPT): 对每个“父配置”的一行就是一个多项分布; 放上狄利克雷先验后, 后验=先验参数 + 计数, 后验均值就是“加平滑的频率” (式 10.32)。
- 例子怎么数 (结点 4 的父是 x_2, x_3):
 - 看所有样本里 (x_2, x_3) 的 4 种组合;
 - 数对应的 $x_4 = 1/0$ 次数 $\rightarrow N_{tck}$;
 - 加上 $\alpha = 1$ 做加一平滑 \rightarrow 得到后验均值 $\bar{\theta}$ (表中分数);
 - 若用 MLE 就是直接用频率 (式 10.33), 但没出现过的组合会给 0 概率 \rightarrow 需要先验来避免。
- 有缺失/隐变量时: 样本不再“独立给各个 CPD 贡献计数”, 似然纠缠在一起且可能有多个局部极值 \rightarrow 常用 EM、变分、采样 等近似方法。

要点总结

- 完整数据 + 图结构 \Rightarrow 似然分解；若先验也分解 \Rightarrow 后验分解（各 CPD 可独立学习）。
- 离散 CPT + 狄利克雷先验 \Rightarrow 共轭：后验参数 = 计数 + 伪计数；后验均值 = “平滑后的频率”。
- MLE = 纯频率；会有零计数风险，建议用先验正则化。
- 缺失/隐变量 \Rightarrow 似然不可分、非凸 \Rightarrow 往往只能做局部最优或近似推断/学习。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

10.5 Conditional independence properties of DGMs
At the heart of any graphical model is a set of conditional independence (CI) assumptions. We write $x_A \perp\!\!\!\perp x_B | x_C$ if A is independent of B given C in the graph G, using the semantics to be defined below. Let $I(G)$ be the set of all such CI statements encoded by the graph. We say that G is an I-map (independence map) for p, or that p is Markov wrt G, iff $I(G) \subseteq I(p)$, where $I(p)$ is the set of all CI statements that hold for distribution p. In other words, the graph is an I-map if it does not make any assertions of CI that are not true of the distribution. This allows us to use the graph as a safe proxy for p when reasoning about p's CI properties. This is helpful for designing algorithms that work for large classes of distributions, regardless of their specific numerical parameters θ . Note that the fully connected graph is an I-map of all distributions, since it makes no CI assertions at all (since it is not missing any edges). We therefore say G is a minimal I-map of p if G is an I-map of p, and if there is no $G' \subseteq G$ which is an I-map of p. It remains to specify how to determine if $x_A \perp\!\!\!\perp x_B | x_C$. Deriving these independencies for undirected graphs is easy (see Section 19.2), but the DAG situation is somewhat complicated, because of the need to respect the orientation of the directed edges. We give the details below.

已思考 6s

下面先把原文逐句完整翻译（保留符号与术语），再给出通俗解释，最后总结要点。

翻译

10.5 有向图模型 (DGM) 的条件独立性质

任何图模型的核心都是一组**条件独立 (CI) **假设。我们记

$$x_A \perp_G x_B \mid x_C$$

表示：在图 G 中，按（稍后给出的）语义，集合 A 与集合 B 在给定 C 的条件下相互独立。令 $I(G)$ 表示由图所编码的全部此类 CI 语句的集合。

若且唯若 $I(G) \subseteq I(p)$ ，我们称 G 是 p 的 I-map (independence map, 独立性映射)，或者说 p 相对于 (wrt) G 是 Markov 的；其中 $I(p)$ 是在分布 p 下实际成立的全部 CI 语句的集合。换言之，如果图不会断言任何不真实的独立性，它就是一个 I-map。这样，当我们讨论 p 的 CI 性质时，就可以把图当成一个安全的替身 (proxy)。这对于设计能适用于一大类分布 (而不依赖具体数值参数 θ) 的算法很有帮助。

注意：完全连通图对所有分布都是 I-map，因为它没有缺边，因此不会做任何 CI 断言。于是我们说：如果 G 是 p 的 I-map，且不存在真子图 $G' \subset G$ 仍然是 p 的 I-map，则 G 是 p 的极小 I-map (minimal I-map)。

还剩下一个问题：如何判断 $x_A \perp_G x_B \mid x_C$ 是否成立？对无向图推导这些独立性很容易（见第 19.2 节）；但对 DAG 情况会稍微复杂一些，因为必须考虑有向边的方向性。我们在下面给出细节。

通俗解释

- **这段在说啥？**

图模型之所以有用，是因为“图上的缺边”代表某种条件独立。作者先把一堆术语/集合搭好，目的是：以后我们只用看图，就能判断哪些变量彼此独立，而不用知道具体参数数值。

- **符号速读**

- x_A, x_B, x_C : 变量集合（可以是一维或多维）。
- $x_A \perp_G x_B \mid x_C$: 在图 G 的语义下，A 与 B 在给定 C 时独立。
- $I(G)$: 图 G “声称”的所有独立性集合；
- $I(p)$: 真实分布 p 里真的成立的独立性集合。

- **I-map 的含义**
 - 要求 $I(G) \subseteq I(p)$: 图说的每一个独立性，在真实分布里都**确实成立**。
 - 这样图就成了一个“**不撒谎的独立性地图**”。我们可以放心地用它来推断独立性，设计通用算法（如分解、消息传递等）。
 - **为什么完全连通图总是 I-map?**
 - 因为它**什么独立性也不宣称**（没有缺边 \Rightarrow 不会说“这俩独立”）。既然不宣称，就不可能出错，自然对任意分布都成立。
 - 当然它太保守，**信息量为零**。我们更希望边尽量少，但仍不说错话。
 - **minimal I-map (极小 I-map)**
 - 在不说错独立性的前提下，**再也删不掉任何一条边**。这类图信息量更大、结构更“稀疏”，常用于建模/结构学习。
 - **如何在 DAG 里判断独立？**
 - 这就是接下来要给的“语义”——**d-分离 (d-separation)**。
 - 直觉：并不是“没有路径就独立”那么简单；DAG 里还有**碰撞点 (v-structure) **等方向性效应：是否独立要看路径在给定 C 下是否被“阻断”。（细节在后续小节展开。）
-

要点总结

- **目标**：用图读出一批**条件独立**；这些独立性不依赖具体数值参数。
- **I-map 定义**：若 $I(G) \subseteq I(p)$ ，图 G 是分布 p 的 I-map (或 p 对 G Markov)。
- **极小 I-map**：在保持 I-map 的前提下**不可再删边**。
- **完全连通图**是所有分布的 I-map，但太保守（不提供任何 CI）。
- **判定** $x_A \perp_G x_B \mid x_C$ ：无向图简单；DAG 需用**方向敏感的规则 (d-分离) **来判断路径是否被条件集阻断。

根据我的理解，minimal I-map，在不说错独立性的前提
下，不应该是越删边越独立吗

已思考 44s

你的理解里有一半对了，但差了一个**边界条件**。

先给结论

- 删边 \Rightarrow 图声称的独立性 (CI) 只会变多 (从数学上, 若 $G' \subseteq G$ 则 $I(G) \subseteq I(G')$)。
- minimal I-map 的含义是: 在不说错独立性 ($I(G) \subseteq I(p)$) 的前提下, 再也不能删任何一条边; 一旦再删, 就会导致新增某些 CI, 而这些 CI 在真实分布 p 里并不成立, 从而违反 $I(G') \subseteq I(p)$ 。

所以, “越删边越独立吗?” ——

- 是的, 形式上删边会让图“更独立” (断言更多 CI)。
- 但 minimal I-map 的停止点 是: 再删就会说错独立性。也就是说, 它已经把能安全删除的边都删完了。

一个直观例子

真实分布满足: $A \perp\!\!\!\perp B$ 且 $A \perp B \mid C$ 。

一个最常见的 minimal I-map 是 叉型 $A \leftarrow C \rightarrow B$ 。

- 这图缺少 $A \leftrightarrow B$ 的边, 正好表达 $A \perp B \mid C$ 。
- 如果你再删任意一条边 (比如去掉 $C \rightarrow B$), 图就会断言 $A \perp B$ (无条件独立), 这在真实分布里是假的; 因此它不再是 I-map。

容易混淆的两个点

1. minimal \neq 最稀疏 (全局最少边)

minimal 只要求“**局部**删不了边” (对任何**单条**边, 删了就错), 但可能还存在**另一张**边更少的 I-map (不是当前图的子图)。寻找“最稀疏 I-map”是更强的目标, 一般更难。

2. minimal I-map \neq perfect map

perfect map 要求 $I(G) = I(p)$ (图恰好刻画所有且仅有真实 CI)。minimal 只保证不说假话, 但可能没把所有真的 CI 都画出来 (还留了多余边)。

一句话版

- 删边会增加图里“宣称的独立性”;
- minimal I-map 就是在不说错独立性前提下能删的边都删光了, 再删一条就会出错;
- 它不一定是“最少边”的那张图, 也不一定把 p 的所有独立性都刻全——那是更强的目标 (最稀疏 I-map / perfect map)。

要点小结

- I-map: $I(G) \subseteq I(p)$ (图不撒谎)。
- 删边单调: $G' \subseteq G \Rightarrow I(G) \subseteq I(G')$ 。
- minimal I-map: 没有任何真子图仍是 I-map (再删就错)。

- 最稀疏 I-map、perfect map 是更强要求；minimal 只是“局部不可再删”的版本。

请举例向我解释I-map和极小I-map

已思考 35s

下面用一个三变量的小例子把 I-map 和 **极小 I-map (minimal I-map) **讲清楚。

例子设定 (一个“共同原因”分布)

- 随机变量： C, A, B 都是二元。
- 生成方式：

$$C \sim \text{Bernoulli}(0.5), \quad A | C \sim \text{Bern}(0.9 \text{ if } C=1; 0.1 \text{ if } C=0)$$

$$B | C \sim \text{Bern}(0.9 \text{ if } C=1; 0.1 \text{ if } C=0)$$

直觉： C 同时影响 A 和 B 。因此

- 条件独立**: $A \perp B | C$ (给定 C 后， A 与 B 独立)。
- 边缘相关**: 通常 $A \perp\!\!\!\perp B$ (不看 C 时两者正相关)。

把这当作“真实分布” p 。

图 1: 完全连通三角形 (不是极小, 但一定是 I-map)

DAG: $C \rightarrow A, C \rightarrow B, A \rightarrow B$ 。

- 这个图**不宣称任何独立性** ($I(G)=\emptyset$)，所以对任意分布都是 I-map。
- 对我们的 p 当然也是 I-map，但**不极小**: 去掉边 $A \rightarrow B$ 仍然不会说错任何独立性 (仍是 I-map)，说明还能删边。

用语: I-map 就是“图里声称的所有 CI 在真实分布里都成立”。完全图总是 I-map，但信息量最少。

图 2: 分叉 (fork) $A \leftarrow C \rightarrow B$ (一个极小 I-map)

- 这个图声称的 CI: **只有** $A \perp B | C$ 。

- 这与 p 的 CI 一致，所以它是 p 的 I-map。
 - **极小性：**再删任何一条边都会出错：
 - 删 $C \rightarrow B$: 图会暗示 B 与 (A, C) **无条件独立**，但在 p 中 B 显然依赖于 C (错)。
 - 删 $C \rightarrow A$ 同理。
因此这张图**再也删不了边而保持 I-map** \Rightarrow 它是 **极小 I-map**。
-

图 3: 链 $A \rightarrow C \rightarrow B$ (也是极小 I-map)

- 这个图同样只断言 $A \perp B \mid C$ ，因此也是 p 的 I-map。
- 删任一边都会断言新的独立性 (如 $A \perp C$ 或 $B \perp C$)，在 p 中都是假的 \Rightarrow 也**极小**。

说明：极小 I-map **不唯一**。图 2 和图 3 的 CI 集相同 (都只包含 $A \perp B \mid C$)，它们是**马尔可夫等价**的两个极小 I-map。

反例帮助理解“极小”的停止点

把图 2 再删一条边得到“孤立的 B ”。这会**新增**“ B 与任何变量独立”的断言，而这在真实分布 p 里**不成立**，所以不再是 I-map。这正是“极小 I-map”的含义：**再删一条就会说错独立性**。

小结 (记忆版)

- I-map：图中所有由 d-分离得到的 CI 都在真实分布里成立 (图不撒谎)。
- **删边会只增不减图的 CI**；因此越删边，图“越独立”。
- **极小 I-map**：在保持 I-map 的前提下**不能再删任何边**；再删就会断言假的 CI。
- **不唯一**：同一个分布可能有多个极小 I-map (马尔可夫等价的 DAG)。
- **完全图**总是 I-map，但从不极小 (几乎总能删边而不说错话)。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 10.9 Bayes ball rules. A shaded node is one we condition on. If there is an arrow hitting a bar, it means the ball cannot pass through; otherwise the ball can pass through. Based on (Jordan 2007).

Figure 10.10 (a-b) Bayes ball boundary conditions. (c) Example of why we need boundary conditions. y is an observed child of y , rendering y "effectively observed", so the ball bounces back up on its way from x to z .

10.5.1 d-separation and the Bayes Ball algorithm (global Markov properties)

First, we introduce some definitions. We say an undirected path P is d-separated by a set of nodes E (containing the evidence) iff at least one of the following conditions hold:

1. P contains a chain, $s \rightarrow m \rightarrow t$ or $s \leftarrow m \leftarrow t$, where $m \in E$
2. P contains a tent or fork, $s \sim m \sim t$, where $m \in E$
3. P contains a collider or v-structure, $s \sim m \sim t$, where m is not in E and nor is any descendant of m .

Next, we say that a set of nodes A is d-separated from a different set of nodes B given a third observed set E iff each undirected path from every node $a \in A$ to every node $b \in B$ is d-separated by E . Finally, we define the CI properties of a DAG as follows:

$$xA \perp\!\!\!G xB | xE \iff A \text{ is d-separated from } B \text{ given } E \quad (10.34)$$

The Bayes ball algorithm (Shachter 1998) is a simple way to see if A is d-separated from B given E , based on the above definition. The idea is this. We "shade" all nodes in E , indicating that they are observed. We then place "balls" at each node in A , let them "bounce around" according to some rules, and then ask if any of the balls reach any of the nodes in B . The three main rules are shown in Figure 10.9. Notice that balls can travel opposite to edge

directions. We see that a ball can pass through a chain, but not if it is shaded in the middle. Similarly, a ball can pass through a fork, but not if it is shaded in the middle. However, a ball cannot pass through a v-structure, unless it is shaded in the middle.

We can justify the 3 rules of Bayes ball as follows. First consider a chain structure $X \rightarrow Y \rightarrow Z$, which encodes

$$p(x, y, z) = p(x)p(y|x)p(z|y) \quad (10.35)$$

When we condition on y , are x and z independent? We have

$$\begin{aligned} p(x, z|y) &= p(x)p(y|x)p(z|y) \\ p(y) &= p(x|y)p(z|y) \end{aligned} \quad (10.36)$$

and therefore $x \perp z|y$. So observing the middle node of chain breaks it in two (as in a Markov chain). Now consider the tent structure $X \leftarrow Y \rightarrow Z$. The joint is

$$p(x, y, z) = p(y)p(x|y)p(z|y) \quad (10.37)$$

When we condition on y , are x and z independent? We have

$$\begin{aligned} p(x, z|y) &= p(x, y, z) \\ p(y) &= p(y)p(x|y)p(z|y) \end{aligned} \quad (10.38)$$

and therefore $x \perp z|y$. So observing a root node separates its children (as in a naive Bayes classifier: see Section 3.5). Finally consider a v-structure $X \rightarrow Y \leftarrow Z$. The joint is

$$p(x, y, z) = p(x)p(z)p(y|x, z) \quad (10.39)$$

When we condition on y , are x and z independent? We have

$$p(x, z|y) = p(x)p(z)p(y|x, z) \quad (10.40)$$

so $x \perp z|y$. However, in the unconditional distribution, we have

$$p(x, z) = p(x)p(z) \quad (10.41)$$

so we see that x and z are marginally independent. So we see that conditioning on a common child at the bottom of a v-structure makes its parents become dependent. This important effect is called explaining away, inter-causal reasoning, or Berkson's paradox. As an example of explaining away, suppose we toss two coins, representing the binary numbers 0 and 1, and we observe the "sum" of their values. A priori, the coins are independent, but once we observe their sum, they become coupled (e.g., if the sum is 1, and the first coin is 0, then we know the second coin is 1).

Finally, Bayes Ball also needs the "boundary conditions" shown in Figure 10.10(a-b). To understand where these

rules come from, consider Figure 10.10(c). Suppose Y is a noise-free copy of Y . Then if we observe Y , we effectively observe Y as well, so the parents X and Z have to compete to explain this. So if we send a ball down $X \rightarrow Y \rightarrow Y \rightarrow Z$, it should “bounce back” up along $Y \rightarrow Y \rightarrow Z$. However, if Y and all its children are hidden, the ball does not bounce back.

For example, in Figure 10.11, we see that $x_2 \perp\!\!\! \perp x_6 | x_5$, since the $2 \rightarrow 5 \rightarrow 6$ path is blocked by x_5 (which is observed), the $2 \rightarrow 4 \rightarrow 7 \rightarrow 6$ path is blocked by x_7 (which is hidden), and the $2 \rightarrow 1 \rightarrow 3 \rightarrow 6$ path is blocked by x_1 (which is hidden). However, we also see that $x_2 \perp\!\!\! \perp x_6 | x_5, x_7$, since now the $2 \rightarrow 4 \rightarrow 7 \rightarrow 6$ path is no longer blocked by x_7 (which is observed). Exercise 10.2 gives you some more practice in determining CI relationships for DGMs.

已思考 13s

好的！我先把原文逐句完整翻译（含符号与图注、编号），再用通俗方式详细解释，最后总结要点。

翻译

图 10.9 贝叶斯球 (Bayes ball) 规则。 阴影结点表示我们对其做了条件化（已观测）。如果有一支箭头撞到了一条小竖线，表示球不能通过；否则球可以通过。图基于 (Jordan, 2007)。

图 10.10 (a–b) 贝叶斯球的边界条件。 (c) 为什么需要边界条件的例子。 y' 是 y 的一个被观测的子结点，这使得 y “等效地被观测”，因此球在从 x 到 z 的路上，会在 $y \rightarrow y'$ 处反弹回去。

10.5.1 d-分离与贝叶斯球算法（全局马尔可夫性质）

先给出一些定义。若无向路径 P 被包含证据结点的集合 E d-分离 (d-separated)，当且仅当下列情形至少有一种成立：

1. P 中包含一段链 (chain)， $s \rightarrow m \rightarrow t$ 或 $s \leftarrow m \leftarrow t$ ，且 $m \in E$ ；
2. P 中包含一段帐篷/叉形 (tent/fork)， $s \leftarrow m \rightarrow t$ ，且 $m \in E$ ；
3. P 中包含一段碰撞点/“V”结构 (collider / v-structure)， $s \rightarrow m \leftarrow t$ ，且 $m \notin E$ ，并且 m 的任何后代也都不在 E 中。

接着，若从集合 A 中每个结点 a 到集合 B 中每个结点 b 的每一条无向路径都被 E d-分离，则称集合 A 与集合 B 在给定 E 时 d-分离。最后，DAG 的条件独立 (CI) 性质定义为

$$x_A \perp_G x_B \mid x_E \Leftrightarrow A \text{ 在给定 } E \text{ 时与 } B \text{ d-分离.} \quad (10.34)$$

贝叶斯球算法 (Shachter, 1998) 是基于上面定义检查“给定 E , A 与 B 是否 d-分离”的一个简单方法。做法：把 E 中的所有结点“涂黑”（表示观测）。然后在 A 的每个结点放一个“球”，让它们按一些规则“弹跳”。最后看是否有球能到达 B 中的任意结点。三条主要规则见图 10.9。注意：球可以沿着边的反方向移动。可以看到：

- 球能穿过链，除非中间结点被涂黑；
- 球能穿过叉形，除非中间结点被涂黑；
- 球不能穿过V 结构，除非中间结点被涂黑（或其某个后代被涂黑）。

我们可以这样说明这三条规则。先看链 $X \rightarrow Y \rightarrow Z$, 其联合分布为

$$p(x, y, z) = p(x)p(y \mid x)p(z \mid y). \quad (10.35)$$

给定 y 时， x 与 z 是否独立？

$$p(x, z \mid y) = \frac{p(x)p(y \mid x)p(z \mid y)}{p(y)} = \frac{p(x, y)}{p(y)} p(z \mid y) = p(x \mid y)p(z \mid y), \quad (10.36)$$

因此 $x \perp z \mid y$ 。也就是说，**观测链的中间结点会把链“截断”**（类似马尔可夫链）。

再看叉形 $X \leftarrow Y \rightarrow Z$, 联合分布

$$p(x, y, z) = p(y)p(x \mid y)p(z \mid y). \quad (10.37)$$

给定 y :

$$p(x, z \mid y) = \frac{p(y)p(x \mid y)p(z \mid y)}{p(y)} = p(x \mid y)p(z \mid y), \quad (10.38)$$

因此 $x \perp z \mid y$ 。这就是**观测共同父结点会使孩子独立**（如朴素贝叶斯）。

最后看 V 结构 $X \rightarrow Y \leftarrow Z$ 。联合分布

$$p(x, y, z) = p(x)p(z)p(y \mid x, z). \quad (10.39)$$

给定 y 时：

$$p(x, z \mid y) = \frac{p(x)p(z)p(y \mid x, z)}{p(y)}, \quad (10.40)$$

所以此时 $x \perp z \mid y$ （一般耦合）；而边缘上

$$p(x, z) = p(x)p(z), \quad (10.41)$$

即 x 与 z 边缘独立。所以：对 V 结构，观测共同子结点会让父结点变得相关。这个重要现象叫 **解释型 (explaining away) / 交叉因果推断 / Berkson 悖论**。例如：抛两枚硬币并观测它们点数之“和”。先验下两枚硬币独立；但一旦观测到“和”，它们就耦合了（如和=1，若第一枚是0，则第二枚必为1）。

最后，贝叶斯球还需要图 10.10(a-b) 的**边界条件**。理解方法：看图 10.10(c)。设 y' 是 y 的无噪声拷贝。若我们观测到 y' ，就等价于也观测到了 y ；于是 y 的父结点 x, z 会“竞争解释”这个观测。所以当球沿 $X \rightarrow Y \rightarrow Y'$ 往下走时，会在 $Y \rightarrow Y'$ 处**反弹回去**，并沿 $Y \rightarrow Z$ 往上走；但是如果 Y 及其所有子结点都**未观测**，则不会发生这种反弹。

举例（见图 10.11）：可以得到 $x_2 \perp x_6 \mid x_5$ ，因为路径 $2 \rightarrow 5 \rightarrow 6$ 被观测的 x_5 阻断；路径 $2 \rightarrow 4 \rightarrow 7 \rightarrow 6$ 被**未观测的碰撞点** x_7 阻断；路径 $2 \rightarrow 1 \rightarrow 3 \rightarrow 6$ 被**未观测的碰撞点** x_1 阻断。然而 $x_2 \perp x_6 \mid x_5, x_7$ ，因为一旦把 x_7 也观测了，路径 $2 \rightarrow 4 \rightarrow 7 \rightarrow 6$ 就被“打开”了。

通俗解释

1) 三种基本结构与是否“打开/关闭”

- 链 $X \rightarrow Y \rightarrow Z$: 给定中间 $Y \Rightarrow$ **关闭**；不观测 $Y \Rightarrow$ **打开**。
- 叉 $X \leftarrow Y \rightarrow Z$: 给定中间 $Y \Rightarrow$ **关闭**（孩子独立）；不观测 \Rightarrow **打开**。
- V 结构 $X \rightarrow Y \leftarrow Z$: 默认 **关闭**；只要观测了 Y 或其后代 \Rightarrow **打开**（解释型）。

“路径被 E d-分离”= 该路径中**至少有一段**满足上述“关闭”条件。

2) 贝叶斯球操作（记一套“能/不能过”的规则就行）

把 E 涂黑，放球在 A 上：

- 到达**非碰撞点**（链/叉）的中间结点：
 - 若此结点被观测 \Rightarrow **挡住**（不能过）；
 - 未观测 \Rightarrow **能穿过**。
- 到达**碰撞点**（V）：
 - 若此结点及其后代都未观测 \Rightarrow **挡住**；
 - 若结点本身或任一后代被观测 \Rightarrow **能穿过**（而且球会向**两个方向**弹开）。
- **球可逆行**（与箭头方向无关）。
- **边界条件**：若从一个结点走向其**被观测的子结点**，球会**反弹回去**（等效把父结点当“被观测”处理）；若该子结点未观测，则不反弹。

若**任何球**能从 A 到达 B ，则“未被 d-分离”（存在依赖）；否则 $A \perp_G B \mid E$ 。

3) “解释型”直觉

V 结构里一旦看到“结果”（子结点）或它的证据，父结点之间就会**牵连起来**：一个父更可能，另一个就更不需要——典型的“**竞争解释**”。

要点总结

- **d-分离定义**：用“链/叉（给定中间 \Rightarrow 关）”与“V 结构（给定中间或其后代 \Rightarrow 开）”判断路径是否被证据集 E 阻断。
- **全局 CI 规则**： $x_A \perp_G x_B \mid x_E \Leftrightarrow$ 从 A 到 B 的所有无向路径都被 E d-分离（式 10.34）。
- **贝叶斯球算法**：把证据结点涂黑，按三条通过/阻断规则（加边界条件）“弹球”，检查能否到达。
- **解释型 (explaining away)**：V 结构中观测到共同子结点（或其后代）会让父结点**相关**。
- **常见陷阱**：
 1. 忘了碰撞点默认关闭；
 2. 忘了观测后代也会打开 V 结构；
 3. 忽略**边界条件**导致误判路径是否被“反弹”。

没看懂这里讲的啥

已思考 21s

没问题！我把这段内容用**最小化的规则 + 小例子**重讲一遍。看完你就能自己判定“给定证据时，两个变量是否独立”。

这段话在讲什么？

它在教你如何只看**有向无环图（DAG）**就判断：

“在给定一些已知变量 E 后，A 和 B 是否独立？”

这叫 **d-分离**。实用的判定工具就是 **贝叶斯球（Bayes Ball）**规则。

必背三种局部结构（只看中间点是否被“观测”）

把从 A 到 B 的任意无向路径抽成三段中的一种：

1. 链: $X \rightarrow Y \rightarrow Z$ 或 $X \leftarrow Y \leftarrow Z$
 - 观测中间 $Y \rightarrow$ 路径被堵 (A 与 B 独立)
 - 不观测 $Y \rightarrow$ 路径畅通
 2. 叉 (共同原因) : $X \leftarrow Y \rightarrow Z$
 - 观测中间 $Y \rightarrow$ 堵
 - 不观测 $Y \rightarrow$ 通
 3. V 结构 (共同结果/碰撞点) : $X \rightarrow Y \leftarrow Z$
 - 默认 堵
 - 只要观测了 Y 或者 Y 的任意后代 \rightarrow 通
- 这就是“解释型 (explaining away) ”：看到了结果，两个原因就会互相牵连。

判定规则：只要某条路径里出现“该堵的局部段”，整条路径就被d-分离 (关掉) 。A 与 B 独立 \Leftrightarrow A 到 B 的所有路径都被关掉。

贝叶斯球怎么用 (口语版)

1. 把证据集 E 里的结点涂黑 (表示已观测) 。
2. 在 A 的每个结点放个“球”，让它沿边任意方向走。
3. 遇链/叉：中点涂黑 \Rightarrow 停；未黑 \Rightarrow 继续。
4. 遇 V 结构：中点及其后代都没黑 \Rightarrow 停；只要中点或后代黑了 \Rightarrow 继续 (向两边散开) 。
5. 若任何球能到 B \Rightarrow 不独立；都到不了 \Rightarrow 独立。

图 10.10 的“边界条件”其实就是第 4 条的实现细节：

如果 Y 的子结点被观测 (比如无噪声的拷贝 Y')，就等效于“观测了 Y ”，于是经 Y 的 V 结构会打开 (球会“反弹”回到另一侧父结点)。

3 个一眼能判的例子

1. 链 $A \rightarrow Y \rightarrow B$
 - 无证据：A 与 B 相关 (路径通) 。
 - 给定 Y : $A \perp B$ (路径被截断) 。
2. 叉 $A \leftarrow Y \rightarrow B$
 - 无证据：A 与 B 相关 (共同原因) 。

- 给定 Y : $A \perp B$ (孩子独立, 朴素贝叶斯的假设)。

3. V 结构 $A \rightarrow Y \leftarrow B$

- 无证据: $A \perp B$ (碰撞点默认关)。
- 给定 Y 或 Y 的子结点: A 与 B 相关 (解释型)。

小故事 (V 结构): 两枚硬币 A 、 B 彼此独立; 但一旦你知道它们“和”为 1 (相当于观测到共同结果 Y), A 为 0 就意味着 B 必为 1——它们被“绑定”了。

一个稍复杂的走球演练 (对应教材里的图意)

- 想判断 x_2 与 x_6 在给定 x_5 时是否独立:
 - 路 $2 \rightarrow 5 \rightarrow 6$: 链中点 x_5 被观测 \Rightarrow 关;
 - 路 $2 \rightarrow 4 \rightarrow 7 \rightarrow 6$: 若 x_7 未观测且是 V 结构的中点 \Rightarrow 关;
 - 路 $2 \rightarrow 1 \rightarrow 3 \rightarrow 6$: 若 x_1 未观测且是 V 结构中点 \Rightarrow 关。
 全关 $\Rightarrow x_2 \perp x_6 \mid x_5$ 。
 若再把 x_7 观测, 第二条路的 V 结构被打开 \Rightarrow 不独立。

速查表 (记忆版)

- 链/叉: 给定中点 \Rightarrow 关; 否则通。
 - V 结构: 默认关; 给定中点或任一后代 \Rightarrow 通。
 - “所有路径都关掉” \Rightarrow 独立; 只要有一条通 \Rightarrow 不独立。
 - 观测一个碰撞点的后代, 等效“激活”该碰撞点 (就是图 10.10 的“反弹”)。
-

要点总结

- d-分离是用来从图上读出“给定证据时的独立性”。
- 贝叶斯球是执行 d-分离的操作化方法: 把证据涂黑、按三条通过/阻断规则让球乱跑, 看能否从 A 到 B。
- 解释型: V 结构一旦观测到共同结果 (或其后代), 两个原因会变相关。
- 牢记三种局部结构的“开/关”规则, 就能快速判断大多数 CI 问题。

AB和XYZ什么关系

已思考 10s

短答：没有固定关系。

在这章里 A、B 用来表示我们关心的“两端”变量（或变量集合），要判断的是 $A \perp B | E$ ；而 X、Y、Z 只是拿来说明**一段三节点的小结构**（链 / 叉 / V 结构）的**局部规则**：X 和 Z 在两侧，Y 是中间点。

更直白地说：

- A、B = 你要判断是否独立的两组结点（全局“起点-终点”）。
- X、Y、Z = 某条从 A 到 B 的路径里的一**小段**三连结点（局部“左-中-右”），用来套 Bayes Ball 的三条规则。

可以把它们对应起来理解：

- 链： $A(=X) \rightarrow Y \rightarrow B(=Z)$
- 叉： $A(=X) \leftarrow Y \rightarrow B(=Z)$
- 碰撞点 (V)： $A(=X) \rightarrow Y \leftarrow B(=Z)$

流程是：从 A 到 B 的每条路径上，遇到的任何一段 X-Y-Z 都按规则判断“开/关”；**所有路径都关** $\Rightarrow A \perp B | E$ ，**只要有一条开** \Rightarrow 不独立。

小结

- A/B：要判独立的两端（全局）。
- X/Y/Z：路径中的局部三元组，便于套规则。
- 没有——映射关系；在某条具体路径上，可以把 X 看成当前段的“左端”、Y 是中点、Z 是“右端”，常常就对应那条路径上的某个 A 与 B。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

10.5.2 Other Markov properties of DGMs

From the d-separation criterion, one can conclude that
 $t \perp nd(t) \setminus pa(t) | pa(t)$ (10.42)
 where the non-descendants of a node $nd(t)$ are all the nodes except for its descendants, $nd(t) = \bigvee \{t \cup desc(t)\}$.

Equation 10.42 is called the directed local Markov property. For example, in Figure 10.11, we have $nd(3) = \{2, 4\}$, and $pa(3) = 1$, so $3 \perp 2, 4 | 1$. A special case of this property is when we only look at predecessors of a node according to some topological ordering. We have

$$t \perp pred(t) \setminus pa(t) | pa(t) \quad (10.43)$$

which follows since $pred(t) \subseteq nd(t)$. This is called the ordered Markov property, which justifies Equation 10.7. For example, in Figure 10.11, if we use the ordering 1, 2, ..., 7, we find $pred(3) = \{1, 2\}$ and $pa(3) = 1$, so $3 \perp 2 | 1$. We have now described three Markov properties for DAGs: the directed global Markov property G in Equation 10.34, the ordered Markov property O in Equation 10.43, and the directed local Markov property L in Equation 10.42. It is obvious that $G \Rightarrow L \Rightarrow O$. What is less obvious, but nevertheless true, is that $O \Rightarrow L \Rightarrow G$ (see e.g., (Koller and Friedman 2009) for the proof). Hence all these properties are equivalent. Furthermore, any distribution p that is Markov wrt G can be factorized as in Equation 10.7; this is called the factorization property F. It is obvious that $O \Rightarrow F$, but one can show that the converse also holds (see e.g., (Koller and Friedman 2009) for the proof).

10.5.3 Markov blanket and full conditionals

The set of nodes that renders a node t conditionally independent of all the other nodes in the graph is called t's Markov blanket; we will denote this by $mb(t)$. One can show that the Markov blanket of a node in a DGM is equal to the parents, the children, and the co-parents, i.e., other nodes who are also parents of its children:

$$mb(t) = ch(t) \cup pa(t) \cup copa(t) \quad (10.44)$$

For example, in Figure 10.11, we have

$$mb(5) = \{6, 7\} \cup \{2, 3\} \cup \{4\} = \{2, 3, 4, 6, 7\} \quad (10.45)$$

where 4 is a co-parent of 5 because they share a common child, namely 7. To see why the co-parents are in the Markov blanket, note that when we derive $p(xt|x-t) = p(xt, x-t)/p(x-t)$, all the terms that do not involve xt will cancel out between numerator and denominator, so we are left with a product of CPDs which contain xt in their scope.

Hence

$$p(xt|x-t) \propto p(xt|xp(t)) \prod_{s \in ch(t)} p(xs|xp(s)) \quad (10.46)$$

For example, in Figure 10.11 we have

$$p(x5|x-5) \propto p(x5|x2,x3)p(x6|x3,x5)p(x7|x4,x5,x6) \quad (10.47)$$

The resulting expression is called t's full conditional, and

will prove to be important when we study Gibbs sampling (Section 24.2).

下面先把原文逐句完整翻译（保留公式与编号），再做通俗解释，最后给出要点总结。

翻译

10.5.2 DGM 的其他马尔可夫性质

由 d-分离准则可以推出

$$t \perp \text{nd}(t) \setminus \text{pa}(t) \mid \text{pa}(t) \quad (10.42)$$

其中结点 t 的**非后代集合** $\text{nd}(t)$ 是除去它自己以及它所有后代之外的所有结点，
 $\text{nd}(t) = V \setminus \{t \cup \text{desc}(t)\}$ 。式 (10.42) 称为**有向的局部马尔可夫性质**。例如，在图
10.11 中， $\text{nd}(3) = \{2, 4\}$ ， $\text{pa}(3) = 1$ ，所以 $3 \perp 2, 4 \mid 1$ 。

上述性质的一个特例：只看某个拓扑序下结点的**所有前驱** (predecessors)。有

$$t \perp \text{pred}(t) \setminus \text{pa}(t) \mid \text{pa}(t) \quad (10.43)$$

因为 $\text{pred}(t) \subseteq \text{nd}(t)$ 。这称为**有序马尔可夫性质**，它正当化了式 (10.7)。例如，在图
10.11 中，如果使用顺序 $1, 2, \dots, 7$ ，则 $\text{pred}(3) = \{1, 2\}$ ， $\text{pa}(3) = 1$ ，所以 $3 \perp 2 \mid 1$
。

到此我们描述了关于 DAG 的三种马尔可夫性质：式 (10.34) 的**有向全局马尔可夫性质** G ，
式 (10.43) 的**有序马尔可夫性质** O ，以及式 (10.42) 的**有向局部马尔可夫性质** L 。显然有
 $G \Rightarrow L \Rightarrow O$ 。更不那么显然但确实成立的是 $O \Rightarrow L \Rightarrow G$ (证明可见如 Koller 和
Friedman 2009)。因此这些性质**彼此等价**。进一步，只要某个分布 p 相对于 G 是马尔可
夫的，它就可以按式 (10.7) 那样进行因子分解；这称为**分解性质** F 。显然 $O \Rightarrow F$ ，而且
也可以证明**反过来也成立** (见 Koller 和 Friedman 2009)。

10.5.3 马尔可夫毯与满条件分布

使结点 t 与图中所有其他结点在条件下相互独立的那一组结点称为 t 的**马尔可夫毯**

(Markov blanket)，记作 $\text{mb}(t)$ 。可以证明，在 DGM 中，一个结点的马尔可夫毯等于
它的**父结点、子结点以及共同父结点 (co-parents)** 的并集——“共同父结点”指也是其子
结点的父结点的其他结点：

$$\text{mb}(t) = \text{ch}(t) \cup \text{pa}(t) \cup \text{copa}(t) \quad (10.44)$$

例如，在图 10.11 中，

$$\text{mb}(5) = \{6, 7\} \cup \{2, 3\} \cup \{4\} = \{2, 3, 4, 6, 7\} \quad (10.45)$$

其中 4 是 5 的共同父结点，因为它们有一个共同的子结点 7。

为什么共同父结点会出现在马尔可夫毯中？注意当我们推导

$$p(x_t | x_{-t}) = \frac{p(x_t, x_{-t})}{p(x_{-t})}$$

时，不含 x_t 的项会在分子和分母中相互抵消，留下的就是那些在其“作用域”里包含 x_t 的 CPD 的乘积。因此

$$p(x_t | x_{-t}) \propto p(x_t | x_{\text{pa}(t)}) \prod_{s \in \text{ch}(t)} p(x_s | x_{\text{pa}(s)}) \quad (10.46)$$

例如，在图 10.11 中有

$$p(x_5 | x_{-5}) \propto p(x_5 | x_2, x_3) p(x_6 | x_3, x_5) p(x_7 | x_4, x_5, x_6) \quad (10.47)$$

上述表达式称为结点 t 的**满条件 (full conditional)**，在我们学习 Gibbs 采样（第 24.2 节）时会非常重要。

通俗解释

1) 三种“马尔可夫性质”是在说什么？

- 全局 (G)**：用 d-分离在整张图上判断“给定 E , A 与 B 是否独立”。
- 局部 (L)**：对单个结点 t ：给定它的父结点， t 与所有非后代（除了父以外）独立。
理解：孩子只通过“父”感知外界，旁边的其他分支对它不起直接作用。
- 有序 (O)**：把结点按拓扑顺序排好，给定父结点后， t 与所有“更早出现但不是父亲”的结点独立。
- 等价**：三种说法其实是同一件事的不同表述；从任一条出发都能推出另外两条。

2) “分解性质 F”是什么？

- 只要分布 p 满足图的 CI（即对 G 马尔可夫），它的联合分布就能写成

$$p(x) = \prod_t p(x_t | x_{\text{pa}(t)})$$

这就是我们在贝叶斯网里常用的**父—子因子分解**。

- 结论更强：**等价于有序性质（甚至可互推）。

3) 马尔可夫毯 (Markov blanket)

- 想知道结点 t 的分布，只需看它的父、子、共同父这三个圈层；把这些固定住， t 与图中其他任何结点都独立。
- 直觉：
 - 父亲直接影响 t ；
 - 子女里含有关于 t 的信息（尤其当子女被观测时）；
 - 共同父会通过**共同的孩子**与 t 发生“解释型”耦合（V 结构被激活时尤为明显）。
- 这也解释了式 (10.46): 在 $p(x_t | x_{-t})$ 里，真正留下的只有涉及 x_t 的 CPD:
 - t 自己的 CPD (父→子)；
 - 每个孩子的 CPD (里面含 x_t)；
 共同父之所以出现，是因为它们在**孩子的** CPD 里与 x_t 同时出现。

4) “满条件”为什么重要？

- Gibbs 采样要轮流从每个结点的 $p(x_t | x_{-t})$ 采样；知道它只依赖**马尔可夫毯**，就能**局部计算**，而不必看整张图。
-

要点总结

- 局部马尔可夫性质 (L)**：给定父结点，结点 t 与其**非后代（除父外）**独立。
- 有序马尔可夫性质 (O)**：给定父结点， t 与拓扑顺序中所有**更早但不是父**的结点独立。
- 全局马尔可夫性质 (G)**：用 d-分离判定任意两组结点的 CI。
- 三者等价**；并与**分解性质 (F)** **等价**：满足这些性质的分布都可写成 $\prod_t p(x_t | x_{\text{pa}(t)})$ 。
- 马尔可夫毯**： $\text{mb}(t) = \text{pa}(t) \cup \text{ch}(t) \cup \text{copa}(t)$ 。给定这组结点， t 与其余所有结点独立。
- 满条件**: $p(x_t | x_{-t}) \propto p(x_t | x_{\text{pa}(t)}) \prod_{s \in \text{ch}(t)} p(x_s | x_{\text{pa}(s)})$ ；它只依赖马尔可夫毯，是 Gibbs 采样的关键。

满条件是什么，如何理解

已思考 26s

下面把“**满条件**”(full conditional)“讲清楚：它是什么、怎么推、怎么用、和 CPD 的区别，并给你一个算例。

是什么

满条件分布指在一个联合分布 $p(x_1, \dots, x_V)$ 中，**某个变量 x_t 在给定其它所有变量 x_{-t} 时的条件分布**：

$$p(x_t | x_{-t}) = \text{“满条件” of } x_t$$

这里 x_{-t} 表示除了 x_t 以外的所有变量。英文也常叫 *complete conditional*。它在 **Gibbs 采样** 里最常见：算法要交替地从每个 $p(x_t | x_{-t})$ 采样。

和 CPD 有啥不同？

- CPD (如贝叶斯网里一条边的分布) 是**模型里**定义好的： $p(x_t | x_{\text{pa}(t)})$ (只依赖父节点)。
 - **满条件是给定数据/证据后的后验**： $p(x_t | x_{-t})$ 。它一般**不等于** CPD，会额外吃进来自**子节点** (以及共同父节点) 的信息。
-

在贝叶斯网 (DAG) 中怎么写？

利用联合分解 $p(x) = \prod_s p(x_s | x_{\text{pa}(s)})$ ：

$$\begin{aligned} p(x_t | x_{-t}) &= \frac{p(x_t, x_{-t})}{\sum_{x'_t} p(x'_t, x_{-t})} = \frac{\prod_s p(x_s | x_{\text{pa}(s)})}{\sum_{x'_t} \prod_s p(x_s | x_{\text{pa}(s)})} \\ &\propto \underbrace{p(x_t | x_{\text{pa}(t)})}_{\text{自己的 CPD}} \times \underbrace{\prod_{s \in \text{ch}(t)} p(x_s | x_{\text{pa}(s)})}_{\text{所有子节点的似然项}} \end{aligned}$$

因为**不含** x_t 的因子在分子分母中抵消了，留下的只有**包含** x_t 的那些因子。

所以满条件只依赖于 t 的**马尔可夫毯**：父 \cup 子 \cup 共同父 (共同父出现在子节点的 CPD 里)。

记忆式：

$$p(x_t | x_{-t}) \propto \text{“prior” } p(x_t | \text{父}) \times \prod_{\text{子}} \text{“likelihood”}$$

再把对所有可能 x_t 的值求和（或积分）做归一化，得到真正的概率。

一个离散小算例（看见“子”信息如何改写信念）

图结构: $A \rightarrow B \leftarrow C$ 且 $B \rightarrow D$, 并且 D 还有一个共同父 E 。

想求 满条件 $p(b | a, c, d, e)$ 。

由上式：

$$p(b | a, c, d, e) \propto \underbrace{p(b | a, c)}_{B \text{ 的 CPD}} \times \underbrace{p(d | b, e)}_{\text{子 } D \text{ 的 CPD}}$$

给个具体数（都二元）：

- $p(b=1 | a=1, c=0) = 0.8 \Rightarrow p(b=0 | a=1, c=0) = 0.2$
- $p(d=1 | b=1, e=1) = 0.9, \quad p(d=1 | b=0, e=1) = 0.2$

观测: $a=1, c=0, d=1, e=1$ 。

则未归一化概率：

- $b=1: 0.8 \times 0.9 = 0.72$
- $b=0: 0.2 \times 0.2 = 0.04$

归一化：总和 0.76。

$$p(b=1 | \dots) = 0.72/0.76 \approx 0.947, \quad p(b=0 | \dots) \approx 0.053$$

你看到：子节点 D 的观测把我们对 B 的信念大幅推向了 1。

连续情形一眼看

- 线性高斯网（每个 CPD 都是高斯回归）：满条件仍是高斯，均值/方差只由马尔可夫毯决定（等价“局部线性回归 + 噪声”的后验）。
-

实操步骤（离散版）

1. 列出 x_t 的所有取值 k 。
2. 计算每个 k 的 未归一化权重

$$\tilde{p}_k = p(x_t=k | x_{\text{pa}(t)}) \prod_{s \in \text{ch}(t)} p(x_s | x_{\text{pa}(s)})$$
3. 归一化: $p_k = \tilde{p}_k / \sum_k \tilde{p}_k$ 。
 (连续变量同理，用密度并做积分归一化。)

常见误区

- 以为满条件就是 CPD：**错**。只有当 t 没有子节点时，满条件才退化为 CPD。
- 以为要看全图：不用，只看**马尔可夫毯**。
- 混淆“满条件”和“给定部分变量的条件分布”：满条件是**给定其它“所有”的变量**。

要点小结

- **满条件**: $p(x_t | x_{-t})$, 给定其它全部变量时 x_t 的分布。
- **贝叶斯网公式**: $p(x_t | x_{-t}) \propto p(x_t | \text{父}) \prod_{\text{子}} p(\text{子} | \text{该子所有父})$ 。
- **只依赖马尔可夫毯** (父、子、共同父)。
- **用途**: Gibbs 采样、坐标式更新、局部推断与学习。
- **与 CPD 区别**: CPD 是建模时的成分；满条件是**后验**，会合入子节点提供的证据。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 10.12 (a) Influence diagram for basic oil wild catter problem. (b) An extension in which we have an information arc from the Sound chance node to the Drill decision node. (c) An extension in which we get to decide whether to perform the test or not.

Figure 10.13 (a) A POMDP, shown as an influence diagram. z_t are hidden world states. We implicitly make the no forgetting assumption, which effectively means

that at has arrows coming into it from all previous observations, $x_1:t$. (b) An MDP, shown as an influence diagram.

Figure 10.14 Some DGMs.

10.6 Influence (decision) diagrams

We can represent multi-stage (Bayesian) decision problems by using a graphical notation known as a decision diagram or an influence diagram (Howard and Matheson 1981; Kjaerulff and Madsen 2008). This extends directed graphical models by adding decision nodes (also called action nodes), represented by rectangles, and utility nodes (also called value nodes), represented by diamonds. The original random variables are called chance nodes, and are represented by ovals, as usual. Figure 10.12(a) gives a simple example, illustrating the famous oil wild-catter problem.³ In this problem, you have to decide whether to drill an oil well or not. You have two possible actions: $d = 1$ means drill, $d = 0$ means don't drill. You assume there are 3 states of nature: $o = 0$ means the well is dry, $o = 1$ means it is wet (has some oil), and $o = 2$ means it is soaking (has a lot of oil). Suppose your prior beliefs are $p(o) = [0.5, 0.3, 0.2]$. Finally, you must specify the utility function $U(d, o)$. Since the states and actions are discrete, we can represent it as a table (analogous to a CPT in a DGM). Suppose we use the following numbers, in dollars:

We see that if you don't drill, you incur no costs, but also make no money. If you drill a dry well, you lose \$70; if you drill a wet well, you gain \$50; and if you drill a soaking well, you gain \$200. Your prior expected utility if you drill is given by

$$EU(d = 1) = 2 \cdot o=0 \cdot p(o)U(d, o) = 0.5 \cdot (-70) + 0.3 \cdot 50 + 0.2 \cdot 200 = 20 \quad (10.48)$$

Your expected utility if you don't drill is 0. So your maximum expected utility is

$$MEU = \max\{EU(d = 0), EU(d = 1)\} = \max\{0, 20\} = 20 \quad (10.49)$$

and therefore the optimal action is to drill:

$$d^* = \operatorname{argmax}\{EU(d = 0), EU(d = 1)\} = 1 \quad (10.50)$$

Now let us consider a slight extension to the model.

Suppose you perform a sounding to estimate the state of

the well. The sounding observation can be in one of 3 states: $s = 0$ is a diffuse reflection pattern, suggesting no oil; $s = 1$ is an open reflection pattern, suggesting some oil; and $s = 2$ is a closed reflection pattern, indicating lots of oil. Since S is caused by O , we add an $O \rightarrow S$ arc to our model. In addition, we assume that the outcome of the sounding test will be available before we decide whether to drill or not; hence we add an information arc from S to D . This is illustrated in Figure 10.12(b).

Let us model the reliability of our sensor using the following conditional distribution for $p(s|o)$:

Suppose we do the sounding test and we observe $s = 0$.

The posterior over the oil state is

$$p(o|s=0) = [0.732, 0.219, 0.049] \quad (10.51)$$

Now your posterior expected utility of performing action d is

$$EU(d|s=0) = \sum_{o=0}^2 p(o|s=0)U(o, d) \quad (10.52)$$

If $d = 1$, this gives

$$\begin{aligned} EU(d=1|s=0) &= 0.732 \times (-70) + 0.219 \times 50 + 0.049 \times 200 \\ &= -30.5 \end{aligned} \quad (10.53)$$

However, if $d = 0$, then $EU(d=0|s=0)=0$, since not drilling incurs no cost. So if we observe $s = 0$, we are better off not drilling, which makes sense. Now suppose we do the sounding test and we observe $s = 1$. By similar reasoning, one can show that $EU(d=1|s=1)=32.9$, which is higher than $EU(d=0|s=1)=0$. Similarly, if we observe $s = 2$, we have $EU(d=1|s=2)=87.5$ which is much higher than $EU(d=0|s=2)=0$. Hence the optimal policy $d^*(s)$ is as follows: if $s = 0$, choose $d^*(0) = 0$ and get \$0; if $s = 1$, choose $d^*(1) = 1$ and get \$32.9; and if $s = 2$, choose $d^*(2) = 1$ and get \$87.5.

You can compute your expected profit or maximum expected utility as follows:

$$MEU = \sum_s p(s)EU(d^*(s)|s) \quad (10.54)$$

This is the expected utility given possible outcomes of the sounding test, assuming you act optimally given the outcome. The prior marginal on the outcome of the test is $p(s) = o p(o)p(s|o) = [0.41, 0.35, 0.24]$ (10.55)

Hence your maximum expected utility is

$$MEU = 0.41 \times 0 + 0.35 \times 32.9 + 0.24 \times 87.5 = 32.2 \quad (10.56)$$

Now suppose you can choose whether to do the test or

not. This can be modelled as shown in Figure 10.12(c), where we add a new test node T. If $T = 1$, we do the test, and S can enter 1 of 3 states, determined by O, exactly as above. If $T = 0$, we don't do the test, and S enters a special unknown state. There is also some cost associated with performing the test.

Is it worth doing the test? This depends on how much our MEU changes if we know the outcome of the test (namely the state of S). If you don't do the test, we have $MEU = 20$ from Equation 10.49. If you do the test, you have $MEU = 32.2$ from Equation 10.56. So the improvement in utility if you do the test (and act optimally on its outcome) is \$12.2. This is called the value of perfect information (VPI). So we should do the test as long as it costs less than \$12.2. In terms of graphical models, the VPI of a variable T can be determined by computing the MEU for the base influence diagram, I, and then computing the MEU for the same influence diagram where we add information arcs from T to the action nodes, and then computing the difference. In other words,

$$VPI = MEU(I + T \rightarrow D) - MEU(I) \quad (10.57)$$

where D is the decision node and T is the variable we are measuring. It is possible to modify the variable elimination algorithm (Section 20.3) so that it computes the optimal policy given an influence diagram. These methods essentially work backwards from the final time-step, computing the optimal decision at each step assuming all following actions are chosen optimally. See e.g., (Lauritzen and Nilsson 2001; Kjaerulff and Madsen 2008) for details. We could continue to extend the model in various ways.

For example, we could imagine a dynamical system in which we test, observe outcomes, perform actions, move on to the next oil well, and continue drilling (and polluting) in this way. In fact, many problems in robotics, business, medicine, public policy, etc. can be usefully formulated as influence diagrams unrolled over time (Raiffa 1968; Lauritzen and Nilsson 2001; Kjaerulff and Madsen 2008).

A generic model of this form is shown in Figure 10.13(a). This is known as a partially observed Markov decision process or POMDP (pronounced "pom-d-p"). This is basically a hidden Markov model (Section 17.3) augmented with action and reward nodes. This can be

used to model the perception-action cycle that all intelligent agents use (see e.g., (Kaelbling et al. 1998) for details). A special case of a POMDP, in which the states are fully observed, is called a Markov decision process or MDP, shown in Figure 10.13(b). This is much easier to solve, since we only have to compute a mapping from observed states to actions. This can be solved using dynamic programming (see e.g., (Sutton and Barto 1998) for details). In the POMDP case, the information arc from x_t to a_t is not sufficient to uniquely determine the best action, since the state is not fully observed. Instead, we need to choose actions based on our belief state, $p(z_t|x_1:t, a_1:t)$. Since the belief updating process is deterministic (see Section 17.4.2), we can compute a belief state MDP. For details on how to compute the policies for such models, see e.g., (Kaelbling et al. 1998; Spaan and Vlassis 2005).

下面先把原文逐句完整翻译（含表/图注与公式、保持编号），再用通俗方式详细解释并把关键数值算一遍，最后给出要点总结。

翻译

图 10.12

- (a) 基本“野猫式找油 (oil wild-catter) ”问题的影响图（又称决策图）。
- (b) 一个扩展版本：从“声波回声 (Sound) ”**机会结点**到“是否钻井 (Drill) ”**决策结点**加上一条信息弧。
- (c) 进一步扩展：我们可以先决定是否做“测试 (Test) ”。

图 10.13

- (a) 以影响图表示的一个 POMDP (部分可观测马尔可夫决策过程)。 z_t 是隐藏的世界状态。我们隐含采用“不遗忘 (no forgetting) ”假设，即 a_t 实际上从所有以往观测 $x_{1:t}$ 都有信息弧指向它。
- (b) 以影响图表示的一个 MDP (完全可观测的马尔可夫决策过程)。

图 10.14 一些有向图模型 (DGMs)。

10.6 影响 (决策) 图

我们可以用一种称为**决策图/影响图** (Howard & Matheson, 1981; Kjaerulff & Madsen, 2008) 的图形记法来表示多阶段 (贝叶斯) 决策问题。这是在有向图模型的基础上加入

- **决策结点** (也叫**动作结点**, 用**矩形**表示) ,
 - **效用结点** (也叫**价值结点**, 用**菱形**表示) 。
- 原本的随机变量称为**机会结点** (用**椭圆**表示) 。

图 10.12(a) 给出了著名的“野猫式找油”问题的简单例子。你要决定是否钻井。动作有两种: $d = 1$ 表示钻, $d = 0$ 表示不钻。假设自然有 3 个状态: $o = 0$ 干井; $o = 1$ 有点油 (wet); $o = 2$ 很多油 (soaking)。先验信念 $p(o) = [0.5, 0.3, 0.2]$ 。最后, 需要指定**效用函数** $U(d, o)$ 。由于状态和动作都是离散的, 可用一张表 (类似 DGM 里的 CPT) 表示。下面是假设的 (美元) 数值:

(第一张表: 当 $d = 0$ 时效用都为 0; 当 $d = 1$ 且 $o = 0, 1, 2$ 时分别为 $-70, 50, 200$ 。)

可见: 不钻既无成本也没收益; 钻到干井亏 \$70; 钻到有油赚 \$50; 钻到很多油赚 \$200。若选择“钻”, 你的**先验期望效用**为

$$EU(d = 1) = \sum_{o=0}^2 p(o) U(d, o) = 0.5 \cdot (-70) + 0.3 \cdot 50 + 0.2 \cdot 200 = 20 \quad (10.48)$$

不钻的期望效用为 0。故**最大期望效用**

$$MEU = \max \{EU(d = 0), EU(d = 1)\} = \max \{0, 20\} = 20 \quad (10.49)$$

因此最优动作为钻井:

$$d^* = \arg \max \{EU(d = 0), EU(d = 1)\} = 1. \quad (10.50)$$

现在稍作扩展: 先做一个**声波探测** (sounding) 来估计井的状态。观测 s 取三值: $s = 0$ 弥散回波 (暗示没油); $s = 1$ 开口回波 (有些油); $s = 2$ 闭合回波 (很多油)。因为 S 由 O 产生, 于是在图中加入 $O \rightarrow S$; 并假设在决定是否钻井之前可看到测试结果, 所以再从 S 到 D 加一条信息弧。如图 10.12(b)。

用下表的条件分布来建模传感器可靠性 $p(s | o)$:

(第二张表: 行 $o = 0, 1, 2$, 列 $s = 0, 1, 2$, 分别为 $[0.6, 0.3, 0.1], [0.3, 0.4, 0.3], [0.1, 0.4, 0.5]$ 。)

若做了探测并观测到 $s = 0$, 则油的**后验**为

$$p(o | s = 0) = [0.732, 0.219, 0.049]. \quad (10.51)$$

此时选择动作 d 的**后验期望效用**

$$EU(d \mid s = 0) = \sum_{o=0}^2 p(o \mid s = 0) U(o, d). \quad (10.52)$$

当 $d = 1$,

$$EU(d = 1 \mid s = 0) = 0.732 \times (-70) + 0.219 \times 50 + 0.049 \times 200 = -30.610.53)$$

若 $d = 0$, 仍为 0。因此观察到 $s = 0$ 时不钻更好, 这符合直觉。

同理可得: 若 $s = 1$, 则 $EU(d = 1 \mid s = 1) = 32.9 > 0$; 若 $s = 2$, 则 $EU(d = 1 \mid s = 2) = 87.5 \gg 0$ 。故最优策略 $d^*(s)$ 为:

- 若 $s = 0$, 选 $d^*(0) = 0$, 收益 \$0;
- 若 $s = 1$, 选 $d^*(1) = 1$, 收益 \$32.9;
- 若 $s = 2$, 选 $d^*(2) = 1$, 收益 \$87.5。

你的**期望利润/最大期望效用**为

$$MEU = \sum_s p(s) EU(d^*(s) \mid s), \quad (10.54)$$

其中测试结果的先验边缘分布

$$p(s) = \sum_o p(o) p(s \mid o) = [0.41, 0.35, 0.24]. \quad (10.55)$$

因此

$$MEU = 0.41 \times 0 + 0.35 \times 32.9 + 0.24 \times 87.5 = 32.2. \quad (10.56)$$

若我们还能选择是否做测试, 如图 10.12(c) 增加“测试”结点 T : 当 $T = 1$ 时进行测试, S 由 O 决定 (同上); $T = 0$ 时不测, S 进入一个“未知”状态。做测试还会产生某个成本。

值不值得测? 看“知道测试结果 (即 S 的状态) 后, MEU 提升了多少”。不测时, 式 (10.49) 给出 $MEU = 20$; 测试并按最优策略行动时, 式 (10.56) 给出 $MEU = 32.2$ 。提升 \$12.2 称为**完全信息的价值 (VPI)**。因此只要测试成本 $< \$12.2$, 就应当做测试。用图模型表述: 变量 T 的 VPI 可通过比较**基础影响图 I** 的 MEU 与在同一图上向所有决策结点添加 T 的信息弧后的 MEU 的差得到:

$$VPI = MEU(I + T \rightarrow D) - MEU(I). \quad (10.57)$$

可以改造**变量消除算法** (第 20.3 节) 来在给定影响图时计算最优策略。这些方法本质上是从最后一步**反向递推**, 在假设后续动作都最优的前提下, 逐步求出当前的最优决策 (参见 Lauritzen & Nilsson, 2001; Kjaerulff & Madsen, 2008)。

我们可以继续扩展模型。例如，考虑一个随时间展开的系统：不断“测试—观测—行动—转到下一口井”。实际上，机器人、商业、医疗、公共政策等许多问题，都可表述为随时间展开的影响图 (Raiffa, 1968; Lauritzen & Nilsson, 2001; Kjaerulff & Madsen, 2008)。

这一类通用模型如图 10.13(a)，称为**部分可观测马尔可夫决策过程 (POMDP)**：它基本上是在隐马尔可夫模型 (第 17.3 节) 上加入动作与奖励结点，用来描述智能体的“感知—行动”循环 (参见 Kaelbling 等, 1998)。其特例是**马尔可夫决策过程 (MDP)**，状态完全可观测，如图 10.13(b)；这更容易求解，因为只需把**观测到的状态映射到动作**，可用**动态规划**求解 (Sutton & Barto, 1998)。在 POMDP 中，仅有从 x_t 到 a_t 的信息弧还不足以唯一确定最优动作，因为状态不可完全观测；需要基于**信念状态** $p(z_t | x_{1:t}, a_{1:t})$ 来选择动作。由于信念更新是确定性的 (第 17.4.2 节)，可以把问题转化为“**信念状态 MDP**”。关于这类模型策略的计算，可参见 Kaelbling 等 (1998)；Spaan & Vlassis (2005)。

通俗解释 + 数值验证

1. 影响图=贝叶斯网 + 决策 + 效用

- 椭圆：不确定的世界（机会结点）。
- 矩形：你能选的动作。
- 菱形：好坏评估（钱/收益/效用）。
- **信息弧**：决策前可“看到”的变量；不改变概率，只改变**可用信息**。

2. 先验直接决策

- 不测、直接选钻：按先验 $p(o)$ 计算 EU 。上面算得 20，优于 0 \Rightarrow 钻。

3. 先测再决策（有信息弧 S→D）

- 先用 $p(s | o) + p(o)$ 算 $p(s)$ ：

$$p(s=0) = 0.6 \cdot 0.5 + 0.3 \cdot 0.3 + 0.1 \cdot 0.2 = 0.41,$$

$$p(s=1) = 0.35, \quad p(s=2) = 0.24 \text{ (与式 10.55 一致)}.$$

- 再用贝叶斯公式算 $p(o | s)$ 。例如 $s = 0$ ：

$$[0.3, 0.09, 0.02]/0.41 \approx [0.732, 0.219, 0.049].$$

- 计算条件期望效用：

$$s = 0 \Rightarrow EU(d=1 | s) = -30.5 \text{ (所以不钻)}$$

$$s = 1 \Rightarrow 32.9 \text{ (钻)}$$

$$s = 2 \Rightarrow 87.5 \text{ (钻)}$$
- 把三种结果按 $p(s)$ 加权： $MEU = 32.2$ 。

4. 做不做测试？—信息价值 VPI

- 不测的 $MEU = 20$ ；测试后的 $MEU = 32.2 \Rightarrow$ **信息价值 VPI = 12.2**。

- 若测试成本 $< 12.2 \Rightarrow$ 值得测；否则不测。

5. POMDP vs. MDP

- MDP：当前状态就能看到 \Rightarrow 直接学/算“状态→动作”的映射（策略）。
- POMDP：状态看不全 \Rightarrow 先维护“信念” $b_t(z) = p(z_t | \text{历史})$ ，策略是“信念→动作”。可把信念当作新的确定性状态，用动态规划/近似方法求解。

要点总结

- 影响图在贝叶斯网基础上加了决策（矩形）和效用（菱形），并用信息弧表示决策前可用的信息。
- 期望效用（EU）：给定信息/证据下，对每个动作求期望效用；MEU选择 EU 最大的动作/策略。
- 观测（测试）会改变后验，从而改变最优动作与 MEU。
- 信息价值（VPI）：加上信息弧后的 MEU 与基础 MEU 的差；若测试成本 $< VPI \Rightarrow$ 值得做测试。
- POMDP = HMM + 决策 + 奖励；策略基于信念状态；MDP 是其完全可观测特例，可用动态规划求解。
- 变量消除 / 逆推（动态规划式）方法可用于从影响图求最优策略。

基于本对话里我贴过的整章原文，输出：①知识点总结
(分点+关键词+用途) ②子章节关联 (依赖/对比/递进+关系句，附Mermaid可选) ③本章总结。

已思考 8s

① 知识点总结 (分点 | 关键词 | 用途)

- 图模型动机
 - 关键词：模块化、抽象、因子分解、平均
 - 用途：把高维联合分布 $p(x)$ 用少量参数与结构性假设（独立/条件独立）紧凑表示，支撑可计算的推断与学习。
- 链式法则与条件独立
 - 关键词：Chain rule、 $p(x_{1:T}) = \prod_t p(x_t | x_{1:t-1})$ 、马尔可夫假设
 - 用途：理解为何没有 CI 时参数爆炸；引出“只依赖父结点”的图结构。

- **图论术语**
 - 关键词：DAG、父/子/祖先/后代、度、拓扑序、团/极大团
 - 用途：建立贝叶斯网的语言，便于读图与写出因子分解。
- **有向图模型 DGM (Bayes Net)**
 - 关键词：**有序马尔可夫性质**、 $p(x) = \prod_t p(x_t | x_{\text{pa}(t)})$ 、CPD/CPT
 - 用途：把“只有父影响子”的 CI 编码进图；显著减少参数与计算量。
- **典型模型与应用**
 - **朴素贝叶斯 / TAN**：条件独立/树增强；文本分类、缺失特征易处理。
 - **马尔可夫链/高阶链**：序列建模；
 - HMM：**隐状态 + 观测模型**；语音、基因、NLP。
 - **医学诊断 (QMR/BN2O)**：Noisy-OR、泄漏节点；多病因→症状。
 - **遗传连锁**：基因型→表型、切换变量、马尔可夫重组；定位致病基因。
 - **有向高斯网**：线性-高斯 CPD，全局仍为多元高斯； $\Sigma = US^2U^\top$ （与 Cholesky 相关）。
- **推断 (Inference)**
 - 关键词：**后验** $p(x_h | x_v)$ 、**边缘化**、**似然/证据**、**树宽**
 - 用途：回答“已知部分变量，未知的是什么”；树/链上线性时间，复杂图上需近似（第20章）。
- **学习 (Learning)**
 - 关键词：MLE/MAP、**贝叶斯学习**、plate notation、**完全数据**、**缺失/隐变量**
 - 用途：
 - 完整数据：似然按图**分解**（每个 CPD 独立学习）；离散用 Dirichlet-**分类分布共轭**；
 - 缺失/隐变量：似然**不分解且非凸**，需 EM/变分/采样等近似。
 - 区分：CPD（建模时的条件分布） vs **满条件** $p(x_t | x_{-t})$ （推断/采样时用，只依赖**马尔可夫毯**）。
- **条件独立判定**
 - 关键词：**d-分离**、Bayes Ball、**解释型 (Explaining away)**
 - 用途：只看图就能断言 $A \perp B | E$ 。链/叉：给定中点→**关**；V 结构：默认**关**，给定中点或其后代→**开**。
- **马尔可夫性质与分解等价**
 - 关键词：**全局(G)**、**局部(L)**、**有序(O)**、**分解(F)**、**等价**

- 用途：三种马尔可夫性质彼此等价，并与因子分解性质互相推出；保证“看图写公式”是正确的。
 - **马尔可夫毯 & 满条件**
 - 关键词： $\text{mb}(t) = \text{pa}(t) \cup \text{ch}(t) \cup \text{copa}(t)$
 - 用途：局部推断/采样 (Gibbs) 只需毯内变量；
 $p(x_t | x_{-t}) \propto p(x_t | \text{pa}(t)) \prod_{s \in \text{ch}(t)} p(x_s | \text{pa}(s)).$
 - **I-map / 极小 I-map**
 - 关键词： $I(G) \subseteq I(p)$ 、minimal、删边单调
 - 用途：结构学习与可解释性；极小 I-map 在“不说假独立”的前提下再删就错。
 - **影响图 (决策图) 与信息价值**
 - 关键词：机会/决策/效用、信息弧、期望效用 EU、MEU、VPI
 - 用途：把“测一看一做—得分”的多阶段决策形式化；
 $VPI = MEU(I + T \rightarrow D) - MEU(I)$, 决定“是否值得测试”。
 - **POMDP / MDP**
 - 关键词：信念状态 $b_t(z) = p(z_t | x_{1:t}, a_{1:t})$ 、动态规划
 - 用途：在不完全可观测环境下选择动作；MDP 为可观测特例，易解；POMDP 需在信念 MDP 上解策略。
-

② 子章节关联 (依赖 | 对比 | 递进)

关系句 (简要)

- 10.1 (链式法则与 CI) → 为 10.1.5 (DGM 分解式) 提供动机与数学基础。
- 10.1.3–10.1.5 (图论与 DGM) → 是 10.2 各模型实例的**结构模板**。
- 10.2 (实例) → 为 10.3 (推断复杂度/方法) 与 10.4 (学习) 提供需求场景。
- 10.4.2 (完整数据分解学习) ≥ 10.5 (CI 判定) **相辅相成**：CI 让似然分解、学习可并行。
- 10.5.1 (d-分离/Bayes Ball) $\Rightarrow 10.5.2$ (G/L/O 等价) $\Rightarrow 10.5.3$ (马尔可夫毯/满条件)；递进给出从全局到局部的推断法则。
- 10.6 (影响图、VPI、POMDP/MDP) 在 DGM 之上加入**决策与效用**，把“认知→行动”闭环化。

(可选) Mermaid 总览

```
graph TD
    A[10.1 基础: Chain rule & CI] --> B[10.1.5 DGM: 因子分解]
    B --> C[10.2 例子: NB/TAN/MC/HMM/Noisy-OR/遗传/高斯网]
    C --> D[10.3 推断: 后验/树宽/近似]
    C --> E[10.4 学习: Plate/完整数据/缺失-隐变量]
    B --> F[10.5 CI判定: d-分离 & ...]
```

Bayes Ball] F --> G[10.5.2 G/L/O 等价 & 分解] F --> H[10.5.3 马尔可夫毯 & 满条件] B --> I[10.6 影响图: EU/MEU/VPI & POMDP/MDP]

③ 本章总结

- 本章以**有向图模型（贝叶斯网）**为核心，展示了如何用**结构化的条件独立**把高维联合分布**因子化**，从而在**参数上更节省、在推断/学习上更可计算**。
- **d-分离 + Bayes Ball**给出“**看图断独立**”的操作法；它与**局部/有序马尔可夫性质**以及**因子分解**互相等价，保证了“**看图写公式**”的正确性。
- 通过一组**代表性模型**（朴素贝叶斯/TAN、马尔可夫链/HMM、Noisy-OR 医学诊断、遗传连锁、有向高斯网），本章说明 DGM 的**表达力与实用性**。
- 在**学习方面**，完整数据时每个 CPD 可**独立估计**（离散用 Dirichlet 共轭）；缺失/隐变量时问题**非凸**，需要 EM/变分/采样。**满条件与马尔可夫毯**奠定了 Gibbs 等近似推断的**局部计算规则**。
- 最后，用**影响图**把“**不确定性建模**”扩展到“**最优决策**”（EU/MEU/VPI），并连接到 MDP/POMDP 等顺序决策框架，形成从**感知→推断→决策**的一体化视角。

一句话：DGM = “**结构 (CI) + 因子分解 + 局部计算**”，再加上影响图就变成“**结构化的不确定性下做最优决策**”。