Dong Wang

现代机器学习技术导论

2018 年 3 月 20 日

Springer

Contents

Chapter 1
机器学习概述

Chapter 2
线性模型

Chapter 3
神经模型

Chapter 4
深度学习

Chapter 5
核方法

Chapter 6
图模型

前面我们已经介绍了神经模型和核方法。神经模型是典型的数据驱动模型，包含大量参数，需要足够的数据量以确定这些参数；核方法中可调节的参数较少，不需要太多数据，但需要较强的先验知识来设计核函数的形式。这两种模型都取得了巨大成功，但都无法实现对复杂知识的处理。例如当数据生成过程比较复杂或维度间的依赖关系比较复杂时，我们希望把对这些过程或关系的先验知识作为输入来提高建模质量。神经模型和核方法都缺少利用复杂先验知识的有效机制。另一方面，在很多任务中我们的目标不仅是回归和分类，还需要对回归和分类的结果做深入分析。神经模型和核方法在很大程度上是“盲学习”，对数据的因果关系分析不足。

概率模型在很大程度上解决了上述问题，通过将先验知识形式化为变量间的相关性，可以对复杂系统进行有效刻画；同时，这一模型也保留了足够的灵活性，可基于数据学习模型中的参数，使之适应目标任务。这意味着概率模型是一种将领域知识和数据结合的有效工具。然而，当系统的变量较多时（如上百个变量），模型设计变得越来越困难。概率图模型以图的形式来描述变量间的关系，不仅可以更直观地描述目标问题，而且衍生出一套统一的推理方法和参数估计方法，极大简化了概率模型的构造和推理过程。本章将介绍概率图模型的基本概念和方法，并介绍基于该模型的推理方法和参数估计算法。

6.1 概率图模型简介

很多实际问题包含众多变量，且变量间具有复杂的依赖关系。如图6.1所示的一个气象预报系统，变量既包括气压、温度、湿度等常规观察量，

也包括季节这样的指示变量和卫星云图这样的图片资料。这些变量决定是否会产生降雨、降雪等天气事件，这些事件又决定是否会发生汛情，是否要提醒大家穿衣御寒等。在实际天气预报系统中，变量可能多达数万，这些变量之间互相影响，互相作用，形成复杂庞大的关系系统。概率模型是描述这些复杂系统的有效工具，通过对变量间的局部概率关系进行建模，即可利用贝叶斯推理方法进行事实推断。例如当气温持续升高时，发生汛情的可能性，或一个地区发生了降雨，临近较高海拔地区发生降雪的可能性，等等。然而，面对如此庞大的变量集合和如此复杂的概率关系，贝叶斯推理变得不再直观，甚至连模型的表达都会变得困难。

概率图模型是概率论和图论的巧妙结合。该模型将变量表示成图的节点，变量之间的相关性表示成图的边，通过定义变量间的局部相关性，即可通过图算法推理出任意两个变量之间的全局性概率关系。这一框架特别适合描述包含多元信息的复杂系统，使变量间的相关性变得一目了然; 同时，该框架提供了一套通用的推理方法和参数估计方法，极大降低了建模复杂性。值得说明的是，图模型本身更关注变量之间的拓朴结构，而不是变量间概率关系的具体形式。不同拓朴结构的推理和参数估计方法有很大区别，图模型关注同一拓扑结构下的通用算法。

概率图模型分为有向图模型和无向图模型两种，前者一般也称为信任网络（Belief Network）或贝叶斯网络（Bayesian Network），后者也称为马尔可夫随机场（Markov Random Field）。有向图的优点在于可直观表示随机变量间的因果关系，无向图则长于表示变量之间的概率独计性。本章将首先介绍这两种图模型，然后讨论这些模型上的推理和训练方法。关于图模型更多信息可参考 [11, 23, 14, 12]。

6.2 有向图模型

图 6.2给出一个简单的有向图模型，其中每一个节点表示一个随机变量，节点间的有向边代表变量间存在的相关性。从定性表达上看，有向图模型可直观表示变量间的因果关系，并通过图中节点的连通性判断变量或变量集合间的概率相关性；从定量表达上看，有向图模型定义了图中所有节点所代表变量的联合概率分布，形式化如下：

$$P(X) = \prod_i P(x_i|Pa(x_i)), \tag{6.1}$$

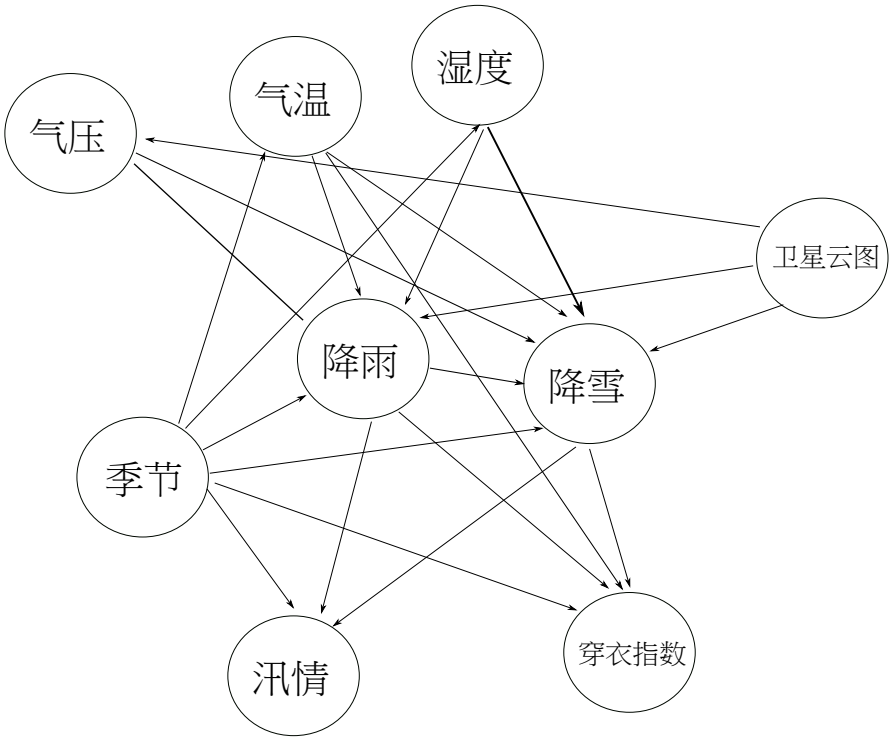


Fig. 6.1 气象预测系统的关系图，其中圆圈代表系统中包含的变量，有向箭头代表变量之间的因果关系。

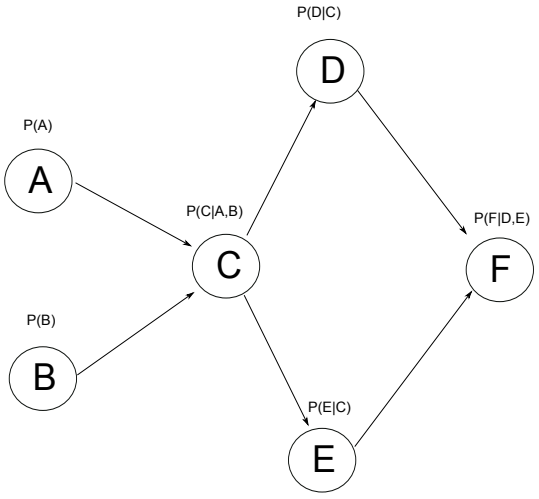


Fig. 6.2 简单的有向图模型，包括A,B,C,D,E,F五个变量，变量间的有向连接表示概率依赖关系。每个节点对应一个先验概率（无父节点）或条件概率（有父节点）。

其中$X = \{x_i\}$是有向图所包含的所有变量，x_i是第i个变量，$Pa(x_i)$是x_i的父节点变量集合。具体而言，有向图所包含的所有变量的联合概率是每个变量条件概率的乘积；如果该条件概率的条件变量集合为空，则意味着该变量不存在父节点，条件概率退化为先验概率。依此原则，图 6.2所表达的联合概率可写成如下形式：

$$P(A,B,C,D,E,F) = P(A)P(B)P(C|A,B)P(D|C)P(E|C)P(F|D,E),$$

其中A和B 没有父节点，因此对应的项为先验概率。

除了上述基础表示方法，有向图模型在应用中还经常包含若干扩展表示：（1）通常用灰色节点代表观测变量，用白色节点代表隐藏变量（Latent Variable）；（2）通常用圆圈代表连续变量，用方框代表离散变量；（3）有时会对某些节点加框，表示一组独立同分布的变量；（4）模型的参数通常用一些实心黑点表示。图 6.3给出一个扩展表示的例子，这一有向图模型对应的联合概率为:

$$P(A,B,C,D,G,E_1,E_2,...,E_N) = P(A)P(B|A)P(C|AB)P(D|B;\theta)P(G|C)\prod_{i=1}^{N}P(E_i|C,D,G).$$

有向图可以用于多种任务中，典型的三种如图 6.4所示，其中（a）表示预测任务，即根据模型预测出某一原因可能导致的结果；（b）表示推理任务，基于结果推理出导致该结果的原因；（c）表示分析任务，原因和结果已知，根据模型分析这些原因导致该结果的机制。值得一提的是，在有向图模型中，所有观察变量，不论是原因还是结果，都统一表示成图中的节点，这意味着传统模式识别概念中的数据x和数据标记t之间的区分变得不再明显，两者都是概率图模型中的观测变量。这事实上打破了监督学习和非监督学习的界限，将这两种方法统一到同一个概率学习和推理框架之中。

6.2.1 变量相关性判断

一个系统中两个变量或变量集合之间的相关性或条件相关性（等价地，独立性或条件独立性），对于系统设计具有很强的指导意义，是重要的领域知识之一。在设计模型时，通常由专家对两个变量之间是否存在概率相关性进行判断，如果存在某种相关性，则在有向图模型中加入相应的边。所有这

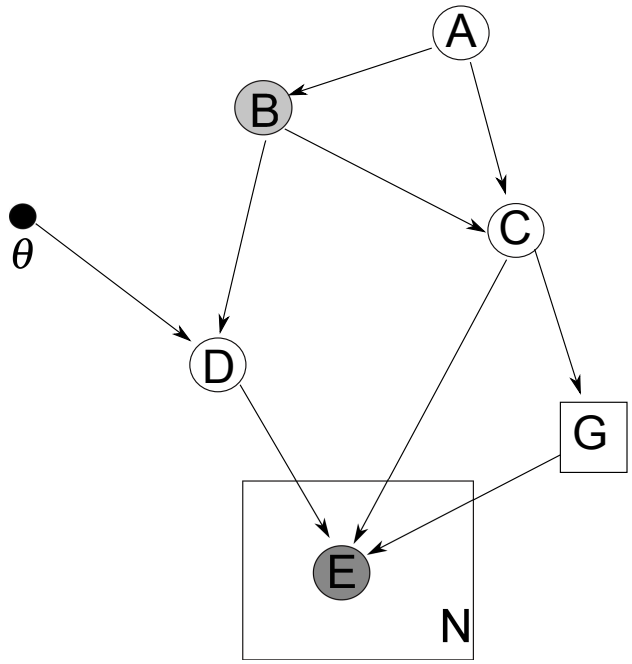


Fig. 6.3 带参数和组变量的有向图。A,C,D,G为隐藏变量，B,E为观测变量; A,B,C,D,E为连续变量，G为离散变量；A,B,C,D,G为单个变量，E为一组N个变量；$\boldsymbol{\theta}$是变量D 的条件概率的参数。

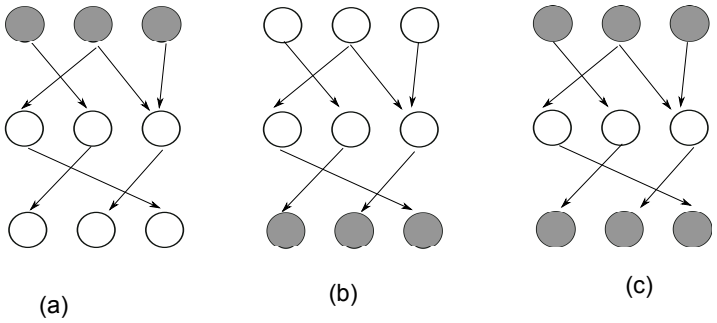


Fig. 6.4 基于有向图可以进行（a）预测，（b）推理，和（c）分析。灰色圆圈表示观测变量，白色圆圈表示隐藏变量。

些局部相关性决定了有向图的全局拓朴结构，而这一拓朴结构即隐性决定了变量或变量集合间的全局相关性。

从有向图模型的拓朴结构中解析变量间的相关性对于理解系统行为具有重要意义，在很多实际应用中具有直接价值。例如在前面讨论的天气预报的例子中，如果我们能从该系统的图模型中直接判断出气压和穿衣指数的相关性，则可迅速知道当气压发生变化时是否要发出穿衣提醒。

从大规模有向图中分析变量之间的独立性或相关性可通过考察连接这两个变量的所有可能路径是否被阻断来确定。所谓阻断，直观上说是某一变量的信息无法通过该路径传递给另一变量。如果两个变量间的所有路径皆被阻断，则这两个变量独立（对应路径中不存在观测变量情况）或条件独立（对应路径中存在观测变量情况），否则这两个变量相关或条件相关。

为了判断一条路径是否被阻断，我们需要判断路径上每个节点是否被阻断，为此需要分析路径节点的典型结构所具有的阻断性质。依中间节点与左右两个节点的依赖关系（即边的方向性），我们可将路径节点分为（a）头-尾结构，（b）尾-尾结构，（c）头-头结构三种。在每种结构中，中间节点可能为观测变量或隐变量两种，前者对应首尾两节点的条件相关性，后者对应（非条件）统计相关性。这三种典型结构如图 6.5所示，对这些结构的相关性分析如下。

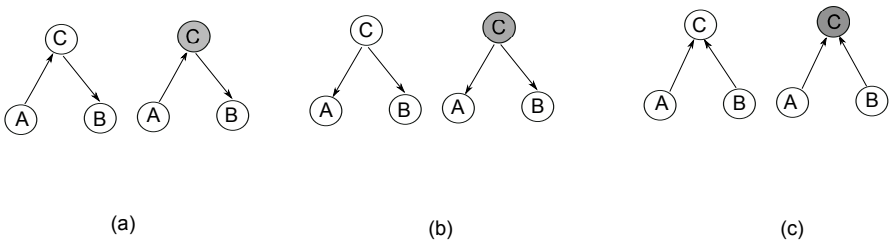


Fig. 6.5 有向图的三种基础结构。（a）头-尾结构，当C不可见时，A与B相关，当C可见时，A与B条件独立；（b）尾-尾结构，当C不可见时，A与B相关，当C可见时，A与B 条件独立；（c）头-头结构，当C不可见时，A与B 独立，当C可见时，A 与B条件相关。

- 对头-尾结构（图 6.5(a)），当中间节点C 没有被观察到时，A与B具有相关性，因为$P(AB) \neq P(A)P(B)$；当中间节点C被观察到时，A与B条件独立，因为

$$P(AB|C) = \frac{P(ABC)}{P(C)} = \frac{P(A)P(C|A)P(B|C)}{P(C)} = \frac{P(AC)P(B|C)}{P(C)} = P(A|C)P(B|C).$$

从直观上说，当C未被观察到时，A 的发生有可能引起C的发生，进而引起B的发生，因此A与B相关；当C被观察到时，C成为定值，A无论发生不发生，都不会引起C的变化，所以A和B条件独立。

- 对尾-尾结构，当图中的C 点没有被观察到时，A、B之间具有相关性，因为$P(AB) \neq P(A)P(B)$；当C被观察到时，A与B条件独立，因为

$$P(AB|C) = \frac{P(ABC)}{P(C)} = \frac{P(C)P(A|C)P(B|C)}{P(C)} = P(A|C)P(B|C).$$

直观来说，在C未被观察到的时候，A事件如果发生的概率很大，说明A事件的原因C发生的概率很大，而C也是B事件的原因，因而B发的概率也很大，因此A和B相关。当C事件被观察到时，A与B发生的概率完全由C的观察值确定，因此A与B条件独立。

- 对头-头结构，当C未被观察到时，A与B独立，因为：

$$P(AB) = \sum_C P(ABC) = P(A)P(B)\sum_C P(C|A)P(C|B) = P(A)P(B).$$

当C被观察到时，A和B条件相关，因为$P(AB|C) \neq P(A|C)P(B|C)$。直观上说，当C未被观察到时，A和B发生的概率完全由其各自的先验概率决定，因此是独立事件。但当C是观测变量时，如果C发生，说明A和B都有一定概率发生从而引发了C。由于A和B共同决定C的发生，那么如果当A发生的概率非常大时，则事件C已经有了合理的解释，因而事件B发生的概率就会减小，反之亦然。这种现象称为Explain Away。

我们用一个实例来理解头-头结构中的Explain Away现象。如图 6.6所示，当母亲在照顾新生婴儿时，如果发现新生儿吮吸自己的手指，那么很大可能性是婴儿饿了。如果此时又观察到婴儿开始哭闹，那么“天气炎热”这件事的概率就会非常小。这是为什么呢？假设我们认为，婴儿哭闹只会由“天气炎热”和“饥饿”这两件事引起，当我们已经观察到婴儿开始哭闹，并且确定了婴儿正处于饥饿状态中，那么我们可以认为，婴儿哭闹就是由于饥饿引起的。此时，“天气炎热”这件事发生的可能性就不大了。

基于上述三种基础结构的相关性分析，我们可以判断一条路径的相关性，进而判断任意两个变量间的相关性。这可以形象理解成一个小球（称为贝叶斯球）从节点A 通过任一路径向节点B滚动，途中会经过上述三种基础结构中的任何一种。当所经基础结构中的两端点独立（C为隐变量）或条件独立（C为观测变量）时，该路径发生阻断，或称为D-Separation。当

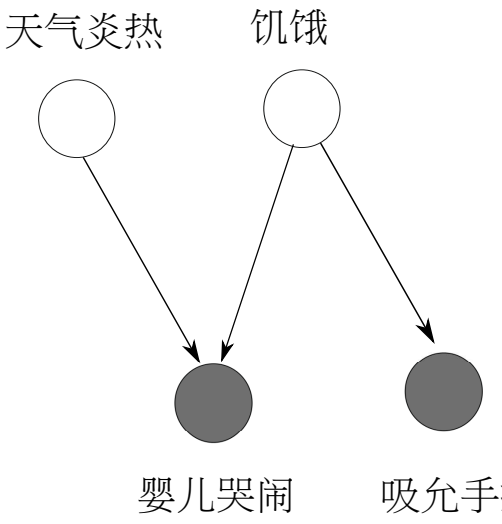


Fig. 6.6 有向图中头-头结构存在Explain Away现象。婴儿吸允手指时，则可判断婴儿饥饿，这时如果发现婴儿哭闹，则很大可能是由于饥饿造成的。这是因为饥饿这一原因已经足够解释婴儿为什么哭闹，因此天气炎热的可能性会相应下降。

从A到B的所有路径都被阻断时，则可判断这两个变量是独立的（对应路径中不包含观测变量情况）的或条件独立的（对应路径中包含观测变量情况）。

图 6.7给出两个相关性判断的例子。在图（a）中，C点已经被我们观测到，贝叶斯球从A点开始滚动，当滚动到E时，由于E到F是头-头结构，所以观察它本身以及其子节点是否被观测到。因为C被观测到，所以可认为E点不被阻断，球可以通过该节点。当球滚到F 时，由于F是尾-尾结构，且F没有被观察到，所以F节点导通，球可以滚到B 。综上所述，球可以从A经过E、F两个节点到达B点，所以A和B关于C条件相关。在图（b）中，贝叶斯球从A点开始滚动，当滚到E时，E是头-头结构且E和E的子节点都不是观测变量，因此路径被阻断，球无法通过，所以A和B独立。不仅如此，即便球在E点没被阻断，滚到F时也会被阻断，因为F是尾尾结构且是观测变量，E和F条件独立。综合起来，A和B关于F点条件独立。

6.3 无向图模型

有向图模型用边的方向表示变量之间的因果关系，无向图模型的边是没有方向的，因此不描述因果关系，仅表示节点之间的相关性。一个典型的无

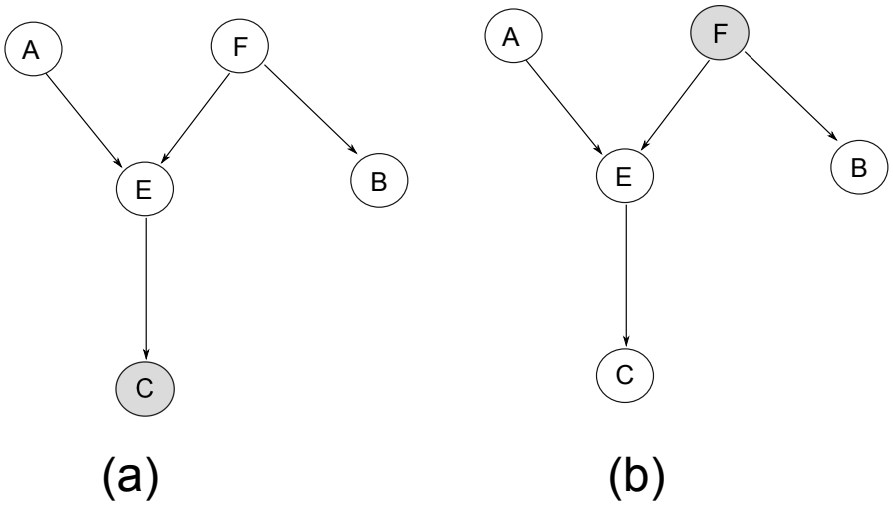


Fig. 6.7 贝叶斯球与D-Separation。在图（a）中，由于节点E存在头-头结构，且其子节点存在观测变量，因此其两个父节点A和F 没有被阻断; 同时，F是隐变量且是尾尾结构，因此也没有被阻断。由此，可判断A与B条件相关；在图（b）中，由于E的子节点中不存在观测变量，因此A和F被阻断，同时，F是观测变量且是尾-尾结构，因此被阻断。综合起来，A和B条件独立。

向图模型如图 6.8所示。同有向图一样，无向图也定义了一个联合概率。我们首先定义Clique的概念。所谓Clique，是指一个全连接节点集合，集合中所有节点之间互相连接。当一个Clique足够大，以致增加任何一个节点都会破坏这种全连接属性时，则称该Clqiue为最大Clique。图 6.8中用带颜色的椭圆标示出了一个有效的Clique划分。

用x_c表示一个Clique中的所有变量（注意不同Clique可能共享变量），$\psi_c(x_c)$是定义在x_c上的势函数（Potential Function），则一个无向图模型表示的联合概率定义为其中所有Clique的势函数的归一化乘积：

$$p(X) = 1/Z \prod_c \psi_c(x_c)$$

其中，Z为归一化因子。如果X是离散变量，则Z定义如下：

$$Z = \sum_X \prod_c \psi_c(x_c),$$

如果X为连续变量，则Z定义为：

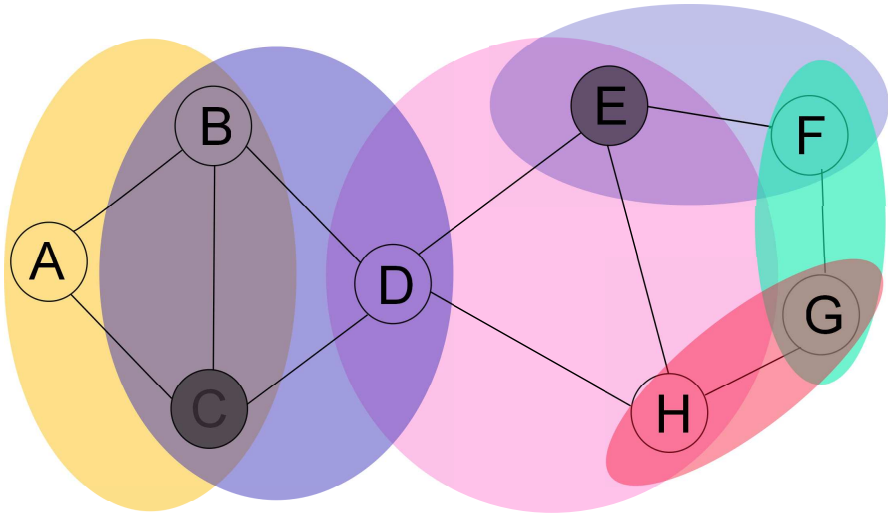
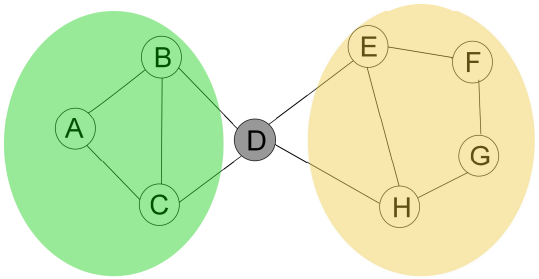


Fig. 6.8 无向图模型的例子。A,B,D,F,G,H为隐藏变量，C和E是观测变量。每种颜色的椭圆表示一个Clique。该图中共包含如下六个Clique: A-B-C, C-B-D, D-E-H, E-F, F-G, G-H。

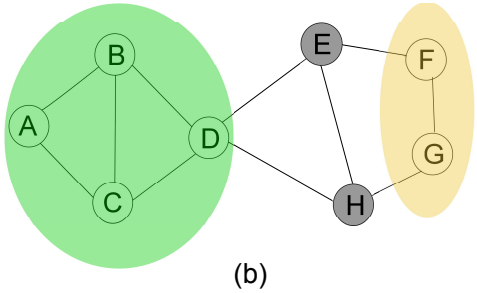
$$Z = \int \prod_c \psi_c(x_c) \mathrm{d}X.$$

6.3.1 变量相关性判断

无向图中两个变量间的相关性判断很直观：如果两个变量间有路径连接，则这两个变量相关，因为无向图中的任意一条边都表示其连接的两个变量是相关的，这一相关性可通过路径传导。然而，如果路径中的某个节点是观测变量，则该路径是阻断的。如果两个节点间的所有路径都是被阻断的，则这两个节点所代表的变量（依所有观测变量）条件独立。图 6.9中给出两个例子：当没有观察变量时，图中所有节点都是互相连接的，因而都具有相关性，但当某些变量成为观测变量后，图中所示的阴影部分即条件独立。相反，图 6.10中的观测变量并未对所有路径造成阻断，因此图中所有节点都是条件相关的。



(a)



(b)

Fig. 6.9 条件独立的无向图。灰色节点是观测变量，阴影部分表示要考察相关性的两个子图。在(a)(b)两种情况下，两个子图间的所有路径都被观测变量阻断，因此具有条件独立性。

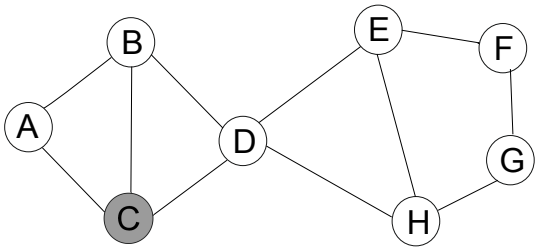


Fig. 6.10 条件相关的无向图。灰色节点是观测变量。由于该节点没有阻断任何路径，因此图中任意两个变量或子图条件相关。

6.3.1.1 有向图向无向图转化

给定一个有向图，我们可以将其转化成无向图，使得这两种概率图模型可以用统一的推理和训练算法来处理。以一个简单的链式结构来讨论这一转化过程，如图 6.11所示。

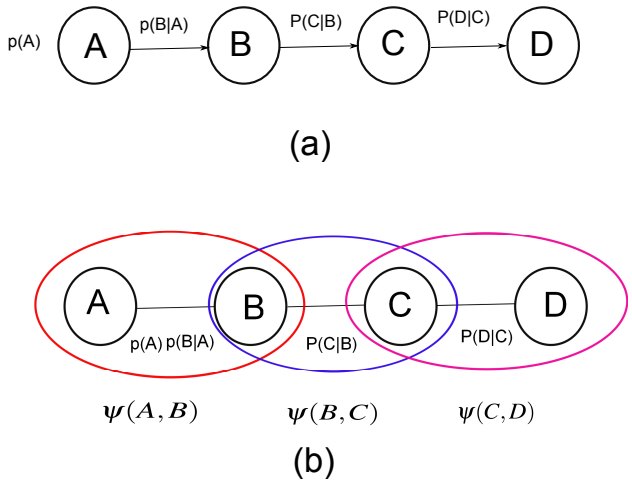


Fig. 6.11 链式有向图(a)转成无向图(b)。有向图联合概率表示中的每一项中所包含的所有变量对应无向图中的一个Clique。先验概率和子节点的条件概率组合成一个Clique。

首先，将有向图的联合概率写作：

$$P(A,B,C,D) = P(A)P(B|A)P(C|B)P(D|C)$$

可以定义Clique及其势函数如下:

$$\psi(A,B) = P(A)P(B|A),$$

$$\psi(B,C) = P(C|B),$$

$$\psi(C,D) = P(D|C).$$

将有向图中的每一项条件概率对应无向图中的一个Clique，从而完成转化。注意对没有父节点的变量，其先验概率需要和其子节点的条件概率结合在一起，对应成一个Clique。经过转化后，得到的无向图的联合概率写作：

$$P(A,B,C,D) = \frac{1}{Z}\psi(A,B)\psi(B,C)\psi(C,D).$$

对非链式结构的有向图可依同样准则将其转成无向图，唯一需要注意的是当有向图中某一节点具有多个父节点时，将该节点及其父节点对应成一个Clique，并在每对父节点间加入一条附加边，使得该节点集合满足Clique的全连接条件。在父节点间加入附加边这一过程称为Moralization, 生成的无向图称为Moralized Graph。图 6.12给出一个包含三个父节点的有向图转化成无向图的例子。

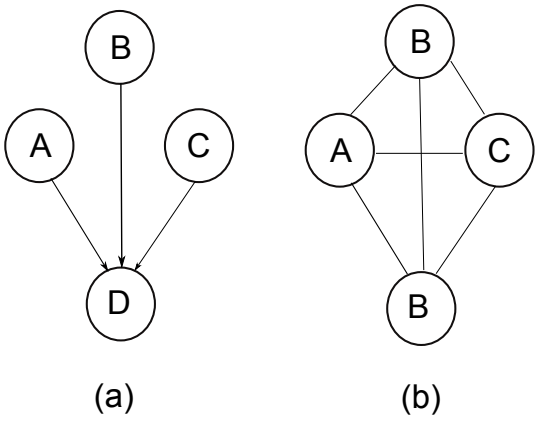


Fig. 6.12 有向图(a)转化成无向图(b)时，如果某个节点具有多个父节点，需要做Moralization。本例中，$P(D|A,B,C)$对应一个Clique，Moralization 操作在父节点A,B,C间加入附加连接，以保证这四个变量成为一个有效的Clique。

总体而言，将有向图转化成无向图需要三个步骤：（1）将有向图按其联合概率的因子分解方式分解成若干Clique，无父节点的节点与其子节点组成一个Clique；（2）如果一个Clique中包含多个父节点，需要在父节点间做Moralization以保证Clique的全连接性；（3）将有向边替换成无向边；（4）将有向图中的概率项作为无向图中对应的Clique的势函数。

值得说明的是，由于Moralization的存在，一些在有向图中存在的条件独立性会在无向图中消失。因此，从拓朴结构来看，转换前后两幅概率图模型所能代表的概率分布集合是不同的。但因为无向图的势函数是由有向图的条件概率定义的，基于这一特殊定义，转换前后两幅图所代表的联合概率分布是等价的。更一般来说，有向图模型和无向图模型所能代表的概率分布集合是不同的，一些概率分布只能由有向图能表示，另一概率分布只能由无向图表示，还有些概率分布两种图模型都无法表示。

6.3.2 有向图和无向图对比

有向图和无向图这两种概率图模型有各自的特点和优势。对于有向图来说，每个节点间具有明确的因果关系，直观形象；联合概率可以直接写出，不需要特别的正规化。因为节点间的因果关系由人为定义，这为引入领域知识提供了可能性，同时也对模型设计提出了更高要求。有向图模型的一个困难在于存在Explain Away问题，导致变量间的条件相关性不很直观，需要用Bayes Ball等方法来判断。

无向图模型不关心变量之间的因果关系，仅考虑它们之间是否具有相关性，因此建模比较简单，且不同变量之间是否条件相关可由两者之间的路径联通性直接得到。这一模型最大的问题在于Clique的势函数不是归一化的概率，需要计算归一因子Z，通常复杂度较高。

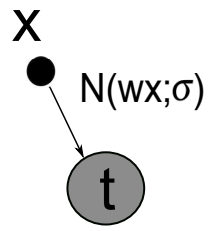
选择有向图模型还是无向图模型很大程度上由任务的性质和领域知识来决定。一般来说，图中节点的物理意义越清晰，对这些节点所代表的变量知识越丰富，越倾向于有向图模型。如本章开始所述的关于房屋中介的例子。反之，如果变量同质化，因果关系不明确，则倾向选择无向图模型，如在统计物理学中用于描述铁磁性的Ising 模型 [9]。

6.4 常用概率图模型

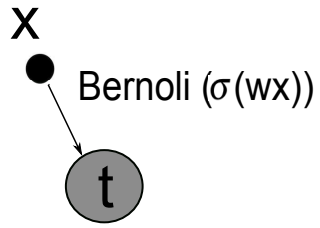
本书前几章提到的很多模型都属于概率图模型，如第 2章介绍过的线性回归和Logistic回归可以认为是包含一个变量的有向图模型，Probabilistic PCA（PPCA）和Probabilistic linear discriminative analysis （PLDA）是包含两个变量的有向图模型。第 3章介绍过受限玻尔兹曼机（RBM）是典型的无向图模型。图 6.13给出上述几种模型的图模型表示。本节将对另外三种常见的概率图模型做简要介绍，即高斯混合模型、隐马尔可夫模型、条件随机场。

6.4.1 高斯混合模型

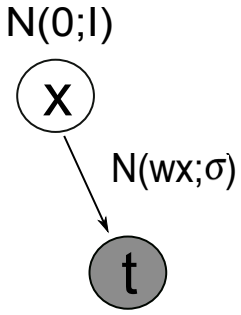
高斯混合模型（Gaussian Mixture Model, GMM）用若干高斯分布的加权平均来描述一个复杂分布。如果混合数足够多，GMM可以逼近任何复杂的



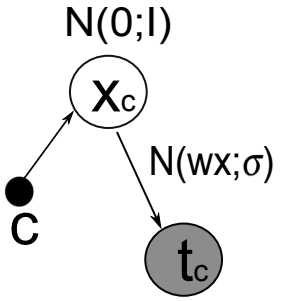
(a) Linear Regression



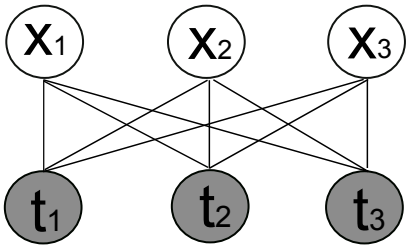
(b) Logistic Regression



(c) PPCA



(d) PLDA



(e) RBM

Fig. 6.13 几种简单的概率图模型。(a)为线性回归模型，其中只有观察变量t是随机的，服从以$\mathbf{w}^T\mathbf{x}$为中心的高斯分布；(b)为Logistic回归模型，随机变量t符合以$\boldsymbol{\sigma}(\mathbf{w}^T\mathbf{x})$为参数的Bernoli分布；(c)为PPCA模型，包括两个随机变量$\mathbf{x}$和$\mathbf{t}$，$\mathbf{x}$的先验概率是标准正态分布，$\mathbf{t}$的条件概率是为以$\mathbf{W}\mathbf{x}$为中心的高斯分布；(d)为PLDA模型，包括一个类别参数c，每一类的类中心$\mathbf{x}$服从先验概率$N(\mathbf{0},\mathbf{I})$，观察值$\mathbf{t}$服从以$\mathbf{W}\mathbf{x}$为中心的高斯分布; (e)为受限玻尔兹曼机，是一个无向图模型，第i个Clique包含一个(x_i,t_j)对，势函数为x_i和t_j的二阶函数。

连续分布，因而被广泛应用在各种实际系统中。特别是，GMM模型中涵盖了有向图模型的基础概念，适合初学者作为分析实例进行研究。

模型表示

我们从最简单的高斯分布开始。一个一维变量x的高斯分布具有如下概率密度函数形式：

$$p(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp[-\frac{1}{2}\frac{(x-u)^2}{\sigma^2}] \tag{6.2}$$

其中μ,σ为模型参数。容易证明，如果x符合上式所示的高斯分布，则其期望和方差分别为：

$$E(x) = \mu,$$

$$Var(x) = \sigma^2.$$

可见高斯分布的性质完全由其一阶和二阶统计量决定，因此通常将高斯分布写成如下形式:

$$x \sim N(\mu,\sigma^2).$$

现在假设某一变量x符合高斯分布，但对高斯分布的具体参数μ和σ 未知，我们可以很容易通过x的一个样本集合对μ和σ进行估计，即模型训练。该训练一般选择最大似然准则，即所选参数应使该模型生成样本集合的概率最大。以$\{x_n\}$表示样本集合，则似然函数有如下形式：

$$L(\mu,\sigma) = \prod_{n=1}^{N} p(x_n;\mu,\sigma).$$

对上式求对μ和σ的偏导数并使之等于零，可得到如下最大似然估计：

$$\tilde{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n,$$

$$\tilde{\sigma} = \frac{1}{N}\sum_{n=1}^{N}\{x_n - \tilde{\mu}\}^2.$$

当样本集足够大时，上述估计得到的参数与实际参数的误差趋近于无穷小。可见，对单高斯模型的参数估计并不存在原则上的困难。

实际任务中完全符合高斯分布的情况不多，特别是很多数据具有多峰性质（概率函数中具有多个极大值点），这时用单峰的高斯分布会产生较大偏差。人们将高斯模型扩展成GMM来描述复杂的数据分布。GMM假设数据由多个高斯源产生，每个高斯源s_k以一定的先验概率π_k被激发，被激发的高斯源再以$N(\mu_k, \sigma_k)$为概率密度函数生成一个观察变量x_n。通过这一生成过程得到的概率密度函数可写成如下形式：

$$p(x) = \sum_{k=1}^{K} \pi_k N(x; \mu_k, \sigma_k^2)$$

或:

$$x \sim \sum_{k=1}^{K} \pi_k N(\mu_k, \sigma_k^2).$$

这一模型中，每个高斯源称为一个高斯成分，先验概率π_k称为第k个高斯成分的权重。

上面讨论都是基于一维数据，但相似结论很容易在多维数据上得到。图6.14给出一个一维数据上GMM的例子。

参数估计

与高斯模型相比，GMM模型的复杂度提高了很多。这一复杂度不在于由一个高斯分布变成了K个，而在于模型中出现了隐变量，即每次数据生成时，哪一个高斯成分被激发是不可见的。隐变量的存在意味着观察到一个采样点x_n时，我们看到只是部分数据，缺失了哪个高斯成分被激发的事件z_n。注意，z_n是离散变量，在1到K之间取值。引入隐变量z后，GMM模型可表示为一个有向图模型，如图 6.15 所示。

隐藏变量z_n的存在使得GMM模型参数的估计变得不再直观。后面我们会看到，隐藏变量是概率模型最重要的优势，同时也是最主要的困难。对于GMM模型，我们可以考虑这样一种参数估计思路：假设我们已经知道了每个数据点x_n所对应的高斯成分z_n ，则可对训练数据依高斯成分划分为K类，对每类分别求参数$\{\mu_k, \sigma_k\}$，先验概率$\{\pi_i\}$ 可依z_n在K类上的分布比例确定。现在的问题是我们不知道z_n的确切值，因此无法对数据集进行划分。一种解决办法是不对数据集依z_n进行硬性划分，而是依后验概率$P(z_n|x_n)$对数据进行“软性划分”。换句话说，x_n不会被硬性归到某一个高斯成分，而是在每个高斯成分中都占一定比例，这一比例由$P(z_n|x_n)$确定。基于这一软性划分，每一个高斯成分即可依前一节所述的单高斯模型参数估计方法进行模型训练。

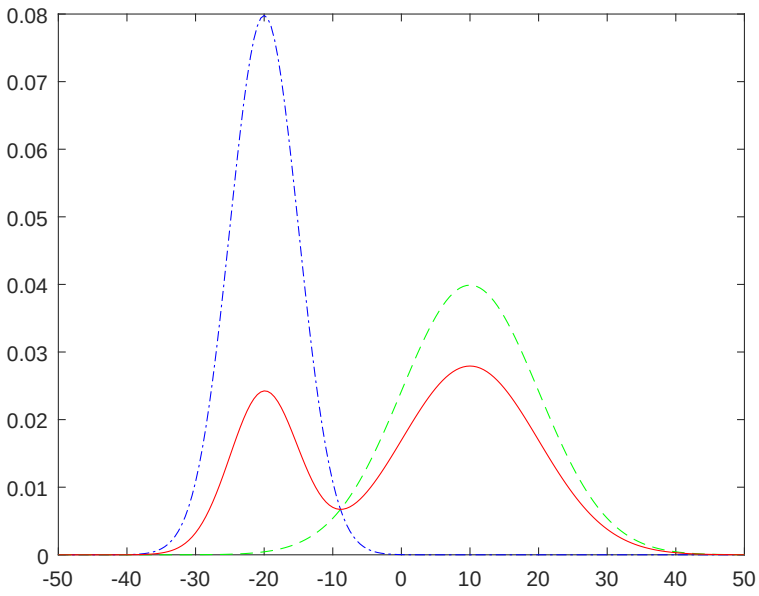


Fig. 6.14 一维数据的混合高斯模型。其中蓝色点划线为高斯成分$N(-20,25)$，绿色虚线为高斯成分$N(10,100)$，红色实线代表这两个高斯成分按$\pi = [0.3,0.7]$混合起来的高斯混合分布。

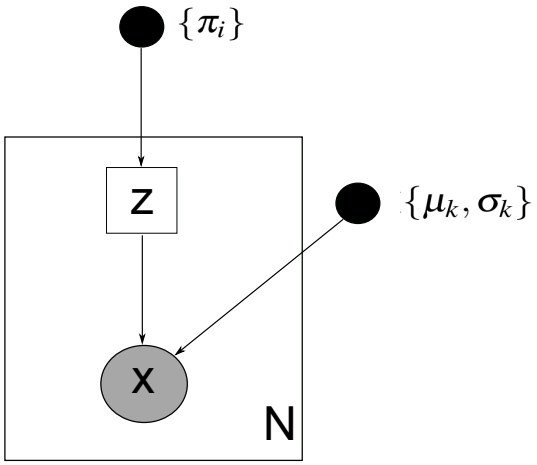


Fig. 6.15 高斯混合模型的有向图模型。

将上述过程形式化为两步：第一步求z_n的后验概率，可理解为对隐变量赋值，从而得到模型优化所需的全数据信息; 第二步依全数据进行模型参数估计。依贝叶斯公式，后验概率的计算方式如下：

$$p(z_n|x_n) = \frac{p(x_n|z_n)p(z_n)}{p(x_n)} = \frac{p(x_n|z_n)p(z_n)}{\sum_k p(x_n|z_n)p(z_n)}.$$

其中$p(z_n = k) = \pi_k$ 为第k个高斯成分的先验概率。基于这一后验概率，即可对高斯成分的参数进行估计。对每个高斯成分进行估计时，训练数据包括训练集中的所有样本，但每个样本以$p(z_n = k|x_n)$ 为比例对第k个高斯成分贡献统计量。为简便起见，记这一比例为r_{nk}。简单推导可得如下参数估计公式：

$$\mu_k = \frac{1}{\sum_n r_{nk}} \sum_{n=1}^{N} r_{nk} x_n,$$

$$\sigma_k = \frac{1}{\sum_n r_{nk}} \sum_{n=1}^{N} r_{nk} (x_n - \mu_k)^2.$$

对先验概率π_k，我们可以用所有数据在每个高斯成分上的贡献比例来估计。

$$\pi_k = \frac{\sum_{n=1}^{N} r_{nk}}{N}.$$

注意上述参数估计公式并不是一个闭式解，因为r_{nk}本身依赖当前模型参数。然而，这些公式确实提供了一种迭代求解方法，从一个初始参数开始，计算r_{nk}和模型参数优化交替进行。这一方法称为Expectation-Maximization （EM）算法，其中Expectation 是指依r_{nk}得到一个似然函数的期望，Maximization是指对该似然函数期望进行最大化 [5]。

值得说明的是，EM算法得到的并不是全局最大似然解。首先，GMM的似然函数并不是一个凸函数，这一点从高斯成分间具有对称性即可看到，因此很难得到一个全局最大似然解；其次，EM算法的M步并不是在最大化似然函数，而是似然函数依当前后验概率得到的期望。后面我们会看到，这一期望事实上是似然函数的一个下界，通过迭代优化这一下界函数，EM 算法会收殓到似然函数的一个局部最大值。在很多情况下，这一局部最优解已经完全可以满足精度要求。

6.4.2 隐马尔可夫模型

模型表示

GMM是静态模型，不能很好描述数据的时序相关性。隐马尔可夫模型（Hidden Markov Model, HMM）是简单的时序概率模型，在语音识别、自然语言处理、金融数据分析等众多任务中有广泛应用。我们首先介绍马可夫链，然后将其扩展到HMM模型。

一个马可夫链$\mathbf{q} = [q_1, ..., q_T]$是具有马尔可夫性质的事件序列。所谓马尔可夫性质，是指某一时刻的状态q_t的概率分布只与前一时刻的状态q_{t-1}有关。因此，一个马尔可夫链的概率由一个初始概率分布$\boldsymbol{\pi}$和一个状态转移矩阵A确定，其中A_{ij}为由状态i跳到状态j的概率。具体概率形式如下：

$$P(q_1 = i) = \pi_i$$

$$P(q_t = j | q_{t-1} = i) = A_{ij}(t) \qquad i, j = 1, ..., T.$$

如果转移概率与时间无关，即$A_{ij}(t)$对所有t都是相等的，则该马尔可夫链称为齐次马尔可夫链。大多数应用中，我们只关心齐次马尔可夫链。齐次马尔可夫链的概率计算如下：

$$P(\mathbf{q}) = \pi_{q_1} \prod_{i=1}^{T-1} A_{q_t q_{t+1}}.$$

隐马可夫模型（HMM）是马尔可夫链的一个扩展。在HMM中，状态取值不可直接观测，只能通过另一个随机变量间接推理得到。具体来说，HMM定义了一个生成模型，该模型首先由一个马尔可夫链随机生成一个状态序列，再由该随机序列的每个状态随机生成一个观测值，从而生成一个可见的观测序列。这一观测序列可以是连续值，也可以是离散值。生成连续观测值的HMM称为连续HMM，生成离散值的HMM称为离散HMM。可见，HMM模型具有双重随机性，既有状态上的不确定性，也有生成观测值上的不确定性。与GMM相比，HMM模型考虑到时序上的前后依赖关系（转移矩阵A），因此是一种动态时序模型，可以用来描述一个依时间发展的随机过程；同时，这一模型引入的马尔可夫假设极大降低了模型复杂度。基于此，HMM模型在很多领域得到广泛应用。图 6.16给出了一个离散HMM的例子，其中每个状态按不同的Multinomial分布生成观测值。

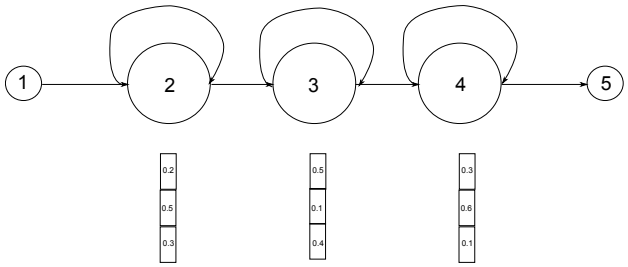


Fig. 6.16 离散链式隐马可夫模型结构，包括一个开始状态，三个输出状态和一个结束状态。其中三个输出状态可生成观测值，每个状态按不同的Multnomial分布生成若干离散观测值。

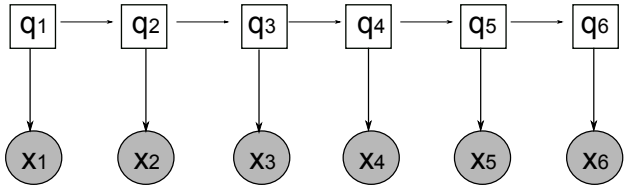


Fig. 6.17 对应状态转移图 6.16的概率图模型。

和GMM模型类似，HMM模型也可以表示成一个有向概率图，其中每一时刻的状态和观测变量都是图中的节点，相邻时刻的状态具有马尔可夫相关性，因此由有向边相连。图 6.17给出对应图 6.16的概率图模型。在这一图模型中，参数包括初始概率分布π和状态转移概率矩阵A，同时定义了给定某一状态的条件概率分布$p(x_t|q_t;b)$，其中b是该概率分布的参数。给定一个观测序列$\mathbf{x}=[x_1,...,x_T]$，如果状态序列为$\mathbf{q}$，则该图模型表示的联合概率分布如下：

$$p(\mathbf{x},\mathbf{q};\theta)=\pi_{q_1}p(x_1|q_1;b)\prod_{t=2}^{T}A_{q_{t-1}q_t}p(x_t|q_t;b). \tag{6.3}$$

对所有可能的$\mathbf{q}$做边缘化，得到$\mathbf{x}$的生成概率如下：

$$p(\mathbf{x};\boldsymbol{\theta})=\sum_{q}\pi_{q_1}p(x_1|q_1;b)\prod_{t=2}^{T}A_{q_{t-1}q_t}p(x_t|q_t;b). \tag{6.4}$$

其中$\boldsymbol{\theta}=\{\pi,A,b\}$是模型参数。$p(x_t|q_t;b)$既可以是离散的，也可以是连续的，前者适用于离散观测值（即离散HMM），后者适用于连续观测值（即连续HMM）。一般离散观测值的条件概率是个Multinomial分布，连续观测值的

条件概率多采用GMM模型。以GMM为输出概率的HMM可以认为是GMM模型的多状态扩展，而GMM可认为是只有一个状态的HMM。

参数估计

隐马可夫模型的参数是一个三元组$\boldsymbol{\theta} = \{\pi, A, b\}$，模型训练的目的是基于一系列观察样本$\{\mathbf{x}_n\}$，确定一个三元组$\boldsymbol{\theta}$使得如下似然函数最大化：

$$L(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n; \boldsymbol{\theta}).$$

显然这一似然函数不是一个凸函数，因此不存在全局最优解。和GMM模型的情形类似，HMM训练中的主要困难在于模型中存在隐变量，即每一个观测值所对应的状态。另外，如果每个状态的条件概率是个GMM模型，每个观测值对应的高斯成分也是个隐变量。在下面推导中，我们假设状态条件概率为单高斯连续分布。

和GMM模型的参数估计类似，我们可以用EM算法对HMM进行参数估计。该算法基于当前模型参数计算隐变量的后验概率和依此概率得到的似然函数的期望，再最大化该期望得到新的参数估计。具体而言，设当前模型参数$\boldsymbol{\theta}'$，对每个观测序列$\mathbf{x}$可以计算状态路径$\mathbf{q}$的后验概率$p(\mathbf{q}|\mathbf{x}; \boldsymbol{\theta}')$，再依此计算似然函数的期望如下：

$$\tilde{L}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{q} p(\mathbf{q}|\mathbf{x}_n; \boldsymbol{\theta}') \ln p(\mathbf{x}_n, \mathbf{q}; \boldsymbol{\theta}). \tag{6.5}$$

上式是一个凸函数，因此可求得全局最优解。注意，当$\boldsymbol{\theta} = \boldsymbol{\theta}'$时，$L(\boldsymbol{\theta}) = \tilde{L}(\boldsymbol{\theta})$，否则$\tilde{L}(\boldsymbol{\theta}) < L(\boldsymbol{\theta})$。这意味着$\tilde{L}(\boldsymbol{\theta})$是$L(\boldsymbol{\theta})$的下界。设对$\tilde{L}(\boldsymbol{\theta})$优化得到的参数是$\hat{\boldsymbol{\theta}}$，显然有：

$$\tilde{L}(\hat{\boldsymbol{\theta}}) \geq \tilde{L}(\boldsymbol{\theta}') = L(\boldsymbol{\theta}')$$

因此，这一优化得到的$\hat{\boldsymbol{\theta}}$是比$\boldsymbol{\theta}'$具有更大似然值的模型参数。

上述EM算法中比较困难的是计算式（6.5）中的后验概率$p(\mathbf{q}|\mathbf{x}; \boldsymbol{\theta}')$，以及处理两个加和操作，因为两者都需要计算大量的可能路径q。由于不同路径间有大量重叠，依基础公式（6.3）（6.4）（6.5）会造成大量计算浪费。

Baum 和Welch 给出了一个基于动态规划的快速计算方法，通常称为Baum-Welch 算法 [3, 1, 2]。将式（6.5）稍作整理如下：

$$\tilde{L}(\boldsymbol{\theta}) = \sum_n \sum_{\mathbf{q}} p(\mathbf{q}|\mathbf{x}_n;\boldsymbol{\theta}') \sum_t \ln f(q_{t-1}, q_t, x_{nt}), \tag{6.6}$$

其中，我们定义：

$$f(q_0, q_1, x_1) = \pi_{q_1} p(x_1|q_1;b),$$

$$f(q_{t-1}, q_t, x_t) = A_{q_{t-1}q_t} p(x_t|q_t;b) \quad t = 2,3,...T.$$

对上式整理可得：

$$\tilde{L}(\boldsymbol{\theta}) = \sum_n \sum_t \sum_q p(\mathbf{q}|\mathbf{x}_n;\boldsymbol{\theta}') \ln f(q_{t-1}, q_t, x_{nt}) \tag{6.7}$$

$$= \sum_n \sum_t \sum_{q_{t-1}, q_t} p(q_{t-1}, q_t|\mathbf{x}_n;\boldsymbol{\theta}') \ln f(q_{t-1}, q_t, x_{nt}), \tag{6.8}$$

其中第二个等式中对所有可能路径$\mathbf{q}$进行了边缘化，因此只保留了$\mathbf{q}$在$t-1$和t时刻分别处于状态q_{t-1} 和q_t的概率分布。经过这一变换，我们不必显式地对所有可能的路径$\mathbf{q}$进行计算和累加，因而极大节约了计算量。

然而，我们依然需要计算$p(q_{t-1}, q_t|\mathbf{x}_n;\boldsymbol{\theta}')$。依原始公式，计算这一后验需要边缘概率$p(\mathbf{x}_n;\boldsymbol{\theta}')$，因而同样要考虑所有可能路径。采用动态规划算法，可以极大减少计算量。对某一序列$\mathbf{x}$，定义前向概率$\alpha(t,s)$为在t时刻以状态s结束的所有路径的概率和：

$$\alpha(1,s) = \pi_s p(x_t|s)$$

$$\alpha(t,s) = \sum_{s'} \alpha(t-1,s') A(s's) p(x_t|s;b); \quad t = 2,3,...,T$$

同样，定义后向概率$\beta(t,s)$为在t时刻以状态s开始的所有路径的概率和：

$$\beta(T,s) = 1$$

$$\beta(t,s) = \sum_{s'} \beta(t+1,s') A_{ss'} p(x_{t+1}|s';b); \quad t = T-1, T-2, ..., 1$$

由此，可方便计算各种边缘概率如下：

$$p(\mathbf{x}, q_t) = \alpha(t,q_t)\beta(t,q_t),$$

$$p(\mathbf{x}, q_{t-1}, q_t) = \alpha(t-1,q_{t-1}) A_{q_{t-1},q_t} p(x_t|q_t;b)\beta(t,q_t).$$

$$p(\mathbf{x}) = \sum_s \alpha(t,s)\beta(t,s) = \sum_s \alpha(T,s) = \sum_s \pi_s p(x_1|s;b)\beta(1,s).$$

基于上述公式，可方便计算每一时刻所处状态的后验概率如下：

$$p(q_t|\mathbf{x}) = \frac{p(x, q_t)}{p(\mathbf{x})},$$

$$p(q_t, q_{t+1}|\mathbf{x}) = \frac{p(\mathbf{x}, q_t, q_{t+1})}{p(\mathbf{x})}.$$

注意上述计算都是基于当前模型参数$\boldsymbol{\theta}'$，因此有

$$p(q_{t-1}, q_t|\mathbf{x}) = p(q_{t-1}, q_t|\mathbf{x}; \boldsymbol{\theta}').$$

基于$p(q_{t-1}, q_t|\mathbf{x}; \boldsymbol{\theta}')$，即可计算似然函数的期望（6.8）并对其进行优化。

总结起来，由于存在隐变量（状态序列），HMM模型的似然函数形式比较复杂，直接优化该似然函数比较困难。EM算法提供了一种方法，在每一步优化时设计一个具有闭式解的下界函数，用迭代法求该下界函数的最优参数，从而实现对原似然函数的优化。然而，优化这一下界函数需要对大量可能的状态路径进行边缘化，因此计算效率较低。为提高效率，Baum-Welch算法基于动态规划原则对路径进行整理，避免重复计算，极大提高了计算效率。

6.4.3 线性条件随机场

模型表示

HMM模型定义的概率关系简单清晰，推理容易，但对复杂关系的描述能力不强。这是很多有向图模型的特点，因为要对条件概率关系做清晰的定义，考虑到推理和参数估计的困难，往往需要选择简单的概率函数，因此会损失一定的表达能力。无向图模型则没有这个问题，因为不对条件概率做直接定义（只定义Clique的势函数），无向图模型通常可用较简单的拓扑结构描述更复杂的分布。

HMM的另一个问题在于其生成模型的本质。给定一个预测任务，其中$\mathbf{x}$是观测序列，$\mathbf{y}$是预测序列（状态序列）。HMM通过描述$\mathbf{y}$中每个元素之间的概率关系$p(y_t|y_{t-1})$ 以及x_t 与y_t 的概率关系$p(x_t|y_t)$来描述系统的概率性质。这种方法本质上是对$p(\mathbf{x}, \mathbf{y})$建模，再基于概率规则推理出$p(\mathbf{y}|\mathbf{x})$，因此是典型的生成模型。这类模型的一个特征是需要对$p(\mathbf{x})$建模，而这一模型并

不是预测任务关注的重点，因为我们更关注后验概率$p(\mathbf{y}|\mathbf{x})$的准确性。如果对$p(\mathbf{x})$ 建模是精确的，显然有助于提高$p(\mathbf{y}|\mathbf{x})$的估计质量，但如果$p(\mathbf{x})$建模不精确，则会影响对$p(\mathbf{y}|\mathbf{x})$的估计。在HMM模型中，为模型简洁引入的链式结构及每个状态下的条件独立同分布假设显然是粗糙的，由此得到的$p(\mathbf{x})$估计具有较大偏差，因而影响$p(\mathbf{y}|\mathbf{x})$的准确性。

上述两个问题（复杂分布描述能力有限和$p(\mathbf{x})$不精确）都可以通过增加模型的复杂度解决，如在观测变量x_t之间建立概率关系，但这将破坏HMM中简洁的链式拓扑结构，给推理和参数估计带来困难。为保持模型结构的简洁同时提高模型表示能力，可从两个角度入手：（1）用无向图模型代替有向图模型，从而可自由设计Clique的势函数，提高对复杂概率分布的表达能力；（2）用区分性模型代替生成模型，即直接对$p(\mathbf{y}|\mathbf{x})$建模，从而避免对$p(\mathbf{x})$估计的困难。线性条件随机场（Linear-Chain CRF, L-CRF）即是这种模型 [22, 16]。图 6.18给出一个L-CRF的例子。L-CRF定义如下后验概率函数：

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \Psi_t(y_{t-1}, y_t, x_t),$$

其中，我们已经定义y_0为一个哑元变量（Dummy Variable），因此$\Psi_1(y_0, y_1, x_1) = \Psi_1(y_1, x_1)$。$Z(\mathbf{x})$是归一化因子。注意这一因子是$\mathbf{x}$ 的函数，这是和标准无向图模型最主要的区别。势函数Ψ_t 具有如下形式，

$$\Psi_t(y_{t-1}, y_t, x_t) = exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t, x_t) \right\}.$$

上式中, $f_k(\cdot)$称为特征，可以是任意连续函数或标记函数，θ_k是对应的参数。如在Shallow Parsing任务中，$f_k(\cdot)$可以是$\delta(y_{t-1} = Noun)$ 或$\delta(y_t = Noun)\delta(x_t = America)$。

CRF 与HMM和Logistic回归的关系

值得说明的是，上述L-CRF模型可以认为是HMM模型的扩展，当取合适的$\{\theta_k\}$和$\{f_k\}$时，上述L-CRF模型将等价于HMM模型。为更清楚看到这一点，我们以一个离散HMM为例，设其状态空间为S，观测空间为O，则可将其联合概率改写如下：

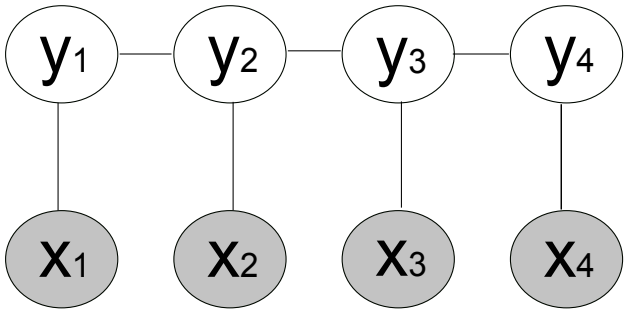


Fig. 6.18 线性条件随机场。该模型定义如下条件概率$p(y|x) = \frac{1}{Z(x)}\Psi_1(y_1,x_1)\Psi_2(y_1,y_2,x_2)\Psi_3(y_2,y_3,x_3)\Psi_4(y_3,y_4,x_4)$。

$$\begin{aligned} p(\mathbf{x},\mathbf{y}) &= \prod_{t=1}^{T} p(y_t|y_{t-1})p(x_t|y_t) \\ &= \prod_{t=1}^{T} exp\left\{\sum_{i,j\in S}\theta_{ij}\delta(y_t=i)\delta(y_{t-1}=j)+\sum_{i\in S}\sum_{o\in O}\mu_{oi}\delta(y_t=i)\delta(x_t=o)\right\}, \end{aligned} \tag{6.9}$$

其中：

$$\theta_{ij} = \ln p(y_t = i|y_{t-1} = j) \tag{6.10}$$

$$\mu_{oi} = \ln p(x_t = o|y_t = i). \tag{6.11}$$

上式可形式化写成：

$$p(\mathbf{x},\mathbf{y}) = \prod_{t=1}^{T} exp\left\{\sum_{k=1}^{K}\theta_k f_k(y_t,y_{t-1},x_t)\right\}.$$

则后验概率有：

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{y},\mathbf{x})}{\sum_{\mathbf{y}'}p(\mathbf{y}',\mathbf{x})} = \frac{\prod_{t=1}^{T} exp\left\{\sum_{k=1}^{K}\theta_k f_k(y_t,y_{t-1},x_t)\right\}}{\sum_{\mathbf{y}'}\prod_{t=1}^{T} exp\left\{\sum_{k=1}^{K}\theta_k f_k(y'_t,y'_{t-1},x_t)\right\}}.$$

取$Z(\mathbf{x}) = \sum_{\mathbf{y}'}\prod_{t=1}^{T} exp\left\{\sum_{k=1}^{K}\theta_k f_k(y'_t,y'_{t-1},x_t)\right\}$，则上式正是L-CRF 的形式。注意在这一形式中，θ_k和f_k都有特别的定义，因此HMM可以认为是L-CRF的特例，L-CRF可通过扩展f_k实现对复杂分布的建模。

同时，L-CRF可以认为是Logistc回归向序列预测任务的扩展。我们知道Logistic回归具有如下形式：

$$p(\mathbf{y}|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x})$$

其中σ是Logistic函数（二分类任务）或Softmax函数（多分类任务）。以多分类任务为例，假设分类任务共有K类，$\mathbf{x}$的维度为D，则上式可写成如下形式：

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} exp\{\sum_{k=1}^{K}\sum_{d=1}^{D}\theta_{k,d} f_{k,d}(\mathbf{x},\mathbf{y})\}, \tag{6.12}$$

其中：

$$\theta_{k,d} = w(k,d),$$

$$f_{k,d}(\mathbf{x},\mathbf{y}) = x_d\delta(y=k),$$

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} exp\{\sum_{k=1}^{K}\sum_{d=1}^{D}\theta_{k,d} f_{k,d}(\mathbf{x},\mathbf{y}')\}.$$

将(k,d)作为一个标记序列，则式（6.12）具有和L-CRF一样的形式，只不过y是一个独立分类，而不是分类序列。

参数估计

L-CRF模型的参数是$\{\theta_k\}$，模型训练的目的是基于一系列训练样本$\{(\mathbf{x}_n,\mathbf{y}_n\}$，确定参数$\{\theta_k\}$使得如下似然函数最大化：

$$\begin{aligned} L(\theta) &= \sum_n \ln p(\mathbf{y}_n|\mathbf{x}_n) \\ &= \sum_n\sum_t \Psi_t(y_{nt}, y_{n(t-1)}, x_{nt}) - \sum_n \ln Z(\mathbf{x}_n) \\ &= \sum_n\sum_t\sum_k \theta_k f_k(y_{nt}, y_{n(t-1)}, x_{nt}) - \sum_n \ln Z(\mathbf{x}_n). \end{aligned}$$

上式可由任意一种最优化方法求解，如SGD，Newton方法。一般常用的方法是BFGS方法 [4]，这一方法用一阶信息来近似二阶信息。对上式求θ_k的导数，有：

$$\frac{\partial L}{\partial \theta_k} = \sum_n \sum_t \sum_k f_k(y_{nt}, y_{n(t-1)}, x_{nt}) - \sum_n \sum_t \sum_{\mathbf{y},\mathbf{y}'} p(y_t, y_{t-1}|\mathbf{x}_n) f_k(y_t, y_{t-1}, x_{nt}). \tag{6.13}$$

上式表明θ_k上的误差来源于以训练集中$\mathbf{y}_n$为参数的f_k和以模型的后验概率$p(y_t, y_{t-1}|\mathbf{x}_n)$做期望得到的$f_k$之差。上式中，$f_k$是可计算的，但$p(y_t, y_{t-1}|\mathbf{x})$很难计算。这是因为$p(y_t, y_{t-1}|\mathbf{x})$是$p(\mathbf{y}|\mathbf{x})$的边缘概率，需要对所有不在$t$和$t-1$时刻的$y_t$做边缘化。由于存在大量可能的路径$\mathbf{y}$，这一概率的计算非常复杂。

我们可用类似HMM中Baum-Welch算法的动态规划方法对$p(y_t, y_{t-1}|\mathbf{x})$求解。类似Baum-Welch算法，我们利用乘法分配律对$Z(\mathbf{x})$展开，可得：

$$\begin{aligned} Z(\mathbf{x}) &= \sum_y \prod_t \Psi_t(y_{t-1}, y_t, x_t) \\ &= \sum_{y_2, y_3, \ldots, y_T} \prod_{t=2}^{T} \Psi_t(y_{t-1}, y_t, x_t) \sum_{y_1} \Psi_1(y_0, y_1, x_1) \\ &= \sum_{y_3, y_4, \ldots, y_T} \prod_{t=3}^{T} \Psi_t(y_{t-1}, y_t, x_t) \sum_{y_2} \Psi_2(y_1, y_2, x_2) \sum_{y_1} \Psi_1(y_0, y_1, x_1) \\ &\ldots \end{aligned} \tag{6.14}$$

定义前向变量如下：

$$\alpha(1, \mathbf{y}) = \Psi_1(y_0, y_1, x_1),$$

$$\alpha(t, \mathbf{y}) = \sum_{\mathbf{y}'} \alpha(t-1, \mathbf{y}') \Psi_t(\mathbf{y}', \mathbf{y}, x_t),$$

其中y_0是前面定义的哑元变量。上述递归计算可避免大量重复路径计算。当递归计算结束时，即有：

$$Z(\mathbf{x}) = \sum_{y_T} \alpha(T, y_T).$$

类似地，对$Z(\mathbf{x})$反向做递归计算，定义后向变量如下：

$$\beta(T, \mathbf{y}) = 1,$$

$$\beta(t, \mathbf{y}) = \sum_{\mathbf{y}'} \beta(t+1, \mathbf{y}') \Psi_{t+1}(\mathbf{y}, \mathbf{y}', x_{t+1}).$$

当递归结束时，同样可计算$Z(\mathbf{x})$如下：

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \Psi_1(y_0, y, x_1)\beta(1, \mathbf{y}).$$

上面定义的前向和后向变量与HMM中定义的前向和后向概率和具有相同形式，只不过在HMM中$\Psi_t(y_{t-1}, y_t, x_t)$定义不同。事实上，如果我们做如下定义：

$$\Psi_t(y_{t-1}, y_t, x_t) = p(y_t|y_{t-1})p(x_t|y_t),$$

则上述推导过程和HMM的Baum-Welch算法是完全一致的。由于$\Psi_t(y_{t-1}, y_t, x_t)$定义具有概率意义，HMM中的前后向变量可理解为路径的概率和，L-CRF中则不具有这种意义。

基于$\alpha(t, \mathbf{y})$和$\beta(t, \mathbf{y})$，可以很容易得到后验概率$p(y_{t-1}, y_t|\mathbf{x})$如下：

$$\begin{aligned} p(y_{t-1}, y_t|\mathbf{x}) &= \frac{\sum_{y_1, y_2, \ldots, y_{t-2}} \sum_{y_{t+1}, y_{t+2}, \ldots, y_T} \prod_{l=1}^{T} \Psi_l(y_{l-1}, y_l, x_l)}{Z(\mathbf{x})} \\ &= \frac{\alpha(t-1, y_{t-1})\Psi_t(y_{t-1}, y_t, x_t)\beta(t, y_t)}{Z(\mathbf{x})}. \end{aligned}$$

由此，公式（6.13）所有项皆可计算。

需要说明的是，上述模型参数训练的前提假设是$\{y_{nt}\}$是已知的，这一般可以通过强制对齐算法实现。基于这一前提，L-CRF中是不存在隐变量的，因此上述参数优化算法是一个相对简单的最优化问题。如果去掉这一假设，则需要考虑所有可能的对齐，这相当于引入了一个类似HMM模型的隐变量序列$\mathbf{q}$，该隐变量序列符合输出序列$\mathbf{y}$（即$\mathbf{q}$是序列$\mathbf{y}$的扩展，其中每个$\mathbf{y}$中的元素可能重复若干次，对应$\mathbf{x}$中同一个元素）。引入隐变量后，模型的目标函数需要对$\mathbf{q}$做边缘化，即：

$$L(\boldsymbol{\theta}) = \sum_{n} \ln \sum_{\mathbf{q} \in \mathscr{E}(\mathbf{y}_n)} p(q|\mathbf{x}_n), \tag{6.15}$$

其中$\mathscr{E}(\mathbf{y})$是所有符合$\mathbf{y}$的状态序列。由于存在隐变量，上述目标函数的优化变得更加困难。我们可以利用HMM中讨论过的EM 算法，以当前参数计算后验概率$p(\mathbf{q}|\mathbf{x}_n)$，再基于该后验概率求似然函数的期望，最后对该似然函数期望求最优化。不论是求后验概率$p(\mathbf{q}|\mathbf{x}_n)$，还是对似然函数期望进行优化，都需要对大量路径做边缘化，因此需要利用前面讨论的前后向递归计算方法。带有隐变量的CRF称为隐状态CRF（HCRF） [21]。这里隐变量不局限于输入变量$\mathbf{x}$与输出变量$\mathbf{y}$的对齐，事实上可以是任意描述数据分布结构的隐变量。

模型扩展

条件随机场（Conditional Random Field，CRF）是一类典型的无向图模型，L-CRF只是其中最简单的链式模型。L-CRF可扩展为更通用的链式结构，其中保持$\mathbf{y}$的链式结构不变，但每个Ψ_t依赖整个输入序列$\mathbf{x}$。由于$\mathbf{y}$保持链式，其因子分解形式不变，参数估计方法也保持不变。L-CRF进一步可扩展成非链式结构，即一般CRF。这时因子分解方式会变得更加复杂，一般需要近似推理方法。

6.5 EM算法

回顾前面介绍的GMM、HMM、HCRF等模型，不难发现，这些模型都有一个共同的特点，即存在隐变量。类似于神经模型中的隐藏层，隐变量的存在使得概率模型对领域知识的表达能力大大提高，极大增强了对复杂概率的表征能力。然而，隐变量的存在使得概率模型的参数估计（即模型训练）更为困难，因为对似然函数优化需要对所有隐变量进行边缘化，这在多数情况下是非常复杂的。EM 模型提供了一种通用的解决方法，用以对包含隐变量的模型求似然函数的局部最优解。这一节我们将讨论通用的EM算法，GMM、HMM、HCRF中的EM算法都是这一通用算法的特例。

假设一个概率模型$p(\mathbf{x},\mathbf{z};\boldsymbol{\theta})$, 其中$\mathbf{x}$代表所有观测变量，$\mathbf{z}$ 代表所有隐变量（假设这里的隐变量为离散变量，当隐变量为连续变量时，下述讨论中的加和变成积分即可）。显然，$\mathbf{z}$在GMM中对应高斯成分，在HMM和HCRF中对应状态序列。这一联合概率分布由一组参数$\boldsymbol{\theta}$ 控制。我们最终希望最大化如下似然函数:

$$L(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n;\boldsymbol{\theta}).$$

对任意一个在$\mathbf{z}$上的分布函数$q(\mathbf{z})$，经过简单推导，可以得到函数$L(\boldsymbol{\theta})$的一个下界：

$$\begin{aligned}
L(\boldsymbol{\theta}) &= \sum_n \sum_{\mathbf{z}} q(\mathbf{z}) \ln p(\mathbf{x}_n) \\
&= \sum_n \sum_{\mathbf{z}} q(\mathbf{z}) \ln \frac{p(\mathbf{x}_n, \mathbf{z})}{p(\mathbf{z}|\mathbf{x}_n)} \\
&= \sum_n \sum_{\mathbf{z}} q(\mathbf{z}) \ln \frac{p(\mathbf{x}_n, \mathbf{z})}{q(\mathbf{z})} \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x}_n)} \\
&= \sum_n \sum_{\mathbf{z}} q(\mathbf{z}) \ln \frac{p(\mathbf{x}_n, \mathbf{z})}{q(\mathbf{z})} + \sum_n \sum_{\mathbf{z}} q(\mathbf{z}) \ln \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x}_n)} \\
&= \tilde{L}(\boldsymbol{\theta}) + \sum_n KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}_n))
\end{aligned}$$

上式中我们定义了

$$\tilde{L}(\boldsymbol{\theta}) = \sum_n \sum_{\mathbf{z}} q(\mathbf{z}) \ln \frac{p(\mathbf{x}_n, \mathbf{z})}{q(\mathbf{z})}.$$

由于$KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}_n)) \geq 0$，因此$\tilde{L}(\boldsymbol{\theta})$是$L(\boldsymbol{\theta})$的下界函数。

通过对$\tilde{L}(\boldsymbol{\theta})$优化，即可得到比当前$\boldsymbol{\theta}$更好的参数。注意$q(\mathbf{z})$可以取任意函数，但过低的下界将失去意义。一种方式是取基于当前参数的后验概率$q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x};\boldsymbol{\theta}')$，记对应的下界函数为$\tilde{L}(\boldsymbol{\theta},\boldsymbol{\theta}')$。注意在$\boldsymbol{\theta}'$点处有$L(\boldsymbol{\theta}') = \tilde{L}(\boldsymbol{\theta}';\boldsymbol{\theta}')$，因此可保证优化$\tilde{L}(\boldsymbol{\theta};\boldsymbol{\theta}')$ 得到的新的$\boldsymbol{\theta}$ 优化值$\boldsymbol{\theta}''$有更好的似然函数值，即：

$$\tilde{L}(\boldsymbol{\theta}'';\boldsymbol{\theta}') \geq \tilde{L}(\boldsymbol{\theta}';\boldsymbol{\theta}') = L(\boldsymbol{\theta}').$$

上述优化过程如图6.19所示：基于当前模型参数$\boldsymbol{\theta}'$得到下界函数$\tilde{L}(\boldsymbol{\theta};\boldsymbol{\theta}')$。这条曲线与似然函数曲线$L(\boldsymbol{\theta})$在$\boldsymbol{\theta}'$点相切。对$\tilde{L}(\boldsymbol{\theta};\boldsymbol{\theta}')$进行优化，得到新的参数$\boldsymbol{\theta}''$，基于这一新的参数得到的$\tilde{L}(\boldsymbol{\theta};\boldsymbol{\theta}'')$ 是更好的下界函数。EM算法可形式化为算法 1.

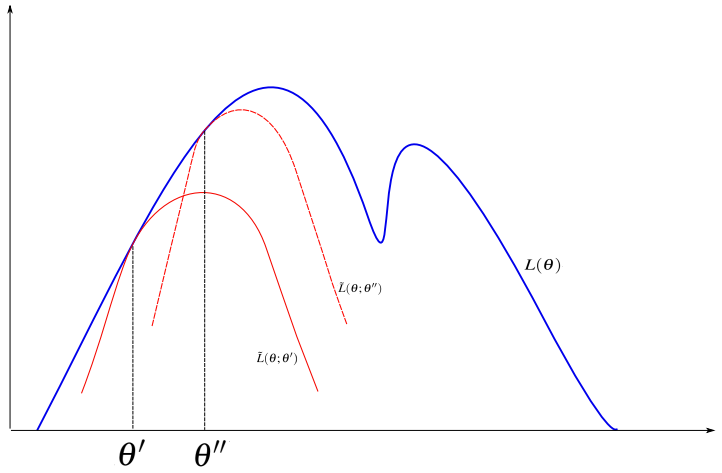


Fig. 6.19 EM算法。$L(\boldsymbol{\theta})$是要逼近的似然函数，用蓝粗线表示。$\boldsymbol{\theta}'$是当前参数，基于此得到的下界函数$\tilde{L}(\boldsymbol{\theta};\boldsymbol{\theta}')$。对该下界函数最优化得到新的参数$\boldsymbol{\theta}''$，基于该参数得到的下界函数$\tilde{L}(\boldsymbol{\theta};\boldsymbol{\theta}'')$ 是更好的下界函数。

```
Random Initialize θ;
while True do
    θ' = θ;
    Expectation:
    Compute p(z|x;θ');
    Compute L̃(θ;θ');
    Maximization:
    θ'' = arg max_θ L̃(θ;θ') ;
    if |θ'' − θ'| < δ then
        Break;
    end
    θ = θ'';
end
```

Algorithm 1: EM算法。

值得说明的是，EM算法提供了一个通用的参数估计框架，但对具体模型还要设计具体算法来计算后验概率$p(\mathbf{z}|\mathbf{x};\boldsymbol{\theta}')$以及优化似然函数的期望$\tilde{L}(\boldsymbol{\theta};\boldsymbol{\theta}')$。对于GMM和HMM模型，不论是后验计算还是对似然函数期望的优化都有较好的闭式解，但对更复杂的模型，这一求解过程并不直观，如

在HCRF中，需要采用数值优化方法，典型的如SGD，Newtown, BFGS等。更多优化方法将在第 11章具体介绍。对于后验概率计算，我们已经在HMM模型一节中讨论了Bauch-Welch算法，这一算法事实上是更通用的加和-乘积（Sum-Product）算法的特例，而后者则是更联合树（Junction Tree）算法的特例。这些算法可以精确计算后验概率，但仅适用于特定的图结构，如链状或树状。对于更复杂的图模型，一般采用近似解法。本章后面两节将讨论包括后验概率计算在内的图模型推理算法。

6.6 精确推理算法

给定一个概率模型，我们可以求某个或某几个变量的边缘概率，或给定某些变量，求另一些变量的后验概率，这些操作称为概率模型的推理（Inference）。推理在概率模型中至关重要，因为只有通过推理，才能从图模型所表示的复杂局部关系中推导出全局相关性。推理对模型训练也十分重要，因为计算后验概率是EM算法的关键步骤。对于简单模型，存在精确推理算法，但对大多数模型，只能做近似推理。本节将介绍精确推理方法，近似推理方法将在下节介绍。本节中我们仅考虑边缘概率，因为有了边缘概率，利用贝叶斯定理即可计算后验概率。

6.6.1 加和-乘积算法

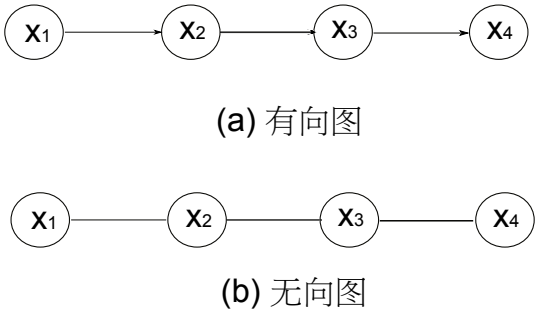


Fig. 6.20 链式结构的概率图模型，其中(a)为有向图模型，（b）为无向图模型。

我们从最简单的链式结构的边缘概率开始讨论，如图 6.20所示。这一模型的联合概率可统一写成如下因子分解形式：

$$p(\mathbf{x}) = \frac{1}{Z}\prod_{t=1}^{T}\Psi_t(x_{t-1}, x_t), \tag{6.16}$$

其中x_0是一个哑元变量，使得

$$\Psi_1(x_0, x_1) = \Psi_1(x_1).$$

对图 6.20 (a)所示的有向图，有

$$Z = 1; \Psi_t(x_{t-1}, x_t) = p(x_t|x_{t-1}).$$

对图 6.20 (b)所示的无向图，则有:

$$Z = \sum_{\mathbf{x}}\prod_{t=1}^{T}\Psi_t(x_{t-1}, x_t),$$

$\Psi_t(x_{t-1}, x_t)$为定义在Clique(x_{t-1}, x_t)上的势函数。

我们关心如下推理任务：给定如式 6.16所示的联合概率，计算任一变量的边缘概率$p(x_i)$。我们将推导一个基于动态规划的快速算法，其基本思路是：当我们对链中的变量进行边缘化时，需要对所有可能的路径计算概率，并将所有可能的路径加和，这里边包含了大量重复计算。用动态规划算法，将部分路径合并，可大量减少计算开销。

现在让我们求链式结构中第i个变量的边缘概率$p(x_i)$。依边缘概率公式，由式（6.16）可得：

$$p(x_i) = \sum_{x_1,\ldots,x_{i-1},x_{i+1},\ldots x_N} p(\mathbf{x}) = \frac{1}{Z}\sum_{x_1,\ldots,x_{i-1},x_{i+1},\ldots x_N}\prod_{t=1}^{T}\Psi_t(x_{t-1}, x_t).$$

将上式右侧的加和与乘积整理可得：

$$p(x_i) = \frac{1}{Z}[\sum_{x_1,\ldots,x_{i-1}}\prod_{t=1}^{i}\Psi_t(x_{t-1}, x_t)][\sum_{x_{i+1},\ldots x_N}\prod_{t=i+1}^{T}\Psi_t(x_{t-1}, x_t)], \tag{6.17}$$

其中第一项为x_i左侧所有以x_i为结束状态的路径之和，第二项为x_i右侧所有以x_i为起始状态的路径之和。利用乘法分配律对第一项做整理，有：

$$\sum_{x_1,\ldots,x_{i-1}}\prod_{t=1}^{i}\Psi_t(x_{t-1},x_t)=\sum_{x_{i-1}}\Psi_i(x_{i-1},x_i)\sum_{x_{i-2}}\Psi_{i-1}(x_{i-2},x_{i-1})\ldots\sum_{x_2}\Psi_3(x_2,x_3)\sum_{x_1}\Psi_2(x_1,x_2)\Psi_1(x_0,x_1).$$

上式表明求一个序列的所有可能路径的概率和时，可将变量依次提取出来做边缘化，从而避免对变量的重复计算。可以定义一个统计量$\alpha_i(x_i)$来表示以x_i为结束状态的所有路径的变量和：

$$\alpha(1,x_1)=\Psi_1(x_0,x_1),$$

$$\alpha(i,x_i)=\sum_{x_{i-1}}\alpha(i-1,x_{i-1})\Psi_t(x_{j-1},x_j).$$

同理，式（6.17）的第二项可写成如下形式：

$$\sum_{x_{i+1},\ldots,x_T}\prod_{t=i+1}^{T}\Psi_t(x_{t-1},x_t)=\sum_{x_{i+1}}\Psi_{i+1}(x_i,x_{i+1})\ldots\sum_{x_{t-1}}\Psi_{T-1}(x_{T-2},x_{T-1})\sum_{x_T}\Psi_T(x_{T-1},x_T)$$

定义统计量$\beta_j(x_j)$为以x_j为起始状态的所有可能路径的概率和:

$$\beta(T,x_T)=1,$$

$$\beta(i,x_i)=\sum_{x_{i+1}}\beta(i+1,x_{i+1})\Psi_{i+1}(x_i,x_{i+1}).$$

由式（6.17），x_i的边缘概率可计算如下：

$$p(x_i)=\frac{1}{Z}\alpha(i,x_i)\beta(i,x_i).$$

其中Z可由前后向概率计算得到：

$$Z=\sum_{x_i}\alpha(i,x_i)\beta(i,x_i)=\sum_{x_T}\alpha(T,x_T)=\sum_{x_1}\beta(1,x_1).$$

基于部分路径的统计量$\alpha(i,x)$和$\beta(i,x)$，可以计算其它边缘概率和后验概率，如：

$$p(x_i,x_{i+1})=\frac{1}{Z}\alpha(i,x_i)\Psi_{i+1}(x_i,x_{i+1})\beta(i+1,x_{i+1}),$$

$$p(x_{i+1}|x_i)=\frac{p(x_i,x_{i+1})}{p(x_i)}.$$

上述算法称为*加和-乘积算法*（Sum-Product Algorithm）。这一算法实质上是利用乘法的分配率，对原来的计算过程调整加乘顺序，从而避免重复计算。如果我们回忆一下HMM中的Baum-Welch算法中求后验概率$p(q_{t-1},q_t|\mathbf{x})$和L-CRF中求边缘概率$p(y_{t-1},y_t|\mathbf{x})$的方法，就可以发现这些

算法事实上都基于类似的思路，只不过基于不同的势函数定义，因此都是加和-乘积算法在具体模型中的特例。

6.6.2 树状结构的加和-乘积算法

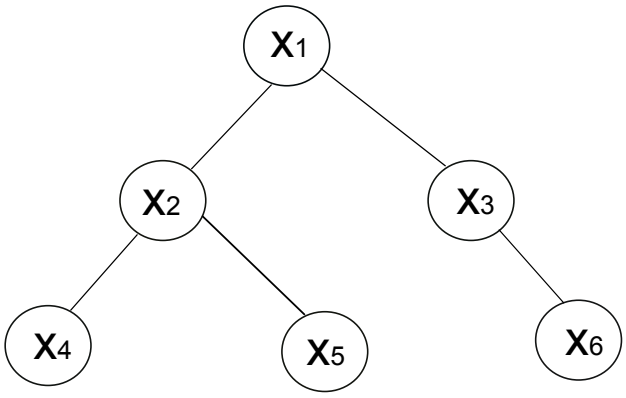


Fig. 6.21 树结构的无向图模型。

加和-乘积算法可以扩展到树状结构。如前所述，有向图的概率因子分解结构和无向图的势函数因子分解可认为是等价的，因此可将两者都统一到无向图，故而我们仅考虑无向图中树结构的加和-乘积算法。以图 6.21 所示的树状模型为例，假设我们要求x_2 节点的边缘概率，有：

$$\begin{aligned} p(x_2) &= \frac{1}{Z}\sum_{x_1,x_3,x_4,x_5,x_6}\Psi_{1,2}(x_1,x_2)\Psi_{1,3}(x_1,x_3)\Psi_{3,6}(x_3,x_6)\Psi_{2,4}(x_2,x_4)\Psi_{2,5}(x_2,x_5) \\ &= \frac{1}{Z}\sum_{x_1}\Psi_{1,2}(x_1,x_2)\sum_{x_3}\Psi_{1,3}(x_1,x_3)\sum_{x_6}\Psi_{3,6}(x_3,x_6)\sum_{x_4}\Psi_{2,4}(x_2,x_4)\sum_{x_5}\Psi_{2,5}(x_2,x_5) \end{aligned}$$

上式把联合概率$p(x)$ 按x_2的邻边分解成三部分:

$$m_{4\to 2}(x_2) = \sum_{x_4}\Psi_{2,4}(x_2,x_4) \tag{6.18}$$

$$m_{5\to 2}(x_2) = \sum_{x_5}\Psi_{2,5}(x_2,x_5) \tag{6.19}$$

$$m_{1\to 2}(x_2)=\sum_{x_1}\Psi_{1,2}(x_1,x_2)\sum_{x_3}\Psi_{1,3}(x_1,x_3)\sum_{x_6}\Psi_{3,6}(x_3,x_6). \tag{6.20}$$

由此可得:

$$p(x_2)=\frac{1}{Z}\prod_{i\in N(x_2)}m_{i\to 2}(x_2)$$

其中$N(x_2)$表示x_2所有相邻节点的下标。注意，之所以可以做这样的分解，是因为以x_2为中心各方向的子图互不重叠（这一性质对树结构中的任意一个节点都适用）。我们可以将不同方向的因子看作从不同方向流向x_2的信息量，当所有信息量计算完成时，即可计算任意一个节点的边缘概率。为有效计算各个节点各个方向的信息量，可以设计一个递归算法，从叶子节点开始累积信息量，自底向上向根节点归约。基于树结构，可保证这一归约过程是可完成的。对树中任意一个节点x_i，等待其所有子节点信息量计算完成后，依下式向其父节点$x_{Pa(x_i)}$传递信息量：

$$m_{i\to Pa(x_i)}(x_{Pa(x_i)})=\sum_{x_i}\prod_{j\in Child(x_i)}m_{j\to i}(x_i)\Psi_{i,Pa(x_i)}(x_i,x_{Pa(x_i)}).$$

当上述信息量传递到达根节点时，我们事实上已经可以计算所有路径的信息量，即Z。这时从根节点自顶向下向所有叶子节点反向传递信息量。在任意一个节点x_i，向其任意一个子节点传送的信息量计算如下：

$$m_{i\to k}(x_k)=\sum_{x_i}m_{Pa(x_i)\to i}(x_i)\prod_{j\in Child(x_i)\setminus k}m_{j\to i}(x_i)\Psi(x_i,x_k)\quad\forall k\in Child(x_i).$$

当反向信息量传递到达所有叶子节点时，即可计算任意一个或几个变量的边缘概率。

6.6.3 联合树算法

前述的加和-乘积算法仅适用于树状结构，对一般概率图模型，因为不能保证每个节点的归约是顺序可完成的，因此无法适用。一个办法是将通用图结构转化成树结构，再基于加和-乘积算法做推理。这一算法称为联合树算法（Junction Tree Algorithm） [17, 24]。同样，我们仅讨论无向图情况。如果要处理一个有向图，则需要先将其转换成无向图。有向图转化成无向图的

方法在第 6.3.1.1节中讨论过，其中重要的一步是Moralization，即对具有多个父节点的节点所对应的Clique增加附加边，使$\{x_i, \{x_j : j \in Pa(x_i)\}\}$成为一个合法的Clique。统一到无向图后，可以将该图表示成一个信息流图，其中每个信息节点对应无向图中的一个Clique，Clique之间的共有变量表示为信息节点间的边。图 6.22给出这一转换的例子。可见，该图中包含环状结构，因此不能顺序计算信息流。为此，我们将该无向图进行三角化，在环中加入附加边，保证不存在超过三个节点的环。一般来说，一个环状结构可通过多种加边方法进行三角化，但较为理想的三角化方案是增加最少的边达到目的。图6.25给出三角化的例子：通过在节点E和节点G之间加入附加边，使得图中不存在超过3个节点的环。三角化后的无向图转换成信息图后没有环结构。值得说明的是，虽然这里加入了附加边，但因为势函数没有改变，所以三角化后的概率图与原概率图是等价的。此时信息图所表示的联合概率分布如下式所示:

$$p(x) = \frac{1}{Z}\psi_{A,B,C}(A,B,C)\psi_{B,C,D}(B,C,D)\psi_{D,E,H}(D,E,H)\psi_{E,G,H}(E,G,H)\psi_{E,F,G}(E,F,G)$$

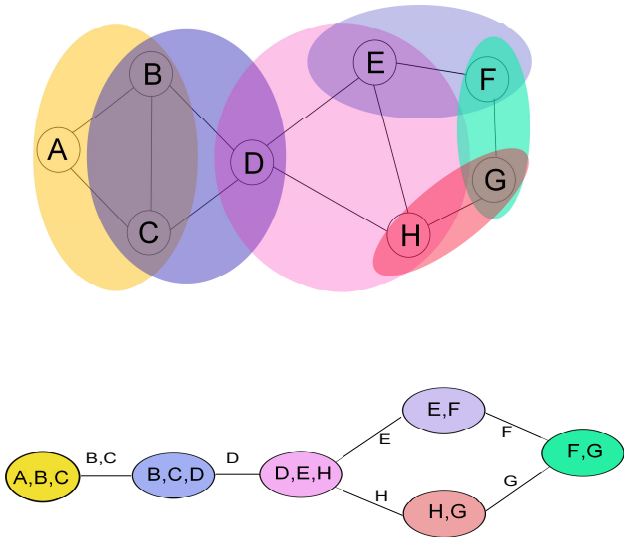


Fig. 6.22 无向图的信息流图，其中每个Clique简化为信息流图中的信息节点，Clique之间的共享变量表示为信息流图的边连接。

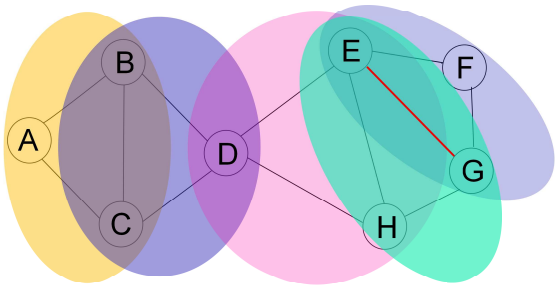


Fig. 6.23 三角化后无向图的信息流图呈现一个链式结构。

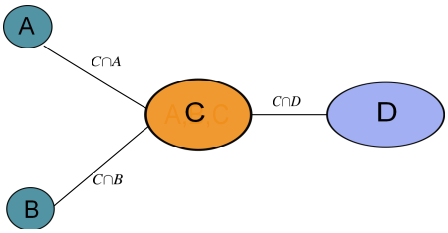


Fig. 6.24 Junction Tree上的Sum-Product 算法。由C到D的信息量由所有C的邻居节点（除去D）传递过来的信息量计算得到。

一般情况下，通过上述Moralization和三角化，我们会得到一个树结构的信息流图，称为联合树，基于此可应用前面讨论过的加和-乘积算法进行推理。和上一节讨论的简单树结构不同，现在每个树节点不是一个变量，而是若干变量组成的集合，节点与节点间的共享变量也可能有多个。因此，当进行递归归约时，每个节点向其父节点或某一子节点传递的信息量可能是多个共享变量的函数。如图 6.24所示的局部结构，我们要计算从Clique C到Clique D的信息，需要对Clique C 中所有不包含在$C \cap D$中的变量做边缘化，得到以$C \cap D$为变量的信息量函数，如下式所示：

$$m_{C \to D}(C \cap D) = \sum_{C - C \cap D} \psi_C(C) m_{A \to C}(x_{C \cap A}) m_{B \to C}(x_{C \cap B}).$$

上式写成更一般的通用形式为：

$$m_{C\to C'}(C\cap C') = \sum_{C-C\cap C'} \psi_C(C) \prod_{C''\in\{N(C)-C'\}} m_{C''\to C}(x_{C\cap C''}).$$

和简单树结构下的加和-乘积算法类似，我们首先从叶子节点向根节点递归归约，再从根节点反向计算向叶子节点传递的信息量。当这两个方向的信息量计算完成时，即可计算任意一个节点中所包含变量的边缘概率:

$$P(C) = \prod_{C'\in N(C)} m_{C'\to C}(C\cap C').$$

6.7 近似推理算法

Junction Tree算法对于简单模型可以实现有效推理，但当模型比较复杂时，计算复杂度将急剧提高。一是复杂网络的树结构也更复杂，计算量随信息节点数线性增长；更严重的是，复杂网络导致联合概率的可分解性下降，对应产生很多包含大量变量的Clique。这些庞大的Clique无法利用动态规划算法简化计算，会带来巨大的计算开销。例如在离散变量情况下，联合树算法的计算复杂度依最大Clique中的变量个数呈指数增长 [6]。本节将介绍两种近似推理算法，一种方法基于采样来估计概率分布，称为采样法；另一种方法用简单概率分布来近似实际复杂的概率分布，称为变分法。这两种方法都可极大提高推理效率，虽然这些推理是不精确的，但在很多实际问题中已经足够了。

6.7.1 采样法

不论是有向图模型还是无向图模型，都是一种生成模型，这意味着给定模型参数，即可从模型中采样出一系列独立同分布样本，这些样本符合该模型所代表的联合概率分布。有了联合概率分布，即可得到以采样表示的边缘概率和后验概率。对于边缘概率，只需在每个采样点中忽略那些不关心的变量，只保留目标边量；对于条件概率，只需丢掉那些不符合条件的采样点。

然而，这种简单采样方法通常是不可行的。首先，对于有向图，采样比较容易，但对无向图，采样通常比较困难；第二，即便可以得到有效采样点，简单采样法效率也很低，在估计边缘概率分布时得到的样本分散缓慢，在估

计条件概率分布时有大量样本被抛弃。一种常用的高效采样方法是马尔可夫链-蒙特卡洛算法（Markov Chain Monte Carlo, MCMC）。

马尔可夫链-蒙特卡洛

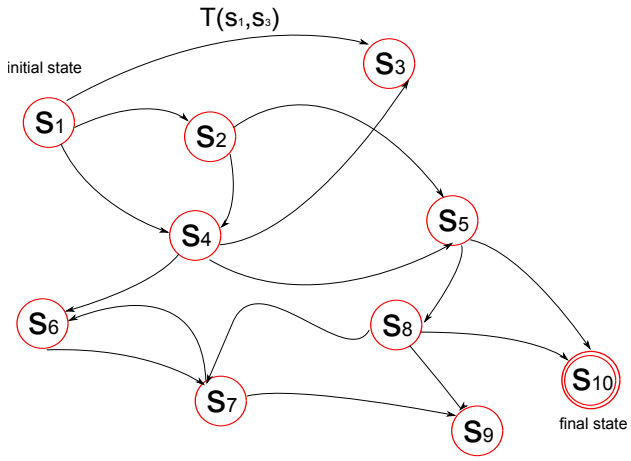


Fig. 6.25 马尔可夫链，每条连接状态S_i和S_j的边对应一个跳转概率$T(S_i,S_j)=P(S_j|S_i)$。

MCMC是一种高效的采样算法，其基本思路是设计一个马尔可夫链，使其每一个状态代表目标概率分布的一个有效采样值。当该马尔可夫链运行无限长时间后，如果达到一个在所有状态上的稳定分布且该分布等于目标概率分布，则通过运行该可马尔夫链即可得到目标概率的采样。对应图模型，该马尔可夫链的一个状态对应于图中所有节点$\mathbf{z}=\{z_i\}$的一种可能取值。

我们首先定义马尔可夫链所代表的稳定概率分布。设某一马尔可夫链的转移概率为：

$$T(\mathbf{z}^t,\mathbf{z}^{t+1})=p(\mathbf{z}^{t+1}|\mathbf{z}^t),$$

其中$\mathbf{z}^t$是离散变量，代表马尔可夫链在t时刻的状态。依马尔可夫假设，$t+1$时刻的概率分布由t时刻的概率分布计算得到：

$$p(\mathbf{z}^{t+1})=\sum_{\mathbf{z}^t}p(\mathbf{z}^{t+1}|\mathbf{z}^t)p(\mathbf{z}^t).$$

显然，当一个马尔可夫链运行到稳定状态时，必然有$\mathbf{z}$的分布保持不变，即：

$$p^*(\mathbf{z}) = \sum_{\mathbf{z}'} T(\mathbf{z}', \mathbf{z}) p^*(\mathbf{z}'),$$

其中$p^*(\mathbf{z})$称为该马尔可夫链的稳定分布。如果我们能设计出一个马尔可夫链，使其稳定分布$p^*(\mathbf{z})$恰好是我们要采样的目标概率分布，则通过无限运行该马尔可夫链即可完成采样任务。注意一个马尔可夫链可能对应多个稳定分布，但当它具有各态遍历属性时（Ergodic），则该马尔可夫链只有一个稳定分布，且不论起始状态如何，运行该马尔可夫链都会收敛到该稳定分布。所谓各态遍历性，是指从任何起始状态出发，马尔可夫链到所有状态的概率都大于零。

如果我们想让$p(\mathbf{z})$作为一个马尔可夫链的稳定分布，则可选择马尔可夫链的跳转概率满足如下条件：

$$p(\mathbf{z}) T(\mathbf{z}, \mathbf{z}') = p(\mathbf{z}') T(\mathbf{z}', \mathbf{z}),$$

上式称为细节平衡条件（Detailed Balance Condition）。当满足这一条件时，有：

$$\sum_{\mathbf{z}'} p(\mathbf{z}') T(\mathbf{z}', \mathbf{z}) = \sum_{\mathbf{z}'} p(\mathbf{z}) T(\mathbf{z}, \mathbf{z}') = p(\mathbf{z}) \sum_{\mathbf{z}'} p(\mathbf{z}'|\mathbf{z}) = p(\mathbf{z}),$$

因此$p(\mathbf{z})$是该马尔可夫链的稳定分布。一个具有状态遍历属性，且满足细节平衡条件的马尔可夫链，不论从哪个状态出发，必然收敛到该细节平衡条件所对应的稳定概率分布。上述用马尔可夫链来代表目标概率的方法称为马尔可夫链-蒙特卡洛方法（MCMC）。

下面我们给出一个实现MCMC的采样算法。我们的目的是设计一个马尔可夫链，使其稳定分布为某一目标概率$p(\mathbf{z})$。我们将假设$p(\mathbf{z})$由一个图模型代表。对有向图，任何一个给定的$\mathbf{z}$，我们可以很容易依该有向图所代表的联合概率计算$p(\mathbf{z})$；对无向图，由于归一化因子Z通常很难得到，直接计算$p(\mathbf{z})$存在困难，但非归一化概率值$\tilde{p}(\mathbf{z}) = Zp(\mathbf{z})$很容易由图中Clique的势函数得到。我们将以$\tilde{p}(\mathbf{z})$可计算为假设进行讨论，有向图可以认为是$Z = 1$的特殊情况。

设计任意一个“建议分布”(Proposal Distribution) $q(\mathbf{z}|\mathbf{z}')$，这一分布具有足够简单的形式，使得给定$\mathbf{z}'$，可实现对$\mathbf{z}$的采样。我们将$q(\mathbf{z}|\mathbf{z}')$作为状态转移概率$T(\mathbf{z}', \mathbf{z})$设计一个马尔可夫链。运行该马尔可夫链，可得到一个采样序列$\mathbf{z}^1, \mathbf{z}^2, ...$。到目前为止，这一马尔可夫链和我们的目标分布$p(\mathbf{z})$没有任何关系。现在我们引入一个采样接受/拒绝机制，即每次依$q(\mathbf{z}|\mathbf{z}')$进行采样时，不

会对得到的样本全盘接受，而是以一定概率接受，该接受概率包含目标概率$p(\mathbf{z})$（严格来说，是非归一化概率$\tilde{p}(\mathbf{z})$），定义如下：

$$A(\mathbf{z}',\mathbf{z}) = \min(1, \frac{\tilde{p}(\mathbf{z})q(\mathbf{z}'|\mathbf{z})}{\tilde{p}(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')}).$$

如果一个采样没有被接受，则原有状态$\mathbf{z}'$被输出成一个新的采样点。这样将生成一个新的马尔可夫链，其跳转概率为$T(\mathbf{z}',\mathbf{z}) = q(\mathbf{z}|\mathbf{z}')A(\mathbf{z}',\mathbf{z})$。简单计算可得：

$$\begin{aligned} p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')A(\mathbf{z}',\mathbf{z}) &= p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')\min(1, \frac{\tilde{p}(\mathbf{z})q(\mathbf{z}'|\mathbf{z})}{\tilde{p}(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')}) && (6.21)\\ &= \min(p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}'), p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})) && (6.22) \end{aligned}$$

$$\begin{aligned} p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})A(\mathbf{z},\mathbf{z}') &= p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})\min(1, \frac{\tilde{p}(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')}{\tilde{p}(\mathbf{z})q(\mathbf{z}'|\mathbf{z})}) && (6.23)\\ &= \min(p(\mathbf{z})q(\mathbf{z}'|\mathbf{z}), p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')) && (6.24) \end{aligned}$$

因而有：

$$p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})A(\mathbf{z},\mathbf{z}') = p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')A(\mathbf{z}',\mathbf{z}),$$

即：

$$p(\mathbf{z})T(\mathbf{z},\mathbf{z}') = p(\mathbf{z}')T(\mathbf{z}',\mathbf{z}).$$

这表明该马尔可夫链满足以$p(\mathbf{z})$为概率的细节平衡条件，因而$p(\mathbf{z})$是其稳定分布。同时，我们可以选择合适的跳转概率$q(\mathbf{z}|\mathbf{z}')$，使得该马尔可夫链具有各态遍历性，因而不管初始状态如何，都会收敛到$p(\mathbf{z})$。这一采样方法称为Metropolis-Hastings采样 [8]。

Gibs 采样

Metropolis-Hastings采样依然会抛弃一些采样点，带来效率下降。Gibs采样是一种特殊形式的Metroplis-Hastings，这种采样会保留所有采样点，实现起来也更容易。

具本来说，Metroplis-Hastings在每一次采样时，会依$q(\mathbf{z}|\mathbf{z}')$采样得到一个样本$\mathbf{z}$，再以$A(\mathbf{z},\mathbf{z}')$为概率保留采样点。Gibs采样时，每一步仅对第$i$个变

量随机取值，而保持其余变量不变，即得到的新采样点$\mathbf{z}$中，有$\mathbf{z}'_{-i} = \mathbf{z}_{-i}$，其中$\mathbf{z}_{-i}$代表除$i$外所有变量的采样值。在对$z_i$进行随机取值时，基于条件概率$p(z_i|\mathbf{z}_{-i})$，其中$p(\mathbf{z})$是我们要采样的目标概率分布。这一采样法称为Gibs采样。显然，在Gibs采样中，马尔可夫链的跳转概率为:

$$T(\mathbf{z}', \mathbf{z}) = p(\mathbf{z}|\mathbf{z}') = p(z_i|\mathbf{z}'_{-i}),$$

而采样点$\mathbf{z}$的联合概率为：

$$p(\mathbf{z}) = p(z_i|\mathbf{z}_{-i})p(\mathbf{z}_{-i}).$$

代入Metropolis-Hastings的采样选择公式，有：

$$\begin{aligned} A(\mathbf{z}', \mathbf{z}) &= \min(1, \frac{\tilde{p}(\mathbf{z})q(\mathbf{z}'|\mathbf{z})}{\tilde{p}(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')}) \\ &= \min(1, \frac{p(z_i|\mathbf{z}_{-i})p(\mathbf{z}_{-i})p(z'_i|\mathbf{z}_{-i})}{p(z'_i|\mathbf{z}'_{-i})p(\mathbf{z}'_{-i})p(z_i|\mathbf{z}'_{-i})} \\ &= 1, \end{aligned}$$

其中我们应用了关系$\mathbf{z}_{-i} = \mathbf{z}'_{-i}$。

因此，Gibs采样可以认为是一种所有采样都会被接受的Metropolis-Hastings采样，其稳定分布为$p(\mathbf{z})$。注意的是，Gibs采样每一步需要计算$p(z_i|\mathbf{z}_{-i})$。对于一个图模型来说，这是相对简单的任务，只需找到和z_i相关的所有变量并列出它们之间的条件概率关系即可。在无向图模型中，相关变量可以从和z_i直接相连的变量得到；有向图模型稍微复杂，相关变量包括z_i的所有父节点，所有子节点，以及所有子节点的父节点。与z_i相关的所有变量集合称为z_i的马尔可夫毯（Markov Blanket）。图 6.26给出无向图和有向图中马尔可夫毯的例子。在计算条件概率$p(z_i|\mathbf{z}_{-i})$时，仅需考虑马尔可夫毯中的变量，极大简化了计算复杂度。在实际实现时，z_i可以按顺序循环选择，也可以随机选择。另外，Gibs采样的相邻样本间具有很强的相关性，为实现独立同分布采样，可以选择每隔若干次采样保留一个样本。

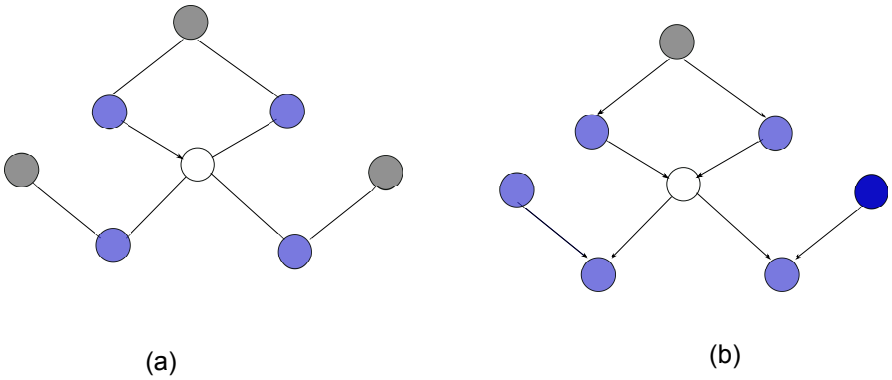


Fig. 6.26 在Gibs采样中，无向图（a）和有向图（b）的Markov Blanket。白色圆圈代表目标变量，染色圆圈代表观测变量，其中蓝色圆圈代表和目标变量相关的变量，需要在条件概率中考虑，灰色圆圈代表和目标变量无关的变量，在条件概率中不予考虑。

基于采样的近似推理

给定一个图模型，我们可以利用Gibs采样完成一系列推理任务，包括求联合概率$p(\mathbf{z})$，边缘概率$p(\mathbf{z}_A)$和后验概率$p(\mathbf{z}_A|\mathbf{z}_B)$，其中$\mathbf{z}_A$和$\mathbf{z}_B$是$\mathbf{z}$的两个子集。

- 联合概率$p(\mathbf{z})$。对于有向图，$p(\mathbf{z})$可以直接由变量的先验概率和变量之间的条件概率得到；对无向图，由于归一化因子Z很难估计，$p(\mathbf{z})$估计并不直观。基于Gibs采样生成$p(\mathbf{z})$的一系列采样$\{\mathbf{z}^{(l)}\}$，则可由下式由$p(\mathbf{z})$:

$$p(\mathbf{z}) = \sum_{\mathbf{z}'} p(\mathbf{z}|\mathbf{z}')p(\mathbf{z}') \approx \sum_{l=1}^{L} T(\mathbf{z}^{(l)}, \mathbf{z}).$$

- 边缘概率$p(\mathbf{z}_A)$。基于Gibs采样生成$p(\mathbf{z})$的一系列采样$\{\mathbf{z}^{(l)}\}$，忽略每个采样中不在$\mathbf{z}_A$中的变量，再利用上述求联合概率的方法计算$p(\mathbf{z}_A)$。
- 条件概率$p(\mathbf{z}_A|\mathbf{z}_B)$。在Gibs采样时，固定$\mathbf{z}_B$中的变量为给定值，对剩余变量循环采样，得到一系列采样$\{\mathbf{z}^{(l)}\}$，忽略不在$\mathbf{z}_A$中的变量，再利用求联合概率的方法计算$p(\mathbf{z}_A|\mathbf{z}_B)$。

事实上，在很多应用中上述概率的确切值并不重要，更重要的是基于这些概率的统计量。例如在贝叶斯线性预测中，我们并不关心模型参数$\mathbf{w}$的后验概率$p(\mathbf{w}|D)$，而是基于$p(\mathbf{w}|D)$对预测概率$p(t|\mathbf{x},\mathbf{w})$的期望，因此可直接用$p(\mathbf{w}|D)$的采样点实现估计：

$$p(t|\mathbf{x};D) = \int p(t|\mathbf{x},\mathbf{w})p(\mathbf{w}|D) \tag{6.25}$$

$$\approx \frac{1}{L}\sum_{l=1}^{L} p(t|\mathbf{x},\mathbf{w}^{(l)}), \tag{6.26}$$

其中$\{\mathbf{w}^{(l)}\}$为后验概率$p(\mathbf{w}|D)$的L个采样点（D为训练数据）。

类似的，在拉普拉斯近似中，我们对后验概率分布用高斯分布近似，这时仅需要后验概率的最大值位置和分布的方差。通过对后验概率进行Gibs采样，得到一系列采样点，即可对最大值位置和方差做近似估计。

在EM算法中，求似然函数的期望是重要一步，即：

$$\tilde{L}(\boldsymbol{\theta}) = \sum_n \left\{ \sum_{\mathbf{z}} \{p(\mathbf{z}|\mathbf{x}_n;\boldsymbol{\theta}') \ln p(\mathbf{z},\mathbf{x}_n;\boldsymbol{\theta})\} + H(p(\mathbf{z}|\mathbf{x}_n;\boldsymbol{\theta}')) \right\},$$

其中H是$p(\mathbf{z}|\mathbf{x};\boldsymbol{\theta}')$的熵，与$\boldsymbol{\theta}$无关。上式可以通过若干$p(\mathbf{z}|\mathbf{x};\boldsymbol{\theta}')$的采样点进行近似：

$$\tilde{L}(\boldsymbol{\theta}) = \sum_n \left\{ \frac{1}{L}\sum_l \ln p(\mathbf{z}^l,\mathbf{x}_n;\theta) + H(p(\mathbf{z}|\mathbf{x}_n;\boldsymbol{\theta}')) \right\},$$

其中$\{\mathbf{z}^l\}$是L个采样点。

6.7.2 变分法

另一种近似推理的思路是用较简单的概率分布近似较复杂目标概率分布，从而简化推理过程。和采样法相比，变分法效率更高，但因为采用简单函数，有可能会误差较大。为了尽量减少误差，我们尽可能选择一组足够丰富的概率分布函数，并从中选择近似误差最小的一个。求最优函数是求最优取值点的扩展，称为变分法（Variational）。

设图模型定义了一个概率分布函数$p(\mathbf{x},\mathbf{z})$，其中$\mathbf{x}$是观察变量，$\mathbf{z}$是隐藏变量。给定$\mathbf{x}$的一组独立同分布的观测值$X = \{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_N\}$，对应的隐变量为$Z = \{\mathbf{z}_1,\mathbf{z}_2,...,\mathbf{z}_N\}$，推理的目标是求后验概率$p(Z|X)$以及边缘概率$p(X)$。

和EM算法类似，我们可以将$P(X)$进行分解如下：

$$\ln p(X) = \tilde{L}(q) + KL(q||p),$$

其中

$$\tilde{L}(q) = \sum_{Z} q(Z) \ln \frac{p(X,Z)}{q(Z)}, \tag{6.27}$$

$$KL(q||p) = -\sum_{Z} q(Z) \ln \frac{p(Z|X)}{q(Z)}, \tag{6.28}$$

其中$\tilde{L}(q)$是个下界函数，不同的q对应不同的下界函数。显然，最好的q是由$P(X,Z)$ 导出的后验概率$p(Z|X)$，但现在我们假设这个函数不可求解，因此需要假定一个足够灵活的函数集，在该函数集中找到最优的q，使其$\tilde{L}(q(Z))$最大或$KL(q(Z)||p(Z|X))$最小。注意，使$KL(q(Z)||p(Z|X))$最小的q同时使$KL(q(Z)||p(X,Z))$ 最小。

为求该优化函数q，一种办法是选择一个以$\boldsymbol{\theta}$为参数的函数族$q(Z;\boldsymbol{\theta})$，通过优化$\boldsymbol{\theta}$来选择与$p(Z|X)$的KL距离最小的函数。另一种方法是对q做一定假定，并基于该假设得到最优q。一个常用假设是$q(Z)$可分解 [12, 10]，这定义了一个非常广泛的函数族，可保证足够的灵活性，因此被广泛使用。

基于可分解概率函数的变分法

设$q(Z)$具有如下因式分解形式：

$$q(Z) = \prod_{i=1}^{M} q_i(Z_i),$$

其中$Z_1, Z_2, ..., Z_M$为M个隐变量集。注意，这里我们并没有定义$q(Z)$的具体形式，因此所有可分解的概率函数都是可选函数。将上述$q(Z)$代入$\tilde{L}(q)$，并将$q_i(Z_i)$写成q_i，有:

$$\begin{aligned}
\tilde{L}(q) &= \sum_{Z} \prod_{i} q_i \{\ln p(X,Z) - \ln \prod_{k} q_k\} \\
&= \sum_{Z_j} q_j \sum_{Z_{i\neq j}} \prod_{i\neq j} q_i ln(X,Z) - \sum_{Z} q \sum_{k} \ln q_k \\
&= \sum_{Z_j} q_j \ln \tilde{p}(X,Z_j) - const - \sum_{k} \sum_{Z_k} q_k \ln q_k \\
&= \sum_{Z_j} q_j \ln \tilde{p}(X,Z_j) - const - \sum_{Z_j} q_j \ln q_j - \sum_{k\neq j} \sum_{Z_k} q_k \ln q_k.
\end{aligned}$$

其中:

$$\ln \tilde{p}(X,Z_j) = \sum_{Z_{i\neq j}} \prod_{i\neq j} q_i \ln(X,Z) + const = \mathbb{E}_{i\neq j}[\ln p(X,Z)] + const,$$

即$\ln(X,Z)$对不在Z_j中的所有隐变量的期望。注意，上面推导中加入一个不变量是为了使$\tilde{p}(X,Z_j)$ 是一个归一化的概率。现在我们保持所有$\{q_{i\neq j}\}$不变，对q_j进行优化，即寻找q_j，使得下式最大化：

$$\tilde{L}(q_j) = \sum_{Z_j} q_j \ln \tilde{p}(X,Z_j) - \sum_{Z_j} q_j \ln q_j + F(q_{i\neq j}) - const$$

其中$F(q_{i\neq j}) - const$ 是与q_j无关的量。注意到上式右侧前两项正好是q_j与$\tilde{p}(X,Z_j)$的KL距离的负值，因此当q_j取$\tilde{p}(X,Z_j)$时$\tilde{L}(q)$最大化，此时有：

$$\ln q_j^* = \ln \tilde{p}(X,Z_j) = \mathbb{E}_{i\neq j}[\ln p(X,Z)] + const. \tag{6.29}$$

因而有：

$$q_j^*(Z_j) = \frac{exp(\mathbb{E}_{i\neq j}[\ln p(X,Z)])}{\sum_{Z_j} exp(\mathbb{E}_{i\neq j}[\ln p(X,Z)])}.$$

上述推导过程给出一个对q的迭代求解方法：将q分解成M个因子q_j，对每个q_j循环优化；在优化某个q_j时，利用其它因子$q_{i\neq j}$的信息求$\ln(X,Z)$对$Z_{i\neq j}$的期望。可以证明 [7]，这一过程迭代进行，将收敛到使下界函数$\tilde{L}(q)$达到最优的q^*。

仔细考察上述推导过程，可以看到式（6.29）事实上来源于对$\mathrm{KL}(q(Z)|p(X,Z))$的最小化（式6.27）。当X给定时，这等价于对$KL(q(Z)|p(Z|X))$的最小化。如果没有观察变量X，则对$KL(q(Z)|p(Z))$的最小化同样可得到形如式(6.29)的解，即：

$$\ln q_j^*(Z_j) = \ln \tilde{p}(Z_j) = \mathbb{E}_{i\neq j}[\ln p(Z)] + const.$$

因此式(6.29)事实上是以$KL(q|p)$最小化为目标，用可分解概率函数q近似目标函数p的基础公式。另一种常用的近似优化目标为$KL(p|q)$，该方法称为Expectation Propagation [18]。

基于变分法的Bayes线性回归

为了使读者对变分法有更清晰的了解，我们以Bayes线性回归模型为例讨论变分法的应用。在贝叶斯线性回归模型中，将每个输入$\mathbf{x}_n$和输出t_n都作为有向图模型中的变量，并有：

$$p(t_n|\mathbf{w}) = N(t_n|\mathbf{w}^T\mathbf{x}_n, \beta^{-1}),$$

其中参数$\mathbf{w}$是一个随机变量，其概率分布定义为：

$$p(\mathbf{w}) = N(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}).$$

另假设α满足Gmma分布，形式化如下：

$$p(a) = Gam(a|a_0, b_0),$$

其中a_0和b_0是Gamma分布的参数。经上述定义的概率模型可表成示如图6.27所示的有向图。

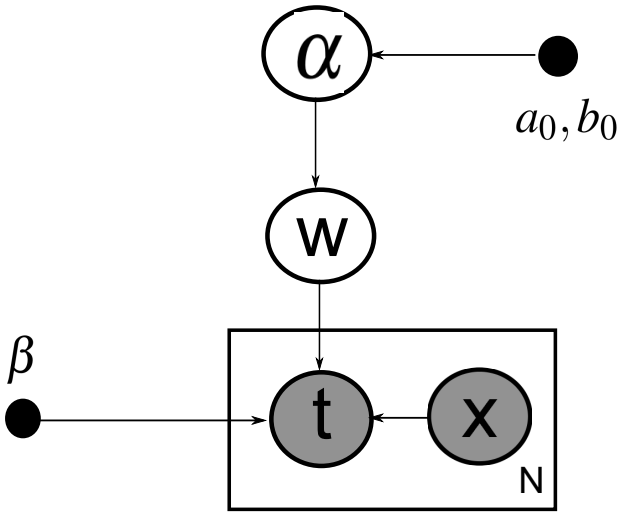


Fig. 6.27 线性回归模型的有向概率图表示。

上述图模型的联合概率有如下形式：

$$p(\{\mathbf{x}_n\}, \{t_n\}, \mathbf{w}, \alpha) = \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}) p(\mathbf{w}|\alpha) p(\alpha).$$

上式中，我们假定$\mathbf{X} = \{\mathbf{x}_n\}$是确切观察值，不具有随机性。现在我们的目标是求后验概率$p(\mathbf{w}, \alpha|\mathbf{t})$，其中$\mathbf{t} = \{t_1, t_2, ..., t_N\}$。取可分解的$q$如下：

$$q(\mathbf{w}, \alpha) = q(\mathbf{w})q(\alpha).$$

基于公式（6.29）求$q^*(\alpha)$，有：

$$\ln q^*(\alpha) = \ln p(\alpha) + \mathbb{E}_{\mathbf{w}\sim q^*(\mathbf{w})}[\ln p(\mathbf{w}|\alpha)] + const \tag{6.30}$$

$$= (\alpha_0 - 1)\ln\alpha - b_0\alpha + \frac{M}{2}\ln\alpha - \frac{\alpha}{2}\mathbb{E}[\mathbf{w}^T\mathbf{w}] + const, \tag{6.31}$$

其中M为$\mathbf{w}$的维度。上式说明α事实上是一个Gamma分布：

$$q^*(\alpha) = Gam(\alpha|a_N, b_N),$$

其中,

$$a_N = a_0 + \frac{M}{2},$$

$$b_N = b_0 + \frac{1}{2}\mathbb{E}[\mathbf{w}^T\mathbf{w}].$$

类似地，应用公式（6.29）求$q^*(\mathbf{w})$，有：

$$\ln q^*(\mathbf{w}) = \ln p(\mathbf{t}|\mathbf{w}) + \mathbb{E}_{\alpha\sim q^*(\alpha)}[\ln p(\mathbf{w}|\alpha)] + const \tag{6.32}$$

$$= -\frac{\beta}{2}\sum_{n=1}^{N}\{\mathbf{w}^T\mathbf{x}_n - t_n\}^2 - \frac{1}{2}\mathbb{E}_{\alpha\sim q^*(\alpha)}[\alpha]\mathbf{w}^T\mathbf{w} + const \tag{6.33}$$

$$= -\frac{1}{2}\mathbf{w}^T(\mathbb{E}_{\alpha\sim q^*(\alpha)}[\alpha]\mathbf{I} + \beta\mathbf{X}\mathbf{X}^T)\mathbf{w} + \beta\mathbf{w}^T\mathbf{X}\mathbf{t} + const. \tag{6.34}$$

由于上式是$\mathbf{w}$的二次型，因此$q^*(\mathbf{w})$符合高斯分布：

$$q^*(\mathbf{w}) = N(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N),$$

其中:

$$\mathbf{m}_N = \beta\mathbf{S}_N\mathbf{X}\mathbf{t}$$

$$\mathbf{S}_N = (\mathbb{E}[\alpha]\mathbf{I} + \beta\mathbf{X}\mathbf{X}^T)^{-1}.$$

注意，在$q^*(\alpha)$中，我们需要求$\mathbb{E}[\mathbf{w}^T\mathbf{w}]$，在$q^*(\mathbf{w})$中，我们需要求$E[\alpha]$。基于Gamma 分布和高斯分布特性，容易验证：

$$\mathbb{E}[\alpha] = a_N/b_N,$$

$$\mathbb{E}[\mathbf{w}\mathbf{w}^T] = \mathbf{m}_N\mathbf{m}_N^T + \mathbf{S}_N.$$

上述推导过程定义了一个迭代优化算法，由一个随机的初始值$\mathbb{E}[\alpha]$和$\mathbb{E}[\mathbf{w}^T\mathbf{w}]$开始，交替更新$q(\alpha)$和$q(\mathbf{w})$，每一次迭代都保证得到更好的下界函数$\tilde{L}(q)$。当收敛时，给定一个新的输入$\mathbf{x}'$，其预测分布为：

$$\begin{aligned} p(t'|\mathbf{x}',\mathbf{t}) &= \int p(\mathbf{t}|\mathbf{x}',\mathbf{w})p(\mathbf{w}|\mathbf{t})\mathrm{d}\mathbf{w} \\ &= \int p(\mathbf{t}|\mathbf{x}',\mathbf{w})q(\mathbf{w})\mathrm{d}\mathbf{w} \\ &= \int N(\mathbf{t}|\mathbf{w}^T\mathbf{x}',\beta^{-1})N(\mathbf{w}|\mathbf{m}_N,\mathbf{S}_N)\mathrm{d}\mathbf{w} \\ &= N(\mathbf{t}|\mathbf{m}_N^T\mathbf{x}',\frac{1}{\beta}+\mathbf{x}'^T\mathbf{S}_N\mathbf{x}'). \end{aligned}$$

可见，该预测函数是一个高斯分布，对t'预测的期望值为$\mathbf{m}_N^T\mathbf{x}'$，这和$\alpha$作为固定参数的贝叶斯线性回归得到的解是一致的，但引入α的随机性改变了$\mathbf{S}_N$的取值，因而增强了模型对复杂数据的描述能力。

变分法应用于概率图模型

我们可以将变分法应用于任何概率图模型。以有向图模型为例，其联合概率分布可表示如下：

$$p(\mathbf{x}) = \prod_i p(x_i|Pa(x_i)),$$

其中x_i表示第i个节点对应的变量，$Pa(x_i)$是该节点的所有父节所代表的变量。我们可以设计如下近似概率分布：

$$q(\mathbf{x}) = \prod_i q_i(x_i),$$

其中每个q_i对应一个变量。应用变分公式（6.29），可以得到：

$$\ln q^*(x_j) = \mathbf{E}_{i\neq j}[\sum_i \ln p(x_i|Pa(x_i))] + const.$$

需要注意的是，上式右侧经过取期望后，将仅包括和x_j相关的项，这些项或者以x_j为变量，或者以x_j为条件，相应的项中将包括和x_j相关的所有节点，即x_j的Markov Blanket。可见，基于变分法，我们只需计算概率图上的局部关系，因而可极大减少计算量。

6.7.3 采样法和变分法比较

采样法和变分法作为两种近似推理方法，在概率图模型中具有重要意义。这两种方法各有优点和缺陷。总体来说，变分法效率较高（虽然仍然需要迭代计算），但需要推导近似函数的迭代公式，如果目标概率比较复杂，推导将比较困难；采样法实现简单（Gibs采样只需考虑局部条件概率即可），但效率通常较低。变分法的收敛精度取决于近似函数的设计，迭代次数再多也无法逼近真实概率分布；采样法的收敛精度取决于采样多少，只要采样足够多，可无限接近真实概率分布。

6.8 本章小结

本章我们讨论了概率图模型。我们从图模型的基本概念开始，介绍了有向概率图模型和无向概率图模型的表示方法、概率意义和相关性判断。基于这些基础定义，我们介绍了几种典型的有向图和无向图模型，特别是介绍了高斯混合模型、隐马尔可夫模型和条件随机场，通过这些模型的参数估计算法，我们总结出图模型的通用参数估计方法，即著名的EM算法。在这一算法中，我们对参数进行迭代估计，在每次迭代时，首先基于当前参数计算隐变量的后验概率，再基于该后验概率计算似然函数的期望，对该期望进行优化即得到新的参数估计。EM算法收敛到似然函数的局部最优解。

推理是指基于给定图模型对事实进行若干推断，包括隐藏变量的后验概率和某些变量的边缘概率。我们首先介绍了精确推理方法，包括适用于树结构的加和-乘积算法及这一算法在通用图模型上的扩展，即Junction-Tree算法。这些算法直观上可以认为是概率信息在节点间的流动，这一思路可以让我们在复杂模型上较容易地设计推理算法。

虽然Junction-Tree算法可以应用于通用概率图模型，但当节点数较多，特别是最大Clique包含的节点数较多时，计算量将急剧增加，使得推理变得不可完成。我们讨论了两种近似推理算法，一种用采样点来模拟实际概率分布，另一种用简单概率分布来近似真实概率分布；前者称为采样法，后者称为变分法。在采样法中，我们特别介绍了MCMC算法，用一个马尔可夫链来代表目标概率分布；MCMC算法的特例，Gibs采样法进一步简化了采样过程。变分法可采用多种近似函数，我们特别介绍了基于可分解概率函数的变分方法，这一方法提供了一种在通用图模型上的近似推理方法，将某一变量

集上的后验概率或边缘概率设计成对该集合中所有变量的可分解函数，并对这些因子函数进行迭代优化。这些推理算法结合到EM框架中，提供了一套完整的模型训练方法。

概率图模型是一种实现复杂推理的基础框架，这一框架可容纳大量领域知识，也可充分利用数据资源，因此具有强大的表达能力和学习能力，我们日常用到的很多模型都属于这种模型。近年来，随着深度学习的兴起，研究者对神经模型的兴趣大幅提高，概率图模型受到的关注有所下降，但这一方法依然具有不可替代的优势，特别是在数据量有限或数据关系复杂的应用场景中，概率图模型依然是最好的选择。另一方面，神经模型和图模型的结合也是必然趋势，如我们前面介绍过的VAE模型 [15]。总体上看，神经模型对于非结构化数据具有很强的学习能力，如感知任务；而概率图模型更适合处理结构化数据，如推理任务。

6.9 相关资源

- 本章参考了Bishop《Pattern Recognition and Machine Learning》一书的第8章，第9章，第10章，第11章。
- 本章部分内容参考了Daphne Koller 的著作《Probabilistic graphical models: principles and techniques》。
- 本章部分内容参考了Eric Xing的课程讲义[1]。
- 关于概率图模型的更多知识可参考相关资料 [13, 12, 23, 11, 20, 19]。

[1] http://www.cs.cmu.edu/ epxing/Class/10708/

Chapter 7
非监督学习

Chapter 8
非参数模型

Chapter 9
遗传学习

Chapter 10
强化学习

Chapter 11
优化方法

References

[1] Baum LE, Eagon JA (1967) An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. Bulletin of the American Mathematical Society 73(3):360–363

[2] Baum LE, Petrie T (1966) Statistical inference for probabilistic functions of finite state markov chains. The annals of mathematical statistics 37(6):1554–1563

[3] Baum LE, Petrie T, Soules G, Weiss N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The annals of mathematical statistics 41(1):164–171

[4] Bertsekas DP (1999) Nonlinear programming. Athena scientific Belmont

[5] Bilmes JA, et al (1998) A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. International Computer Science Institute 4(510):9–10

[6] Bodlaender HL (1994) A tourist guide through treewidth. Acta cybernetica 11(1-2):1

[7] Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge university press

[8] Hastings WK (1970) Monte carlo sampling methods using markov chains and their applications. Biometrika 57(1):97–109

[9] Ising E (1925) Beitrag zur theorie des ferromagnetismus. Zeitschrift für Physik A Hadrons and Nuclei 31(1):253–258

[10] Jaakkola T (2001) 10 tutorial on variational approximation methods. Advanced mean field methods: theory and practice p 129

[11] Jensen FV (1996) An introduction to Bayesian networks, vol 210. UCL press London

[12] Jordan MI (1998) Learning in graphical models, vol 89. Springer Science & Business Media

[13] Jordan MI, Bishop C (2001) An introduction to graphical models. unpublished book

[14] Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999) An introduction to variational methods for graphical models. Machine learning 37(2):183–233

[15] Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:13126114

[16] Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the eighteenth international conference on machine learning, ICML, vol 1, pp 282–289

[17] Lauritzen SL, Spiegelhalter DJ (1988) Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society Series B (Methodological) pp 157–224

[18] Minka TP (2001) Expectation propagation for approximate bayesian inference. In: Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., pp 362–369

[19] Murphy K (1998) A brief introduction to graphical models and bayesian networks

[20] Pearl J (2014) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann

[21] Quattoni A, Wang S, Morency LP, Collins M, Darrell T (2007) Hidden conditional random fields. IEEE transactions on pattern analysis and machine intelligence 29(10)

[22] Sutton C, McCallum A, et al (2012) An introduction to conditional random fields. Foundations and Trends® in Machine Learning 4(4):267–373

[23] Wainwright MJ, Jordan MI, et al (2008) Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning 1(1–2):1–305

[24] Williams C (2009) The junction tree algorithm. URL http://www.inf.ed.ac.uk/teaching/courses/pmr/slides/jta-2x2.pdf